# Lectures for Advanced Control Systems

**January 12, 2025**

EE 60655

Department of Electrical Engineering

University of Notre Dame

# Contents

# Preface

This book grew out of lectures I gave for a course in *advanced control* for first year engineering graduate students at the University of Notre Dame. It was intended as a survey course that covers a range of topics in control theory in a slightly more sophisticated mathematical level than what is found in many first year graduate control courses. Early versions of these lectures were written 5-6 years ago and contained material from more detailed graduate control courses that I had taught in the past. This most recent version was written after having been asked to teach this course again after a 3 year hiatus. The pre-requisite for this course is a graduate level course in linear systems theory and an undergraduate level course in feedback control systems.

The book is essentially divided into five parts. The first part reviews undergraduate SISO control of linear systems with an emphasis on frequency-domain design methods such as loopshaping. This follows the same sort of thread as found in Doyle et al. (2013) and Rohrs et al. (1992). The second part reviews basic approaches to optimal control systems using the classical variational methods and dynamic programming. This part is drawn from Liberzon (2012), Bertsekas (1995), Fleming and Rishel (1972), and Puterman (1994). The third part discusses robust optimal control of linear systems, primarily through the lens of $\mathcal{H}_\infty$ control based on material in Zhou et al. (1996), Sanchez-Pena and Sznaier (1998), Dorato et al. (1994), and Green and Limebeer (2012) The fourth part discusses constructive methods for nonlinear control using backstepping and passivity methods. This part was drawn from Khalil (2002), Freeman and Kokotovic (2008), Krstic et al. (1995), and Sepulchre et al. (2012). The final part is a very brief overview on the recent topic of data-driven control. This part covers a range of methods including indirect methods based on dynamic model decomposition (DmD) and Koopman operators, adaptive control using Control Lyapunov functions, and Machine learning approaches to data-driven control; in particular Reinforcement Learning. The material on Koopman operators and DmD comes from Brunton and Kutz (2022). The material on adaptive control using control Lyapunov functions comes from Freeman and Kokotovic (2008). The material on Reinforcement learning is based on Sutton and Barto (2018).

These lecture notes are a work in progress, having been revised and reorganized several times over the past decade.

M. D. Lemmon
Department of Electrical Engineering
University of Notre Dame
Summer, 2024

CHAPTER 1

# Controller Synthesis for SISO Linear Control Systems

There are two basic control problems of interest to us. The first problem is a *steering problem* that designs an input that drives a dynamical system to a desired operating point in *finite time*. The second problem is a *stabilization problem* that forces the system state or output to remain in a neighborhood of the desired operating point for *all time*. This chapter examines these two problems for single-input single-output (SISO) linear time-invariant (LTI) systems. This problem is often the subject of elementary undergraduate courses in control theory and so this chapter may be seen as a review of those undergraduate topics in control.

This chapter is organized as follows. We first review the use of transfer functions in modeling LTI systems and then discuss how these models are used in pole-placement design of control systems. We then review a frequency-domain controller design method known as loopshaping and use that method to highlight the inherent tradeoff that control systems make between closed-loop performance and performance robustness to model uncertainty. We then review state-space modeling of LTI systems and review material on state feedback and observer-based controllers. This last topic is often covered in many linear systems theory courses, and can be taken as a launch pad for the more advanced techniques covered in this course.

## 1. Operator-theoretic view of Dynamical Systems:

Before reviewing the transfer function concept, let us first introduce the notion of a linear system as an abstract linear transformation over a linear space of signals. We consider a continuous-time system with signals, $x$ :

$\mathbb{R} \rightarrow \mathbb{R}^n$ that are functions mapping the real-valued time, $t \in \mathbb{R}$, onto a real-valued vector, $x(t) \in \mathbb{R}^n$. These signals form a *linear space* that we denote as $L(\mathbb{R}^n)$, with respect to the binary operations of vector addition and dilation (scalar-vector multiplication).

We view a system, $\mathbf{G} : L(\mathbb{R}^p) \rightarrow L(\mathbb{R}^m)$, as an *operator* that maps an input signal, $w \in L(\mathbb{R}^p)$, onto an output signal, $y \in L(\mathbb{R}^m)$. The value that this operator takes for a particular input $w$ will be denoted as $\mathbf{G}[w]$, which is a function of time. The value of the output signal at a particular time $t \in \mathbb{R}$ will be denoted as $\mathbf{G}[w](t)$.

We need to restrict the operator $\mathbf{G}$ to accurately model the forward flow of time. Let us consider a signal $w \in L(\mathbb{R})$ and define the *truncation* of $w$ with respect to a specified time instant, $T$, as the function $w_T : \mathbb{R} \rightarrow \mathbb{R}^m$ that takes values

$$w_T(t) = \begin{cases} w(t) & \text{if } t \leq T \\ 0 & \text{otherwise} \end{cases}$$

The system operator, $\mathbf{G} : L(\mathbb{R}) \rightarrow L(\mathbb{R})$ is said to be *causal* if and only if for any $T \in \mathbb{R}$ we have

$$\mathbf{G}\left[w_T\right](t) = \mathbf{G}[w](t), \quad \text{for all } t \leq T$$

Informally this means that the output of the system prior to time $T$ given a non-truncated input, $w$, is identical to the system's output prior to time $T$ under the truncated input $w_T$. Since $w_T$ is zero for $t > T$ this means that nonzero inputs after time $T$ have no impact on outputs prior to $T$. In other words, future inputs have no impact on past outputs. Throughout this book, we confine our attention to system operators that are *causal*.

We will also find it useful to distinguish between *time-invariant* and *time-varying* systems. Consider a causal system operator $\mathbf{G} : L(\mathbb{R}) \rightarrow L(\mathbb{R})$ whose response to an input $w$ is the function

$$y = \mathbf{G}[w]$$

Let $t_0 \in \mathbb{R}$ denote a specified time index used to shift the input in time. In other words, we consider an input signal $v$ that takes values $v(t) = w(t-t_0)$ for all $t$. If the output of $\mathbf{G}$ to this time-shifted input takes values

$$\mathbf{G}[v](t) = \mathbf{G}[w](t-t_0) = y(t-t_0)$$

for all $t \in \mathbb{R}$, then we say the system is *time-invariant*. A system that is not time invariant is *time-varying*. Time invariance means that the system's behavior in response to an input is independent of when that input was applied to the system. To some extent this may be seen as asserting that time-invariant systems do not "age".

We will also find it convenient to define a special class of *linear systems*. A system $\mathbf{G} : L(\mathbb{R}) \to L(\mathbb{R})$ is said to be *linear* if for any two inputs $w_1, w_2 \in L(\mathbb{R})$ we have

$$\mathbf{G}\left[\alpha w_1 + \beta w_2\right](t) = \alpha \mathbf{G}\left[w_1\right](t) + \beta \mathbf{G}\left[w_2\right](t)$$

for all $t \in \mathbb{R}$ and any $\alpha, \beta \in \mathbb{R}$. This is also referred to as the *principle of superposition* and it essentially says that a linear system is a *linear transformation* between the linear spaces of input and output signals.

So far our operator theoretic view of a dynamical system treats the system as an *algebraic* object; a linear transformation between two linear spaces. It has no notion of distance to tell us how "close" one output signal might be to another. We will therefore find it useful to introduce a topology or metric on these linear signal spaces. Such topologies can be introduced in many ways, but for our purposes we will do so using the concept of a *norm*.

Consider a linear space $L(\mathbb{R}^n)$ of integrable continuous-time functions, $x : \mathbb{R} \to \mathbb{R}^n$. We define the $\mathcal{L}_p$ norm of $x$ where $p$ is a positive integer as

$$\|x\|_{\mathcal{L}_p} \overset{\text{def}}{=} \lim_{T\to\infty} \left( \int_0^T |x(\tau)|^p d\tau \right)^{1/p}$$

where $|x(\tau)|$ is the Euclidean 2-norm of the vector $x(\tau) \in \mathbb{R}^n$. We define the normed linear space, $\mathcal{L}_p$, as the linear space consisting of all functions

$x \in L(\mathbb{R}^n)$ such that $\|x\|_{\mathcal{L}_p}$ is finite. The most commonly used $\mathcal{L}_p$ norms are for $p = 1, 2$, and $\infty$. For $p = \infty$, the norm can be shown to be

$$\|x\|_{\mathcal{L}_\infty} \stackrel{\text{def}}{=} \lim_{p \to \infty} \|x\|_{\mathcal{L}_p} = \max_i \left\{ \sup_{t \in \mathbb{R}} |x_i(t)| \right\}$$

where $|x_i(t)|$ is the absolute value of the $i$th component of vector $x(t)$. The $\mathcal{L}_\infty$ space is then the space of all integrable functions with a finite $\mathcal{L}_\infty$ norm.

One can show that the set of linear transformations we use to represent an LTI SISO system also form a linear space. So we introduce a topology or metric on these linear spaces as well. Let us consider a system $\mathbf{G} : \mathcal{L}_2 \to \mathcal{L}_2$ mapping finite energy input signals onto finite energy output signals. The amount of energy gained or lost between the input and output is sometimes called a *gain* for the system. We can therefore define the system's $\mathcal{L}_2$-induced gain as

$$\begin{aligned}
\|\mathbf{G}\|_{\mathcal{L}_2-\text{ind}} &\stackrel{\text{def}}{=} \sup_{w \neq 0} \frac{\|\mathbf{G}[w]\|_{\mathcal{L}_2}}{\|w\|_{\mathcal{L}_2}} \\
&= \sup_{\|w\|_{\mathcal{L}_2}=1} \|\mathbf{G}[w]\|_{\mathcal{L}_2} \\
&= \inf \{\gamma \in \mathbb{R} \,:\, \|\mathbf{G}[w]\|_{\mathcal{L}_2} \leq \gamma \|w\|_{\mathcal{L}_2}\}
\end{aligned}$$

we say $\|\mathbf{G}\|_{\mathcal{L}_2-\text{ind}}$ is an induced gain because it is induced by our selection of norms for the input and output spaces. The choice of signal space norms is usually driven by the application.

## 2. Impulse Response Function

The operator-theoretic view of dynamical systems provides a high level abstraction without a concrete computational way of representing a system. There are several concrete representations for such systems. The first one of interest to us is based on an LTI system's *impulse response function*.

Let us consider a causal continuous-time SISO linear system $\mathbf{G} : L(\mathbb{R}) \to L(\mathbb{R})$ to which we apply an input signal $w \in L(\mathbb{R})$. Note that one can approximate this function as

$$w(t) \approx w_h(t) = \sum_{k=0}^{\infty} w(kh)h\delta_h(t-kh)$$

for all $t$ where $\delta_h : \mathbb{R} \to \mathbb{R}$ is an *impulse-like* function taking values

$$\delta_h(t) = \begin{cases} \frac{1}{h} & \text{if } 0 \leq t < h \\ 0 & \text{otherwise} \end{cases}$$

The parameter $h$ is a positive real constant representing the duration of a regular sampling interval. In particular, as the length of the sampling interval goes to zero, i.e. $h \to 0$, the values of the approximate function $w_h(t)$ converge to the original function's values $w(t)$ provided $w$ is smooth enough.

If we were to apply the impulse $\delta_h(t-\tau)$ to the linear system, $\mathbf{G} : L(\mathbb{R}) \to L(\mathbb{R})$, it would generate an output response, $g_h(t, \tau)$. The impulse is time shifted so the impulse starts at time $\tau$ and this means that the output response, $g_h$, is not only a function of time, $t$, it is a function of the time, $\tau$, when that impulse was applied. Since the system is linear we can deduce that the system's output to the approximate input signal $w_h$ will be a function $y_h$, that can be written as

$$\begin{aligned} y_h(t) &= \mathbf{G}\left[\sum_{k=0}^{\infty} \delta_h(t-kh)w(kh)h\right] \\ &= \sum_{k=0}^{\infty} hw(kh)g_h(t, kh) \end{aligned}$$

In the limit as the sampling interval, $h$, goes to zero, we can take the limit inside the above summation to deduce that the system's response $y = \mathbf{G}[w]$

to the original input $w$ is

$$\begin{aligned} y(t) &= \lim_{h\to 0} y_h(t) \\ &= \lim_{h\to 0} \sum_{k=0}^{\infty} hw(kh)g_h(t,kh) \\ &= \int_0^{\infty} w(\tau)g(t,\tau)d\tau \end{aligned}$$

where $g : \mathbb{R}^2 \to \mathbb{R}$ is the limiting value of $g_h$ as $h \to 0$. In other words, $g$'s values are

$$g(t,\tau) = \lim_{h\to 0} g_h(t,\tau)$$

This function $g$ is called the *impulse response function* of the linear system $\mathbf{G}$ because as $h \to 0$ the impulse-like functions $\delta_h$ converge to the classical Dirac delta function, $\delta$.

A linear system's impulse response function has a number of useful properties that we itemize below without formal proof.

- If the linear system, $\mathbf{G}$, is causal, then $g(t,\tau) = 0$ for $\tau > t$.
- If the linear system is time invariant then $g(t,\tau) = g(t-\tau)$.
- For causal LTI systems, one can see that

$$y(t) = \int_{-\infty}^{t} g(t-\tau)w(\tau)d\tau = [g * w](t) \tag{1}$$

where $g * w$ denotes the *convolution integral* of the two functions.

Note that the impulse response function, $g$, plays a critical role in equation (1) in computing a system's output. We can therefore see that one way of concretely representing an LTI system is by first specifying what its impulse response function is.

## 3. Transfer Function Modeling of LTI Systems

Impulse response functions provide a concrete time-domain representation of an LTI system. Determining this system's response to any input requires the solution of a convolution integral that can be tedious and difficult to compute directly. It is common practice to analyze such systems by first transforming them in a way that turns the calculus operations of integration and differentiation into the algebraic operations of multiplication and division. These algebraic operations are much easier to work. This section discusses one such transform-based representation known as the system's *transfer function*; namely the single sided Laplace transform of the system's impulse response function.

Given the impulse response function, $g$, equation (1) provides a way to concretely compute the response of a SISO LTI system to any input $w$. This computation, however, is easier to do if we first transform the time-domain signals in the equation into functions of a complex variable using single sided Laplace transforms. The Laplace transform of a real function $x : \mathbb{R} \to \mathbb{R}$ maps that function $x$ onto a function of a complex variable $X : \mathbb{C} \to \mathbb{C}$ through the integral equation

$$X(s) = \int_0^\infty x(\tau) e^{-s\tau} d\tau$$

where $s \in \mathbb{C}$. The transform may be seen as an operator $\mathcal{L} : L(\mathbb{R}) \to L(\mathbb{C})$ mapping the linear space of real functions onto the linear space of complex functions, so that $X(s) = \mathcal{L}[x](s)$.

Laplace transforms are often seen in undergraduate courses in signals and systems and ordinary differential equations, so we will not review that material here. Instead we will highlight some of the more useful aspects of those transforms. In particular, we know that Laplace transforms are invertible transforms so that $X$ may be seen as an equivalent concrete representation of the function $x$.

- The transformed function $X(s)$ is said to be a rational function if it can be written as the ratio, $X(s) = \frac{n(s)}{d(s)}$, of two polynomials in $s$ with real coefficients. If the degree of the numerator polynomial, $n(s)$, is strictly less than the degree of the denominator polynomial, $d(s)$, then $X(s)$ is said to be strictly proper and we can assert that $|X(s)| \to 0$ as $|s| \to \infty$.
- If the transformed function $X(s)$ is a strictly proper rational function, then we can factor the denominator polynomial, $d(s)$, into first order factors and use a *partial fraction expansion* to express $X(s)$ as

$$\begin{aligned} X(s) &= \frac{b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_1 s + b_0}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_1 s + a_0} \\ &= \frac{b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_1 s + b_0}{(s-p_1)(s-p_2)\cdots(s-p_{n-1})(s-p_n)} \\ &= \frac{K_1}{s-p_1} + \frac{K_2}{s-p_2} + \cdots + \frac{K_n}{s-p_n} \end{aligned}$$

where $p_i$ ($i = 1, \ldots, n$) are the $n$ distinct roots of the polynomial equation $d(s) = 0$. We refer to these roots as the *finite poles* (removable singularities) of $X(s)$. We refer to the roots of the polynomial equation $n(s) = 0$ formed from the transfer function's numerator polynomial as the *finite zeros* of $X(s)$.
- If the transformed function $X(s)$ is a strictly proper rational function with partial fraction expansion,

$$X(s) = \frac{K_1}{s-p_1} + \frac{K_2}{s-p_2} + \cdots + \frac{K_n}{s-p_n}$$

Then its *inverse transform* gives the time domain signal, $x$,

$$x(t) = K_1 s^{p_1 t} u(t) + K_2 s^{p_2 t} u(t) + \cdots + K_{n-1} s^{p_{n-1} t} u(t) + K_n s^{p_n t} u(t)$$

where $u(t)$ is the unit step function.
- If we consider the convolution integral, $[g * w](t)$, of two signals, $g(t)$ and $w(t)$, then the Laplace transform of $g * w$ is

$$\mathcal{L}[g * w](s) = \mathcal{L}[g](s)\mathcal{L}[w](s) = G(s)W(s) \tag{2}$$

The operational transform formula in equation (2) means that if we have a causal linear LTI system whose response to a given input is given as

$$y(t) = [g * w](t)$$

then we can compute the Laplace transform of that output as

$$Y(s) = \mathbf{G}(s)W(s)$$

We know that for causal LTI systems the Laplace transform of the impulse response function is going to be a rational function. If we then confine our input to also be signals with rational Laplace transforms, then the output will also be rational and we can use Partial Fraction Expansion methods to easily compute the time domain output of the system to any known input. The function $\mathbf{G}(s)$ in this case can be viewed as a concrete representation of the linear transformation $\mathbf{G}$ and the removable singularities of $\mathbf{G}$ then become the *poles* of the system $\mathbf{G}$. We refer to $\mathbf{G}(s)$ as the *transfer function* of the system $\mathbf{G}$ and we can readily see that it is nothing more than the Laplace transform of the original system's impulse response function, $g$. We will often use boldfaced notation to denote transfer functions.

Transfer functions play an important role in the analysis and design of dynamical systems. In the first place, note that if we apply to the system $\mathbf{G}$ a sinusoidal input

$$w(t) = A\cos(\omega t + \phi)$$

of frequency $\omega$ with amplitude $A$ and phase shift $\phi$, then the output can be shown to be

$$y(t) = A|\mathbf{G}(j\omega)|\cos(\omega t + \phi + \arg(\mathbf{G}(j\omega)))$$

In other words the output is still a sinusoid of frequency $\omega$ but its amplitude is obtained by multiplying the input's amplitude by the modulus of the transfer function $|\mathbf{G}(j\omega)|$ evaluated at the specified frequency $s = j\omega$ and its phase is shifted by the argument of $\mathbf{G}(s)$ at $s = j\omega$. The functions $|\mathbf{G}(j\omega)|$ and $\arg(\mathbf{G}(j\omega))$ are commonly called the system's *frequency response function*. One important aspect of this is that the frequency response of a system can be readily measured experimentally by simply applying a sinusoidal input and measuring the gain and phase of the resulting sinusoidal output. So transfer functions provide useful empirical representations of a system's input/output response.

Another important feature of transfer functions is that the poles of $\mathbf{G}(s)$ play a major role in characterizing the system's transient response to any input. In particular, if we know that the input is a bounded function that asymptotically goes to zero, then the poles of $\mathbf{G}(s)$ govern how quickly the output goes to zero. In particular, we require that the real part of these poles be negative to ensure exponential rate of decay to the output. In many cases, we want our controlled systems, $\mathbf{G}(s)$, to exhibit this asymptotic behavior. As a result transfer functions play an important role in control system design based on placing the poles of the control system. A pole-placement design methodology for SISO LTI systems will be discussed in the next section.

## 4. Pole Placement Design for LTI Control Systems

This section examines pole-placement methods for designing control systems whose output, $y$, asymptotically track a desired "reference" input signal, $r$. These pole-placements methods are often discussed in undergraduate control courses. Let us consider a *one-parameter control system* whose block diagram is shown in Fig. 1.

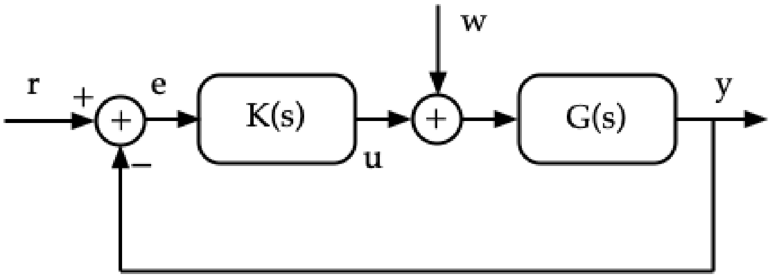


FIGURE 1. One Parameter Control System

The original plant we wish to control has the transfer function $\mathbf{G}(s)$. The output of this plant, $y$, is then subtracted off of a reference input, $r$, that has been supplied to the control system. The reference, $r$, represents the signal we want the output, $y$, to track. The resulting error signal $e = r - y$ is then fed into a feedback controller, $\mathbf{K}(s)$, to create the commanded control signal, $u$. We assume there is an additional *disturbance*, $w$, that is added to the control input, $u$. The resulting combined signal, $u + w$, is then what drives the plant. Our problem is to design $\mathbf{K}(s)$ so the output $y$ asymptotically tracks $r$ (i.e. $e(t) \rightarrow 0$ as $t \rightarrow \infty$). The approach we will take is to design $\mathbf{K}(s)$ to place the poles of the closed loop transfer function from $r$ to $e$ (i.e. $\mathbf{T}_{re}$) and the transfer function from $w$ to $u$ (i.e., $\mathbf{T}_{wu}$). In particular we want $e$ to asymptotically go to zero when the disturbance $w = 0$ and we want to constrain how large the control input $u$ will be in response to disturbances, $w$, that are not zero.

We start by deriving an expression for the closed loop transfer functions. Note that

$$Y(s) = \mathbf{G}(s)(W(s) + \mathbf{K}(s)(R(s) - Y(s)))$$

We can rewrite this as

$$(1 + \mathbf{G}(s)\mathbf{K}(s))Y(s) = \mathbf{G}(s)W(s) + \mathbf{G}(s)\mathbf{K}(s)R(s)$$

to get

$$Y(s) \quad = \quad \frac{\mathbf{G}(s)}{1 + \mathbf{G}(s)\mathbf{K}(s)}W(s) + \frac{\mathbf{G}(s)\mathbf{K}(s)}{1 + \mathbf{G}(s)\mathbf{K}(s)}R(s)$$

Note that

$$\begin{aligned} E(s) &= R(s) - Y(s) \\ U(s) &= \mathbf{K}(s)E(s) \end{aligned}$$

which means

$$\begin{aligned} E(s) &= R(s) - \frac{\mathbf{G}}{1+\mathbf{GK}}W - \frac{\mathbf{GK}}{1+\mathbf{GK}}R \\ &= -\frac{\mathbf{G}}{1+\mathbf{GK}}W + \frac{1}{1+\mathbf{GK}}R(s) \\ U(s) &= \mathbf{K}(s)E(s) = -\frac{\mathbf{GK}}{1+\mathbf{GK}}W(s) + \frac{\mathbf{K}}{1+\mathbf{GK}}R(s) \end{aligned}$$

Our first objective is to manage the behavior of the error signal in response to the reference input $R(s)$ assuming zero input disturbance $W(s) = 0$. In particular, we want to ensure that the error $e(t)$ asymptotically goes to zero as $t \to \infty$. We know this will be guaranteed if we select $\mathbf{K}(s)$ to ensure all poles of

$$\mathbf{T}_{re}(s) = \frac{1}{1+\mathbf{G}(s)\mathbf{K}(s)}$$

have negative real parts.

The second objective is to manage the behavior of the control signal $U(s)$ in response to the disturbance $W(s)$ assuming a zero reference input, $R(s) = 0$. In particular, we will require that $u(t)$ asymptotically goes to zero as $t \to \infty$ for any bounded $W(s)$ we might have. This requires that we select $\mathbf{K}(s)$ to also ensure that all poles of

$$\mathbf{T}_{wu}(s) = \frac{\mathbf{G}(s)\mathbf{K}(s)}{1+\mathbf{G}(s)\mathbf{K}(s)}$$

have negative real parts.

Since we are dealing with SISO LTI systems, we can assume $\mathbf{G}(s)$ and $\mathbf{K}(s)$ are all strictly proper rational functions. In particular, we let

$$\mathbf{G}(s) = \frac{n_g(s)}{d_g(s)}, \quad \mathbf{K}(s) = \frac{n_k(s)}{d_k(s)}$$

This means that

$$\begin{aligned}\mathbf{T}_{re}(s) &= \frac{1}{1+\frac{n_g(s)n_k(s)}{d_g(s)d_k(s)}} = \frac{d_g(s)d_k(s)}{d_g(s)d_k(s)+n_g(s)n_k(s)} \\ \mathbf{T}_{wu}(s) &= \frac{\frac{n_g(s)n_k(s)}{d_g(s)d_k(s)}}{1+\frac{n_g(s)n_k(s)}{d_g(s)d_k(s)}} = \frac{n_g(s)n_k(s)}{n_g(s)n_k(s)+d_g(s)d_k(s)}\end{aligned}$$

The poles of both transfer functions are determined by our choice of the controller polynomials $n_k(s)$ and $d_k(s)$. If we let $d_d(s)$ denote the desired denominator polynomial then we need to select $n_k(s)$ and $d_k(s)$ to satisfy the Diophantine equation

$$d_d(s) = n_g(s)n_k(s) + d_g(s)d_k(s)$$

Suppose that the plant $\mathbf{G}(s)$ is strictly proper with no pole-zero cancellations. One can then show if there is an $m$th order proper controller $\mathbf{K}(s) = n_k(s)/d_k(s)$ solving the Diophantine equation, it must have a degree $m \geq n-1$.

As an example, let us consider the plant

$$\mathbf{G}(s) = \frac{n_g(s)}{d_g(s)} = \frac{1}{s^3+s}$$

So we have $n = 3$ and so the controller must be of order

$$m \geq n-1 = 3-1 = 2$$

We therefore choose a proper controller

$$\mathbf{K}(s) = \frac{c_2s^2+c_1s+c_0}{s^2+d_1s+d_0} = \frac{n_k(s)}{d_k(s)}$$

We now form the Diophantine equation

$$d_g(s)d_k(s)+n_g(s)n_k(s) \quad = \quad s^5+d_1s^4+(1+d_0)s^3+(d_1+c_2)s^2+(d_0+c_1)s+c_0$$

We will let the desired poles be at $-3 \pm 3j$ with two poles at $-5$ and one at $-10$ so the desired denominator polynomial is

$$\begin{aligned}d_d(s) &= (s+3-3j)(s+3+3j)(s+5)^2(s+10) \\ &= s^5+26s^4+263s^3+1360s^2+3750s+4500\end{aligned}$$

equating the coefficients in the two polynomials gives the following set of linear algebraic equations

$$\begin{bmatrix} 26 \\ 263-1 \\ 1360 \\ 3750 \\ 4500 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_0 \\ d_1+c_2 \\ d_0+c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_0 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

The solution to this LAE has $d_1 = 26$, $d_0 = 262$, $c_2 = 1334$, $c_1 = 3488$, and $c_0 = 4500$. So the controller is

$$\mathbf{K}(s) = \frac{1334s^2 + 3488s + 4500}{s^2 + 26s + 262}$$

The linear algebraic equation we formed from the Diophantine equation has a special structure. To see this structure, let us return to our example but simply specify the plant as

$$\mathbf{G}(s) = \frac{n_g(s)}{d_g(s)} = \frac{a_2s^2 + a_1s + a_0}{s^3 + b_2s^2 + b_1s + b_0}$$

The controller will be the same as before

$$\mathbf{K}(s) = \frac{n_k(s)}{d_k(s)} = \frac{c_2s^2 + c_1s + c_0}{s^2 + d_1s + d_0}$$

and we'll leave the desired polynomial to be

$$d_d(s) = s^5 + \beta_4s^4 + \beta_3s^3 + \beta_2s^2 + \beta_1s + \beta_0$$

The Diophantine equation can then be written out as

$$\begin{aligned} n_g(s)n_k(s) + d_g(s)d_k(s) &= (s^3 + b_2s^2 + b_1s + b_0)(s^2 + d_1s + d_0) \\ &\quad +(a_2s^2 + a_1s + a_0)(c_2s^2 + c_1s + c_0) \\ &= s^5 + \beta_4s^4 + \beta_3s^3 + \beta_2s^2 + \beta_1s + \beta_0 \end{aligned}$$

Multiplying out and equating coefficients gives rise to the following system of linear algebraic equations

$$\begin{bmatrix} 1 \\ \beta_4 \\ \beta_3 \\ \beta_2 \\ \beta_1 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ b_2 & 1 & 0 & a_2 & 0 & 0 \\ b_1 & b_2 & 1 & a_1 & a_2 & 0 \\ b_0 & b_1 & b_2 & a_0 & a_1 & a_2 \\ 0 & b_0 & b_1 & 0 & a_0 & a_1 \\ 0 & 0 & b_0 & 0 & 0 & a_0 \end{bmatrix} \begin{bmatrix} 1 \\ d_1 \\ d_0 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

The matrix is called a Sylvester matrix. The condition $m \geq n-1$ is required to ensure there are enough equations so any vector $[1, \beta_4, \beta_3, \beta_2, \beta_1, \beta_0]^T$ lies in the range space of the Sylvester matrix (i.e. a solution exists).

Where should we ”place” poles of a closed loop system? In classical undergraduate courses, we can first places specifications on the steady-state and transient response of the closed-loop system to various test signals (step, ramp inputs) and then determines the pole locations that are consistent with those specifications. In general, this can only be done for systems whose transient response is characterized by a dominant pole pair. In this case, there are specific formulae relating pole location to such specifications on the closed-loop system’s rise time, peak overshoot, and settling time.

**Steady-State Tracking Requirements:** Consider a one-parameter control system in Fig. 1. The tracking error may be written as $e(t) = r(t) - y(t)$ where $r$ is the reference input and $y$ is the output. Taking the Laplace transform of the error signal gives

$$E(s) = R(s) - Y(s) = R(s) - \mathbf{G}(s)\mathbf{K}(s)E(s)$$

assuming $W(s) = 0$. Solving for $E(s)$ gives

$$E(s) = \frac{1}{1 + \mathbf{G}(s)\mathbf{K}(s)} R(s)$$

Using the final value theorem for Laplace transforms, we can now evaluate the steady state tracking error as

$$e_{\text{ss}} = \lim_{s \to 0} sE(s) = \lim_{s \to 0} \frac{sR(s)}{1 + \mathbf{G}(s)\mathbf{K}(s)}$$

We can classify control systems by the *type* of test signal they can track with zero steady state error. The *type* is defined to be the number of poles at zero in the loop function $\mathbf{G}(s)\mathbf{K}(s)$. Let us consider a step input so that $R(s) = \frac{1}{s}$. In this case we see that

$$e_{\mathrm{ss}} = \lim_{s \to 0} \frac{1}{1 + \mathbf{G}(s)\mathbf{K}(s)}$$

Note that this goes to zero if $\lim\limits_{s \to 0} \mathbf{G}(s)\mathbf{K}(s) = \infty$. This will occur if the loop function has a pole at the origin. Since there is one pole at the origin this is called a ”type 1” systems.

Now let us consider a ramp input as a test input. In this case $R(s) = \frac{1}{s^2}$ and we have

$$e_{\mathrm{ss}} = \lim_{s \to 0} s \frac{1}{s^2} \frac{1}{1 + \mathbf{G}(s)\mathbf{K}(s)} = \frac{1}{\lim\limits_{s \to 0} s\mathbf{G}(s)\mathbf{K}(s)}$$

As we can see for $e_{\mathrm{ss}} = 0$ to be zero for a ramp input, we need $\mathbf{G}(s)\mathbf{K}(s)$ to have two poles at the origin. Hence such as system would be a type 2 control system.

What these observations mean is that if we require the closed loop system to have zero steady-state tracking error to a type $n$ test input, then we need to select the controller $\mathbf{K}(s)$ so the loop function $\mathbf{G}(s)\mathbf{K}(s)$ has $n$ poles at the origin. This is sometimes known as the *internal model principle*, because it means the desired ”test input” must be embedded within the loop function as an internal model of the signal we wish to track.

**Transient Response Specifications:** Specifications on a system’s transient response are usually characterized with respect to a system’s *step response* (i.e. its response to a unit step input). The common measures we use to characterize this performance are

- Peak Overshoot, $M_p$, the peak value $y_{\max}$ that the output takes which is usually expressed as a percentage equal to $100 \times \frac{y_{\max} - y_{\mathrm{ss}}}{y_{\mathrm{ss}}}$.
- Rise Time, $t_r$, is the amount of time it takes the system response to go from 0 to first reach 90 percent of its final value.
- Settling Time, $t_s$, is the interval of time it takes the response to go from 0 until it is within 10% of its final value for all future times.

Fig. 2 illustrates where how these characteristics of a second order system's transient response might be measured from a plot of its step response.

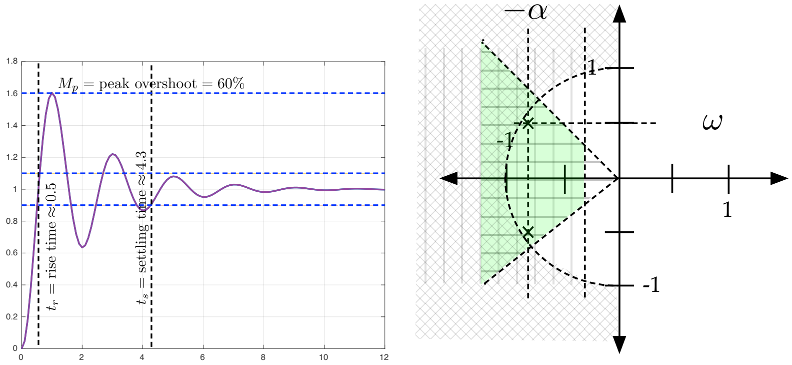


FIGURE 2. (left) step response of second order system (right) desired location of second order system's poles

In general, it is only possible to develop formulae relating rise time, settling time, and overshoot to the poles of first or second order systems. In the following we focus on second order systems of the form

$$\mathbf{G}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

we can derive explicit formulae that can be used in selecting pole locations. In this case, one can write the step response as

$$y(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin(\omega_n \sqrt{1-\zeta^2} t + \cos^{-1}\zeta)$$

for $t \geq 0$. The parameters $\omega_n$ and $\zeta$ are standardized parameters known as the system's *natural frequency* and *damping ratio*, respectively. For this system the system poles are

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -\alpha \pm j\omega = -\omega_n\zeta \pm j\omega_n\sqrt{1-\zeta^2} = re^{j\theta}$$

where $\theta = \cos^{-1}\zeta$ and $r = \omega_n$. For such a system the peak overshoot, rise and settling times can be explicitly approximated as follows.

$$\begin{aligned} M_p &= 100 \times e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \\ t_r &\approx \frac{0.8 + 2.5\zeta}{\omega_n} \\ t_s &\approx \frac{3.2}{\omega_n \zeta}, \quad \text{when } 0 < \zeta < 0.69 \end{aligned}$$

We then usually place specifications on these metrics in the form of inequality constraints. These inequalities then define a region of ”acceptable” pole locations in the complex plane, which we then use to place the system’s closed-loop poles.

As an example, let’s consider a second order closed loop system where that requires $M_p \leq 10\%$, $t_s < 1$ seconds and $t_r$ is as fast as possible. The peak overshoot requirement is

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} < 0.1 \quad \Rightarrow \quad \zeta > 0.59$$

This places a constraint on the damping ratio. In particular since we know the ”angle”, $\theta$, of the pole is $\theta = \cos^{-1}(\zeta)$, we actually define the sector shown above in Fig. 2. The requirement on the settling times means

$$t_s = \frac{3.2}{\zeta\omega_n} < 1 \quad \Rightarrow \zeta\omega_n > \frac{1}{3.2}$$

Since the real part of the pole is $-\zeta\omega_n$, this means all poles meeting the settling time constraint must be to the left of the $-1/3$. This region is also shown in Fig. 2 and the intersection of the two identified regions represents a feasible set of pole locations. A requirement on the rise time that is less than $0.1$ seconds means

$$t_r \approx \frac{0.8 + 0.25\zeta}{\omega_n} > 0.1, \quad \Rightarrow \quad \omega_n > \frac{0.1}{0.3 + 0.25\zeta}$$

where $\zeta$ was chosen to enforce the peak overshoot constraint. Note that $\omega_n$ is the modulus of the poles, so these poles must lie outside of a circular sector determined by that modules. When put together we have a feasible set of locations for pole placement.

This is a classical approach used in designing low order control systems for undergraduate courses. But it is extremely limited in its usefulness. In the first place it really requires that the system’s response is dominated by a pole pair, which may not be the case in practice. In addition to this, the method relies heavily on particular test inputs like steps and ramps, which may not also reflect the actual environment the system may face. This method confines itself to SISO systems which means it is of little use in more complicated MIMO control system design. Finally, this method presumes an accurate prior model of the plant. Since there can be a great deal of uncertainty in our prior models, these methods provide little insight into how to ensure our control system's performance is *robust* to model uncertainty. To address these issues, we will examine two other approaches for designing SISO control systems, that we can then leverage to design more complex robust MIMO control systems. The first approach is a frequency-domain design method known as loopshaping. Even though we'll discuss this method in the context of SISO control systems, it turns out that loopshaping insights can also be used to help design MIMO $\mathcal{H}_\infty$ controllers. We will also examine the use of state-based methods (in particular, observer-based controllers) to provide a systematic framework that works well for MIMO systems.

## 5. Conflicting Control Objectives in Feedback Loops

How do we decide where to place the poles of a closed-loop system? In general, they are chosen to assure the asymptotic stability of the system, but there are many ways of formulating these regulation objective on various parts of the closed loop system and it is important that we formulate these objectives so they don't conflict with each other. Recall we identified two types of inputs, $r$ and $w$, for our earlier one-parameter control system. We also considered two different outputs of interest to us, the tracking error $e$ and the control effort $u$. What we showed above is that

$$\begin{aligned} E(s) &= -\frac{\mathbf{G}(s)}{1+\mathbf{G}(s)\mathbf{K}(s)}W(s) + \frac{1}{1+\mathbf{G}(s)\mathbf{K}(s)}R(s) \\ U(s) &= -\frac{\mathbf{G}(s)\mathbf{K}(s)}{1+\mathbf{G}(s)\mathbf{K}(s)}W(s) + \frac{\mathbf{K}(s)}{1+\mathbf{G}(s)\mathbf{K}(s)}R(s) \end{aligned}$$

We will find it convenient to define a loop function

$$\mathbf{L}(s) = \mathbf{G}(s)\mathbf{K}(s)$$

and two sensitivity functions. The first sensitivity function is

$$\mathbf{S}(s) = \frac{1}{1+\mathbf{L}(s)}$$

and the complementary sensitivity function

$$\mathbf{T}(s) = \frac{\mathbf{L}(s)}{1 + \mathbf{L}(s)}$$

We refer to the transfer function $\mathbf{T}(s)$ as being complementary since it is apparent that

$$\mathbf{S}(s) + \mathbf{T}(s) = 1$$

This complementary relationship places severe constraints on the control objectives we can enforce with regard to the two outputs.

When we discussed pole placement, the only requirement we placed on the poles was that they had negative real parts (stable). But in general, we can be a bit more concrete about ”how stable” we want these poles to be using the system’s *induced gain*. Let us consider a system that is $\mathcal{L}_2$ stable in the sense that it maps $\mathcal{L}_2$ signals (finite energy) onto $\mathcal{L}_2$ signals (finite energy). One can show that for an SISO LTI system $\mathbf{G}$, its induced $\mathcal{L}_2$ gain is

$$\|\mathbf{G}\|^2_{\mathcal{L}_2-\mathrm{ind}} = \max_{\omega} |\mathbf{G}(j\omega)| \equiv \|\mathbf{G}\|_{\mathcal{H}_\infty}$$

Note that since

$$\|y\|_{\mathcal{L}_2} \leq \|\mathbf{G}\|_{\mathcal{L}_2-\mathrm{ind}} \|w\|_{\mathcal{L}_2} = \|\mathbf{G}\|_{\mathcal{H}_\infty} \|w\|_{\mathcal{L}_2}$$

we have that for an input $w$ of unit energy, that the total energy in the output signal is determined by the induced gain of the system itself. In most cases we want the signal $w$ to be ”rejected” at the output. This is the case in our particular control system where we wanted the tracking error $e$ to be small for any reference input $r$ and we want the impact of the disturbance $w$ on the control signal $u$ to be small. In other words, we require for unit input $\|w\|_{\mathcal{L}_2} = 1$ that

$$\begin{aligned} \|e\|_{\mathcal{L}_2} &\leq \|\mathbf{S}\|_{\mathcal{H}_\infty} \leq \gamma_r \\ \|u\|_{\mathcal{L}_2} &\leq \|\mathbf{T}\|_{\mathcal{H}_\infty} \leq \gamma_w \end{aligned}$$

where $\gamma_r$ and $\gamma_w$ are two small positive constants representing the specification on system performance and $\mathbf{S}$ and $\mathbf{T}$ are the two sensitivity functions for this control system. Our objective is now to select $\mathbf{K}(s)$ so that both closed loop systems, $\mathbf{S}$ and $\mathbf{T}$, are stable *and* so that their induced gains satisfy the limits imposed by the design parameters $\gamma_r$ and $\gamma_w$.

Note that both of these sensitivity functions are complementary to each other since

$$\mathbf{S}(s) + \mathbf{T}(s) = 1$$

for any $s \in \mathbb{C}$. Let us suppose we've selected a controller $\mathbf{K}$ that kept the tracking error small

$$\frac{1}{\gamma_r} \|\mathbf{S}\|_{\mathcal{H}_\infty} < 1$$

For good tracking performance we obviously want $\gamma_r \ll 1$. Because $\mathbf{S} + \mathbf{T} = 1$, we have

$$\|\mathbf{T}\|_{\mathcal{H}_\infty} = \|1 - \mathbf{S}\|_{\mathcal{H}_\infty} \geq 1 - \|\mathbf{S}\|_{\mathcal{H}_\infty} \geq 1 - \gamma_r$$

Since $\gamma_r \ll 1$ we can readily see that $\|\mathbf{T}\|_{\mathcal{H}_\infty}$ is close to one, which means we cannot make $\gamma_w$ arbitrarily small. There is a limit on how aggressively we can enforce the second requirement if we also require very good reference tracking. This is a fundamental limitation on what one can achieve in designing feedback control systems. We need to find a way around this limitation to design real-life control systems.

To address the conflicting nature of multiple control objectives, we note that requiring the $\mathcal{H}_\infty$ norm of the closed loop map $\mathbf{S}$ to be less than $\gamma_r$ means that the gain magnitude $|\mathbf{S}(j\omega)|$ of the transfer function is less than $\gamma_r$ for all frequencies $\omega$. The input signal, $r$, however, may not have significant energy in all frequency bands. A reference signal, $r$, often has most of its energy in low frequency harmonics. In a similar way requiring the $\mathcal{H}_\infty$ norm of the complementary sensitivity $\mathbf{T}$ to be less than $\gamma_w$ means that $|\mathbf{T}(j\omega)| < \gamma_w$ for all $\omega$. Again if we think of the input disturbance as being ”noise”, then it usually does not have significant energy in all frequencies. In particular, many noise processes have most of their energy in high frequency harmonics.

What this means is that we do not really require the sensitivity and complementary sensitivity functions to have their gain magnitudes small over all frequencies. We therefore introduce a weighting system whose gain magnitude is large for those frequencies we care about and essentially goes to zero for those less important frequencies. In this regard, the performance requirement is transformed from an unweighted set of constraints into the following weighted constraints

$$\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} < 1 \quad \text{and} \quad \|\mathbf{W}_\Delta\mathbf{T}\|_{\mathcal{H}_\infty} < 1$$

where $\mathbf{W}_p(s)$ is a stable minimum phase transfer function that is close to $1/\gamma_r$ for frequencies where input $r$ has significant energy and is close to zero elsewhere. The other system $\mathbf{W}_\Delta(s)$ is another stable minimum phase transfer function that is close to $1/\gamma_w$ for those frequencies where the disturbance $w$ has significant energy content. In this regard, we are treating the weighting systems, $\mathbf{W}_p$ and $\mathbf{W}_\Delta$, as *performance specifications* on the system that represent the frequency weighted levels of tracking performance and performance specifications we require of our system.

Note that we are not free to pick these weighting systems to be anything. In particular, they too must satisfy the underlying complementary nature of the sensitivity functions. In particular this means they should satisfy

$$|\gamma_r\mathbf{W}_p(j\omega) + \gamma_w\mathbf{W}_\Delta(j\omega)| < 1$$

for all $\omega$. In other words, the frequencies where $|\gamma_t\mathbf{W}_p(j\omega)| \approx 1$ must be frequencies where $|\gamma_w\mathbf{W}_\Delta(j\omega)|$ is close to zero. So we do not attempt to enforce the tracking requirement and the disturbance rejection requirements at the same frequency $\omega$. We identify an interval of frequencies over which the tracking constraint is aggressively enforced and this interval must be disjoint from the interval of frequencies over which the disturbance rejection requirement is aggressively enforced. Specifications that satisfy these conditions are said to be *well posed*.

## 6. Frequency-based Controller Design - Loopshaping

The preceding section asserted that one way of designing controllers for the feedback system in Fig. 1 is to select a controller $\mathbf{K}(s)$ so the weighted objectives

$$\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} < 1 \quad \text{and} \quad \|\mathbf{W}_\Delta\mathbf{T}\|_{\mathcal{H}_\infty} < 1 \tag{3}$$

are satisfied subject to the weighting systems being well posed and the closed loop system being stable. The requirement that the weighting systems are well-posed means we are not attempting to enforce tracking and disturbance rejection objectives at the same frequency. The stability condition requires that for all bounded inputs, the outputs of the closed loop system are bounded *and* that any internal signals of the loop (such as $u$) remain bounded as well.

The problem with these specifications is that they may be difficult to certify because they require closed-form representations of the closed loop sensitivity functions. In many real-life applications we don't actually have concrete representations of $\mathbf{S}$ and $\mathbf{T}$. What we usually have is a concrete representation for the open loop plant's transfer function $\mathbf{G}(s)$. This is usually in the form of its Bode plot (see this chapter's appendix in section 11). So rather than trying to directly certify equations (3), we see if there is a way to certify these objectives by direct inspection of the *loop function* $\mathbf{L}(s) = \mathbf{G}(s)\mathbf{K}(s)$. Procedures that do this are referred to as *loopshaping*.

Let us first look at the weighted sensitivity function $\mathbf{W}_p\mathbf{S}$ and assume that at a specified frequency $\omega_0$ we know $|\mathbf{W}_p(j\omega_0)\mathbf{S}(j\omega_0)| \approx 1$. Because these weights are well-posed, we already know that the disturbance rejection objective satisfies $|\mathbf{W}_\Delta(j\omega_0)\mathbf{T}(j\omega_0)| \approx 0$ and so we can ignore it.

We now translate our condition on the weighted sensitivity function into a constraint on the loop function, $\mathbf{L}(s)$. To do this note that

$$|\mathbf{W}_p(j\omega_0)\mathbf{S}(j\omega_0)| = \left|\frac{\mathbf{W}_p(j\omega_0)}{1+\mathbf{L}(j\omega_0)}\right| \leq \frac{|\mathbf{W}_p(j\omega_0)|}{|1+\mathbf{L}(j\omega_0)|}$$

Let us confine our attention to those frequencies $\omega_0$ such that $|\mathbf{L}(j\omega_0)| > 1$. In this case, the above bound on the weighted sensitivity function becomes

$$\begin{aligned}|\mathbf{W}_p(j\omega_0)\mathbf{S}(j\omega_0)| &\leq \frac{|\mathbf{W}_p(j\omega_0)|}{|1+\mathbf{L}(j\omega_0)|}\\ &\leq \frac{|\mathbf{W}_p(j\omega_0)|}{|\mathbf{L}(j\omega_0)|-1}\end{aligned}$$

In particular if the loop gain is large enough so that

$$|\mathbf{L}(j\omega_0)| > 1 + |\mathbf{W}_p(j\omega_0)| > |\mathbf{W}_p(j\omega_0)|$$

then this clearly means at this frequency, $\omega_0$, we have the weighted sensitivity constraint satisfied

$$|\mathbf{W}_p(j\omega_0)\mathbf{S}(j\omega_0)| < 1$$

In other words if $|\mathbf{L}(j\omega_0)| > |\mathbf{W}_p(j\omega_0)| > 1$, then the weighted condition on the sensitivity function is satisfied at this frequency. To enforce the tracking constraint, therefore, all we need to do is ensure the loop gain is greater than $|\mathbf{W}_p(j\omega_0)|$.

A similar argument applies to the disturbance rejection requirement that

$$|\mathbf{W}_\Delta(j\omega_0)\mathbf{T}(j\omega_0)| < 1.$$

Again because we know the weights are well-posed, we know $|\mathbf{W}_p(j\omega_0)| \approx 0$ and can therefore be ignored. In this case, we focus on frequencies, $\omega_0$, where $|\mathbf{L}(j\omega_0)| < 1$ and ask what condition must be placed on $\mathbf{L}$ to ensure the weighted condition on the complementary sensitivity function is satisfied. In this case we note that

$$\begin{aligned} |\mathbf{W}_\Delta(j\omega_0)\mathbf{T}(j\omega_0)| &= \frac{|\mathbf{W}_\Delta(j\omega_0)\mathbf{L}(j\omega_0)|}{1+\mathbf{L}(j\omega_0)|} \leq \frac{|\mathbf{W}_\Delta(j\omega_0)|\,|\mathbf{L}(j\omega_0)|}{|1+\mathbf{L}(j\omega_0)|} \\ &\leq \frac{|\mathbf{W}_\Delta(j\omega_0)|\,|\mathbf{L}(j\omega_0)|}{1-|\mathbf{L}(j\omega_0)|} \end{aligned}$$

where the last inequality holds because we confined our attention to frequencies where $|\mathbf{L}(j\omega_0)| < 1$. Now note that if the loop function's gain satisfies

$$|\mathbf{L}(j\omega_0)| \leq \frac{|\mathbf{W}_\Delta^{-1}(j\omega_0)|}{1-|\mathbf{W}_\Delta^{-1}(j\omega_0)|} \approx \frac{1}{|\mathbf{W}_\Delta(j\omega_0)|} < 1$$

then we must have

$$|\mathbf{W}_\Delta(j\omega_0)\mathbf{T}(j\omega_0)| < 1$$

In other words, if we make the loop gain $|\mathbf{L}(j\omega_0)|$ less than $\dfrac{1}{|\mathbf{W}_\Delta(j\omega_0)|}$ for frequencies where $|\mathbf{L}(j\omega_0)| < 1$ then the disturbance rejection requirement is met at this frequency.

Combining both of the preceding observations, we see that a controller that internally stabilizes the closed-loop system will enforce the well-posed weighted specifications provided the open-loop gain $|\mathbf{L}(j\omega_0)|$ is greater than $|\mathbf{W}_p(j\omega_0)|$ for frequencies where the tracking objective is active and $|\mathbf{L}(j\omega_0)| < \dfrac{1}{|\mathbf{W}_\Delta(j\omega_0)|}$ over frequencies where the disturbance rejection objective is active. Since one usually has a Bode plot of the loop function, we can directly check the extent to which these constraints are satisfied by the open loop plant $\mathbf{G}$ and then directly use that knowledge to introduce a controller $\mathbf{K}$ that forces the loop function $\mathbf{L} = \mathbf{GK}$ to satisfy the constraints on $\mathbf{L}$.

Recall that these loopshaping constraints are only valid if we know that the closed loop system is (internally) stable. In other words, we need to know if for any inputs $r, w \in \mathcal{L}_2$ we know that $u$ and $y$ are also in $\mathcal{L}_2$. To obtain conditions ensuring the internal stability of the closed loop system, let us consider the following irreducible factorizations of the plant and controller transfer functions

$$\mathbf{G}(s) = \frac{n_g(s)}{d_g(s)}, \quad \text{and} \quad \mathbf{K}(s) = \frac{n_k(s)}{d_k(s)}$$

The closed loop transfer function for this system may be written as

$$\mathbf{T}_{\text{cl}}(s) = \begin{bmatrix} \frac{n_g(s)n_k(s)}{d_g(s)d_k(s)+n_g(s)n_k(s)} & \frac{n_g(s)d_k(s)}{d_g(s)d_k(s)+n_g(s)n_k(s)} \\ \frac{n_k(s)d_g(s)}{d_g(s)d_k(s)+n_g(s)n_k(s)} & \frac{n_g(s)n_k(s)}{d_g(s)d_k(s)+n_g(s)n_k(s)} \end{bmatrix}$$

From the preceding equation we see that internal stability of this system is assured if the polynomial

$$p(s) = d_g(s)d_k(s) + n_g(s)n_k(s)$$

has no roots with positive real parts. One can show that if $\mathbf{L}(s)$ is a stable minimum phase transfer function then for every frequency $\omega_0$ we have

$$\angle\mathbf{L}(j\omega_0) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{d\ln|\mathbf{L}(j\omega)|}{d\nu} \ln\coth\frac{|\nu|}{2} d\nu \tag{4}$$

where the variable of integration is $\nu = \ln(\omega/\omega_0)$. The formula in equation (4) is known as the Bode Gain-Phase Formula. This formula asserts that the phase of a stable minimum phase transfer function is determined from its gain magnitude plot. In particular, let $\frac{d\ln|\mathbf{L}(j\nu)|}{d\nu} = c$ for frequencies about $\omega_0$, then the gain-phase formula reduces to

$$\angle\mathbf{L}(j\omega_0) \approx -\frac{c\pi}{2}$$

where we used the fact that $\ln\coth$ essentially looks like an impulse-like function centered at $\omega_0$.

This last relation means that for a gain-magnitude plot that has a sustained $20c$ dB/decade roll off around the frequency $\omega_0$ that the phase at $\omega_0$ will be $-90c$ degrees. We can therefore relate the roll off $|\mathbf{L}(j\omega_0)|$ about the frequency $\omega_0$ to the system's phase. In particular, if we choose $\omega_0$ to be the gain crossover frequency (i.e. the frequency, $\omega_0$ where $|\mathbf{L}(j\omega_0)| = 1$), then this means the *phase margin* of the system will be $180 - 90c$ degrees. From the Nyquist criterion (see this chapter's appendix in section 11) we know that a closed loop system is robustly stable to phase variations if its phase margin is positive and greater than $60^\circ$. From the gain-phase formula we therefore see that $c$ can only be about $1$ to ensure a healthy phase margin for the closed-loop system. In other words, to ensure the closed-loop map is robustly internally stable we need to ensure the loop function $\mathbf{L}$ exhibits a sustained roll off of no more than 20 dB/decade about the gain crossover frequency.

In view of our earlier discussion regarding tracking and disturbance rejection, we can now propose a loopshaping design method for the loop function $\mathbf{L}(s)$ that ensures internal stability while also meeting the two control objectives. As discussed above, this method presumes that $\mathbf{G}(s)$ and $\mathbf{K}(s)$ are irreducible stable minimum phase transfer functions so we can use Bode's gain-phase formula. We assume there exist two frequency intervals $[0, \omega_\ell)$ and $[\omega_u, \infty)$ such that $\omega_\ell < \omega_u$. Then the loopshaping procedure requires us to select a stable minimum phase controller $\mathbf{K}(s)$ such that

- *Tracking Requirement:* $|\mathbf{L}(j\omega)| > |\mathbf{W}_p(j\omega)|$ for frequencies where $\omega < \omega_\ell$ and $|\mathbf{L}(j\omega)| > 1$.
- *Disturbance Rejection:* $|\mathbf{L}(j\omega)| < |\mathbf{W}_\Delta^{-1}(j\omega)|$ for frequencies where $\omega > \omega_u$ and $|\mathbf{L}(j\omega)| < 1$.
- *Internal Stability:* For frequencies $\omega_\ell < \omega < \omega_u$ the loop function $|\mathbf{L}(j\omega)|$ exhibits a sustained 20 dB/decade roll off about the gain crossover frequency.

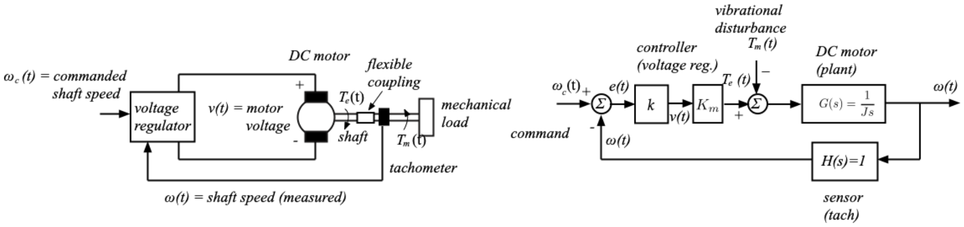


FIGURE 3. DC servo motor example

We now apply these rules to a DC servomotor system shown in Fig. 3. We have two control objectives the first is that the motor's speed, $\omega$, tracks a commanded speed $\omega_c(t) = \cos(2\pi f_c t)$ which is a unit amplitude sinusoid whose frequency is

$f_c$. We will assume that $f_c$ is anything less than $\frac{0.1}{2\pi}$ cycles/sec. The second objective is that vibrational disturbances $T_m$ that are injected at the motor shaft should generate small variations in the voltage output by the controller. These torques are generated by flexible coupling of the shaft to the load and are also sinusoidal disturbances with an amplitude of $0.1$ and frequency, $f_m$ that is greater than $\frac{10}{2\pi}$ cycles/sec. The left side of Fig. 3 shows the physical layout of the controlled system. The DC motor is shown as a circle in the middle of the picture with a shaft that is connected through a flexible coupling to the mechanical load. The voltage, $v$, over the motor's terminals generates a torque, $T_e$, that accelerates the motor shaft. We assume this torque is proportional to the applied voltage $v$ so that $T_e = K_m v$ where $K_m$ is a proportionality constant. The mechanical load, $T_m$, acts in opposition to $T_e$ to decelerate the motor shaft. A tachometer is used to measure the shaft speed and the sensor measurement is fed back to a voltage regulator that takes the error $e(t) = \omega_c(t) - \omega(t)$ between the commanded speed and the measured speed to generate the voltage on the motor terminals. Let us assume that this applied voltage is proportional to the tracking error, $e$, so that

$$v(t) = k(\omega_c(t) - \omega(t))$$

where $k$ is a controller parameter we select to achieve the desired control objectives.

From the preceding considerations we see that the motor speed, $\omega$, must satisfy the following ODE

$$\dot{\omega}(t) = T_e(t) - T_m(t) = K_m k(\omega_c - \omega) - T_m(t)$$

To make this example concrete, let us take $K_m = 1$ and we then derive the closed loop sensitivity functions for this system.

The sensitivity function from $\omega_c$ (reference input) to the error is

$$\mathbf{S}(s) = \frac{1}{1 + k/s}$$

The complementary sensitivity function from the load disturbance $T_m$ to the controller voltage, $v$ is

$$\mathbf{T}(s) = \frac{k/s}{1 + k/s}$$

The loop function associated with these two sensitivity functions is

$$\mathbf{L}(s) = \frac{k}{s}$$

We now formalize the control objectives informally described above

- The tracking specification requires the motor speed track the commanded input $\omega_c$. This command is a unit amplitude sinusoid with frequency $f_c < \frac{0.1}{2\pi}$ cycles/sec. Let us assume we want to reject this sinusoidal command by $40$ dB to obtain a $1\%$ tracking error. Based on this description we identify the performance weight as an ideal low pass filter

  $$|\mathbf{W}_p(j\omega)| = \begin{cases} 100 & \text{for } 0 < \omega < 0.1 \\ 0 & \text{otherwise} \end{cases}$$

  Note that in reality we would have selected a minimum phase stable transfer function that approximated this gain-magnitude.
- The disturbance rejection specification is that the disturbance torque, $T_m$, which is also a sinusoid with maximum amplitude $0.1$ and frequency greater than $\frac{10}{2\pi}$ cycles/seconds be rejected at the controller's voltage by $40$ dB also. Based on these requirements we then see that the gain magnitude of the weighting function should be a high pass filter that approximates

  $$|\mathbf{W}_\Delta(j\omega)| = \begin{cases} 100 & \text{for } \omega > 10 \\ 0 & \text{otherwise} \end{cases}$$

We now see that we should constrain the loop function $\mathbf{L}(s)$ to ensure

$$\begin{array}{rl} |\mathbf{W}_p(j\omega)\mathbf{S}(j\omega)| < 1 & \text{for } 0 < \omega < 0.1 \\ |\mathbf{W}_\Delta(j\omega)\mathbf{T}(j\omega)| < 1 & \text{for } \omega > 10 \end{array}$$

Based on earlier discussion these weighted constraints on the sensitivity function will be achieved if the closed loop maps are internally stable and the loop function satisfies

$$\begin{array}{rl} |\mathbf{L}(j\omega)| > |\mathbf{W}_p(j\omega)| & \text{for } 0 < \omega < 0.1 \\ |\mathbf{L}(j\omega)| < \frac{1}{|\mathbf{W}_\Delta(j\omega)|} & \text{for } \omega > 10 \end{array}$$

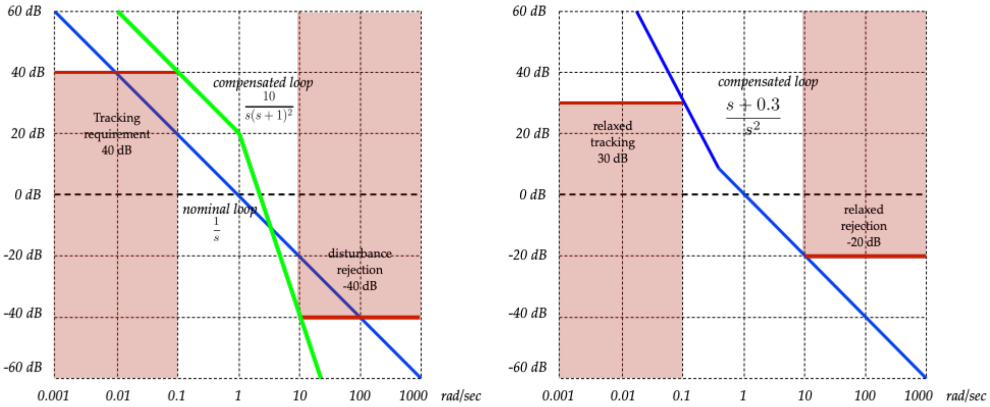


FIGURE 4. Loop Function for DC servo (left - a ) original design that does not preserve internal stability (right - b) design with relaxed specifications that preserve internal stability

Both of these constraints are shown by the shaded red region in the Bode plot in Fig. 4a. For a nominal control gain $k = 1$, the loop function becomes $\mathbf{L}(s) = \frac{1}{s}$ whose gain magnitude is plotted in Fig. 4a. What should be apparent here is that the nominal loop shape does not satisfy the constraints. If we are to design a controller that achieves the objectives we need to reshape the loop function so it satisfies the constraints.

In particular we can see that the requirements will never be met by simply adjusting the gain $k$. So instead we propose using a frequency dependent controller, $\mathbf{K}(s)$. Note that if we let

$$\mathbf{K}(s) = \frac{10}{(s+1)^2}$$

then the resulting loop shape does satisfy the high and low frequency design constraints. Essentially what we have done is use the controller to reshape the loop function in a manner that enforces the desired bounds on the sensitivity function. This method, therefore, is called loopshaping.

However, our prior work presumes the closed loop map is stable and this is not actually the case. In particular, if we look at the nominal loop function, $\mathbf{L}(s)$, it is easy to see that both sensitivity functions

$$\mathbf{S}(s) = \frac{1}{1+1/s} \quad \text{and} \quad \mathbf{T}(s) = \frac{1/s}{1+1/s}$$

will be stable. But this is not the case for our reshaped loop function $\mathbf{L}(s) = \frac{10}{s(s+1)^2}$. For this loop function the sensitivity functions are

$$\mathbf{S}(s) = \frac{s(s+1)^2}{s(s+1)^2+10} \quad \text{and} \quad \mathbf{T}(s) = \frac{10}{s(s+1)^2+10}$$

The denominator polynomial has roots at $-2.8675$ and $0.4337 \pm 1.8154j$ which is clearly unstable.

This is why we also need to enforce the third requirement that the ”slope” of the loop function be around 20dB/decade across the gain crossover point. Clearly that requirement is not satisfied by our reshaped loop. However, we can readily see that this requirement can never be met with the specifications we’ve been given. The change in the loop function has to be 80 dB over 2 decades to meet the high and low frequency requirements. This is an average roll off of 40 dB/decade.

One way of addressing the impasse is to relax the requirements. So we reduce the requirements so that $|\mathbf{L}(j\omega)|$ is greater than 30 dB for $\omega < 0.1$ and we relax the disturbance rejection constraints so that $|\mathbf{L}(j\omega)|$ is less than $-20$ dB for $\omega > 10$. We choose a loop shape that rolls off at 40 dB/decade for low frequencies and then uses a zero to shift to a 20 dB/decade roll off just before the gain crossover frequency at 1 rad/sec. The resulting loop shape is shown in Fig. 4b with a loop function of

$$\mathbf{L}(s) = \frac{s+0.3}{s^2}$$

The sensitivity functions for this system are

$$\mathbf{S}(s) = \frac{s^2}{s^2+s+0.3} \quad \text{and} \quad \mathbf{T}(s) = \frac{s+0.3}{s^2+s+0.3}$$

both of which have poles at $-0.5 \pm .2236j$. If we look at the Bode plot, we see we have a roll off of about 20 dB/decade about the gain cross over at 1 rad/sec. So this is an internally stable controller that meets the relaxed design requirements.

**Loopshaping Blocks:** In loopshaping we take the original plant, $\mathbf{G}(s)$ and place it in series with a controller $\mathbf{K}(s)$ that alters the loop function in a manner to meet the desired loopshaping conditions. This controller or desired loop shape can be built up in an incremental manner using *loopshaping blocks* that most undergrad control students have seen before. This subsection reviews these blocks and shows how they are used to shape the loop function in a systematic manner.

The most elementary loopshaping block is the proportional gain,

$$\mathbf{K}(s) = K$$

The effect of this block is to raise the loop gain at all frequencies. This can increase the system bandwidth, thereby making the system "faster", but by increasing bandwidth it can also make the system more sensitive to high frequency noise.

The next basic loopshaping block is the *lag network*. The transfer function for the lag network is

$$\mathbf{K}(s) = \frac{b}{a}\left(\frac{s+a}{s+b}\right)$$

where $a > b \geq 0$. The action of the lag network is to raise the loop gain at frequencies below $b$ (the compensator's pole) and add phase in the transition region between $b$ and $a$. This tends to increase the low frequency loop gain which implies better rejection of low frequency disturbances, but the additional phase lag can destabilize the system if it occurs around the gain crossover frequency. As a result lag compensators are usually design so the compensator's pole and zero are placed about a decade below the gain cross-over frequency. Closely related to the lag compensator is the proportional-integral (PI) compensator. The transfer function of the PI controller is

$$\mathbf{K}(s) = K_p + \frac{K_I}{s} = \frac{sK_p + K_I}{s}$$

The PI controller may therefore be viewed as a lag compensator whose pole is at the origin.

The lead compensator is another widely used loopshaping block. It has the transfer function

$$\mathbf{K}(s) = \frac{b}{a}\left(\frac{s+a}{s+b}\right)$$

where $b > a \geq 0$. The action of the lead network is to raise the loop gain at frequencies above $b$ (the compensator's pole) and add phase lead in the transition region between $a$ and $b$. This compensator will increase the system's bandwidth, thereby resulting in a faster system and when the compensator adds phase lead around the gain crossover frequency it can improve the system's overall phase margin. The larger high frequency gain, however, also has the effect of making the system more sensitive to noise and modeling error. Lead networks are used primarily to add phase margin to a system. For this purpose there is a well known design procedure in which the designer first identifies the frequencies over which phase needs to be added and then places the transition region of the lead network in that interval.

The phase lead, $\phi_\ell$, that the lead network adds will be

$$\sin(\phi_\ell) = \frac{m-1}{m+1}, \quad \text{where} \quad m = \frac{b}{a} = \frac{1+\sin\phi_\ell}{1-\sin(\phi_\ell)}$$

This phase lead will be added at the frequency

$$\omega_\ell = \sqrt{ab} = a\sqrt{m}$$

The preceding equations can be used in a systematic way to design lead networks. Proportional/derivative controllers are special cases of lead networks. The transfer function for a PD controller is

$$K(s) = K_p + \frac{K_D s}{s+p} = \frac{(K_p + K_D)s + K_p p}{s+p}$$

**Lead-Lag Loopshaping Block Example:** Based on our earlier discussion, loopshaping assumes there are two disjoint frequency intervals $[0, \omega_\ell]$ and $[\omega_u, \infty)$ where $\omega_\ell < \omega_u$. The weighting systems are chosen so that

- $|\mathbf{W}_p(j\omega)| > 1$ and $|\mathbf{W}_\Delta(j\omega)| \ll 1$ for all $\omega \in [0, \omega_\ell)$
- $|\mathbf{W}_\Delta(j\omega)| > 1$ and $|\mathbf{W}_p(j\omega)| \ll 1$ for all $\omega \in [\omega_u, \infty)$

to ensure the specifications are well-posed. The first interval $[0, \omega_\ell)$ is the set of frequencies over which the *tracking objective* is to be enforced. For convenience we call it the *performance* region (P-region) of the frequency space. The second interval $[\omega_u, \infty)$ is the set of frequencies over which the *disturbance rejection objective* is to be enforced, so we call it the *rejection* region (R-region). Finally, those frequencies in the interval $[\omega_\ell, \omega_u]$ form the *transition region* (T-region) of the frequency space.

Loopshaping is done with respect to the three regions identified above. In particular, we need to ”build” a stable minimum phase controller $\mathbf{K}(s)$ such that

- *Tracking Objective:* $|\mathbf{L}(j\omega)| > |\mathbf{W}_p(j\omega)|$ for all $\omega$ in the P-region $([0, \omega_\ell))$,
- *Disturbance Rejection Objective:* $|\mathbf{L}(j\omega)| < |\mathbf{W}_\Delta^{-1}(j\omega)|$ for all $\omega$ in the R-region $([\omega_u, \infty))$,
- *Closed-loop Stability:* $|\mathbf{L}(j\omega)|$ has an average roll off of 20 dB/decade about the gain crossover frequency $\omega_g$ in the T-region.

then we know there is a good chance that the preceding performance requirements will be satisfied and that the resulting closed-loop sensitivity functions $\mathbf{S}(s)$ and $\mathbf{T}(s)$ will be stable. This design methodology is approximate since some judgement is required in selecting $\omega_\ell$ and $\omega_u$ that define the P frequency region and R frequency region. There is also some judgement involved in determining how much one can relax the 20 dB/decade roll off in the T-region before the closed-loop system is no longer stable.

We are going to use the loopshaping blocks identified above to help find a controller $\mathbf{K}(s)$ that meets the requirements. The plant is a voltage regulator whose circuit diagram is shown in Fig. 5. This circuit uses an operational amplifier[1] to generate a correction voltage on the base of a power transistor, Q1, transferring the energy in an unregulated voltage source to an RC load. The voltage source attached to Q1’s emitter is an unregulated device such as a battery whose voltage, $V_{\text{in}}(t)$, varies with the battery’s state of charge. The operational amplifier’s output voltage, $V_a(t)$, is equal to the difference of a nominal set point voltage $V_{\text{nom}}(t)$. and the voltage $V_{\text{out}}(t)$ over the RC load, namely the voltage from the transistor’s collector terminal to ground. This means that $V_a(t)$ acts as an error signal measuring how much the load voltage differs from the set point voltage. By applying the error voltage $V_a(t)$ to the transistor’s base, one can increase or decrease the current $I_a(t)$ delivered to the load, thereby, providing a feedback mechanism which keeps $V_{\text{out(t)}}$ close to $V_{\text{nom}}(t)$. In other words, this circuit acts to “regulate” the load

[1] J.K. Roberge, *Operational amplifiers: theory and practice*, John Wiley $ Sons, NY, 1975.

voltage about the reference nominal voltage $V_{\text{nom}}(t)$ with respect to variations (i.e. disturbances) in the unregulated source's voltage level, $V_{\text{in}}(t)$.

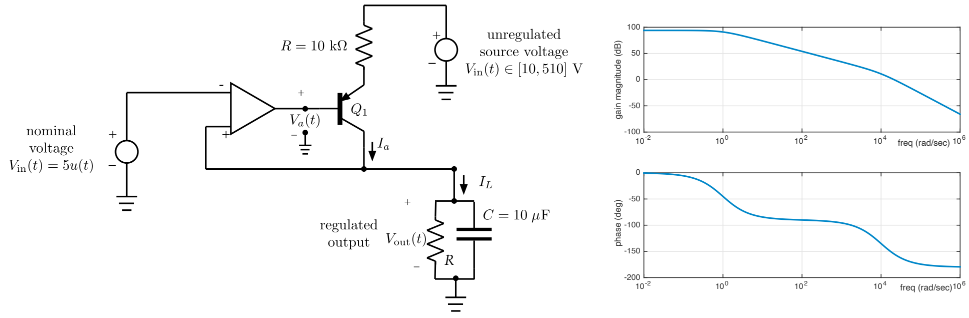


FIGURE 5. Op-amp based Voltage Regulator

The performance of this circuit is defined with respect to its *line regulation*

$$\%\text{ line regulation} = 100\frac{V_{\text{out,max}} - V_{\text{out,min}}}{V_{\text{nom}}}$$

where the variation in $V_{\text{out}}(t)$, is generated by bounded variations of the source voltage, $V_{\text{in}}$. In this case, we assume the unregulated source has a voltage

$$V_{\text{in}}(t) \in [10, 510]\text{ V}$$

We take $V_{\text{nom}}(t) = 5u(t)$ V where $u$ is a unit step function, thereby simulating the turning on of the regulator at time $t = 0$.

The issue we have with this circuit lies in the operational amplifier. Introductory circuit textbooks often take the operational amplifier's input/output relationship to be

$$V_{\text{a}}(t) = a(V_{\text{out}}(t) - V_{\text{nom}})$$

where $a$ is a real and very large positive constant. But in reality, operational amplifiers are linear dynamical systems. The frequency response for this particular op-amp is shown on the right side of Fig. 5. This means that the input/output relationship in the Laplace domain is

$$\widehat{V}_{\text{a}}(s) = \mathbf{a}(s)(\widehat{V}_{\text{out}}(s) - \widehat{V}_{\text{nom}}(s))$$

The op-amp's transfer function can be deduced by an inspection of the Bode plot in Fig. 5 to be

$$\mathbf{a}(s) = \frac{5 \times 10^4}{(s+1)(10^{-4}s+1)}$$

Let us see how well this regulator performs with respect to line regulation for the circuit values shown in the figure. For the moment we'll ignore the hum and simply examine how the output voltage varies when $V_{\text{in}}$ is a constant between 10 to 510 volts. The output voltage for a given constant $V_{\text{in}}(t)$ for "large" $t$ will be

$$V_{\text{out}} = V_{\text{in}} - (\mathbf{a}(0))(5 - V_{\text{out}}) \quad \Rightarrow \quad V_{\text{out}} = \frac{V_{\text{in}} - 5\mathbf{a}(0)}{1 - \mathbf{a}(0)}$$

Since $V_{\text{in}}$ is a constant between 10 and 510 V, and $\mathbf{a}(0) = 5 \times 10^4$, this means

$$\begin{aligned} V_{\text{out,max}} &= \frac{10 - 25 \times 10^4}{1 - 5 \times 10^4} = 4.9999 \text{ V} \\ V_{\text{out,min}} &= \frac{510 - 25 \times 10^4}{1 - 5 \times 10^4} = 4.9899 \text{ V} \end{aligned}$$

So the percent line regulation is

$$\begin{aligned} \text{percent line regulation} &= 100 \times \frac{V_{\text{out,max}} - V_{\text{out,min}}}{V_{\text{nom}}} \\ &= 100 \times \frac{4.9999 - 4.9899}{5} = 0.2 \text{ percent regulation} \end{aligned}$$

This preceding analysis assumed that $V_{\text{in}}(t)$ was a constant somewhere between 10 and 510 volts. But in reality it will be a time varying function, which means we need to take into account the dynamics introduced by the RC load and the op-amp's transfer function, $\mathbf{a}(s)$. Let us look at the circuit schematic and apply KVL along the path from the unregulated source to ground through the transistor and load.

$$\widehat{V}_{\text{in}}(s) = \widehat{I}_a(s)R + \mathbf{a}(s)(\widehat{V}_{\text{nom}}(s) - \widehat{V}_{\text{out}}(s))$$

The current $\widehat{I}_a(s)$ is equal to the load current, since the op-amp input currents are very small.

$$\widehat{I}_a(s) = \widehat{I}_{\text{L}}(s) = \widehat{V}_{\text{out}}(s)\frac{RCs+1}{R}$$

Inserting this into the first equation gives

$$\widehat{V}_{\text{in}}(s) = (RCs+1)\widehat{V}_{\text{out}}(s) + \mathbf{a}(s)(\widehat{V}_{\text{nom}}(s) - \widehat{V}_{\text{out}}(s))$$

which we rewrite as

$$\widehat{V}_{\text{out}}(s) = \frac{1}{RCs+1}\left(\widehat{V}_{\text{in}}(s) - \mathbf{a}(s)(\widehat{V}_{\text{nom}}(s) - \widehat{V}_{\text{out}}(s))\right)$$

This last equation serves as the basis for a block diagram of the circuit that is shown in Fig. 6(a).

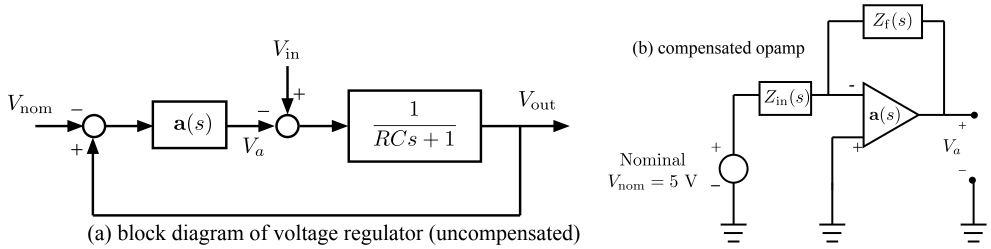


FIGURE 6. (a) Block Diagram of Op-amp Based Voltage Regulator (b) inverted compensated op-amp

This diagram clearly shows that our voltage regulator is a unity gain SISO control system just like we had drawn in Fig. 1. In this case the controller is $\mathbf{a}(s)$, the operational amplifier, the plant is $\frac{1}{RCs+1}$. The input voltage $V_{\text{nom}}$ is the reference input we want to track and the disturbance we want to reject is the unregulated voltage, $V_{\text{in}}(t)$.

Using the values in Fig. 5 ($R = 10^4$ ohms and $C = 10\,\mu$F) we obtain the loop function

$$\mathbf{L}(s) = \frac{\mathbf{a}(s)}{RCs+1} = \frac{5\times 10^4}{(s+1)(10^{-4}s+1)(10^{-1}s+1)}$$

The Bode plot for this loop function is plotted in Fig. 7, where we've marked the gain crossover frequency and associated phase margin. This system has a gain crossover of 332 rad/sec and its phase margin is $-3°$. Since the phase margin is negative, we can conclude that this closed-loop system is unstable. It means that the circuit as drawn will not work and that the line regulation computed in the preceding paragraph is meaningless since the circuit is unstable.

In the block diagram of Fig. 6(a), the plant is $\mathbf{G}(s) = \frac{1}{RCs+1}$ and the controller is the op-amp. Without compensation, the op-amp's transfer function, $\mathbf{a}(s)$, was shown in the preceding paragraph to destabilize the circuit. So obviously, we need

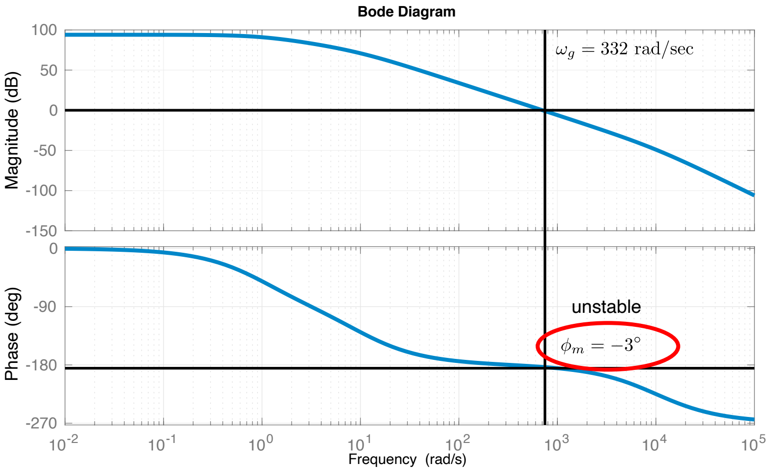


FIGURE 7. Bode plot of voltage regulator's loop function (uncompensated)

to compensate the op-amp to change its transfer function. The standard compensation scheme is shown in Fig. 6(b), where passive RC networks with impedances $Z_{\rm f}(s)$ and $Z_{\rm in}(s)$ are connected to the op-amp as shown. Assuming the op-amp's gain is sufficiently large, then the compensated transfer function may be taken as being

$$\mathbf{a}_c(s) = -\frac{Z_{\rm f}(s)}{Z_{\rm in}(s)}$$

which is something that we can build to provide any set of poles and zeros. So our problem now involves identifying the desired compensated op-amp transfer function, $\mathbf{a}_c(s)$, that is needed to realize a stable closed-loop system meeting a pair of tracking and disturbance rejection objectives.

We still need to specify the control objectives. In general, these objectives are stated in terms of desirable steady-state and transient behaviors. In particular, we will assume the following

- Achieve $0.01\%$ line regulation assuming $V_{\rm nom}(t) = 5$ and $V_{\rm in}(t)$ is a constant voltage taking value in the interval $[10, 510]$ V.
- Let $V_{\rm in}(t) = V_0 + \sin(\omega_d t)$ and reject the hum $(\sin(\omega_d t))$ where $\omega_d > 10^4$ rad/sec at the op-amp's output, $V_a(t)$ by 40 dB

- If $V_{\text{nom}} = 5u(t)$ ($u$ is a step function), then make sure $V_{\text{out}}(t)$ settles to 5 volts with at most $20\%$ of overshoot.

The first objective is a tracking objective in the presence of a constant, though unknown $V_{\text{in}}$. The second objective is a disturbance rejection requirement that a "hum" on the line be attenuated to 1 percent ($-40$ dB) of its maximum amplitude. The third objective is a transient response requirement which is not really handled by constraints on the gain magnitude of the loop function. This constraint is handled, indirectly, through the phase margin and where the gain crossover frequency occurs.

The tracking objective is for line regulation of a constant $V_{\text{in}}$ lying within 10 to 510 volts assuming a constant 5 volt reference voltage, $V_{\text{nom}}$. so we have

$$\begin{aligned}
\text{desired percent line regulation} &= 0.01 \\
&\geq 100 \times \left( \frac{V_{\text{out,max}} - V_{\text{out,min}}}{V_{\text{nom}}} \right) \\
&= \frac{100}{5} \left( \frac{10 - 5\mathbf{a}_c(0) - 510 + 5\mathbf{a}_c(0)}{1 - \mathbf{a}_c(0)} \right) \\
&= \frac{(20)(-500)}{1 - \mathbf{a}_c(0)}
\end{aligned}$$

which places a constraint on the DC gain of the compensated op-amp's transfer function,

$$\mathbf{a}_c(0) \geq 100 \times 2 \times 500 - 1 \approx 10^5$$

So the DC gain of $a_c(0)$ should be $10^5$. Since the DC gain of the plant, $\frac{1}{RCs+1}$, is 1, this means we need the loop gain, $|\mathbf{L}(j0)|$ to be greater than $20\log_{10}(10^5) = 100$ dB. Note that this is only a constraint at DC.

We will enlarge the P-region (tracking) by selecting a range of frequencies from 0 to $\omega_\ell$ for which the loop gain should be greater than 100 dB. Note that since we require disturbances with frequencies greater than $10^4$ rad/sec to be attenuated by 40 dB, this means that the R-region (rejection) is the interval $[10^4, \infty)$ and so the transition (T) region is $[\omega_\ell, 10^4]$. The sustained roll off over the transition region will be set at approximately 30 dB/decade to ensure closed-loop stability. So the total change in loop gain over the transition region is $100 + 40 = 140$ dB and this suggests the transition region should be around 5 decades long. This means $\omega_\ell$

should be chosen to be $10^{-1}$ rad/sec. With the P-region now set, we can define the weighting systems as

$$|\mathbf{W}_p(j\omega)| = \begin{cases} 100 \text{ dB} & \text{for } 0 \leq \omega < 0.1 \text{ rad/sec} \\ -\infty \text{ dB} & \text{otherwise} \end{cases}$$

and the R-region's function will be

$$|\mathbf{W}_\Delta(j\omega)| = \begin{cases} 40 \text{ dB} & \text{for } \omega > 10^4 \text{ rad/sec} \\ -\infty \text{ dB} & \text{otherwise} \end{cases}$$

We are now ready to design the compensated loop function.

Note that since the DC gain of the uncompensated op-amp is $5 \times 10^4$ and since the RC circuit's DC gain is 1, we will need to increase the DC gain of the op-amp from $50,000$ to $100,000$. This corresponds to a proportional gain, $K_p = 2$. The compensated loop function is now

$$\begin{aligned} \mathbf{L}(s) &= 2\mathbf{a}(s) \\ &= \frac{100000}{10^{-5}s^3 + 0.1001s^2 + 1.1s + 1} \end{aligned}$$

The Bode plot for this compensated loop function is shown in Fig. 8.

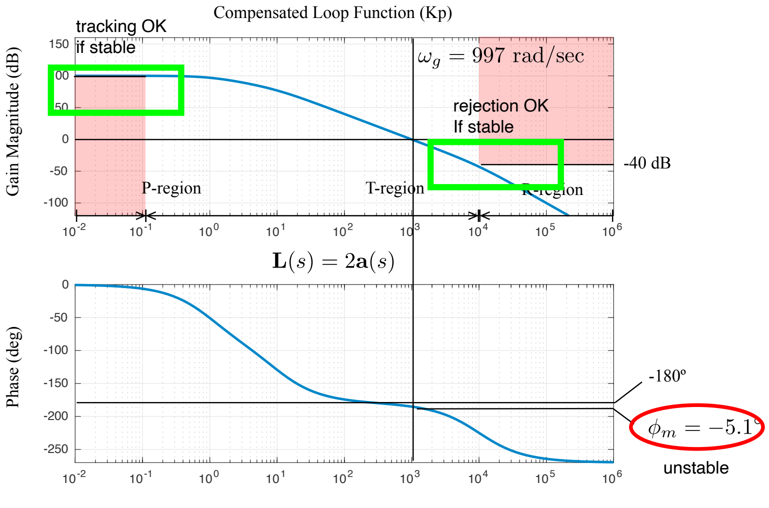


FIGURE 8. Bode plot of voltage regulator's loop function with proportional gain block, $K_p = 2$

What this figure shows is that the tracking constraint and the disturbance rejection constraints are both met. The loop function has a gain crossover at 997 rad/sec with a phase margin of $-5.1^\circ$. Since this phase margin is negative, the closed-loop system would be unstable. We therefore need to add compensation that can address the negative phase margin. We propose using a lead network for this purpose.

Because the lead network will shift the gain crossover point to the right, we propose designing our lead network so it adds phase lead at 3000 rad/sec. At this point the phase of the uncompensated loop is about $-200^\circ$, which means we would have to add $20+50=70^\circ$ of phase lead at this frequency so the lead compensated loop function has a phase margin of $50^\circ$. The compensated loop function is now

$$\begin{aligned}
\mathbf{L}(s) &= 2\mathbf{K}_{\text{lead}}(s)\mathbf{a}(s) \\
&= 2\left(\frac{32.16s + 1.7\times 10^4}{s+1.7\times 10^4}\right)\mathbf{a}(s) \\
&= \frac{3.216\times 10^6 s + 1.8 - 1\times 10^9}{10^{-5}s^4 + 0.2702s^3 + 1704s^2 + 1.872\times 10^4 s + 1.701\times 10^4}
\end{aligned}$$

The Bode plot for this lead compensated loop is in Fig. 9. The figure shows that the tracking objective is still satisfied, that the phase margin has increased to $57.6^\circ$ with a gain crossover at 1914 rad/sec. But the lead network violates the disturbance rejection objective by providing only about $-20$ dB of attenuation at $10^4$ rad/sec.

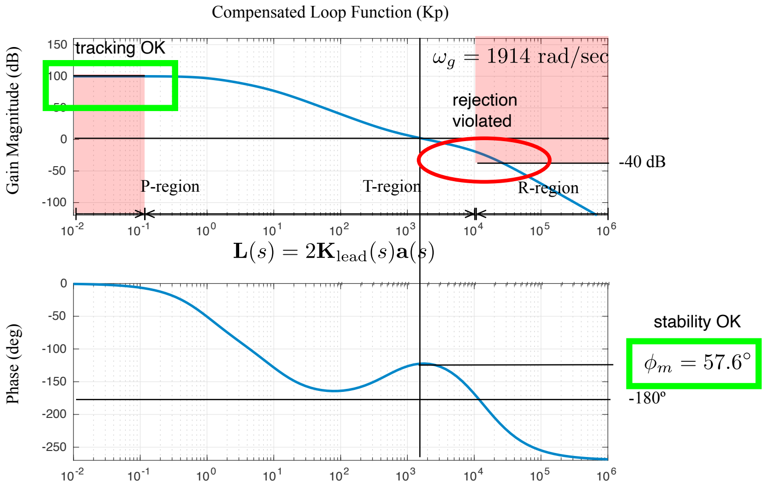


FIGURE 9. Bode plot of voltage regulator's loop function with proportional gain, $K_p = 2$, and lead compensator, $\mathbf{K}_{\text{lead}}(s) = \frac{32.16s+1.7\times 10^4}{s+1.7\times 10^4}$).

So we need to find a way to reduce this by an additional 20 dB. This may be done by reducing the loop gain by a factor of 10. So we modify our loop function to

$$\mathbf{L}(s) \quad = \quad \frac{2}{10}\mathbf{K}_{\mathrm{lead}}(s)\mathbf{a}(s)$$

The Bode plot for this loop function is shown in Fig. 10. What we see here is that this change led to a violation of the tracking constraint and a reduced though positive phase margin of 31.8° at a gain crossover frequency of 345 rad/sec.

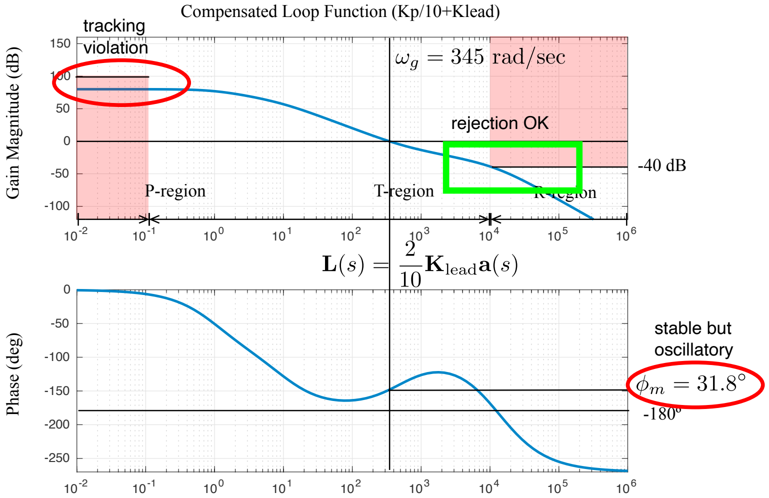


FIGURE 10. Bode plot of voltage regulator's loop function with reduced proportional gain, $K_p = 2/10$, and lead compensator, $\mathbf{K}_{\mathrm{lead}}(s) = \frac{32.16s+1.7\times10^4}{s+1.7\times10^4}$

This reduction in phase margin, however, occurs because the gain crossover moved to 345 rad/sec. This suggests we should have designed our lead network to place the phase lead at this frequency. So we redesign the lead network to provide 60° of phase lead at 600 rad/sec. The loop function obtained using the redesigned lead network is now

$$\begin{aligned}\mathbf{L}(s) \quad &= \quad \left(\frac{2}{10}\right)\mathbf{K}_{\mathrm{lead1}}(s)\mathbf{a}(s) \\ &= \quad \frac{2}{10}\left(\frac{13.93s+2239}{s+2239}\right)\mathbf{a}(s)\end{aligned}$$

The Bode plot for the resulting loop function is shown in Fig. 11. What we see here is that we have a good positive phase margin of 57.3° at a gain crossover frequency of 618 rad/sec. The disturbance rejection requirement is still satisfied, but we still violate the tracking objective.

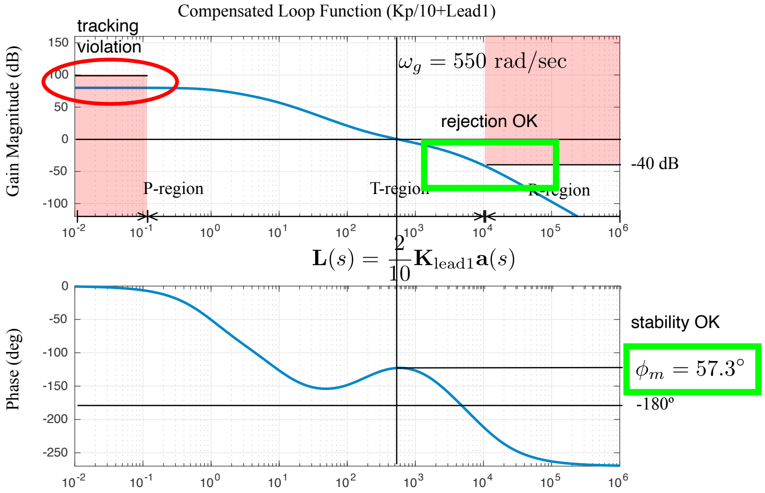


FIGURE 11. Bode plot of voltage regulator's loop function with reduced proportional gain, $K_p = 2/10$, and redesigned lead compensator, $\mathbf{K}_{\text{lead1}}(s) = \frac{13.93s+2239}{s+2239}$

We can improve tracking performance through a lag network that is placed a decade below the gain crossover frequency (800 rad/sec). This lag network will need to provide an additional 20 dB of gain at frequencies below 0.1 rad/sec. Both conditions are satisfied by a lag network whose zero is at 8 rad/sec and whose pole is at 0.8 rad/sec. The loop function now becomes

$$\begin{aligned}
\mathbf{L}(s) &= \frac{2}{10}\mathbf{K}_{\text{lead1}}(s)\mathbf{K}_{\text{lag}}(s)\mathbf{a}(s) \\
&= \frac{2}{10}\left(\frac{13.93s + 2239}{s + 2239}\right)\left(\frac{s+8}{s+0.8}\right)\mathbf{a}(s)
\end{aligned}$$

The Bode plot for this lead-lag compensated loop function is shown in Fig. 12. This figure shows that the tracking and disturbance rejection objectives are met. The phase margin is now 56.8° at 618 rad/sec. So this loop shape satisfies the tracking, rejection, and stability objectives.

The last requirement is that we settle quickly with a 20% overshoot. Loop-shaping can indirectly effect these transient metrics through the phase margin. A larger margin will result in smaller overshoot and the settling time can be reduced

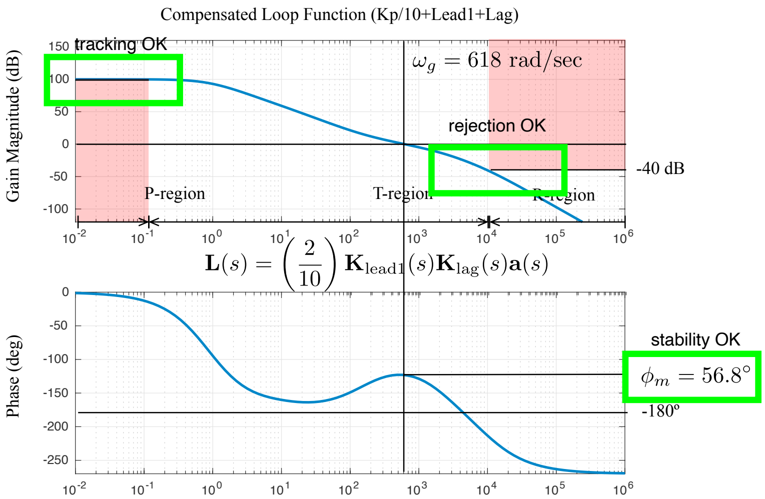


FIGURE 12. Bode plot of voltage regulator's loop function with reduced proportional gain, $K_p = 2/10$, redesigned lead-lag compensator redesigned lead compensator, $\mathbf{K}_{\text{lead1}}(s) = \frac{13.93s+2239}{s+2239}$, and lag compensator, $\mathbf{K}_{\text{lag}}(s) = \frac{s+8}{s+0.8}$

by increasing the gain crossover frequency. Fig. 13 shows the step response for the closed loop system. We see that the overshoot condition is satisfied, and so we are lucky in the sense that we don't have to do any redesign to try and meet the transient response requirements.

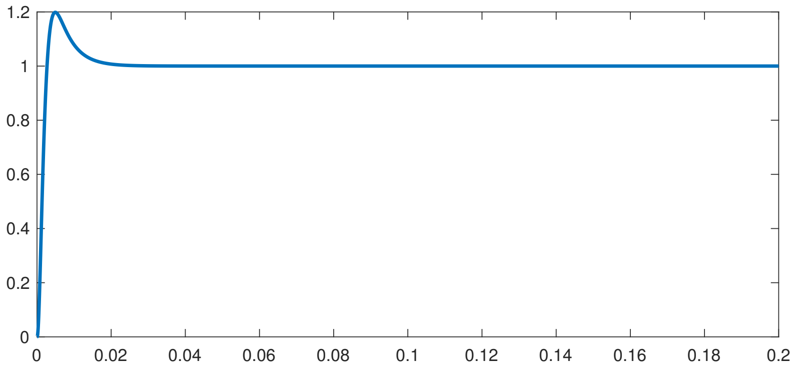


FIGURE 13. Closed loop step response of final design

## 7. State Space Modeling of LTI Systems

Pole placement and frequency response methods are often used in the design of SISO control systems, but they are difficult to use when the LTI system has multiple inputs and multiple outputs (MIMO). MIMO LTI control systems are better designed using state space models of the system. A *state space realization* is a concrete representation of an LTI system $\mathbf{G}$ formed from the following set of equations

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}w(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}w(t) \end{aligned}$$

The signal $x : \mathbb{R} \rightarrow \mathbb{R}^n$ is an internal signals called the system's *state*. The input is the signal $w : \mathbb{R} \rightarrow \mathbb{R}^m$ and the output is $y : \mathbb{R} \rightarrow \mathbb{R}^p$. The other objects $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ are real valued matrices of appropriate dimensions. Such state space realizations are often written in a *packed matrix notation*

$$\mathbf{G} \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]$$

Note that if we know the initial state, $x(0) = x_0$, then if we take the Laplace transform of the above equations we obtain

$$\begin{aligned} sX(s) - x(0) &= \mathbf{A}X(s) + \mathbf{B}W(s) \\ Y(s) &= \mathbf{C}X(s) + \mathbf{D}W(s) \end{aligned}$$

We can then solve the first equation for $X(s)$ to obtain

$$X(s) = (s\mathbf{I} - \mathbf{A})^{-1}(\mathbf{B}W(s) + x(0))$$

Inserting this expression into the second state equation yields,

$$\begin{aligned} Y(s) &= \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}W(s) + \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}x(0) \\ &= \mathbf{G}(s)W(s) + \mathbf{G}_0(s)x_0 \end{aligned}$$

The first term is the product of the *transfer function*, $\mathbf{G}(s)$, with the Laplace transform of the input $W(s)$. This term represents the system's zero-state or forced response to an external input $w$. The second term multiplies $\mathbf{G}_0$ with the Laplace transform of an impulse $\delta(t)x_0$ applied at time 0. This term therefore represents the zero-input or natural response of the system with respect to non-zero initial conditions. What we have just shown is that any state space realization $\mathbf{G} \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]$ has the transfer function

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

Unlike our earlier characterization of a SISO system's transfer function, we can see that for MIMO system's, the transfer function $\mathbf{G}(s)$ will be a matrix of rational functions. The $ij$th component of that matrix being a transfer function from the $j$th input to the $i$th output of the system, assuming all other inputs are zero.

Every state space realization has a unique transfer function. However each transfer function may have an infinite number of possible state space realizations. In particular, given a transfer function $\mathbf{G}(s)$, let us assume that it has a state space realization

$$\mathbf{G} \overset{\mathrm{s}}{=} \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]$$

Let $\mathbf{Q}$ be any nonsingular square matrix with the same dimensions as $\mathbf{A}$. This means there exists a matrix $\mathbf{Q}^{-1}$ such that $\mathbf{Q}^{-1}\mathbf{Q} = \mathbf{I}$. If $x \in \mathbb{R}^n$ is the state vector for $\mathbf{G}$, we can create new "state, $z = \mathbf{Q}x$ by passing $x$ through the linear transformation $\mathbf{Q}$. This would also mean that $x = \mathbf{Q}^{-1}z$ and if we take the time derivative of $x$ we get

$$\begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}w \\ &= \mathbf{A}\mathbf{Q}^{-1}z + \mathbf{B}w \\ &= \mathbf{Q}^{-1}\dot{z} \end{aligned}$$

the last two equations can be rewritten as

$$\dot{z} = \mathbf{Q}\mathbf{A}\mathbf{Q}^{-1}z + \mathbf{Q}\mathbf{B}w$$

Note also that

$$\begin{aligned} y &= \mathbf{C}x + \mathbf{D}w \\ &= \mathbf{C}\mathbf{Q}^{-1}z + \mathbf{D}w \end{aligned}$$

This gives rise to the following state space realization of the $z$-system

$$\begin{aligned} \dot{z} &= \mathbf{QAQ}^{-1}z + \mathbf{QB}w \\ y &= \mathbf{CQ}^{-1}z + \mathbf{D}w \end{aligned}$$

We claim that the state space realization $\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]$ has the same transfer fucntion as the realization $\left[\begin{array}{c|c} \mathbf{QAQ}^{-1} & \mathbf{QB} \\ \hline \mathbf{CQ}^{-1} & \mathbf{D} \end{array}\right]$. This assertion can be directly verified by computing the transfer function for both realizations and using the fact that $\mathbf{Q}^{-1}\mathbf{Q} = \mathbf{I}$. Note that since this is true for *any* nonsingular $\mathbf{Q}$ we chose, so we have an infinite number of realizations associated with a given transfer function. These realizations are indistinguishable from each other on the basis of the system's input/output behavior, but different realizations may be more convenient to work with from an analytical standpoint. Such realizations are said to be canonical and several such important realizations are discussed in linear systems textbooks. Commonly used canonical realizations are the "companion" forms, modal forms, and balanced realizations. The companion matrix realizations are notable because of their close relationship to the controllability and observability of the given realization. Modal realizations where the $\mathbf{A}$ matrix is in Jordan form are useful because of their numerical stability and their clear identification of the fundamental modes of the system. Balanced realizations are important because of the role they play in model reduction. We are assuming that the reader is already somewhat familiar with some of these canonical realizations. In the next section, we discuss how they are used in developing observer-based control laws from the system's state-space realizations.

## 8. State Feedback

Consider the LTI system

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}x(t) \end{aligned}$$

We say this realization can have its eigenvalues arbitrarily assigned by state feedback if for any $n$th order polynomial, $\alpha_d(s)$, there is a matrix $\mathbf{F}$ such that the eigenvalues of $\mathbf{A}+\mathbf{BF}$ are the roots of the polynomial equation, $\alpha_d(s)=0$. In other words there is a state feedback law,

$$u(t)=\mathbf{F}x(t)$$

such that when this is applied to our system we obtain the closed-loop system

$$\begin{aligned}\dot{x}(t) &= (\mathbf{A}+\mathbf{BF})x(t)\\ y(t) &= \mathbf{C}x(t)\end{aligned}$$

A necessary and sufficient condition for the existence of this **F** is that the original system $(\mathbf{A},\mathbf{B})$ is controllable. Note that we often do not need full state assignment of the system's closed loop poles, since the main requirement may be for stability. In this case, we say that $(\mathbf{A},\mathbf{B})$ is stabilizable if and only if all of its uncontrollable eigenvalues already have negative real parts. In this case, a necessary and sufficient condition for a stabilizing **F** is that $(\mathbf{A},\mathbf{B})$ is stabilizable.

In many application, one does not have direct access to the full state. In this case, we would like to find a way to *estimate* the full state $x$, from the available observed outputs, $y$. Such a system is called an *observer*. A classical observer known as the Luenberger observer has the following form. It uses the outputs $y(t)$ and inputs $u(t)$ to generate an estimate $\widehat{x}$ of the system's true state assuming this state is generated by a known LTI system realization

$$\begin{aligned}\dot{x}(t) &= \mathbf{A}x(t)+\mathbf{B}_1w(t)+\mathbf{B}_2u(t)\\ y(t) &= \mathbf{C}x(t)+\mathbf{D}_1w(t)+\mathbf{D}_2u(t)\end{aligned}$$

In this case $w$ is a disturbance input that is generally not known and $u$ is a control input that we do know. We assume that the state space realization $\mathbf{G}\overset{\mathrm{s}}{=}\left[\begin{array}{c|cc}\mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2\\ \hline \mathbf{C} & \mathbf{D}_1 & \mathbf{D}_2\end{array}\right]$ is known. The associated Luenberger observer then generates an estimate $\widehat{x}$ that satisfies the following state equations

$$\begin{aligned}\dot{\widehat{x}}(t) &= \mathbf{A}\widehat{x}(t)+\mathbf{B}_2u(t)+\mathbf{L}(y(t)-\widehat{y}(t))\\ \widehat{y}(t) &= \mathbf{C}\widehat{x}(t)+\mathbf{D}_2u(t)\end{aligned}$$

where $\mathbf{L}$ is a matrix of *observer gains*. We choose these gains to ensure that the state estimation error $\widetilde{x}(t)=x(t)-\widehat{x}(t)$ asymptotically goes to zero in the absence of any external disturbance, $w$ and remains "small" when $w$ is a bounded disturbance.

To see when a ”stable” observer exists, we write out the state equations for the state estimation error $\widehat{x}$.

$$\begin{aligned}\dot{\widetilde{x}}(t) &= \dot{x}(t) - \dot{\widehat{x}}(t) \\ &= \mathbf{A}x(t) + \mathbf{B}_1 w(t) + \mathbf{B}_2 u(t) \\ &\quad -\mathbf{A}\widehat{x}(t) - \mathbf{B}_2 u(t) - \mathbf{L}(\mathbf{C}x(t) + \mathbf{D}_1 w(t) + \mathbf{D}_2 u(t) - \mathbf{C}\widehat{x} - \mathbf{D}_2 u) \\ &= (\mathbf{A} - \mathbf{L}\mathbf{C})\widetilde{x}(t) + (\mathbf{B}_1 + \mathbf{L}\mathbf{D}_1)w(t)\end{aligned}$$

A necessary and sufficient condition for the existence of observer gains, $\mathbf{L}$, that arbitrarily assign the eigenvalues of $\mathbf{A} - \mathbf{L}\mathbf{C}$ is that the pair $(\mathbf{A}, \mathbf{C})$ is observable. If all we require is that the observer’s eigenvalues are stable, then we require that all unobservable eigenvalues of $(\mathbf{A}, \mathbf{C})$ have negative real parts, or rather that $(\mathbf{A}, \mathbf{C})$ is *detectable*.

## 9. Observer-based Controllers

An observer based controller is a feedback control system that uses a Luenberger observer to estimate the inaccessible system states and then uses those state estimates to compute the control signal that is re-injected back into the plant. We take the plant, $\mathbf{P}$, to have state equations

$$\begin{aligned}\dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}_1 w(t) + \mathbf{B}_2 u(t) \\ z(t) &= \mathbf{C}_1 x(t) + \mathbf{D}_{12} u(t) \\ y(t) &= \mathbf{C}_2 x(t) + \mathbf{D}_{21} w\end{aligned}$$

which in packed matrix form is $\mathbf{P} \overset{\mathrm{s}}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{0} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right]$. The first output $z$ is a virtual signal measuring how well the system is ”performing”. If we are thinking in terms of a regulation problem, then $z$ might be the tracking error and the control effort. The other output $y$ is the output signal used by an observer to generate a state estimate $\widehat{x}$. The input $w$ is a disturbance and the input $u$ is the ”control” law that satisfies

$$u(t) = \mathbf{F}\widehat{x}$$

where $\mathbf{F}$ is chosen so the eigenvalues of $\mathbf{A}+\mathbf{B}_2\mathbf{F}$ have negative real parts and the state estimate satisfies

$$\dot{\widehat{x}}(t) \quad = \quad \mathbf{A}\widehat{x}(t)+\mathbf{B}_2u+\mathbf{L}(y(t)-\mathbf{C}_2\widehat{x})$$

where $\mathbf{L}$ is chosen so that $\mathbf{A}-\mathbf{L}\mathbf{C}_2$ is Hurwitz.

We can view the closed-loop equations as the feedback combination of the plant $\mathbf{P}$ with a *dynamic* controller $\mathbf{K}$ that has $\widehat{x}$ as its states, takes $y$ as its input and generates the output $u$. The state equations for the controller, therefore are

$$\begin{aligned}\dot{\widehat{x}} &= \mathbf{A}\widehat{x}+\mathbf{B}_2u+\mathbf{L}(y-\mathbf{C}_2\widehat{x}) \\ &= (\mathbf{A}+\mathbf{B}_2\mathbf{F}-\mathbf{L}\mathbf{C}_2)\widehat{x}+\mathbf{L}y \\ u &= \mathbf{F}x\end{aligned}$$

In packed matrix form the controller's state space realization becomes

$$\mathbf{K} \overset{\mathrm{s}}{=} \left[\begin{array}{c|c} \mathbf{A}+\mathbf{B}_2\mathbf{F}-\mathbf{L}\mathbf{C}_2 & \mathbf{L} \\ \hline \mathbf{F} & \mathbf{0} \end{array}\right]$$

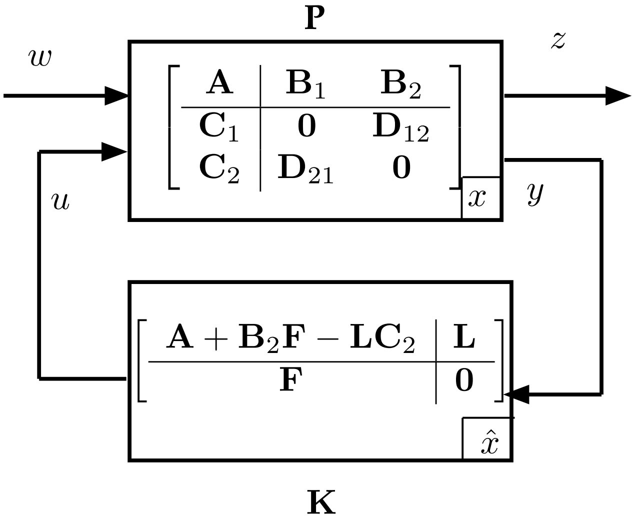

P
w
z
$\left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{0} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right]$
x
u
y
$\left[\begin{array}{c|c} \mathbf{A}+\mathbf{B}_2\mathbf{F}-\mathbf{L}\mathbf{C}_2 & \mathbf{L} \\ \hline \mathbf{F} & \mathbf{0} \end{array}\right]$
$\hat{x}$
K

FIGURE 14. Observer-based Control System's LFT

The feedback connection is portrayed in Fig. 1 as a *linear fractional transformation* (LFT). The transfer function for the original plant $\mathbf{P}$ can be written as

$$\begin{aligned}
\mathbf{P}(s) &= \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} (s\mathbf{I}-\mathbf{A})^{-1} \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \end{bmatrix} + \begin{bmatrix} 0 & \mathbf{D}_{12} \\ \mathbf{D}_{21} & 0 \end{bmatrix} \\
&= \left[\begin{array}{c|c} \mathbf{C}_1(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}_1 & \mathbf{C}_1(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}_2+\mathbf{D}_{12} \\ \hline \mathbf{C}_2(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}_1+\mathbf{D}_{21} & \mathbf{C}_2(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}_2 \end{array}\right] \\
&= \left[\begin{array}{c|c} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \hline \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{array}\right]
\end{aligned}$$

With a slight abuse of notation, we can now write out the relationship between the inputs $w$ and $u$ and $z$ and $y$ as

$$\begin{aligned}
z &= \mathbf{P}_{11}(s)w + \mathbf{P}_{12}(s)u \\
y &= \mathbf{P}_{21}(s)w + \mathbf{P}_{22}(s)u
\end{aligned}$$

Since the controller is also an LTI system, it has a transfer function

$$\mathbf{K}(s) = \mathbf{F}(s\mathbf{I}-\mathbf{A}-\mathbf{B}_2\mathbf{F}+\mathbf{L}\mathbf{C}_2)^{-1}\mathbf{L}$$

and we further know that $u = \mathbf{K}(s)y$. So if we substitute this back into the equation above we get

$$\begin{aligned}
z &= \mathbf{P}_{11}w + \mathbf{P}_{12}\mathbf{K}y \\
y &= \mathbf{P}_{21}w + \mathbf{P}_{22}\mathbf{K}y
\end{aligned}$$

Solving the second equation for $y$ yields

$$y = (\mathbf{I}-\mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}w$$

provided the inverse exists. Taking this last relation and putting it into the first equation gives the transfer funtion for our closed loop observer-based control system

$$\begin{aligned}
z &= \left[\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I}-\mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\right] w \\
&\overset{\text{def}}{=} \mathcal{F}_\ell(\mathbf{P},\mathbf{K})w
\end{aligned}$$

We refer to $\mathcal{F}_\ell(\mathbf{P},\mathbf{K})$ as a lower linear fractional transformation formed from the plant $\mathbf{P}$ and the controller $\mathbf{K}$. This LFT is said to be internally stable if for any bounded input $w$ we have that the internal signals $u$ and $y$ are also bounded (with respect to an assumed signal norm).

With regard to this closed-loop representation as an LFT, we now pose the controller synthesis problem as an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & \|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\| \\ \text{with respect to} & \mathbf{K} \\ \text{subject to} & \mathcal{F}_\ell(\mathbf{P},\mathbf{K}) \text{ being internally stable} \end{array}$$

where $\|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\|$ is the induced gain of the system. If $w$ is a white noise input and the norm of output $z$ is in $\mathcal{L}_2$, then this is called the $\mathcal{H}_2$ optimal controller. If the input and output are both $\mathcal{L}_2$ signals then this is called the $\mathcal{H}_\infty$ optimal controller.

## 10. Summary

This chapter reviewed classical methods taught in undergraduate control courses [Ogata (2009)] for the design of stabilizing feedback controllers. The chapter first considered pole-placement methods and then discussed a frequency-domain design method called *loopshaping*. Our coverage of loopshaping is similar to that found in Rohrs et al. (1992), Doyle et al. (2013), and Astrom and Murray (2010) all of which might be seen as providing a more mature view of frequency-domain design than is found in the older texts. Loopshaping provides considerable insight into what constitutes a ”good” feedback control for a SISO plant. Surprisingly, loopshaping insights are also valuable in selecting weights for modern MIMO robust control methodologies such as the $\mathcal{H}_\infty$ controllers discussed in chapter 3. The chapter closed with a discussion of state feedback methods for MIMO linear systems. Much of that discussion is drawn from linear systems theory textbooks such as Antsaklis and Michel (2006). Our treatment, however, reframes much of that discussion in terms modern MIMO feedback control texts such as Zhou et al. (1996) in which controller synthesis is viewed as an optimization problem subject to internal stability conditions. The next chapter takes a deeper look at the ”optimal control” problems for general classes of nonlinear systems.

## 11. Appendix: Bode Plots and the Nyquist Criterion

There are two ways of graphically representing the information in a system's, $\mathbf{G}(s)$, frequency response; the Bode plot and the Nyquist plot.

- *Bode plots* graph $20\log_{10}|\mathbf{G}(j\omega)|$ (units of decibels (dB)) and $\arg(\mathbf{G}(j\omega))$ versus $\log_{10}\omega$. This first plot is called the *gain-magnitude* plot and the second is called the *phase plot* of the system. Bode plots are useful because one can readily sketch these plots for rational transfer functions and use them to help design stabilizing feedback control systems.
- *Nyquist plots* graph $\mathrm{Re}(\mathbf{G}(j\omega))$ versus $\mathrm{Im}(\mathbf{G}(j\omega))$. These plots are extremely useful in certifying whether or not a closed-loop system has any unstable poles as well as characterizing how close a given feedback system is to being unstable.

This appendix reviews methods for sketching asymptotic approximations to a Bode plots. It then reviews how Bode plots and Nyquist plots are used to evaluate the asymptotic stability of closed-loop systems.

**Sketching Bode Plots:** Bode plots are most easily generated using software functions such as MATLAB's `bode` function. For example, the following script generates the Bode plot for transfer function $\mathbf{G}(s) = \frac{160(s+1)}{s(s^2+s+16)}$.

```
s = tf('s')
G = 160*(s+1)/(s*(s^2+s+16)
bode(G)
grid on
```

whose resulting plot is shown in Fig. 15. The solid blue line shows the Bode plot generated by the `bode` command. The dashed black line shows the Bode plot that was hand sketched using the guidelines discussed below.

Consider the transfer function

$$\mathbf{G}(s) = \frac{K(1+T_1 s)}{s(1+T_a s)(1+2\frac{\eta}{\omega_n}s+\frac{s^2}{\omega_n^2})}$$

This transfer function consists of several factor; a constant factor $K$, a pole at the origin, a simple pole/zero, and a complex pole pair. Let us determine the gain-magnitude and phase plots for this transfer function and discuss how each of these factors might be sketched by hand.

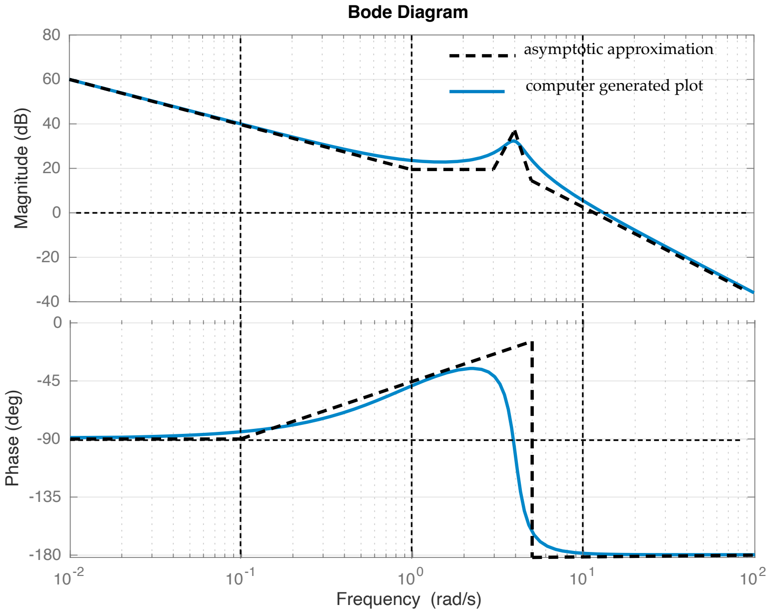


FIGURE 15. Bode plot (computer generated and hand sketched) for $\frac{160(s+1)}{s(s^2+s+16)}$

The gain magnitude of the transfer function is defined as

$$|\mathbf{G}(j\omega)|_{\text{dB}} = 20\log_{10}|\mathbf{G}(j\omega)|$$

If we take logarithms on both sides and expand out the individual terms we see

$$\begin{aligned}|\mathbf{G}(j\omega)|_{\text{dB}} &= 20\log_{10}|K| + 20\log_{10}|1+j\omega T_1| - 20\log_{10}|j\omega| \\ &\quad -20\log_{10}|1+j\omega T_a| - 20\log_{10}\left|1+2\frac{j\eta\omega}{\omega_n} - \frac{\omega^2}{\omega_n^2}\right|\end{aligned}$$

and the phase can be expanded out as

$$\begin{aligned}\arg\mathbf{G}(j\omega) &= \arg K + \arg(1+j\omega T_1) - \arg j\omega \\ &\quad -\arg(1+j\omega T_a) - \arg\left(1+2\frac{j\eta\omega}{\omega_n} - \frac{\omega^2}{\omega_n}^2\right)\end{aligned}$$

Note that these expressions have 4 different types of factors; constant factors, $K$, poles/zeros at the origin, simple poles/zeros, and complex pole/zero pairs. The Bode plot is easily sketched by hand due to the additivity of the terms. In other words, we first take a transfer function and factor the numerator and denominator polynomials. The Bode plot for each factor can be sketched by hand using the methods described below and because of the additivity of the terms under the logarithm, we can graphically add the plots for these factors to obtain an approximate Bode plot for the entire transfer function. Since controllers can also be seen as adding additional factors into the Bode plot, this graphical procedure can be used to easily identify candidate controller architectures in a rapid manner.

**Constant Factors**, $K$, have the following gain magnitude and phase

$$\begin{aligned} |K|_{\mathrm{dB}} &= 20\log_{10} K = \text{constant} \\ \arg\, K &= \begin{cases} 0 & K > 0 \\ \pi & K < 0 \end{cases} \end{aligned}$$

The Bode plot for this constant factor is easily drawn as shown on left side of Fig. 16. The gain magnitude is a straight horizontal line that intersects the $y$-axis at $20\log_{10} K$. The phase is also a straight horizontal line whose $y$-coordinate is either 0 or $-180°$ depending whether $K$ is positive or negative, respectively.

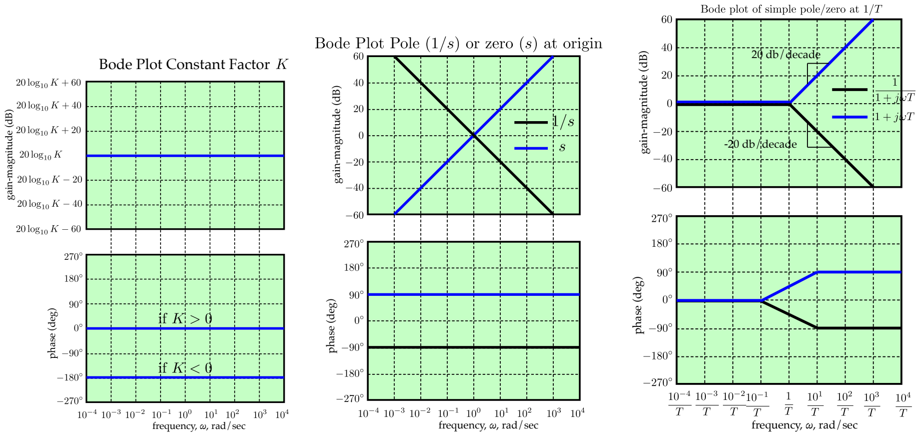


FIGURE 16. (left) $K$ factor Bode plot - (middle) pole/zero origin Bode Plot - (right) simple first order factor

**Pole/zero at Origin** $\frac{1}{j\omega}$ or $j\omega$. Let us first look at a pole at the origin.

$$\begin{aligned} \left|\frac{1}{j\omega}\right|_{\mathrm{dB}} &= -20\log_{10}|(j\omega)| \\ \arg\left(\frac{1}{j\omega}\right) &= -\frac{\pi}{2} \quad \text{rad} = -90° \end{aligned}$$

Note that over a decade, when the frequency, $\omega$, changes by a factor of 10, then the gain magnitude decreases by 20 dB. So we say the gain magnitude has a roll-off of 20 dB/decade. The gain magnitude shown in the middle of Fig. 16 for this factor $(1/s)$ is therefore a straight line with a slope of $-20$ dB/decade that passes through 0 dB at $\omega = 1$ rad/sec. The phase plot for the simple pole at the origin is a straight horizontal line whose $y$-coordinate is $-90°$. Fig. 16 also plots the Bode plot for a zero at the origin $(s)$. In this case the gain-magnitude plot becomes a straight line that increases with a slope of 20 dB/decade and that passes through 0 dB at $\omega = 1$ rad/sec. The phase plot for the zero at the origin is a horizontal line that intersects the $y$-axis at $+90°$.

**Simple pole/zero:** $(1+j\omega T)^{\pm 1}$ where $T > 0$ is a real constant. Let us consider the simple zero first. In this case the factor is $\mathbf{G}(j\omega) = 1 + j\omega T$. The gain magnitude and phase are

$$\begin{aligned} |\mathbf{G}(j\omega)|_{\text{dB}} &= 20\log_{10}|1+j\omega T| = 20\log_{10}\sqrt{1+\omega^2T^2} \\ \arg \mathbf{G}(j\omega) &= \arg(1+j\omega T) = \tan^{-1}\omega T \end{aligned}$$

Let us first examine the *asymptotic behavior* of the *gain-magnitude* when $\omega T \ll 1$ (or rather $\omega \ll \frac{1}{T}$). In this case $|\mathbf{G}(j\omega)| \approx 0$ dB. This can be drawn as a horizontal line for $\omega \ll \frac{1}{T}$ at 0 dB. At the other asymptotic end when $\omega T \gg 1$ we have $|\mathbf{G}(j\omega)|_{\text{dB}} \approx 20\log_{10}\omega T$. This can be drawn as a straight line with a positive slope of 20 dB/decade. If we draw both of these asymptotic lines, we see they intersect at the frequency $\omega_c = 1/T$ rad/sec. This is called the *corner frequency* of the simple zero factor. These straight lines form the *asymptotic approximation* of the simple zero factor's gain-magnitude plot. The ”hand” drawn gain-magnitude plot for this factor is shown on right side of Fig. 16. That plot also shows the gain magnitude of the simple pole at $1/T$. In this case, the asymptotic gain-magnitude differs in that the gain-magnitude rolls off at $-20$ dB/decade after the corner frequency $1/T$.

Now let us examine the asymptotic behavior of the phase plot. When $\omega \ll \frac{1}{T}$, the phase is nearly zero degrees. When $\omega = 1/T$ the phase is $\pm 45°$, with a positive phase for the simple zero and a negative phase for the simple pole. Asymptotically the phases of these factors go to $\pm 90°$ for $\omega \gg 1/T$. The phase of the simple factor therefore switches from $0°$ to $\pm 90°$, with that switch starting around the corner frequency. In particular, that change begins about a decade below the corner frequency and ends about a decade above the corner frequency. The left side of Fig. 17 used these rules to sketch the phase of the simple first order factor.

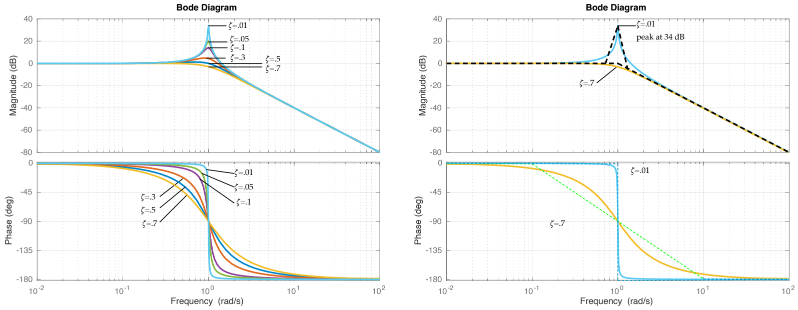


FIGURE 17. (left) Complex Factor Bode plot - (right) Sketch of Complex Pole Bode Plot

**Complex Poles:**. Consider the transfer function

$$\mathbf{G}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\omega_n$ and $\zeta$ are parameters. The frequency response function then becomes

$$\mathbf{G}(j\omega) = \frac{1}{\left(1 - \left(\frac{\omega}{\omega_n}\right)^2\right) - j2\zeta\left(\frac{\omega}{\omega_n}\right)}$$

Evaluating the gain-magnitude and phase of $\mathbf{G}(j\omega)$ yields

$$\begin{aligned}
20\log_{10}|\mathbf{G}(j\omega)| &= -20\log_{10}\sqrt{\left(1 - \left(\frac{\omega}{\omega_n}\right)^2\right)^2 + 4\zeta^2\left(\frac{\omega}{\omega_n}\right)^2} \\
&\approx \begin{cases} 0 \text{ dB} & \omega/\omega_n \ll 1 \\ -40\log_{10}\left(\frac{\omega}{\omega_n}\right) & \omega/\omega_n \gg 1 \end{cases} \\
\arg\mathbf{G}(j\omega) &= -\tan^{-1}\left(\frac{2\zeta}{\omega_n}\omega\left(1 - \left(\frac{\omega}{\omega_n}\right)^2\right)\right)
\end{aligned}$$

The second line for the gain magnitude is the asymptotic approximation of the factor's gain magnitude $\omega \ll \omega_n$ and $\omega \gg \omega_n$. The frequency $\omega_n$ corresponds to a corner frequency and we see that the gain-magnitude can again be drawn using straight line approximations in these asymptotic regions.

The left side of Fig. 17 shows the Bode plot for this complex factor for a range of damping ratios $\zeta$. In these plots $\omega_n = 1$. As $\zeta$ decreases below $0.7$, we see a *resonant peak* form around $\omega_n$. The gain magnitude plot shows that this peak increases as $\zeta$ gets smaller and in the limit when $\zeta = 0$ this peak is infinite. One the phase plots we see the phase start at $0°$ and then switch to $-180°$. That switch occurs around the corner frequency $\omega_n$, but the speed of that transition is a function of the damping ratio. For large damping ratios ($\zeta > .7$) the phase transition starts a decade below the corner frequency and ends a decade above the corner frequency. For damping ratios on the order of $0.05$ or $0.01$ we see an abrupt switch occur so for small damping ratios we ”sketch” the phase as an instantaneous shift occurring at $\omega_n$. The right side of Fig. 17 shows the sketched Bode plots for the case where $\zeta = 0.01$ and $\zeta = 0.7$. We can actually estimate the size of the resonant peak using approximation $\frac{1}{2\zeta}$, which is shown in the figure. For $\zeta$ greater than .5 or .7 we don't put in the resonant peak.

**Nyquist Criterion:** The Nyquist criterion is used to assess the stability of the closed loop system in Fig. 18 using the Nyquist plot of the open loop plant. The criterion is based on a well-known theorem from complex analysis known as the *principle of the argument*.

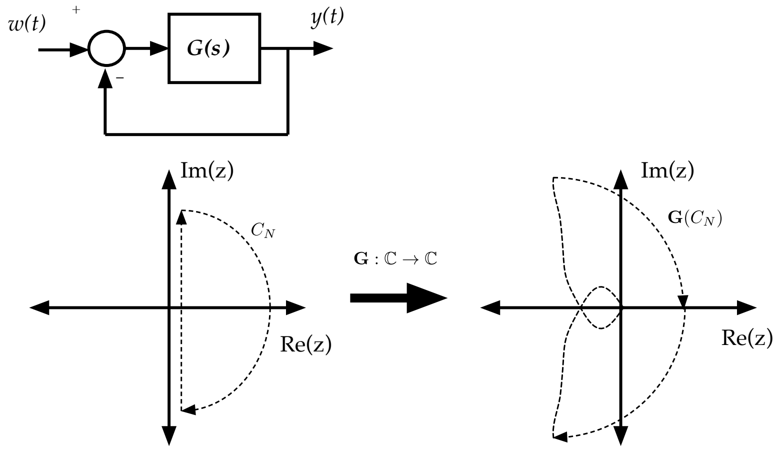


FIGURE 18. Nyquist Contour

Consider a function of a complex variable, $\mathbf{L} : \mathbb{C} \to \mathbb{C}$ that is a rational function. Now consider a simple closed contour in the complex plane defined by the union of two types of contour segments

(1) $s = j\omega$ for $-R < \omega < R$
(2) $s = Re^{j\phi}$ for $-\pi/2 \leq \phi \leq \pi/2$

where $R$ is large. This contour is denoted as $C_N$ and is shown on the left hand side of Fig. 18. We refer to $C_N$ as a *Nyquist contour* and we assume $R$ is large enough so it encloses all removable poles and zeros of $\mathbf{L}(s)$. We pass the Nyquist contour through the loop function to generate the new contour, $\mathbf{L}(C_N)$ that is shown on the right hand side of Fig. 18. This contour is actually the Nyquist plot of $\mathbf{L}(s)$. If we let

$$\begin{aligned} Z &= \text{number of finite zeros of } \mathbf{L} \text{ encircled by } C_N \\ P &= \text{number of finite poles of } \mathbf{L} \text{ encircled by } C_N \end{aligned}$$

The *Principle of the Argument* states that $\mathbf{L}(C_N)$ encircles the origin $(Z-P)$-times in the same direction as $C_N$ provided $\mathbf{L}$ has no zeros or poles on $C_N$.

The Principle of the Argument can be used to assess the closed loop stability of the closed loop map

$$\mathbf{T}(s) = \frac{\mathbf{L}(s)}{1+\mathbf{L}(s)} = \frac{n(s)/d(s)}{1+(n(s)/d(s)} = \frac{n(s)}{n(s)+d(s)}$$

where the open loop map $\mathbf{L}(s) = \frac{n(s)}{d(s)}$ and $n(s), d(s)$ are polynomials in $s$. This closed loop map corresponds to the feedback system shown in Fig. 18 where $\mathbf{L}(s)$ is the *loop function*. Note that $\mathbf{L}$ and $1+\mathbf{L}$ have the same denominator polynomials so the poles of $\mathbf{L}$ and $1+\mathbf{L}$ are the same. The zeros of $\mathbf{L}$ are the roots of $n(s)$ and the zeros of $1+\mathbf{L}$ are the zeros of $n(s)+d(s)$. Therefore the poles of the closed loop map, $\mathbf{T}(s)$ are the zeros of $1+\mathbf{L}(s)$ and so the closed loop map, $\mathbf{T}(s)$ is stable if and only if $1+\mathbf{L}(s)$ has no non-minimum phase zeros.

So let's consider the Nyquist plot, $\mathbf{L}(C_N)$ of the loop function and with regard to this plot let

$$\begin{aligned}
N_0 &= \text{number of encirclements that } \mathbf{L} \text{ makes of the origin} \\
Z_0 &= \text{number of zeros of } \mathbf{L} \text{ encircled by } C_N \\
P_0 &= \text{number of poles of } \mathbf{L} \text{ encircled by } C_N \\
Z_{-1} &= \text{number of zeros of } 1+\mathbf{L} \text{ encircled by } C_N \\
P_{-1} &= \text{number of poles of } 1+\mathbf{L} \text{ encircled by } C_N \\
N_{-1} &= \text{number of encirclements that } 1+\mathbf{L} \text{ makes of origin} \\
&= \text{number of encirclements that } \mathbf{L} \text{ makes of } (-1,0)
\end{aligned}$$

The loop function is stable if and only if $P_0 = 0$. Since the poles of the closed loop map, $\mathbf{T}(s)$, are the roots of $n(s) + d(s)$ (also the zeros of $1 + \mathbf{L}(s)$) we can deduce the closed loop map is stable if and only if $Z_{-1} = 0$.

By the principle of the argument, the number of Nyquist plot will make the number of encirclements of the origin satisfy $N_0 = Z_0 - P_0$. In a similar way the number of encirclements of $(-1, 0)$ will satisfy $N_{-1} = Z_{-1} - P_{-1}$. So if we know how many non-minimum phase zeros the loop function has (i.e. we know $Z_0$) then we can conclude that the number of poles encircled by the Nyquist contour (i.e. the unstable poles) will satisfy

$$P_0 = Z_0 - N_0$$

Because $\mathbf{L}$ and $1 + \mathbf{L}$ have the same finite poles, we also know $P_{-1} = P_0$. This implies that the number of zeros encircled by $1 + \mathbf{L}$ (i.e. the poles of the closed loop map) will be

$$Z_{-1} = N_{-1} + P_{-1} = N_{-1} + P_0 = N_{-1} + Z_0 - N_0.$$

Closed loop stability requires $Z_{-1} = 0$, so we can conclude that $0 = N_{-1} + P_0$ or rather that

$$N_{-1} = -P_0$$

In other words for the closed loop system to be stable, the Nyquist plot for $\mathbf{L}(s)$ must encircle $(-1, 0)$ as many times as the number of RHP (non-minimum phase) poles of $\mathbf{L}(s)$ and these encirclements must be in the clockwise (CW) direction. Note that if we already know $\mathbf{L}$ is minimum phase and stable, then $Z_0 = 0$ and

$P_0 = 0 = P_{-1}$ and we have $N_{-1} = Z_{-1}$. In other words for this special case of loop function (stable and minimum phase) the closed loop system is stable if the Nyquist plot of $\mathbf{L}(s)$ has no encirclements of $(-1, 0)$. This therefore provides a graphical way of using the loop function's Nyquist plot to determine the stability of the closed loop map, $\mathbf{T}(s)$.

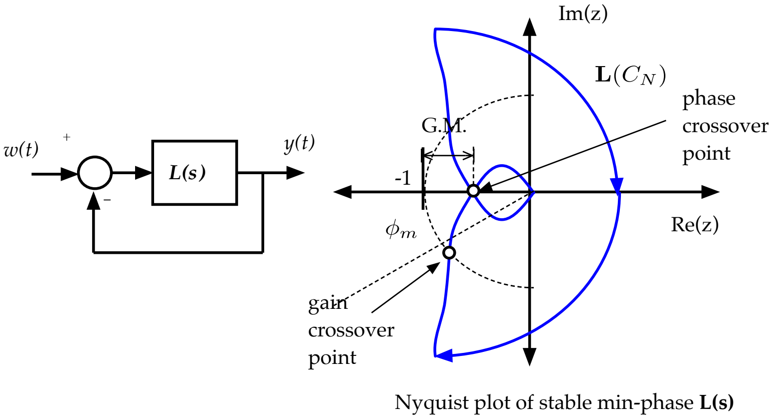


FIGURE 19. Definition of Gain/Phase Margin from Nyquist plot

**Gain/Phase Margin:** We can use the loop function's Nyquist plot to characterize how close the closed loop system is to being unstable. To do this we first define the following points on the Nyquist plot as shown in Fig. 19. We define the *phase crossover point* as that point at which the Nyquist plot crosses the negative real axis. The phase crossover frequency, $\omega_c$, is the frequency where the Nyquist plot crosses the negative real axis (i.e. the frequency when the phase equals $180°$). We define the *Gain Margin* as

$$G.M. = 20 \log_{10} \frac{1}{\mathbf{L}(j\omega_c)}$$

which is measured in units of decibels. Note that if $\mathbf{L}(s)$ is minimum phase we can identify four different cases

(1) If there is no phase crossover point then $G.M. = \infty$.
(2) If a phase crossover point occurs between $0$ and $-1$, then the gain margin is positive
(3) If a phase crossover occurs at $-1$ then the gain margin is zero

(4) If the phase crossover occurs beween $-\infty$ and $-1$ then the gain margin is negative

These observations imply that the gain margin is the amount of gain that can be added to the loop gain, $\mathbf{L}(s)$ of a stable closed loop system which just causes the closed loop system to be unstable. In particular, this means that for stable min-phase loop functions the gain margin has to be positive to ensure the closed loop system is stable. If the closed loop system is stable, then the gain margin is how much we can raise the loop function's gain before we cause the closed-loop system to be unstable.

The gain margin, however, is not the only way we can perturb the open loop function $\mathbf{L}(s)$. We can also introduce a pure phase variation to $\mathbf{L}(s)$ that may result in a change in Nyquist plot's number of encirclements of $(-1, 0)$. To formally define this *phase margin*, we first introduce the *gain crossover point* as that point on the loop function's Nyquist plot where the gain magnitude equals one (i.e. where the Nyquist plot intersects a unit circle centered at the origin). This gain crossover point is shown in Fig. 19. The gain crossover frequency, $\omega_g$ is the frequency when $|\mathbf{L}(\omega_g)| = 1$. We define the *phase margin*, $\phi_m$, of the loop function as

$$\phi_m = \arg(\mathbf{L}(j\omega_g)) - 180^\circ$$

If $\mathbf{L}$ is a stable minimum phase system, then adding $\phi_m$ extra phase to the loop function at $\omega_g$ will force the Nyquist plot to encircle $(-1, 0)$ thereby changing the number of encirclements and causing the closed loop system to be unstable. Phase margin, therefore, may be seen as the additional phase (or lag) that can be added to the loop function that makes the closed loop system unstable.

We can also read the gain and phase margins off of the Bode plot of the loop function. Let us consider the Bode plot for

$$\mathbf{L}(s) = \frac{20}{s(s^2 + 10s + 100)}$$

But rather than using `bode` we use the command `margin` to plot the Bode plot. The resulting plot is shown in Fig. 20.

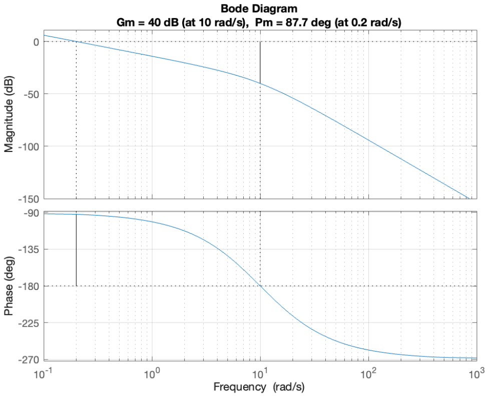


FIGURE 20. Gain/Phase Margin from Bode Plot

CHAPTER 2

# Optimal Control

The prior chapter concluded with a formulation of controller design as an optimization problem. We refer to this as *optimal control*. This basic problem is posed within the following framework where $\widehat{C}[0,T]$ is the linear space of piecewise continuous functions over time interval $[0,T]$.

- *Dynamical Control System* is modeled as an initial value problem

  $$\dot{x}(t) = f(t,x,u), \quad x(t_0) = x_0$$

  where $x : [0,T) \to \mathbb{R}^n$ is the state, $u; [0,T) \to \mathbb{R}^m$ is the control, $t$ is time, and $T$ is a desired final time (deadline). We assume that the differential equation admits unique causal solutions for $t \geq t_0$ once the initial condition $x_0$ has been fixed.
- *Target Set*, $\Omega \subset \mathbb{R}^n$ is a set of states that must be reached by the specified final time $T$.
- *Admissible Control* is the set of piecewise continuous functions, $u \in \widehat{C}[0,T]$, that takes values in a known set $U \subset \mathbb{R}^m$.
- *Cost Functional* is a functional $J : \widehat{C}[0,T] \to \mathbb{R}$ whose value $J[u]$ for a given $u \in \widehat{C}[0,T]$ is the ”cost” incurred by the system when using the control input $u$. We assume that the cost functional is additive in the sense that

  $$J[u_{[0,T]}] = J[u_{[0,t]}] + J[u_{[t,T]}]$$

  for all $t \in [0,T]$.

Within this framework the optimal control problem is to find $u^* \in \widehat{C}[0,T]$ such that $J[u^*] \leq J[u]$ for all $u \in \widehat{C}[0,T]$. This is a constrained optimization problem to be solved over an *infinite* dimensional linear space, $\widehat{C}[0,T]$.

The remainder of this chapter provides a quick tour of the fundamental concepts in optimal control over a finite horizon. We first start by reviewing necessary conditions for optimality of finite dimensional problems. We then extend these necessary conditions to infinite dimensional problems and obtain the Euler-Lagrange equations used in the classical Calculus of Variations (CoV). We then examine necessary conditions for optimal controls based on this variational calculus. These methods are sometimes used to numerically solve finite horizon optimal control problems used in a popular method known as *model predictive control*. We then look at an important approach to optimal control known as *dynamic programming*. We will show how dynamic programming is used to derive the linear quadratic regulator (LQR) and how it is used in the optimal control of Markov Decision Processes (MDP).

## 1. Mathematical Programming

This section reviews basic methods from mathematical programming over finite and infinite dimensional linear spaces. The main problem is of the form

$$\begin{array}{lrl}
\text{minimize:} & f(x) & \text{objective} \\
\text{with respect to:} & x & \text{decision variable} \\
\text{subject to:} & g(x) \leq 0 & \text{inequality constraints} \\
 & h(x) = 0 & \text{equality constraints} \\
 & x \in D & \text{domain of } f
\end{array} \tag{5}$$

For finite dimensional mathematical programs we take $f : D \to \mathbb{R}$, $g : D \to \mathbb{R}^m$, and $h : D \to \mathbb{R}^p$ to be functions taking values in a finite dimensional linear space and we require domain, $D \subset \mathbb{R}^n$, to either be compact or convex. After reviewing necessary conditions for optimal solutions of finite dimensional mathematical programs we will explore similar conditions for infinite dimensional optimization.

**1.1. Finite Dimensional Mathematical Programs:** For the moment let us confine our attention to finite dimensional mathematical programs [Bazaraa et al. (2006)] . This means the constraints in equation (5) use functions $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^p$. A vector $x \in D$ that satisfies all of these constraints ($g(x) \leq 0$ and $h(x) = 0$) is called a *feasible solution* and the set of all feasible solutions is called the *feasible region*. The basic mathematical problem is to find a feasible vector $x^* \in D$ such that $f(x^*) \leq f(x)$ for all feasible $x \in D$.

We first consider the *unconstrained* problem where there are no inequality or equality constraints to be satisfied. This unconstrained problem takes the form

$$\begin{aligned} \text{minimize:} \quad & f(x) \\ \text{subject to:} \quad & x \in D \end{aligned}$$

where $D \subset \mathbb{R}^n$ is now the feasible region. A point $x^* \in D$ is a *strict global minimum* if $f(x) > f(x^*)$ for all $x \in D$ where $x \neq x^*$. A point $x^* \in D$ is a *strict local minimum* of $f$ on $D$ if there exists an $\epsilon > 0$ such that $f(x) > f(x^*)$ for all $x \in N_\epsilon(x^*) \cap D$ where $N_\epsilon(x^*)$ is an open neighborhood of $x^*$ of radius $\epsilon$

$$N_\epsilon(x^*) = \{x \in \mathbb{R}^n \ : \ |x - x^*| < \epsilon\}$$

Note that an optimal solution may not always exist. Figure 1 shows three situations where a solution may fail to exist for the unconstrained problem. In the first case (a), the domain $D$ is an open interval $(a, b)$. The infimum of $f(x)$ is at the left hand boundary point $b$, but since $b$ is not in the domain it is not a solution. The second case (b) shows discontinuous $f$. This function is continuous from the right hand side at point $c$. The infimum of $f$ is attained as we approach $c$ from the left. But since $f$ is right continuous the actual value of $f(c)$ is greater than this infimum and so the problem does not have a solution. The last case (c) shows an $f$ that is unbounded from below and so again the solution does not exist.

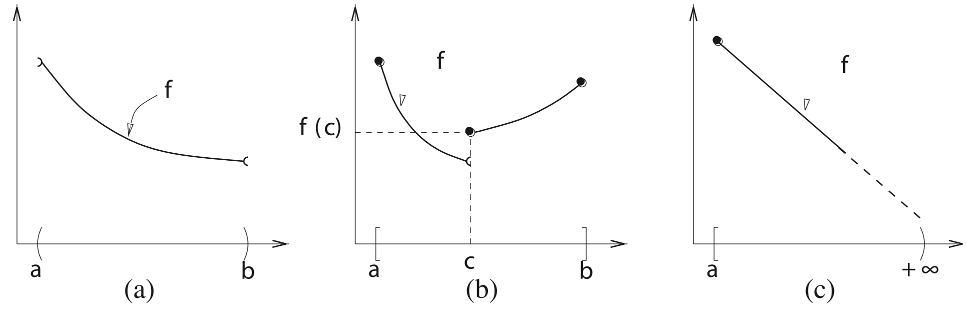


FIGURE 1. Three cases where a mathematical programming problem $\min_{x \in D} f(x)$ has no solution

Given the fact that it is relatively easy to formulate mathematical programs that have no solution, we need to find conditions that can be used to verify if a solution actually exists. Necessary conditions are conditions that must be satisfied by a solution whereas sufficient conditions imply a solution exists. We will present necessary and sufficient conditions for optimal solutions of unconstrained and constrained mathematical programs. The necessary and sufficient conditions are often used to find solutions to the mathematical program.

We first state conditions for solutions of the unconstrained problem

$$\min_{x \in D} f(x)$$

These statements will place various restrictions on $f$ (continuous, differentiable, convex). We denote the gradient of $f$ at $\overline{x} \in D$ as

$$\nabla f(\overline{x}) = \left[ \frac{\partial f(\overline{x})}{\partial x} \right]^T = \left[ \begin{array}{ccc} \frac{f(\overline{x})}{\partial x_1} & \cdots & \frac{f(\overline{x})}{\partial x_n} \end{array} \right]$$

If $f$ is twice differentiable we denote the Hessian of $f$ at $\overline{x} \in D$ as

$$H(\overline{x}) = \left[ \begin{array}{ccc} \frac{\partial^2 f(\overline{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\overline{x})}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\overline{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\overline{x})}{\partial x_n \partial x_n} \end{array} \right]$$

A set $C \subset \mathbb{R}^n$ is *convex* if for all $x, y \in C$ the points $z = \lambda x + (1 - \lambda) y$ lie in $C$ for all $\lambda . \in [0, 1]$. A function $f : C \to \mathbb{R}^n$ is *convex* if and only if

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

for all $x, y \in C$ and all $\lambda \in [0, 1]$.

The main conditions for solutions of the unconstrained mathematical program are enumerated below without proof.

(1) Assume that $f$ is continuous on $D$ and $D$ is a compact set, then there exists $x^* \in D$ such that $f(x^*) = \min\limits_{x \in D} f(x)$

(2) Assume that $f$ is differentiable, if $x^*$ is a local minimum then $\nabla f(x^*) = 0$.

(3) Assume $f$ is twice differentiable, if $x^*$ is a local minimum then $\nabla f(x^*) = 0$ and $H(x^*)$ is positive semidefinite.

(4) Suppose $f$ is differentiable at $x^*$ and is convex on $\mathbb{R}^n$. If $\nabla f(x^*) = 0$ then $x^*$ is a global minimizer of $f$ on $\mathbb{R}^n$.

(5) Suppose $f$ is twice differentiable at $x^*$. If $\nabla f(x^*) = 0$ and $H(x^*)$ is positive definite, then $x^*$ is a local minimizer of $f$

The first condition states that having $f$ continuous and $D$ compact is sufficient for the existence of an optimal solution. A set is compact [Rudin (1964)] if every infinite sequence in $D$ has a convergent subsequence. When $D$ is a subset of $\mathbb{R}^n$, then requiring $D$ to be compact simply means it is a closed and bounded set. Conditions two and three are necessary conditions for optimality with condition three being somewhat ”tighter” due to its requirement that the Hessian is positive semidefinite. The last two results are sufficient conditions for an optimal solution. Strengthening the necessary condition into a sufficient condition requires additional restrictions on the problem, either that $D$ and $f$ are convex or that the Hessian of $f$ is positive definite.

When the underling mathematical program has inequality constraints $g(\overline{x}) \leq 0$ or equality constraints $h(\overline{x}) = 0$ we need to augment the gradient conditions for optimality. This augmentation essentially tries to turn the constrained problem into an unconstrained problem.

We now present the Karush-Kuhn-Tucker (KKT) conditions for the optimal solution of a constrained mathematical program. As before we present this result without formal proof since its development is usually taught in mathematical programming courses [Bazaraa et al. (2006)].

$$\begin{array}{rll} \text{minimize:} & f(x) & \\ \text{subject to:} & g_i(x) \leq 0, & \text{for } i = 1, \ldots, m \\ & h_i(x) = 0, & \text{for } i = 1, \ldots, p \\ & x \in D & \end{array}$$

where $D$ is a nonempty open set in $\mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, $g_i : \mathbb{R}^n \to \mathbb{R}$ ( $i = 1, \ldots, p$), and $h_i : \mathbb{R}^n \to \mathbb{R}$ ($i = 1, \ldots, q$) are continuously differentiable functions.

Consider a point $x \in \mathbb{R}^n$ and let let the set of active inequality constraints at $x$ be denoted as

$$\mathcal{A} = \{i \in \{1, \ldots, p\} \,:\, g_i(x) = 0\}$$

If $\nabla g_i(x)$ for $i \in \mathcal{A}$ and $\nabla h_i(x)$ for $i = 1, 2, \ldots, p$ are linearly independent then the feasible point $x$ is said to be *regular*).

If $x^*$ is a local minimum of $f$ that satisfies the constraints and is a regular point, then there exist $q$ unique vectors $v^* \in \mathbb{R}^q$ and $\lambda^* \in \mathbb{R}^p$ such that

$$\begin{aligned}
\nabla f(x^*) + \nabla \mathbf{g}(x^*)^T v^* + \nabla \mathbf{h}(x^*)^T \lambda^* &= 0 \\
v^* &\geq 0 \\
\mathbf{g}(x^*) &\leq 0 \\
\mathbf{h}(x^*) &= 0 \\
(v^*)^T \mathbf{g}(x^*) &= 0
\end{aligned}$$

The preceding conditions are called KKT *necessary* conditions for $x^*$ to be a local minimum of $f$ subject to the inequality and equality constraints. We refer to the additional variables $v_i$ $(i = 1, \ldots, p)$ and $\lambda_i$ $(i = 1, \ldots, q)$ as Lagrange multipliers.

**Example:** Consider the following mathematical program

$$\begin{aligned}
\text{minimize:} \quad & x_1^2 + x_2^2 \\
\text{subject to:} \quad & x_1^2 + x_2^2 - 4 \leq 0 \\
& -x_1 \leq 0 \\
& -x_2 \leq 0 \\
& x_1 + 2x_2 - 2 = 0
\end{aligned}$$

In this case we have $f(x) = x_1^2 + x_2^2$, $h(x) = x_1 + 2x_2 - 2$ and

$$g(x) = \begin{bmatrix} x_1^2 + x_2^2 - 4 \\ -x_1 \\ -x_2 \end{bmatrix}$$

The gradients of the functions are

$$\nabla f(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}, \quad \nabla g(x) = \begin{bmatrix} 2x_1 & 2x_2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \nabla h(x) = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

So the first KKT condition is

$$\begin{aligned}\begin{bmatrix}0\\0\end{bmatrix} &= \nabla f(\overline{x}) + \nabla g(x)^T v + \nabla h(x)^T \lambda \\ &= \begin{bmatrix}2x_1\\2x_2\end{bmatrix} + \begin{bmatrix}2x_1 & -1 & 0\\2x_2 & 0 & -1\end{bmatrix}\begin{bmatrix}v_1\\v_2\\v_3\end{bmatrix} + \begin{bmatrix}1\\2\end{bmatrix}\lambda \\ &= \begin{bmatrix}2x_1(1+v_1) - v_2 + \lambda\\ 2x_2(1+v_1) - v_3 + 2\lambda\end{bmatrix}\end{aligned}$$

The second KKT condition requires

$$\begin{aligned}0 &= v^T g(x) \\ &= \begin{bmatrix}v_1 & v_2 & v_3\end{bmatrix}\begin{bmatrix}x_1^2 + x_2^2 - 4\\ -x_1\\ -x_2\end{bmatrix} \\ &= v_1(x_1^2 + x_2^2 - 4) - x_1 v_2 - x_2 v_3\end{aligned}$$

and the last KKT requires $v_1, v_2, v_3 \geq 0$.

Let us make a convenient choice for $v_i$ that assumes $v_1, v_2 v_3 = 0$. This is essentially considering the case when $x$ is in the interior of the feasible region (namely none of the inequality constraints are active). The second and third KKT conditions are clearly satisfied and the first KKT condition becomes

$$\begin{aligned}0 &= 2x_1 + \lambda \\ 0 &= 2x_2 + 2\lambda\end{aligned} \quad \Rightarrow \quad 0 = \begin{bmatrix}2 & 0 & 1\\0 & 2 & 2\end{bmatrix}\begin{bmatrix}x_1\\x_2\\\lambda\end{bmatrix}$$

We also need $x^*$ to satisfy the equality constraint $x_1 + 2x_2 - 2 = 0$, which gives a third equation for $x^*$. Appending this to the linear algebraic equation formed from the KKT condition gives

$$\begin{bmatrix}0\\0\\2\end{bmatrix} = \begin{bmatrix}2 & 0 & 1\\0 & 2 & 2\\1 & 2 & 0\end{bmatrix}\begin{bmatrix}x_1\\x_2\\\lambda\end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix}x_1^*\\x_2^*\\\lambda^*\end{bmatrix} = \begin{bmatrix}2/5\\4/5\\-4/5\end{bmatrix}$$

and so we see $x^* = \begin{bmatrix}2/5\\4/5\end{bmatrix}$.

Fig. 2 shows this optimization problem. The figure shows the level curves of the objective function, $f$, and the linear equality constraint $h(x)$. Note that $x^*$ lies within the feasible region so considering the case where all inequality constraints was warranted. If that had not been the case, we would have had to select $v$ to consider the various active cases. We also need $x^*$ to be regular. This occurs if $\nabla g_i(x^*)$ are linearly independent for the active constraints $i \in \mathcal{A}(x^*)$. Since $x^*$ is in the interior, the point $x^*$ is already regular.

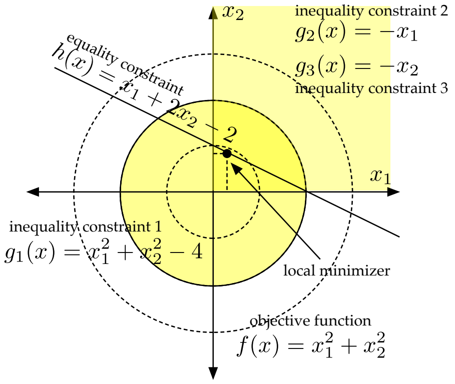


FIGURE 2. Example of mathematical program with equality and inequality constraints

**1.2. Unconstrained Infinite Dimensional Optimization:** The main necessary condition for unconstrained finite dimensional optimization problems was that the gradient, $\nabla J$, of the cost function vanish at the optimal point. Optimal control problems are solved over infinite dimensional linear spaces and so we need to extend our finite dimensional necessary conditions to the infinite dimensional problem.

We will be working in the function space $C^k([a,b], \mathbb{R}^n)$ which is the linear space of $k$-times continuously differentiable functions from $[a,b] \in \mathbb{R}$ to $\mathbb{R}^n$. We will equip this function space with a norm. The most commonly used norms are the 0-norm for $x \in C^k$ are

$$\|x\|_0 = \max_{\tau \in [a,b]} |x(\tau)|$$

or the 1-norm

$$\|x\|_1 = \max_{\tau\in[a,b]} |x(\tau)| + \max_{\tau\in[a,b]} |x'(\tau)|$$

where $x'$ is the first derivative of $x$.

Now let $\mathcal{L}$ be a linear space of functions equipped with the norm $\|\cdot\|$. Let $A$ be a subset of $\mathcal{L}$ and let $J[\cdot] : \mathcal{L} \to \mathbb{R}$ be a real-valued functional defined on $A$. A function $x^* \in A$ is a local minimum of $J$ over $A$ if there exists $\epsilon > 0$ such that for all $x \in A$ with $\|x^* - x\| < \epsilon$ we have $J[x^*] \leq J[x]$. We are interested in determining necessary conditions for the local minimizer of functional $J$. This is an unconstrained problem since we merely require $x \in A \subset \mathcal{L}$. We want to obtain necessary conditions that are similar to those we had for finite dimensional problems, but to do that we first have to define what we mean by the derivative of a functional.

Consider functional $J : \mathcal{L} \to \mathbb{R}$ and for a given $y \in \mathcal{L}$ we define its first variation (derivative) as the functional $\delta J|_y [\cdot] : \mathcal{L} \to \mathbb{R}$ such that for all $\eta \in \mathcal{L}$ and any $\epsilon > 0$ we have

$$J[y + \epsilon\eta] = J[y] + \delta J|_y [\eta]\epsilon + o(\epsilon)$$

with $\lim_{\epsilon\to 0} \frac{o(\epsilon)}{\epsilon} = 0$. This first variation is sometimes called the *Gateaux derivative* of $J$

$$\delta J|_y [\eta] = \lim_{\epsilon\to 0} \frac{J[y + \epsilon\eta] - J[y]}{\epsilon}$$

The following theorem gives a necessary condition for $y^* \in \mathcal{L}$ to be a local minimum of the cost functional $J$.

THEOREM 1. *Let $A \subset \mathcal{L}$ and let $y^* \in \mathcal{L}$ be a local minimum of the functional $J : A \to \mathbb{R}$, then for all perturbations $\eta \in \mathcal{L}$ such that $y^* + \epsilon\eta$ remains in $A$ for $\epsilon$ sufficiently small (an admissible perturbation) we have $\delta J|_{y^*} [\eta] = 0$.*

**Proof:** This proof relies on the fact for a fixed $y$ and $\eta$ we can define a real-valued *function* $g : \mathbb{R} \to \mathbb{R}$ such that

$$g(\epsilon) = J[y + \epsilon\eta]$$

For these fixed functions, the first variation then becomes

$$\delta J\big|_y [\eta] = g'(0)$$

and so we can generate a first order Taylor series expansion

$$g(\epsilon) = g(0) + g'(0)\epsilon + o(\epsilon)$$

Let us suppose that $g'(0) \neq 0$, then since $o(\epsilon)$ is a little-$o$ function there must exist $\bar{\epsilon} > 0$ such that $|o(\epsilon)| < |g'(0)\epsilon|$ whenever $|\epsilon| < \bar{\epsilon}$. For these values we see that

$$g(\epsilon) - g(0) < g'(0)\epsilon + |g'(0)\epsilon|$$

If we choose $\epsilon$ to have the opposite sign of $g'(0)$ so that $g(\epsilon) < g(0)$, then this would imply $y^*$ is not a local minimum, thereby generating a contradiction to our assumption that $g'(0) \neq 0$. So necessarily when $y^*$ is a local minimizer we must also have $g'(0) = 0$. $\diamondsuit$

## 2. Calculus of Variations

The Calculus of Variations provides a well known characterization of a functional's minimizer in terms of a set of differential equations known as the *Euler-Lagrange equations*. In this framework, the functional $J$ is an integral of a function called the *Lagrangian*. The approach has deep connections with classical mechanics [Goldstein (1959)] which also uses Euler-Lagrange equations to determine the equations of motion for mechanical systems. This section considers the the Calculus of Variations (CoV) Free and Fixed Endpoint problems.

**2.1. Free Endpoint Problem:** Consider a function $L : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and define the functional

$$J[y] = \int_a^b L(x, y(x), y'(x))dx$$

where $y$ is some function in $C^1([a,b], \mathbb{R})$. We refer to $L$ as the *Lagrangian* or running cost. The free endpoint problem is to find $y \in C^1[a,b]$ that minimizes $J[y]$ subject to the fixed initial constraint, $y(a) = y_0$, where the terminal constraint $y(b)$ is free to vary for a fixed terminal time $b$.

We can also define a *fixed endpoint problem* by simply requiring that $y(b) = y_1$ with $y_1$ being fixed. The difference between the fixed and free endpoint problems

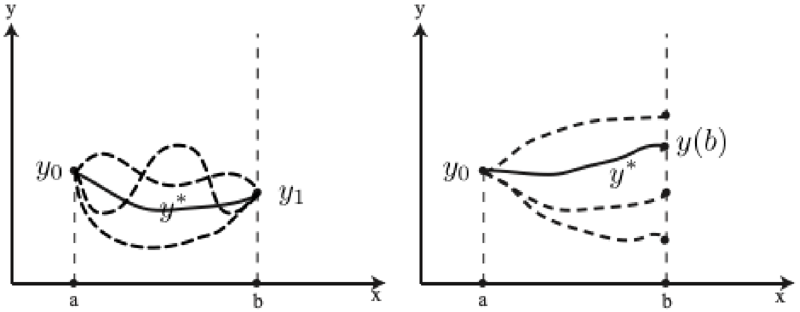


FIGURE 3. (left) fixed endpoint problem - (right) free endpoint problem

is shown in Fig. 3. The left figure plots the minimizer $y^*(x)$ for a fixed endpoint problem and admissible perturbations (dashed), $y(x) = y^*(x) + \epsilon\eta(x)$, of that minimizer. What we can see here is that the perturbation $\eta \in \mathcal{L}$ at the two boundary points is fixed, $\eta(a) = \eta(b) = 0$. The right figure plots the minimizer, $y^*$, for the free endpoint problem with its admissible perturbations (dashed). What you can see here is that for the free endpoint problem the perturbation at $a$ still vanishes $\eta(a) = 0$, but that the perturbation at $b$, $\eta(b)$, is free to move along the vertical line.

From the prior section we know that a necessary condition for optimality is that the first variation of the cost functional vanishes. So we first compute an expression for the first variation. For our choice of the cost functional we can find its first variation from a Taylor expansion about $y$. We fix $y \in \mathcal{L}$ and introduce a fixed perturbation $\eta \in \mathcal{L}$ that we dilate with the scalar $\epsilon \in \mathbb{R}$. This means that $J[y + \epsilon\eta]$ can be written as

$$\begin{aligned} J[y+\epsilon\eta] &= J[y] + \delta J\big|_y [\eta]\epsilon + o(\epsilon) \\ &= \int_a^b L(x, y(x) + \epsilon\eta(x), y'(x) + \epsilon\eta'(x))dx \end{aligned} \tag{6}$$

where $y'$ and $\eta'$ are the first derivatives of $y$ and $\eta$, respectively. A first order Taylor series expansion of the Lagrangian in equation (6) may be written as

$$J[y+\epsilon\eta] = \int_a^b \left( \begin{array}{l} L(x,y,y') + L_y(x,y,y')\epsilon\eta \\ +L_{y'}(x,y,y')\epsilon\eta' + o(\epsilon) \end{array} \right) dx \tag{7}$$

Matching $\epsilon$ terms in equations (6) and (7) allows us to pull out the following expression for the first variation of $J$

$$\delta J\big|_y[\eta] = \int_a^b \left(L_y(x,y,y')\eta + L_{y'}(x,y,y')\eta'\right) dx \tag{8}$$

The second term in equation (8) is a function of $\eta'$ and we can remove this dependency on $\eta'$ by integrating by parts.

Recall that integration by parts means for two functions $u$ and $v$ we have

$$\int u dv = uv - v\int du$$

In this case, let $u = L_{y'}$ and $dv = d\eta$. Integrating the differential yields $v = \eta$. Differentiating $u$ yields $du = \frac{d}{dx}L_{y'}dx$. Inserting these expression into the integration by parts formula gives

$$\int_a^b L_{y'}(x,y,y')\eta' dx \quad = \quad L_{y'}(x,y,y')\eta\big|_a^b - \int_a^b \eta\frac{d}{dx}L_{y'}(x,y,y')dx$$

which allows us to write the first variation at $y$ for any admissible perturbation, $\eta$, as

$$\delta J\big|_y[\eta] \quad = \quad \int_a^b \left(L_y(x,y,y') - \frac{d}{dx}L_{y'}(x,y,y')\right)\eta dx + L_{y'}(x,y,y')\eta\big|_a^b \tag{9}$$

For the free endpoint problem we require $\eta(a) = 0$ but $\eta(b)$ is not necessarily zero. For $y$ to be a local minimizer we require for any admissible perturbation that $\delta J\big|_y[\eta] = 0$ for all admissible $\eta$. Since admissible perturbations require $\eta(a) = 0$, setting equation (9) to zero yields

$$\begin{aligned} 0 \quad = \quad & \int_a^b \left(L_y(x,y,y') - \frac{d}{dx}L_{y'}(x,y,y')\right)\eta dx \\ & + L_{y'}(b,y(b),y'(b))\eta(b) \end{aligned} \tag{10}$$

when $y$ is a local minimizer.

Equation (10) must hold for all admissible perturbations. While $\eta(b)$ is free, it is certainly possible that $\eta(b)$ could be zero. So for equation (10) to be satisfied for this admissible perturbation we require

$$\int_a^b \left(L_y(x,y,y') - \frac{d}{dx}L_{y'}(x,y,y')\right)\eta dx$$

By the Fundamental Lemma of the Calculus of Variations [Liberzon (2012)], this will occur when

$$L_y(x, y(x), y'(x)) = \frac{d}{dx} L_{y'}(x, y(x), y'(x)) \tag{11}$$

This is our first necessary condition for optimality and it is called the *Euler-Lagrange* equation.

For admissible perturbations where $\eta(b)$ is not zero, then we would also require the last term in equation (9) to vanish. This will occur if

$$L_{y'}(b, y(b), y'(b)) = 0 \tag{12}$$

This is the second necessary condition a local minimizer of the free endpoint problem needs to satisfy and it is known as a *transversality condition*. The preceding discussion may be summarized in the following theorem that characterizes the local minimizer for the free endpoint problem.

THEOREM 2. *Consider the problem of minimizing*

$$J[y] = \int_a^b L(x, y(x), y'(x))dx$$

*on the set* $\mathcal{D} = \{(y, b) \in C^1[a, T] \times [a, T]\}$ *with* $y(b)$ *free and* $L : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *being* $C^1$*. Suppose that* $y^*$ *is a local solution, then* $y^*$ *satisfies the Euler Lagrange equation*

$$L_y(x, y(x), y'(x)) - \frac{d}{dx} L_{y'}(x, y(x), y'(x)) = 0$$

*for all* $x \in [a, b]$ *and the transversality condition*

$$L_{y'}(b, y(b), y'(b)) = 0$$

**Example:** Let us find a parameterized curve $y \in C^1$ that has the shortest arc length. The differential arc length is $ds = \sqrt{dy^2 + dx^2}$ and so the functional we want to minimize is

$$J[y] = \int_a^b \sqrt{1 + (y'(x))^2} dx$$

with Lagrangian $L(y, y') = \sqrt{1 + (y')^2}$. We are solving the free endpoint problem over the interval $[a, b]$ so $y(a) = y_0$ and $y(b)$ is free to vary. Our minimizer must satisfy two conditions; the Euler-Lagrange equation and the transversality condition. The Euler-Lagrange equation can be written as

$$\frac{\partial L}{\partial y} = 0 = \frac{d}{dx}\frac{\partial L}{\partial y'} = \frac{d}{dx}\left(\frac{y'}{\sqrt{1+(y')^2}}\right)$$

This implies that $\dfrac{y'(x)}{\sqrt{1+(y'(x))^2}}$ is constant which also means that $y'(x)$ is constant as well. Integrating $y'$ therefore yields the following family of solutions

$$y^*(x) = k_1 x + k_2$$

where $k_1$ and $k_2$ are real valued parameters we need to determine. In particular we know that $y^*(a) = y_0$ which implies $k_2 = y_0$. The other parameter, $k_1$, is determine from the transversality condition

$$L'_y(b, y(b), y'(b)) = \frac{y'(b)}{\sqrt{1+(y'(b))^2}} = 0$$

which is satisfies when $y'(b) = 0$. We therefore know that the slope of our line is 0, which means $k_1 = 0$. So our local minimizer is the straight horizontal line $y^*(x) = y_0$.

**2.2. Fixed Endpoint Problem with Constraints:** The fixed endpoint problem seeks a function $y \in C^1[a,b]$ that minimizes a functional $J$ subject to the constraint that $y(a) = y_0$ and $y(b) = y_1$ with $y_0$ and $y_1$ being fixed. We are going to augment the fixed endpoint problem with a path constraint. There are two types of constraints we consider; integral and non-integral constraints.

**Fixed Endpoint Problem with Integral Constraints:** We are going to augment the fixed endpoint problem with an integral constraint of the form

$$C[y] = \int_a^b M(x, y(x), y'(x))dx = C_0 \tag{13}$$

where $C : C^0 \to \mathbb{R}$ is a continuous constraint functional, $M$ is a $C^1$ function, and $C_0$ is a real constant. This means that the local minimizer, $y^*$, minimizes $J$ subject to $C[y^*] = C_0$.

Our development of the necessary conditions for the local minimizer will be based on heuristic arguments using the Lagrange multipliers we introduced when solving finite-dimensional mathematical programs with equality constraints. A more rigorous development requires a more sophisticated approach to the problem that is covered in many textbooks on the Calculus of Variations [Liberzon (2012)], but which is beyond the scope of what I want to cover in this book.

Consider a feasible solution $y \in \mathcal{L}$ and consider an admissible perturbation of the form $y + \epsilon\eta$ where $\eta$ is in $C^1$ and $\epsilon \in \mathbb{R}$. Since we are augmenting the fixed endpoint problem, we still require $\eta(a) = \eta(b) = 0$ for any admissible perturbation $\eta$. We will still need the first variation of $J$ to satisfy $\delta J|_y = 0$ for any admissible $\eta$. So we require

$$0 = \int_a^b \left( L_y(x, y, y') - \frac{d}{dx} L_{y'}(x, y, y') \right) \eta dx$$

where we've dropped the transversality condition since $\eta(b) = 0$. To be admissible, however, the perturbed curve $y + \epsilon\eta$ most also satisfy the integral constraint so that

$$C[y + \epsilon\eta] = C_0$$

and since $C[\cdot]$ is an integral constraint, the necessary condition for this to be satisfied is that its first variation also vanish with $\eta(a) = \eta(b) = 0$. So in a similar way to our earlier derivation of the first variation of $J$ we have

$$\begin{aligned} 0 &= \delta C|_y [\eta] \\ &= \int_a^b \left( M_y(x, y, y') - \frac{d}{dx} M_{y'}(x, y, y') \right) \eta dx \end{aligned}$$

for all admissible perturbations $\eta$.

This is similar to the constrained finite-dimensional mathematical programs with equality constraints that we described above. This suggests that solving our problem is equivalent to finding a Lagrange multiplier $\lambda \in \mathbb{R}$ such that

$$\left( L_y - \frac{d}{dx} L_{y'} \right) + \lambda \left( M_y - \frac{d}{dx} M_{y'} \right) = 0$$

for all $x \in [a, b]$. We will find it convenient to define an *augmented Lagrangian* of the form

$$\overline{L}(x, y, y', \lambda) = L(x, y, y') + \lambda M(x, y, y')$$

and the preceding equation becomes

$$\overline{L}_y(x, y, y', \lambda) = \frac{d}{dx} \overline{L}_{y'}(x, y, y', \lambda)$$

which is again an Euler-Lagrange equation, but this time for the augmented Lagrangian, $\overline{L}$ which must be solved for $y$ *and* $\lambda$.

The preceding discussion leading up to this theorem is heuristic in nature. It ignores a number of important technical issues that a more careful derivation of the theorem would require. In the first place we did not formally justify our use of Lagrange multipliers. The infinite dimensional nature of the problem requires different arguments than the earlier KKT conditions used in finite dimensional problems. In particular, our earlier use of Lagrange multipliers required the notion of a regular point. A rigorous justification for using such multipliers will require something similar to the regular point condition.

**Example:** We now consider a problem that looks for an arc, $y(x)$ from $a$ to $b$ where $y(a) = y(b) = 0$ that minimizes the area under the curve, $y(x)$, subject to an integral constraint that fixed the arc's length to a constant $C_0$. This classical problem is also known as Dido's problem. In this case our cost functional becomes

$$J[y] = \int_a^b y(x)dx$$

So $L(x) = y(x)$ and the constraint functional is

$$C[y] = \int_a^b \sqrt{1+(y')^2}dx = C_0$$

so that $M(y') = \sqrt{1+(y')^2}$. This is a fixed endpoint problem since $y(a) = y(b) = 0$. For this problem we let $C_0 = 1$, $a = 0$, and $b = 1/2$. The augmented Lagrangian then becomes

$$\overline{L}(y, y', \lambda) = y + \lambda\sqrt{1+(y')^2}$$

and the associated partial derivatives are

$$\begin{aligned} \overline{L}_y &= \frac{\partial}{\partial y}\left(y + \lambda\sqrt{1+(y')^2}\right) = 1 \\ \overline{L}_{y'} &= \frac{\partial}{\partial y'}\left(y + \lambda\sqrt{1+(y')^2}\right) = \frac{\lambda y'}{\sqrt{1+(y')^2}} \end{aligned}$$

The Euler-Lagrange equation for $\overline{L}$ then becomes

$$1 = \frac{d}{dx}\left(\frac{\lambda y'}{\sqrt{1+(y')^2}}\right)$$

Integrating this derivative gives

$$\frac{\lambda y'}{\sqrt{1+(y')^2}} = x + K$$

where $K$ is a constant of integration. Solving for $y'$ then yields the ordinary differential equation

$$\frac{dy}{dx} = \frac{\lambda(x+K)}{\sqrt{\lambda^2+(x+K)^2}}$$

for $x \in [0, 1/2]$. Integrating this differential equation gives

$$\begin{aligned} y(x) &= \int_0^x \frac{\lambda(z+K)}{\sqrt{\lambda^2+(z+K)^2}} dz = \lambda\sqrt{\lambda^2+(z+K)^2}\Big|_0^x \\ &= \lambda\left(\sqrt{\lambda^2+(x+K)^2} - \sqrt{\lambda^2+K^2}\right) \end{aligned}$$

With the boundary conditions $y(0) = y(1/2) = 0$, we find $K = -1/4$ and the solution is then

$$y(x) = \lambda\left(\sqrt{\lambda^2+(x-1/4)^2} - \sqrt{\lambda^2+1/16}\right)$$

which is the equation for a half circle.

**Fixed Endpoint with Non-integral Constraints:** Suppose we have an equality constraint that must hold pointwise

$$M(x, y(x), y'(x)) = 0$$

for all $x \in [a, b]$. Let $y$ be a test curve and note that the first-order necessary conditions for optimality are similar to those for integral constraints. The main difference is that now the Lagrange multiplier is a function of $x$. In other words the Euler Lagrange equation holds for the augmented Lagrangian

$$\overline{L}(x, y, y', \lambda) = L(x, y, y') + \lambda^*(x)M(x, y, y')$$

where $\lambda^* : [a, b] \to \mathbb{R}$.

The earlier integral constraint was global in the sense that it applied to the entire test curve, $y$. The non-integral constraint, on the other hand, is local since it applies at each point in the interval $[a, b]$. This means that for each $x \in [a, b]$ there should be a Lagrange multiplier, so that $\lambda^*$ is a function of $x \in [a, b]$. So here when we consider the optimization of an integral of the augmented Lagrangian we are trying to minimize

$$\int_a^b L(x,y,y')dx + \int_a^b \lambda(x)M(x,y,y')dx$$

over $y$ *and* $\lambda$.

## 3. Variational Method for Optimal Control

The preceding section showed how the Calculus of Variations can be used to minimize the functional

$$J[y] = \int_a^b L(x,y(x),y'(x))dx$$

over a family of spatial curves, $y(x)$, subject to constraints. We can also view these curves as functions of time, $t \in \mathbb{R}$ so that $y(x)$ becomes $y(t)$. The derivative $\frac{dy}{dx}$ now becomes the velocity vector $\dot{y} = \frac{dy}{dt}$ and we treat it as the decision variable that we call the *control.* So in this case, we seek a control $u = \dot{y}$ that minimizes the cost functional $J[y]$ subject to $(y, \dot{y})$ satisfying a differential equation of the form

$$M(t,y(t),\dot{y}(t)) = 0, \quad y(a) = y_0$$

This clearly has the same structure as the Calculus of Variations problem we considered in the preceding section, but because the decision variable $\dot{y}$ is treated as a ”control” input, it becomes an *optimal control problem.* This section shows how the Calculus of Variations is used to solve such optimal control problems.

We now want to minimize

$$J[u] = \int_{t_0}^{t_f} L(t,x(t),u(t))dt + K(x(t_f))$$

subject to the constraints

$$\dot{x}(t) = f(t,x(t),u(t)), \quad x(t_0) = x_0$$

with fixed time points $t_0 < t_f$. We assume $L$ and $f$ are continuous in $t$ and have continuous derivatives in $x$ and $u$. The Lagrangian, $L$, in the cost functional is also called the problem’s *running cost* and the function $K$ is called the problem’s *terminal cost*. The optimal control minimizes the total running cost with a terminal penalty $K(x(t_f))$ that is applied at the fixed terminal time $t_f$. This optimal control must satisfy the differential equation constraint $\dot{x} = f(t,x,u)$ in a point-wise manner and so it is similar to the non-integral constraint problems we discussed in the preceding section. In that section we saw the original constrained problem was equivalent to an unconstrained problem in which the Lagrangian was augmented by the equality constraint having Lagrange multipliers that are functions. We will also refer to these Lagrange multipliers as *costates* and denote them as $p : [t_0, t_f] \to \mathbb{R}^n$. The augmented cost functional then becomes

$$J_a[u,p] = \int_{t_0}^{t_f} \left[ L(t, x(t), u(t)) + p^T(t) \left( \dot{x}(t) - f(t, x(t), u(t)) \right) \right] dt + K(x(t_f))$$

The costate function $p$ plays the same role as the Lagrange multiplier did in the finite dimensional optimization problem with equality constraints.

To solve this problem, we will find it convenient to introduce a function called the *Hamiltonian*

$$H(t, x, u, p) = p^T(t) f(t, x, u) - L(t, x, u)$$

This allows us to rewrite the cost functional in terms of the Hamiltonian

$$J[u] = \int_{t_0}^{t_f} \left[ p^T \dot{x} - H(t, x, u, p) \right] dt + K(x(t_f)) \tag{14}$$

and to find necessary conditions for optimality we compute the first variation of this form of the cost functional at a local minimizing solution $u^*$.

Equation 14 has three distinct terms and so our expression for the first variation, $\delta J|_{u^*}[\eta]$, of $J$ with respect to the minimizer $u^*$ will also have three terms. Let $\xi \in C^1$ denote an admissible perturbation of the control $u^*$. This means

$$u(t) = u^*(t) + \alpha \xi(t)$$

This perturbation in $u$ will generate an admissible perturbation, $\eta$, of the state so that

$$x(t) = x^*(t) + \alpha \eta(t) + o(\alpha)$$

The state perturbation $\eta$ depends on the control perturbation $\xi$ through a differential equation. In particular, we can see from our equation for $x(t)$ that the derivative of

$x$ with respect to $\alpha$ will be $x_\alpha(t,0) = \eta(t)$. So the derivative of $\eta$ is

$$\begin{aligned}
\dot{\eta}(t) &= x_{\alpha t}(t,0) = x_{t\alpha}(t,0) = \left.\frac{d}{d\alpha}\right|_{\alpha=0} \dot{x}(t,\alpha) \\
&= \left.\frac{d}{d\alpha}\right|_{\alpha=0} f(t, x(t,\alpha), u^* + \alpha\xi(t)) \\
&= f_x(t, x(t,0,u^*(t))x_\alpha(t,0) + f_u(t, x(t,0), u^*(t))\xi(t) \\
&= f_x(t, x(t,0), u^*(t))\eta(t) + f_u(t, x^*(t), u^*(t))\xi(t) \\
&= f_x|_* \, \eta + f_u|_* \, \xi \\
&= A_*\eta + B_*\xi
\end{aligned}$$

We used the notation $f_x|_*$ to denote the partial derivative of $f$ evaluated along the optimal trajectory $x^*$. Note, however that $f_x$ and $f_u$ are time varying linear operators. So the state perturbation $\eta$ is generated by the control perturbation $\xi$ through the linear time-varying differential equation given above.

This variation in the state and control is the basis for computing the first variation of the cost functional $J$. From equation (14) we see that $J$ consists of three terms, so we will compute the variation of each term separately and then combine them. The first variation of the third term in equation (14) will be

$$\begin{aligned}
K(x(t_f)) - K(x^*(t_f)) &= K(x^*(t_f) + \alpha\eta(t_f) + o(\alpha)) - K(x^*(t_f)) \\
&= K(x^*(t_f)) + \langle K_x(x^*(t_f)), \alpha\eta(t_f)\rangle + o(\alpha) - K(x^*(t_f)) \\
&\approx \langle K_x(x^*(t_f)), \alpha\eta(t_f)\rangle
\end{aligned}$$

The first variation of the second term in equation (14) is obtained by integrating the variation in the Hamiltonian. In particular, we can see that

$$\begin{aligned}
H(t,x,u,p) - H(t,x^*,u^*,p) &= H(t, x^* + \alpha\eta + o(\alpha), u^* + \alpha\xi, p) - H(t,x^*,u^*,p) \\
&\approx \langle H_x(t,x^*,u^*,p,\alpha\eta\rangle + \langle H_u(t,x^*,u^*,p), \alpha\xi\rangle
\end{aligned}$$

The variation in the first term of (14) will take the form

$$\begin{aligned}
\int_{t_0}^{t_f} \langle p(t), \dot{x}^*(t)\rangle dt &= \langle p(t), x(t) - x^*(t)\rangle|_{t_0}^{t_f} - \int_{t_0}^{t_f} \langle \dot{p}, x(t) - x^*(t)\rangle dt \\
&\approx \langle p(t_f), \alpha\eta(t_f)\rangle - \int_{t_0}^{t_f} \langle \dot{p}(t), \alpha\eta(t)\rangle dt
\end{aligned}$$

Summing the last three expressions yields the following formula for $\delta J$,

$$\begin{aligned}\delta J|_{u^*}[\xi] &= -\int_{t_0}^{t_f} \left(\langle \dot{p} + H_x(t,x^*,u^*,p), \eta\rangle + \langle H_u(t,x^*,u^*,p),\xi\rangle\right) dt \\ &\quad + \langle K_x(x^*(t_f) + p(t_f), \eta(t_f)\rangle\end{aligned}$$

where the state perturbation, $\eta$, is related to the control perturbation $\xi$ through the time-varying differential equation

$$\dot{\eta}(t) = A_*(t)\eta(t) - B_*(t)\xi(t), \quad \eta(t_0) = 0$$

Our usual first order necessary condition at the local minimizer $u^*$ requires $\delta J|_{u^*}[\xi] = 0$. This should hold for any $p(t)$ we choose. So we will find it convenient to select a $p(t)$ which greatly simplifies the problem. In particular, let's assume that $p(t)$ satisfies

$$\dot{p}(t) = -H_x(t, x^*(t), u^*(t), p(t)), \quad p(t_f) = -K_x(x^*(t_f))$$

We will denote this particular choice for $p$ (aka co-state) as $p^*$. With this choice, the first variation of $J$ in equation (15) simplifies to

$$\delta J|_{u^*}[\xi] = -\int_{t_0}^{t_f} \langle H_u|_*, \xi\rangle dt = 0$$

where $H|_* = H(t, x^*, u^*, p^*)$. From the fundamental lemma of the calculus of variations we know this means

$$H_u(t, x^*(t), u^*(t), p^*(t)) = 0$$

for all $t \in [t_0, t_f]$. In light of our definition for the Hamiltonian we can rewrite our control system and the differential equation for $p^*$ as

$$\begin{aligned}\dot{x}^* &= H_p|_* \\ \dot{p}^* &= -H_x|_*\end{aligned}$$

We now have a complete characterization of the necessary conditions for this "optimal control problem" which is summarized in the following theorem

THEOREM 3. *Consider the problem of minimizing*

$$J[u] = \int_{t_0}^{t_f} L(t, x(t), u(t))dt + K(x(t_f))$$

*subject to*

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_0) = x_0$$

*by selection a control $u \in C[t_0, t_f]$ assuming $t_0 < t_f$. Assume $L$ and $f$ are continuous in $t$ and $C^1$ in $x$ and $u$. Assume $K$ is $C^1$ in $x$. If $u^* \in C[t_0, t_f]$ is a local minimizer and $x^* \in C^1[t_0, t_f]$ denotes the state trajectory generated by $u^*$, then there is a vector function $p^* \in C^1[t_0, t_f]$ such that the triple $(u^*, x^*, p^*)$ satisfies*

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)), \quad x(t_0) = x_0 \\ \dot{p}(t) &= -L_x(t, x(t), u(t)) - (f_x)^T p(t), \quad p(t_1) = -K_x(x(t_f)) \\ 0 &= L_u(t, x, u) + (f_u)^T p(t) \end{aligned}$$

*for all $t \in [t_0, t_f]$.*

**Remark:** The optimality conditions in the above theorem consist of $m$ algebraic equation (for $u \in \mathbb{R}^m$) and $2n$ ordinary differential equations with boundary conditions. There are $2n + m$ unknowns ($x$, $u$, and $p$) that must be determined from the equations in the theorem.

One issue we have with this problem formulation is that the $n$ boundary conditions for the state equation $\dot{x} = f(t, x, u)$ are specified at the initial time $t_0$ while the boundary conditions for the co-state equation $\dot{p} = -L_x - (f_x)^T p(t)$ are specified at the terminal time $t_f$. Such ODE's are called two point boundary value problems (TPBVP) and their solution can be difficult to find.

**Example:** Consider the problem of minimizing

$$J[u] = \int_0^1 \left( \frac{1}{2} u^2(t) - x(t) \right) dt$$

subject to

$$\dot{x}(t) = 2(1 - u(t)), \quad x(0) = 1$$

The Hamiltonian is

$$H(x, u, p) = \frac{1}{2} u^2 - x + 2p(1 - u)$$

and the two differential equations we need $x^*$ and $p^*$ to satisfy are

$$\begin{aligned} \dot{x}^*(t) &= H_p = 2(1 - u^*), \quad x^*(0) = 1 \\ \dot{p}^*(t) &= -H_x = 1, \quad p^*(1) = 0 \end{aligned}$$

The co-state's differential equation implies

$$p^*(t) = t - 1$$

and from the last condition in the theorem using this $p^*$ we get

$$0 = L_u + f_u p^* = u^* + (-2)p^*$$

which implies the optimal control is

$$u^*(t) = 2(t-1)$$

for all $t \in [0,1]$.

The preceding discussion showed how the variational methods in the calculus of variations give rise to equations whose satisfaction is necessary for the local minimizer. There are, however, significant limitations to the method we used to derive this result. The first difficulty is our assumption that $H$ is differentiable. This would also mean that $f$ is differentiable and there are numerous optimal control problems where $f$ is discontinuous. We assumed that the control perturbations $\alpha\xi(t)$ were small (i.e. $\alpha$ is small). This is overly restrictive because this neglects "bang-bang" solutions where the control switches back and forth between extreme values in $U$. These issues greatly limit the utility of the variational method in solving optimal control problems. A more general framework that draws on some nonstandard techniques can address these issues. It is known as Pontryagin's maximum principle (PMP) [Liberzon (2012)]. PMP actually shows that the basic idea of using co-states can still be used to find the optimal control, though the necessary conditions are a bit different and will not be covered in this course.

## 4. Numerical Methods for Optimal Control

Optimal control problems are often solved using numerical methods [Rao (2009)]. This is particularly true in *model predictive control* (aka *receding horizon control*). Receding horizon control (RHC) [Mayne and Michalska (1988)] seeks to optimally control a system

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_0) = x_0$$

in a manner that minimizes an infinite horizon cost functional

$$J[u] = \int_{t_0}^{\infty} L(t, x(t), u(t))dt$$

subject to inequality constraints on the state and control. RHC solves this problem by solving the simpler finite horizon problem over the finite horizon of length $T$ with cost functional

$$J_T[u] = \int_{t_0}^{t_0+T} L(t, x(t), u(t))dt + K(x(t_0 + T))$$

The resulting control, $u_T : [t_0, t_0+T] \to \mathbb{R}^m$, is used over a shorter time interval of duration $\Delta < T$. Then at time $t_0 + \Delta$, we reset the initial time to this current time and then re-solve the finite horizon problem again. A key issue in the use of RHC methods is the stability of the resulting control. But this problem is addressed through the appropriate selection of the terminal cost $K$. A key to the practical use of this method, however, is the ability to numerically solve the finite horizon control problem, which is the topic to be covered below.

There are two types of numerical methods for optimal control; *direct* and *indirect* solution methods. Direct solution methods find the local minimizer by constructing a sequence of solutions that converge to that minimum. Indirect solution methods solve the problem by numerically solving the necessary conditions for optimality. In many of these methods, one needs to compute the values of the functionals subject to differential equation constraints and one needs to compute the functional's gradient. So we first review methods for evaluating functionals and their gradients.

**Numerical Evaluation of Functional and its Gradient:** Let us first consider the functional

$$J[u] = \int_{t_0}^{t_f} L(t, x, u)dt + K(x(t_f)) \tag{15}$$

This version of the functional is used in the traditional Bolza formulation of the optimal control problem. We are going to show that this Bolza form can be simplified to a form known as the Mayer formulation which is more convenient for numerical solution.

We reduce the Bolza problem to a Mayer problem by first noting that the terminal cost into the integral can be written as

$$\begin{aligned} K(x(t_f)) &= K(x(t_0)) + \int_{t_0}^{t_f} \frac{d}{dt} K(x(t))dt \\ &= K(x(t_0)) + \int_{t_0}^{t_f} \langle K_x(x(t)), f(t,x,u)\rangle dt \end{aligned}$$

The first term is constant and can be neglected since it has no impact on the optimization. The second term obviously can be folded into the first term of equation (15) to obtain a new Lagrangian. This would mean that

$$J[u] = \int_{t_0}^{t_f} \overline{L}(t,x,u)dt$$

When we define our optimal control problem using this functional we obtain the Lagrange formulation of the problem. But let us introduce a new variable $\phi$ such that

$$\dot{\phi}(t) = \overline{L}(t, x(t), u(t)), \quad \phi(t_0) = 0$$

This would imply that the the integral term in the above functional can be written is

$$J[u] = \int_{t_0}^{t_f} \overline{L}(t,x,u)dt = \phi(t_f)$$

In other words, our cost functional is simply a function of the terminal time $t_f$. If we use this form of the functional for the optimal control problem we obtain the Mayer formulation. The key point is that all of these forms are equivalent to each other so we can use the simpler one for the purposes of numerical convenience. In our case, we will start from the Mayer formulation.

In our case, the control, $u$, is what we are trying to find. So if we are given a set of basis functions $\{u_{bi}(t)\}_{i=1}^{N}$, then $u$ can be written as

$$u = \sum_{i=1}^{N} \theta_i u_{bi}(t)$$

In this case the Mayer problem's functional is now parameterized with respect to the real vector $\theta$

$$J[u] = J(\theta) = \phi(x(t_f), \theta)$$

where the state satisfies

$$\dot{x}(t) = f(t, x, \theta)$$

where $x(t_0) = h(\theta)$ and $t \in [t_0, t_f]$. Our problem is to compute the value of $J(\theta)$ as well at its gradient $\nabla J(\theta)$. Evaluating the functional requires that we first numerical solve the initial value problem. We then present three methods for computing $\nabla J(\theta)$.

Evaluating the functional for a given parameter vector $\theta$ requires that we compute $x(t_f)$ that satisfies $\dot{x} = f(t, x, \theta)$ with initial value $x(t_0) = h(\theta)$. This is an initial value problem that we solve by first discretizing time with respect to a fixed step size $h$. This means that we compute a discrete time approximation of the state trajectory. A popular method for computing this trajectory is the Runge-Kutta (RK) method. The RK method is implemented in Matlab's `ode` function and can be used to numerical solve the IVP with a variety of properties.

The simplest way of determining $\nabla J(\theta)$ uses finite difference approximations of the form

$$\nabla_{\theta_j} J(\theta) \approx \frac{J(\theta_1, \ldots, \theta_j + \delta\theta_j, \ldots \theta_M) - J(\theta)}{\delta\theta_j}$$

for each $j = 1, 2, \ldots, M$ where $\delta\theta_j$ is chosen perturbation of the parameter $\theta_j$. A major limitation of this approach is its accuracy.

An alternative approach to address this accuracy problem is based on a sensitivity method. Consider the initial value problem for a given parameter $\theta$

$$\dot{x}(t; \theta) = f(t, x(t; \theta), \theta), \quad x(t_0; \theta) = h(\theta)$$

The first order state sensitivity function measures the sensitivity of the state with regard to parameter perturbations. In particular, $x_{\theta_j}(t; \theta)$ for $j = 1, 2, \ldots, M$ is a function of time that is the partial derivative of the state with respect to parameter $\theta_j$. This state sensitivity function satisfies the differential equation

$$\begin{aligned} \dot{x}_{\theta_j}(t; \theta) &= f_x(t, x(t; \theta), \theta) x_{\theta_j}(t; \theta) + f_{\theta_j}(t, x(t; \theta), \theta) \\ x_{\theta_j}(t_0; \theta) &= h_{\theta_j}(\theta) \end{aligned}$$

The preceding equations are called the *sensitivity equation with respect to parameter $\theta_j$*. They are, in general, linear differential equations that we can integrate forward to get the terminal state sensitivity $x_\theta(t_f)$. Once the sensitivity functions are known at time $t_f$, and since the Mayer function $\phi$ is $C^1$, the gradient, $\nabla_{\theta_j} J(\theta)$ can be computed as

$$\nabla_{\theta_j}\phi(x(t_f),\theta) = J_x(x(t_f;\theta),\theta)^T x_{\theta_j}(t_f,\theta) + \phi_{\theta_j}(x(t_f;\theta),\theta)$$

for each $j = 1, 2, \dots, M$.

This leads to the following procedure for calculating both the value and the gradient of $J$ at $\theta$ as follows:

- *state and sensitivity numerical integration:* $t_0 \to t_f$

$$\begin{array}{rcllrcl}
\dot{x}(t) & = & f(t,x(t);\theta) & & x(t_0) & = & h(\theta) \\
\dot{x}_{\theta_1}(t) & = & f_x(t,x(t),\theta)x_{\theta_1}(t) + f_{\theta_1}(t,x(t),\theta), & & x_{\theta_1}(t_0) & = & h_{\theta_1}(\theta) \\
\vdots & & & & \vdots & & \\
\dot{x}_{\theta_M}(t) & = & f_x(t,x(t),\theta)x_{\theta_M}(t) + f_{\theta_M}(t,x(t);\theta), & & x_{\theta_M}(t_0) & = & h_{\theta_M}(\theta)
\end{array}$$

- *Functional and Gradient Evaluation:*

$$\begin{array}{rcl}
J(\theta) & = & \phi(x(t_f),\theta) \\
\nabla_{\theta_1} J(\theta) & = & \phi(x(t_f),\theta)^T x_{\theta_1}(t_f) + \phi_{\theta_1}(x(t_f),\theta) \\
 & \vdots & \\
\nabla_{\theta_M} J(\theta) & = & \phi_x(x(t_f),\theta)^T x_{\theta_M}(t_f) + \phi_{theta_M}(x(t_f),\theta)
\end{array}$$

Note that the state and sensitivity equations are solved simultaneously so that local error control can be performed on both the state and state sensitivity variables. The size of this state/sensitivity system grown as $(n+1)M$, which can be computationally intractable if both $n$ and $M$ are large. Methods have been developed to address these issues and there are a number of codes available for forward sensitivity analysis of IVPs.

**Indirect Methods:** Indirect methods use iterative procedures based on successive linearization to find a solution to a system of necessary optimality conditions (NOCs). A nominal solution is chosen that satisfies part of the NOCs, then this nominal solution is modified by successive linearization to meet the remaining NOCs. One approach for doing this are the indirect shooting methods which we describe below.

Consider a problem to find $u^* \in C^1[t_o, T]$ and $t_f^* \in [t_0, T)$ to

$$\begin{array}{rl} \text{minimize:} & J[u,t_f] = \int_{t_0}^{t_f} L(t,x,u)dt + \phi(t_f, x(t_f)) \\ \text{subject to:} & \dot{x}(t) = f(t,x(t),u(t)), \quad x(t_0) = x_0 \\ & \psi_i(t_f, x(t_f)) = 0, \quad k = 1,2,\ldots,n_\psi \end{array}$$

If this problem has a solution $(u^*, t_f^*)$, then there must exist $u^*, x^*, \lambda^*, \nu^*$, and $t_f^*$ that satisfy the Euler-Lagrange equations

$$\begin{aligned} \dot{x}^*(t) &= H_\lambda(t, x^*(t), u^*(t), \lambda^*(t)), \quad x^*(t_0) = x_0 \\ \dot{\lambda}^*(t) &= -H_x(t, x^*(t), u^*(t), \lambda^*(t)), \quad \lambda^*(t_f^*) = \Phi_x(t_f^*, x^*(t_f^*)) \\ 0 &= H_u(t, x(t), u(t), \lambda(t)) \end{aligned}$$

for all $t \in [t_0, t_f^*]$ along with the transversal conditions

$$\begin{aligned} \psi(t_f^*, x^*(t_f^*)) &= 0 \\ \Phi_t(t_f^*, x(t_f^*)) + H(t_f^*, x^*(t_f^*), u^*(t_f^*), \lambda^*(t_f^*)) &= 0 \end{aligned}$$

with $\Phi = \phi + (\nu^*)^T\psi$ and $H = L + (\lambda^*)^T f$.

Observe that if the costate values $\lambda^*(t_0)$, the Lagrange multipliers $\nu^*$, and the terminal time $t_f^*$ were known, the Euler-Lagrange equations could be integrated forward in time. So the idea of indirect shooting is to guess the values of the initial costate, Lagrange multipliers, and terminal time and then iteratively improve these estimates to satisfy the adjoint terminal conditions and the transversal conditions. In other words one wants to find $(\lambda^*(t_0), \nu^*, t_f^*)$ such that

$$b(\lambda^*(t_0), \nu^*, t_f^*) = \begin{bmatrix} \lambda^* + \phi_x + (\nu^*)^T\phi_x \\ \psi \\ L + (\lambda^*)^T f + \phi_t + \nu^*\phi_t \end{bmatrix}_{t=t_f^*} = 0$$

This can be done using a Newton-Raphson type algorithm.

**Direct Method:** Direct methods discretize the control problem and then apply mathematical programming codes to the resulting finite-dimensional optimization problem. These methods use only control and/or state variables as decision variables and dispense completely with the costate.

One direct method is known as the sequential method. We consider the following optimal control problem to illustrate its use

$$\begin{array}{rl} \text{minimize:} & J[u] = \int_{t_0}^{t_f} L(t, x(t), u(t), \theta)dt + \phi(x(t_f, \theta)) \\ \text{with respect to:} & u^* \in \widehat{C}^1[t_0, t_f], \quad \theta^* \in \mathbb{R}^{n_\theta} \\ \text{subject to:} & \dot{x}(t) = f(t, x(t), u(t), \theta), \quad x(t_0) = h(\theta) \\ & \psi_j(x(t_f), \theta) = 0, \quad j = 1, \dots, n_\psi \\ & \kappa_j(x(t_f), \theta) \leq 0, \quad j = 1, \dots, n_\kappa \\ & g_j(t, x(t), u(t), \theta) \leq 0, \quad j = 1, \dots, n_g \\ & u(t) \in [\underline{u}, \overline{u}], \quad v \in [\underline{v}, \overline{v}] \end{array}$$

In direct sequential methods, the control variables $u$ are parameterized by a finite set of parameters and the optimization is carried out in the parameter space. A convenient way to parameterize the controls is by subdividing the optimization horizon, $[t_0, t_f]$ into $n_s \geq 1$ control stages

$$t_0 < t_1 < t_2 < \cdots < t_{n_s} = t_f$$

and using low-order polynomials, $\tilde{u}(t, \omega)$ on each interval with coefficient vector $\omega$ so that

$$u(t) = \tilde{u}^k(t, \omega^k), \text{for } t_{k-1} \leq t \leq t_k$$

with $\omega^k$ being the parameters for that subinterval. In practice, Lagrange polynomials are used to approximate the controls so in stage $k$ the $j$th control variable is

$$u_j(t) = \tilde{u}_j^k(t, \omega^k) = \sum_{i=0}^{M} \omega_{ij}^k \phi_i^{(M)}(\tau^{(k)}), \quad t \in [t_{k-1}, t_k]$$

where $\tau^{(k)} = \frac{t - t_{k-1}}{t_k - t_{k-1}} \in [0, 1]$ denotes a normalized time in stage $k$ and $\phi_i^{(M)}(\cdot)$ denotes the Lagrange polynomial of order $M$

$$\phi_i^{(M)}(\tau) = \begin{cases} 1 & \text{if } M = 0 \\ \displaystyle\prod_{q=0, q\neq i}^{M} \frac{\tau - \tau_q}{\tau_i - \tau_q} & \text{if } M \geq 1 \end{cases}$$

With this parameterization of the controls, the original problem is transformed into a finite dimensional optimization problem

$$\begin{array}{ll} \text{minimize:} & \sum_{k=1}^{n_s} \int_{t_{k-1}}^{t_k} L(t, x(t), \tilde{u}(t, \omega^k), \theta) dt + \phi(x(t_{n_s}), \theta) \\ \text{subject to:} & \dot{x}(t) = f(t, x(t), \tilde{u}(t, \omega^k), \theta), \quad t \in [t_{k-1}, t_k], \quad k = 1, \dots, n_s \\ & x(t_0) = h(\theta) \\ & \psi(x(t_{n_s}), \theta) = 0 \\ & \kappa(x(t_{n_s}), \theta) \leq 0 \\ & g(t, x(t), \tilde{u}(t, \omega^k), \theta) \leq 0, \quad t \in [t_{k-1}, t_k], \quad k = 1, \dots, n_s \\ & \omega^k \in [\underline{\omega}, \overline{\omega}], \theta \in [\underline{\theta}, \overline{\theta}] \end{array}$$

where the decision variables are the parameters $(\omega^1, \dots, \omega^{n_s}, \theta)$.

## 5. Dynamic Programming

The preceding sections examined the variational approach to optimal control. This section examines an alternative approach to the problem called *dynamic programming*. Dynamic programming uses a *value function*, $V : \mathbb{R} \times X \to \mathbb{R}$, whose values $V(x, t)$ equal the optimal cost if the system started in state $x$ at time instant $t$. The state space is denoted as $X$ and in our following discussion it may either be $\mathbb{R}^n$, $\mathbb{Z}^n$, or event a finite discrete set. The resulting value function allows us to identify sufficient conditions for optimality and more importantly the optimal controls take the form of a state feedback law. Equations for the Value Function are derived from a *principle of optimality* that is extremely general so it can be used for deterministic, discrete-time, and discrete-event systems. For continuous-time systems the Value function is the solution to a partial differential equation known as the Hamilton-Jacobi-Bellman (HJB) equation.

**5.1. Generalized Principle of Optimality:** Let $X$ and $U$ be linear space called the state and control space, respectively. Let $G$ be a group[1] with a strong order relation, $<$, such that for all $s, t \in G$, one of the following relations always holds: $s < t$, $t < s$, or $s = t$ (aka trichotomy law). Let $\mathcal{X}$ denote all functions $x : G \to X$ and $\mathcal{U}$ denote all functions $u : G \to U$. Let $\mathcal{U}_{[s,t]}$ and $\mathcal{X}_{[s,t]}$ denote

[1] A group is a mathematical system $(X, +)$ with a binary operation of addition that satisfies certain properties.

the restriction of $\mathcal{X}$ and $\mathcal{U}$ to functions whose domain is $[s,t]$. The group, $G$, represents *time*, but because of the generality of our framework, the results apply to discrete-time and continuous-time systems.

Consider a doubly indexed family of operators for indices $s, t \in G$

$$\Phi_{ts} \,:\, X \times \mathcal{U}_{[s,t]} \to X$$

and define any ordered pair $(t_0, x_0) \in G \times X$ as an *event*. The state trajectory generated by $u \in \mathcal{U}_{[t_0,t_f]}$ for event $(t_0, x_0)$ under system $\Phi_{ts}$ is the function $x : [t_0, t_f] \to X$ such that

$$x(t) = \Phi_{t,t_0}(x_0 \,|\, u)$$

for all $t_0 \leq t \leq t_f$. Clearly $\Phi$ is what we think of as a *transition operator* for the system. If $G = \mathbb{Z}$ (the set of integers), then the system is discrete time. If $G = \mathbb{R}$, the system is continuous-time. The spaces $X$ and $U$ can either be subsets of a Euclidean space, or they can be finite sets (discrete). In some cases we can equip $X$ and $U$ with a probability measure so they become probability spaces and our system becomes a Markov process.

Let us now introduce a payoff/penalty functional

$$J \,:\, \mathcal{U}_{[t_0,t_f]} \times G \times G \times X \to \mathbb{R}$$

such that $J[u \,|\, t_0, t_f, x_0]$ is the payoff/penalty received by system $\Phi_{s,t}$ using $u \in \mathcal{U}_{[t_0,t_f]}$. Given a control $u \in \mathcal{U}_{[t_0,t_f]}$ let $x$ be the state trajectory generated by $\Phi_{t_f,t_0}$ under $u$. We assume there exists a function $M : X \to \mathbb{R}$ and functional $Q : \mathcal{U}_{[t_0,t_f]} \times G \times G \times X \to \mathbb{R}$ such that the total payoff (reward) for any $u \in \mathcal{U}_{[t_0,t_f]}$ with $x$ generated under $u$ is

$$J[u \,|\, t_0, t_f, x_0] = M(x(t_f)) + Q[u \,|\, t_0, t_f, x_0]$$

We also assume that $Q$ is additive over time so that for all $s \in [t_0, t_f]$ we have

$$Q[u \,|\, t_0, t_f, x_0] = Q[u \,|\, t_0, s, x_0] + Q[u \,|\, s, t_f, x_0]$$

One obvious choice for $Q$ would be the integral functionals we considered in our calculus of variation problems

$$Q[u \,|\, t_0, t_f, x_0] = \int_{t_0}^{t_f} L(x(\tau), u(\tau))d\tau$$

Given the 2 parameter group of system transition operators, $\Phi_{s,t}$, and a payoff functional $J$, the *value function* is a function $V : G \times X \to \mathbb{R}$ such that

$$V(t,x) = \inf_{u \in \mathcal{U}_{[t,t_f]}} J[u \,|\, t, t_f, x]$$

If the infimum is achieved by some $u^* \in \mathcal{U}_{[t,t_f]}$ then $u^*$ is the optimal control.

Note that $u^*$ is a function of the event $(t,x)$. But since we determine $V$ for all possible events, we can see that $u^*$ is actually a function of time $t$ and the state $x$. In other words, having the value function allows us to identify a state-feedback control law, rather than an open-loop control signal as was done with the variational methods. The following theorem characterizes the behavior of the value function under any admissible control using a recursive relationship. This recursion essentially says that the Value function at event $(t_0, x)$ is bounded above by the sum of the value function at $(s, \Phi_{s,t_0}(x : u_{[t_0,s)})$ and the running cost $Q[u_{[t_0,s)} : t_0, s, x]$ incurred in going from $(t_0, x)$ to $(s, \Phi_{s,t_0}(x : u_{[t_0,s)})$.

THEOREM 4. *Let $s, t_0, t_f \in G$ be such that $t_0 \leq s \leq t_f$. For any admissible control, we have*

$$V(t_0, x) \leq Q\left[u_{[t_0,s)} : t_0, s, x\right] + V(s, \Phi_{s,t_0}(x : u_{[t_0,s]}))$$

**Proof:** Assume $s, t_0, t_f \in G$ and $u \in \mathcal{U}_{[t_0,t_f]}$ are given. For any $\epsilon > 0$ there exists a control $v \in \mathcal{U}_{[s,t_f)}$ such that

$$J[v : s, t_f, \Phi_{s,t_0}(x : u_{[t_0,s)}) < V_s(\Phi_{s,t_0}(x : u_{[t_0,s)}) + \epsilon$$

Now let $u_\epsilon \in \mathcal{U}_{[t_0,t_f]}$ be a control obtained by concatenating $u_{[t_0,s)]}$ with $v$ so that

$$u_\epsilon(t) = \begin{cases} u(t) & t_0 \leq t < s \\ v & s \leq t \leq t_f \end{cases}$$

From the definition of the value function $V(t_0, x)$ we have

$$\begin{array}{rcl} V(t_0,x) & \leq & J[u_\epsilon : t_0, t_f, x] \\ & = & Q[u_{[t_0,s)} : t_0, s, x] + J[v : s, t_f, \Phi_{s,t_0}(x : u_{[t_0,s)})] \end{array}$$

wich implies that

$$V(t_0, x) \leq Q\left[u_{[t_0,s]} : t_0, s, x\right] + V(s, \Phi_{s,t_0}(x : u_{[t_0,s]})) + \epsilon$$

In the limit as $\epsilon \to 0$ we obtain the theorem's conclusion. $\diamondsuit$

The recursion relationship given in the preceding theorem is necessarily satisfied with equality by the optimal control $u^*$. This equation is also known as the *principle of optimality*. Because we have proven it with regard to very general assumptions on the state/control sets and the time set, $G$, we refer to it as a *"generalized" principle of optimality* [Fleming and Rishel (1972)].

THEOREM 5. *Let the initial time $t_0$ and the final time $t_f$ be given. If the value function is achieved by $u^* \in \mathcal{U}_{[t_0,t_f]}$ then for any $s \in [t_0, t_f]$ we have*

$$V(t_0, x) = Q\left[u^*_{[t_0,s]} \,:\, t_0, s, x\right] + V(s, \Phi_{s,t_0}(x \,:\, u^*_{[t_0,s]}))$$

**Proof:** Let $u^*$ be optimal for event $(t_0, x)$. By the additivity of $Q$ we rewrite $V(t_0, x)$ as

$$\begin{aligned} V(t_0,x) &= \min_{u\in\mathcal{U}_{[t_0,t_f]}} J[u \,:\, t_0,t_f,x] = J[u^* \,:\, t_0,t_f,x] \\ &= \min_{u\in\mathcal{U}_{[t_0,t_f]}} [M(x(t_f)) + Q\,[u \,:\, t_0,t_f,x]] \\ &= \min_{u\in\mathcal{U}_{[t_0,t_f]}} \left\{M(x(t_f)) + Q\,[u \,:\, t_0,s,x] + Q\left[u \,:\, s,t_f,\Phi_{s,t_0}(x \,:\, u_{[t_0,s]})\right]\right\} \\ &= \min_{u\in\mathcal{U}} \left\{Q\,[u \,:\, t_0,s,x] + J\left[u_{[s,t_f]} \,:\, s,t_f,\Phi_{s,t_0}(x \,:\, u_{[t_0,s]})\right]\right\} \end{aligned}$$

Assume there exists another control $\widehat{u}_{[s,t_f]}$ such that

$$J[\widehat{u}_{[s,t_f]} : t_f, \Phi_{s,t_0}(x : u_{[t_0,s]})] \leq J[u^*_{[s,t_f]} : s, t_f, \Phi_{s,t_0}(x : u_{[t_0,s]})]$$

Then we can construct $v \in \mathcal{U}_{[t_0,t_f]}$ such that

$$v(t) = \begin{cases} u^*(t) & \text{for } t \in [t_0, s) \\ \widehat{u}(t) & \text{for } t \in [s, t_f] \end{cases}$$

for which

$$V(t_0, x) \geq J[v : t_0, t_f, x]$$

which would violate the optimality of $u^*$, thereby generating a contradiction. $\diamondsuit$

The following theorem provides conditions under which our recursive equation for the value function becomes *sufficient* for optimality. This result is also known as the verification theorem since it verifies the optimality of a given control.

THEOREM 6. *Assume $\widehat{V} \,:\, [t_0, t_f] \times X \to \mathbb{R}$ is any function such that*

- *for all $s$ and $t$ with $t_0 \leq t \leq s \leq t_f$ and all $x \in X$ we have*

  $$\widehat{V}(t,x) \leq Q\left[u_{[t,s]} : t, s, x\right] + \widehat{V}(s, \Phi_{s,t}(x : u_{[t,s]}))$$

  *for all admissible $u_{[t,s]}$*
- *For all $s$ and $t$ with $t_0 \leq t \leq s \leq t_f$ and all $x \in X$, there exists some control $u^*_{[t_0,t_f]}$ such that*

  $$\widehat{V}(t,x) = Q\left[u^*_{[t,s]} : t, s, x\right] + \widehat{V}(s, \Phi_{s,t}(s : u^*_{[t,s]}))$$

- *and $\widehat{V}(t_f, x) = M(x)$ for all $x \in X$*

*Then the value function $V(t,x)$ is achieved by $u^* \in \mathcal{U}_{[t_0,t_f]}$ and $V(t,x) = \widehat{V}(t,x)$.*

**Proof:**. Fix any $t \in [t_0, t_f]$ and $x \in X$. Consider a particular case of the first condition $(s = t_f)$ and use the third condition to deduce that for any $u \in \mathcal{U}_{[t,t_f]}$ that

$$\widehat{V}(t,x) \leq Q\left[u : t, t_f, x\right] + M(\Phi_{t_f,t}(x : u))$$

By the second condition there is some $u^* \in \mathcal{U}_{[t_0,t_f]}$ such that

$$\begin{aligned}\widehat{V}(t,x) &= Q\left[u^* : t, s, x\right] + \widehat{V}(s, \Phi_{s,t}(x : u^*_{[t,s]})) \\ &= Q\left[u^* : t, s, x\right] + M(\Phi_{t_f,t}(x : u^*))\end{aligned}$$

From the definition of the value function we obtain

$$\begin{aligned}V(t,x) &\leq J[u^* : t, t_f, x] \\ &= M(\Phi_{t_f,t}(x : u^*)) + Q[u^* : t, t_f, x] \\ &= \widehat{V}(t,x)\end{aligned}$$

So this shows that $V(t,x) \leq \widehat{V}(t,x)$. We also know that for any $\epsilon > 0$ there exists $u_\epsilon \in \mathcal{U}_{[t_0,t_f]}$ such that

$$\begin{aligned}V(t,x) + \epsilon &> J[u_\epsilon : t, t_f, x] \\ &= M(\Phi_{t_f,t}(x : u_\epsilon)) + Q[u_\epsilon : t, t_f, x] \\ &\geq \widehat{V}(t,x)\end{aligned}$$

Since this holds for all $\epsilon > 0$ we can conclude $V(t,x) \geq \widehat{V}(t,x)$. So we know that $V(t,x) = \widehat{V}(t,x)$ with

$$\begin{aligned} V(t,x) &= \min_u \left\{ Q[u : t, t_f, x] + M(\Phi_{t_f,t}(x : u)) \right\} \\ &= Q\left[u^* : t, t_f, x\right] + M(\Phi_{t_f,t}(x : u^*)) \end{aligned}$$

So that the value function is achieved by $u^*$. $\diamondsuit$

**5.2. Hamilton-Jacobi-Bellman Equation:** This subsection applies the principle of optimality to continuous-time systems. The main result is a local characterization of the principle that takes the form of a partial differential equation known as the *Hamilton-Jacobi-Bellman (HJB)* equation.

So consider the continuous time system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0$$

If $x : [t_0, t_f] \to \mathbb{R}^n$ is a solution to this IVP, then we know it can be written as

$$x(t) = \Phi_{0,t}(x_0 : u)$$

We are interested in finding the optimal control, $u^*$, that minimizes the cost functional

$$J[u \, : \, t_0, t_f, x_0] = \int_{t_0}^{t_f} L(x(\tau), u(\tau)) d\tau + M(x(t_f))$$

subject to $x$ being the state trajectory generated by the system under $u$.

We first develop necessary conditions for optimality for an infinitesimal variation of the value function. In particular, the principle of optimality says that for each optimal control $u^*$ starting from event $(t,x)$ the following relation holds

$$V(t,x) = \int_t^s L(x(\tau), u(\tau)) d\tau + V(s, x(s))$$

for all $s > t$. If the value function is differentiable then clearly

$$\frac{dV(s, x(s))}{ds} = \frac{\partial V}{\partial s} + \frac{\partial V}{\partial x}\dot{x} + o(\cdot)$$

which implies that

$$\frac{\partial V(t,x)}{\partial t} = -L(x(t), u^*(t)) - \frac{\partial V(t,x)}{\partial x} f(x(t), u^*(t))$$

Along arbitrary state trajectories (not necessarily generated by the optimal $u^*$), our earlier theorems say the value function satisfies

$$\frac{\partial V(t,x)}{\partial t} \geq -L(x(t),u(t)) - \frac{\partial V(t,x)}{\partial x} f(x,u)$$

Note that only the value of the control at time $t$ appears in these formulae. If the value function is achieved and if the functions are ”smooth enough” then the value function will satisfy

$$-\frac{\partial V}{\partial t} = \min_{u \in U} \left\{ L(x(t),u) + \frac{\partial V(t,x(t))}{\partial x} f(x(t),u) \right\}$$

with the value function $V$ at time $t_f$ being

$$V(t_f, x) = M(x)$$

The preceding observations can be summarized in the following theorem.

THEOREM 7. **HJB Theorem:** *Assume the value function $V$ is a $C^1$ function of variables $(x,t)$. Then $V$ satisfies the equation*

$$-\frac{\partial V}{\partial t}(x,t) = \min_{u \in U} \left\{ \frac{\partial V}{\partial x}(t,x) f(x,u) + L(x(t),u(t)) \right\}$$

*for all $x \in \mathbb{R}^n$ and $0 \leq t < T$ with terminal condition*

$$V(x,t_f) = M(x)$$

One interesting aspect of the HJB equation is that it generates state feedback controllers in a very natural way. This construction takes two steps

(1) Solve the HJB equation for all $(x,t)$ events. In other words determine the value function
(2) For each point $x \in \mathbb{R}^n$ and each time $t$ define a function $k : \mathbb{R}^n \times \mathbb{R} \to U$ that takes values

$$k(x,t) = \arg\min_{u \in U} \left\{ L(x,u) + \frac{\partial V(t,x)}{\partial x} f(x,u) \right\}$$

The control $u = k(x,t)$ generated in this way will achieve the value function and therefore be optimal.

**5.3. Linear Quadratic Regulator:** One of the best known examples illustrating the use of the HJB equation is the derivation of the linear quadratic regulator (LQR). In our linear systems lecture notes we derived the LQR control law using a completing the square argument. But this derivation assumed the controller was a state feedback controller. Using the HJB equation, we can prove the stronger result that shows the state feedback control law is indeed the ”optimal” law under all possible controllers. Consider the linear system

$$\dot{x}(t) \quad = \quad \mathbf{A}x(t) + \mathbf{B}u(t), \quad x(0) = x_0$$

and a quadratic cost functional

$$J[u] = \int_0^T (x^T \mathbf{Q} x + u^T \mathbf{R} u) dt + x^T(T) \mathbf{M} x(T)$$

where $\mathbf{Q}$, $\mathbf{M}$, and $\mathbf{R}$ are positive definite matrices. We will solve this problem using the HJB equation. In this case the Lagrangian is

$$L(x,u) = x^T \mathbf{Q} x + u^T \mathbf{R} u$$

and the system’s $f$ function is

$$f(x,u) = \mathbf{A}x + \mathbf{B}u$$

The terminal cost is

$$M(x(T)) = x^T(T) \mathbf{M} x(T)$$

The associated HJB equation for this problem is

$$-\frac{\partial V}{\partial t} \quad = \quad \min_{u \in \mathbb{R}^m} \left\{ x^T \mathbf{Q} x + u^T \mathbf{R} u + \frac{\partial V}{\partial x} (\mathbf{A}x + \mathbf{B}u) \right\}$$

Since this is a convex problem we see the optimal control must satisfy

$$\begin{aligned} 0 &= \frac{\partial}{\partial u} \left\{ x^T \mathbf{Q} x + u^T \mathbf{B} u + \frac{\partial V}{\partial x} (\mathbf{A}x + \mathbf{B}u) \right\} \\ &= 2\mathbf{R}u + \mathbf{B}^T \frac{\partial V}{\partial x} \end{aligned}$$

Solving for $u$, we find the optimal control is

$$u^*(t) = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{B}^T \frac{\partial V(t, x^*(t))}{\partial x}$$

We now need to determine the value function. Since the Lagrangian is quadratic we will assume that the value function is also quadratic with the form

$$V(t,x) = x^T \mathbf{P}(t) x$$

where $\mathbf{P} : [0,T] \to \mathbb{R}^{n\times n}$ is a symmetric matrix valued function of time. Note that

$$\frac{\partial V(t,x)}{\partial x} = 2\mathbf{P}(t)x$$

Putting this back into our previous equation yields

$$-x^T \dot{\mathbf{P}}(t) x = x^T (\mathbf{A}^T \mathbf{P}(t) + \mathbf{P}(t)\mathbf{A} + \mathbf{Q} - \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t))x$$

with boundary condition

$$V(T,x) = x^T \mathbf{P}(T) x = x^T \mathbf{M} x$$

Since these two equations must hold for all $x$ we can conclude that $\mathbf{P}(t)$ is the solution to the following matrix differential Riccati equation

$$-\dot{\mathbf{P}}(t) \quad = \quad \mathbf{A}^T \mathbf{P}(t) + \mathbf{P}(t)\mathbf{A} + \mathbf{Q} - \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t), \quad \mathbf{P}(T) = \mathbf{M}$$

Note that the boundary conditions for the ODE are at the terminal time, $T$. Solving this terminal value problem would give $\mathbf{P}(t)$ and the optimal control then takes the form

$$u^*(t) = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t)x(t)$$

Note that this is a state-feedback controller whose gains are time-varying. Because we were solving the finite horizon problem over $[0,T]$, the gains will tend to get larger as the system approaches the terminal time. We can also consider what happens as the final time $T$ goes to infinity. In this case, one can prove that the matrix-valued function of time, $\mathbf{P}(t)$, converges to a constant matrix, $\mathbf{P}$. Since this matrix is constant, we know $\dot{\mathbf{P}} = 0$ and so the matrix satisfies a matrix Riccati equation of the form

$$0 = \mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$$

This equation is nonlinear in the decision variable $\mathbf{P}$, but there are well known transformations that turn it into a linear function of $\mathbf{P}$ that can then be solved by conventional methods. We refer to this infinite horizon LQR controller as a *steady-state* LQR control law.

## 6. Stochastic Dynamic Programming

The previous derivation of the LQR assumed that the system state is fully available to the controller and that it is not corrupted by noise. This section considers the optimality of the control when the state is disturbed by a random process. Dynamic programming can still be applied to such systems through a stochastic version of the Hamilton-Jacobi Bellman equation. This section uses that stochastic HJB to solve the stochastic version of the linear quadratic regulator problem. The stochastic HJB is a stochastic differential equation (SDE) and so we first need to review basic results regarding stochastic differential equations.

**Stochastic Differential Equations:** Let $\mathbf{x}_t$ be a random process where $t \in [s,t]$ and consider the following partition of $[s,t]$,

$$s = t_0 < t_1 < t_2 < \cdots < t_n = t$$

We refer to the random process formed by

$$\mathbf{x}_{t_1} - \mathbf{x}_{t_0},\ \mathbf{x}_{t_2} - \mathbf{x}_{t_1},\ \cdots\ , \mathbf{x}_{t_n} - \mathbf{x}_{t_{n-1}}$$

as *increments*, $\Delta \mathbf{x}_i$ of $\mathbf{x}_t$. We say that $\mathbf{x}_t$ has independent increments if for any finite partition, $\Delta \mathbf{x}_i$ are statistically independent. We say the increments are *stationary* if $\mathbf{x}_{t+w} - \mathbf{x}_t$ has the same distribution for every $t$.

A stochastic process $\{\mathbf{x}_t\}_{t\geq 0}$ is called a *Brownian motion* or *Wiener process* if

- $\mathbf{x}_0 = 0$
- $\{\mathbf{x}_t\}_{t\geq 0}$ has stationary and independent increments,
- for all $t > 0$, $\mathbf{x}_t$ is normally distributed with mean $0$ and variance $c^2 t$.

To discuss stochastic differential equations, we need to develop a *calculus* for stochastic processes. One such calculus is the *Ito calculus* [Gikhman and Skorokhod (1972)] based on the *Ito stochastic integral*. Let us introduce a partition $0 < t_1 < t_2 < \cdots < t_n = t$ and define $\widehat{x}(t) = x(t_i)$ for $t \in [t_i, t_{i+1}]$. Define the integral

$$\int_0^t \widehat{x} dw = \sum_{i=1}^n x(t_i)(w(t_{i+1}) - w(t_i))$$

where $w$ is a standard Brownian motion. We let $\widehat{x}_n(t)$ be a sequence of random variables such that

$$\lim_{n\to\infty}\int_0^t (x-\widehat{x}_n)^2\,dw = 0$$

with probability one. This means that $\int_0^t \widehat{x}_n(t)dw$ converges with probability 1 to a limit that we denote as $\int xdw$. We call this limit the *Ito stochastic integral*.

So let us consider a random process $\{\mathbf{x}_t\}_{t\geq 0}$ such that for arbitrary $t_1$ and $t_2$

$$\mathbf{x}_{t_2} - \mathbf{x}_{t_1} = \int_{t_1}^{t_2} m(t)dt + \int_{t_1}^{t_2} \sigma dw$$

where $m$ and $\sigma$ are functions of time, $t$. The first integral is a standard (Riemann or Lebesgue) integral and the second integral is the Ito stochastic integral we defined above. This integral expression is sometimes called a *stochastic differential equation* and is often written in the more suggestive form,

$$d\mathbf{x} = m(t)dt + \sigma(t)dw$$

So when discussing the optimal control for a stochastic system, we will assume that the system can be represented as a stochastic differential equation. In particular, this means the original differential equation is driven by a Brownian motion.

**Remark:** Engineers are more acquainted with differential equations driven by *white noise processes*. We cannot really use this approach for continuous-time systems because the probability measure is not well defined over an uncountable set (i.e. the real line, time). This is why we use the notion of ”increments” and Brownian motions to formally describe how ”random noise” drives a differential equation.

**Stochastic Bellman Equation:** Let us define the following performance measure for the stochastic control problem

$$J[u] = \mathbb{E}\left\{\int_t^T L(x(\tau),u(\tau))d\tau + m(x(T)) \mid \mathbf{x}_t = x\right\}$$

We assume that $T$ is fixed and that $L$ and $m$ have the same properties they have in the deterministic LQR problem. The optimal stochastic control problem looks for a control $u$ that minimizes $J$ (the expected value of the cost) subject to the constraint that the state $\mathbf{x}_\tau$ satisfy the stochastic differential equation

$$d\mathbf{x} = f(\mathbf{x}, u, t)dt + G(\mathbf{x}, u, t)d\mathbf{w}$$

where $d\mathbf{w}$ is a Wiener increment with covariance matrix $W(t)dt$.

With regard to the above problem it is possible to develop a *stochastic* version of the principle of optimality. This stochastic optimality problem states that if $u^*(\tau)$ is optimal over the interval $[t, T]$, conditioned on the initial state $x(t)$, then $u^*(\tau)$ is necessarily optimal over the subinterval $[t + \Delta t, T]$ for any $\Delta t$ such that $T - t \geq \Delta t > 0$. As before we use this optimality principle to develop a stochastic version of the Hamilton-Jacobi Bellman equation.

Recall that the value function, $V^*$, is

$$V^*(t,x) = \min_u \mathbb{E}\left\{ \int_t^{t+\Delta t} Ld\tau + \mathbb{E}\left\{ \left( \int_{t+\Delta t}^T Ld\tau + m(x(T)) \right) \mid x(t+\Delta t) \right\} \mid x(t) \right\}$$

which is the average cost incurred by a system described by the above SDE using the "optimal" control. The stochastic optimality principle allows us to express $V^*$ in a recursive manner

$$(16)\ V^*(t,x) = \min_u \mathbb{E}\left\{ \int_t^{t+\Delta t} Ld\tau + V^*(x(t+\Delta t), t+\Delta t) \mid x(t) \right\}$$

where $x(t + \Delta t)$ is a random vector given by $x(t + \Delta t) = x + \Delta x$ with $\Delta x$ being a stochastic increment approximated as

$$\Delta x = f\Delta t + G\Delta w$$

We use a Taylor series expansion of $V^*$ about $(t, x)$ and because the covariance of a Wiener process is linear in $\Delta t$, we keep the quadratic terms of that expansion. This means that

$$\begin{aligned} V^*(t+\Delta x, t+\Delta t) &= V^*(t,x) + \frac{\partial V^*}{\partial t}\Delta t \\ &+ \left[\frac{\partial V^*}{\partial t}\right]^T \Delta x + \frac{1}{2}(\Delta x)^T H(\Delta x) \end{aligned}$$

where $H$ is the Hessian of $V^*$. Inserting this expression for $V^*(t+\Delta x, t+\Delta t)$ into equation (16) gives

$$\begin{aligned} V^*(t,x) &= \min_{u(t)} \mathbb{E}\left\{ L(x,u,t)\Delta t + V^*(t,x) + \frac{\partial V^*}{\partial t}\Delta t \right. \\ &\qquad \left. + \left[\frac{\partial V^*}{\partial x}\right]^T \Delta x + \frac{1}{2}(\Delta x)^T H(\Delta x) \mid x \right\} \end{aligned}$$

Using the fact that $\Delta w$ is zero mean, $\Delta x = f\Delta t + G\Delta w$, and the fact that $x^T H x = \mathrm{trace}(Hxx^T)$ we can rewrite the above expression for $V^*(t,x)$ as

$$-\frac{\partial V^*}{\partial t} = \min_u \left\{ L(x,u,t) + \left[\frac{\partial V^*}{\partial x}\right]^T f(x,u) + \frac{1}{2}\mathrm{trace}\left(HGWG^T\right) \right\}$$

with boundary condition $V^*(T,x) = m(x)$. This last equation is the *stochastic Hamilton-Jacobi Bellman equation*. Note that if there is no disturbance (i.e. $W = 0$) then this reverts to the deterministic HJB equation.

**Stochastic LQR with additive disturbances:** Consider the cost functional

$$J[u] = \mathbb{E}\left\{ \int_t^T (x^T \mathbf{Q} x + u^T \mathbf{R} u) d\tau + x^T(T)\mathbf{M}x(T) \mid x \right\}$$

where

$$d\mathbf{x} = (\mathbf{A}\mathbf{x} + \mathbf{B}u)dt + \mathbf{G}d\mathbf{w}$$

where $\mathbf{w}$ is a standard Brownian motion with covariance $Wdt$. The minimization step in the stochastic HJB minimizes

$$u^T \mathbf{R} u + \left[\frac{\partial V^*}{\partial x}\right]^T \mathbf{B} u$$

since these are the only terms involving $u$. Note that these are precisely the same terms that were minimized in the deterministic case. So we can say the optimal control is

$$u^* = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{B}^T \frac{\partial V^*}{\partial x}$$

We then need to solve for $V^*$. In the deterministic case, we assumed $V^* = x^T \mathbf{P}(t) x$, but this won't work in the stochastic case. A more appropriate candidate for the value function is

$$V^* = x^T \mathbf{P}(t) x + c(t)$$

where we now need to determine $\mathbf{P}$ and $c$. If substitute this into the stochastic HJB equation using our control, we get

$$-x^T\dot{\mathbf{P}}x - \dot{c} = x^T(\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P})x + \mathrm{trace}(\mathbf{P}GWG^T)$$

If the coefficients powers of $x$ are equated we finally get

$$\begin{aligned} -\dot{\mathbf{P}} &= \mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \\ -\dot{c} &= \mathrm{trace}(\mathbf{P}GWG^T) \end{aligned}$$

with boundary conditions $\mathbf{P}(T) = \mathbf{M}$ and $c(T) = 0$. So the optimal control becomes

$$u^*(t) = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t)x(t)$$

where $\mathbf{P}(t)$ is given as the solution of the differential Riccati equation.

Note that the optimal control for this stochastic problem is identical to that for the deterministic case. In other words, the optimal control for the stochastic LQR is identical to what happens when $W = 0$. While the control law is the same, the optimal cost is not the same due to the additional $c$ term. The integration of the $\dot{c}$ equation gives

$$c(t) = \int_t^T \mathrm{trace}\left(\mathbf{P}(\tau)GWG^T\right) d\tau$$

where $W dt$ is the covariance matrix for the Brownian increment $d\mathbf{w}$. So the optimal cost has an additional term reflecting the fact due to the state's random walk induced by the Brownian motion.

## 7. Optimal Control of Markov Decision Processes

Markov Decision Processes (MDPs) play an important role in a popular trial-by-error learning method known as Reinforcement Learning [Sutton and Barto (2018)]. An MDP consists of a decision maker (agent) itneracting with an external environment. That interaction may be viewed as seen in Fig. 4. Our problem is to determine the agent's action policy, $\pi$, that maximizes the expected total discounted reward that the agent receives during its interaction with the environment.

We formally define the MDP as a tuple, $(S, A, p, r, S_0, S_K)$ where $S$ is a finite set of environmental states and $A$ is a finite set of agent actions. We denote the state

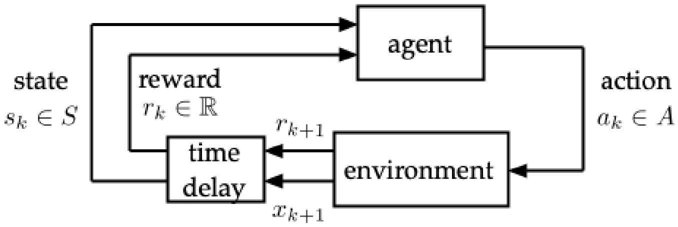


FIGURE 4. MDP models an agent's interaction with an external environment

at time instant $k \in \mathbb{Z}$ as $s_k$ and the action at time $k$ as $a_k$. The sets $S_0, S_K \subset S$ are called the initial and terminal state sets. The map $p : S \times A \to \mathcal{P}(S)$ maps the current state action pair $(s_k, a_k) \in S \times A$ onto the next state, $s_{k+1}$, through a conditional probability distribution function

$$p(y \,|\, x, a) = \Pr\{\mathbf{s}_{k+1} = y \,|\, \mathbf{s}_k = x, \mathbf{a}_k = a\}$$

This conditional distribution defines the *dynamics* of the MDP. The other map, $r : S \times A \times S \to \mathbb{R}$ maps the current state-action-next-state triple, $(s_k, a_k, s_{k+1})$ onto a numerical reward $r_{k+1} \in \mathbb{R}$.

The agent and environment interact over a sequence of time steps, $k = 0, 1, 2, 3, \ldots$. The environmental state at time 0 is in the initial state $S_0$. At each time instant, $k$, the agent selects an action $a_k$ using a *policy*, $\pi : S \to \mathcal{P}(A)$. The policy randomly uses the current state, $s_k$, to select the current action, $a_k$, by sampling from the policy distribution, $\pi(\mathbf{a}_k \,|\, \mathbf{s}_k = s)$. The set of all admissible policies will be denoted as $\mathbf{\Pi}$. The environment takes the agent's selected action and returns the environment's next state $s_{k+1} \sim p(\cdot \,|\, s_k, a_k)$ and the next reward $r_{k+1} = r(s_k, a_k, s_{k+1})$. This interaction therefore generates a sequence, $\{(s_k, a_k, r_{k+1})\}_{k=0}^{K-1}$, of state-action-reward triples that we sometimes refer to as the agent's *trajectory*. This sequence has a random stopping time $K$ that occurs when the system state trajectory first enters the terminal set, $S_K$.

Our problem is to find an action policy $\pi : S \to \mathcal{P}(A)$ that maximizes the expected total discounted reward that the agent receives over a trajectory through the environment. The total reward under a given policy may be written as a function $V^\pi : S \to \mathbb{R}$ that maps the initial state $s$ that the agent started in and whose value $V^\pi(s)$ is the expected total discounted reward received by the agent for using $\pi$

until it reaches the terminal set $S_K$. The expectation is taken over all possible trajectories that were randomly generated under $\pi$ for the given initial state $s$. This means the value at $s$ under $\pi$ can be written as

$$V^{\pi}(s) = \mathbb{E}^{\pi}\left\{\sum_{k=0}^{K-1} \gamma^k r(s_k, \pi(s_k), s_{k+1}) \,|\, s_0 = s\right\}$$

where $K$ is the stopping time when the state enters the terminal set, $S_K$ for the first time and $\gamma \in (0,1)$ is a *discount factor*.

We seek a policy, $\pi^* \, : , S \to \mathcal{P}(A)$ such that

$$V^{\pi^*}(s) \geq V^{\pi}(s)$$

for all $s \in S$ and over all possible policies, $\pi \in \mathbf{\Pi}$. For simplicity, we refer to the optimal value function as $V^*$, rather than $V^{\pi^*}$.

Finding the optimal policy, $\pi^*$, can be done using the generalized principle of optimality through a discrete time version of the Bellman equation

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} p(s'|s,a) \left(r(s,a,s') + \gamma V^{\pi}(s')\right)$$

This equation provides the basis for developing recursive algorithms that are used to algorithmically compute $V^*(s)$.

Directly finding $\pi^*$ from $V^*(s)$ is difficult to do. It is more convenient to find the policy, $\pi^*$ from the state-action value function $Q^{\pi} : S \times A \to \mathbb{R}$. The state action function under $\pi$ takes the value $Q^{\pi}(s,a)$ and it is the expected total reward received by the agent after it takes action $a$ while in state $s$. Since it is a value function it too satisfies a discrete-time Bellman equation of the form

$$Q^{\pi}(s,a) = \sum_{s' \in S} p(s'|s,a) \left(r(s,a,s') + \gamma \max_{a \in A} Q^{\pi}(s',a)\right)$$

The optimal $Q$-function is then

$$Q^*(s,a) = \max_{\pi \in \mathbf{\Pi}} Q^{\pi}(s,a)$$

Because $a \sim \pi(a|s)$ we can readily see that the optimal action policy when the current state is $s$ will be

$$\pi^*(s) = \arg\max_{a} Q^*(s,a)$$

If the environment's dynamics are completely known, then the Bellman equation is a system of simultaneous linear equations where the number of unknowns equals the cardinality of the state space $S$. Directly solving this system of linear equations will be difficult if the cardinality of $S$ is high (usually the case). So one usually uses successive approximation methods to find the value function. In particular, a sequence $\left\{\widehat{V}_\ell^\pi(s)\right\}_{\ell=0}^\infty$ of functions, $\widehat{V}_\ell^\pi : S \to \mathbb{R}$ is generated whose components are approximations of the true value function $V^\pi$ under the given policy $\pi$. These approximations are computed through the recursion

$$\begin{aligned}\widehat{V}_{\ell+1}^\pi(s) &= \mathbb{E}^\pi\left\{r(s_k,\pi(s_k),s_{k+1}) + \gamma\widehat{V}_\ell^\pi(s_{k+1}) \mid s_k = s\right\} \\ &= \sum_a \pi(a|s)\sum_{s'\in S} p(s' \mid s,a)\left(r(s,a,s') + \gamma\widehat{V}_\ell^\pi(s')\right)\end{aligned}$$

The actual value function, $V^\pi$, is a fixed point for the recursive update. One can prove that if the discount factor is between 0 and 1 then this recursion converges to the policy's value function $V^\pi$. This algorithm is known as *iterative policy evaluation*.

Once we use this policy to compute $V^\pi$ for a given policy $\pi_0$ we need to perturb the policy to find a better one. So let us assume that for some state $s$ that instead of picking the action $\pi_0(s)$, we pick an alternative action $a \neq \pi_0(s)$. After that we simply continue using the original policy $\pi_0$. This is sometimes referred to as a *needle* perturbation of the policy. The state-action value function from $s$ under this perturbed policy would be

$$\begin{aligned}Q^{\pi_0}(s,a) &= \mathbb{E}^\pi\left\{r(s,a,s') + \gamma V^{\pi_0}(s')\right\} \\ &= \sum_{s'\in S} p(s' \mid s,a)\left\{r(s,a,s') + \gamma V^{\pi_0}(s')\right\}\end{aligned}$$

The key thing is whether this is greater than or less than $V^{\pi_0}(s)$. Clearly we can use the above equation to compute $Q^{\pi_0}(s,a)$ when the policy is perturbed at time $k$ by using when the policy is perturbed at time $k$ by using $a$ instead of $\pi_0(s)$. Since there are a finite number of actions, we compute this value for each $a$ and come up with an improved policy that uses an $a$ that maximizes $Q^\pi(s,a)$. So we end up with a *greedy* policy of the form

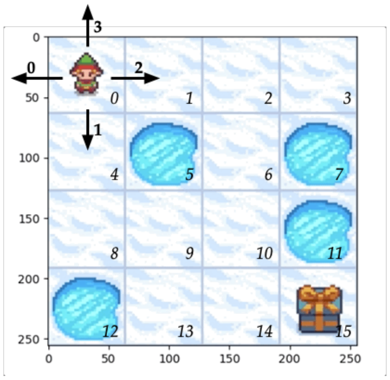
$$\begin{aligned}\pi_1(s) &= \arg\max_a Q^{\pi_0}(s,a) \\ &= \arg\max_a \sum_{s' \in S} p(s' \mid s,a)\left\{r(s,a,s') + \gamma V^{\pi_0}(s')\right\}\end{aligned}$$

Once a policy has been improved using $V^{\pi_0}$ to obtain $\pi_1$, we recompute $V^{\pi_1}$ and improve it again to obtain a better $\pi_2$. We therefore generate a sequence of monotonically improving policies and value functions.

$$\pi_0 \overset{\text{eval}}{\to} V^{\pi_0} \overset{\text{improve}}{\to} \pi_1 \overset{\text{eval}}{\to} V^{\pi_1} \overset{\text{improve}}{\to} \cdots \overset{\text{improve}}{\to} \pi^* \overset{\text{eval}}{\to} V^* \tag{17}$$

Because a finite MDP only has a finite number policies, this process must converge to an optimal policy and an optimal value function after a finite number of recursions. This search strategy for the optimal policy is called the *Policy iteration*.

Note that the policy iteration is not often used because the evaluation step requires that we converge to the value function $V^\pi$ before proceeding to the improvement step. In practice, one does not need to do this. In particular another algorithm called the *Value Iteration* simply executes a single update step of the iterative policy evaluation and then executes the improvement step. The Value Iteration also converges when the discount factor $\gamma \in (0,1)$ and in practice it converges much more quickly than the Policy Iteration [Puterman (1994)].



On Slipper Lake means 1/3 chance of moving to wrong state

Episode Ends when s=15 (home) or s=5,7,11,12 (fall through ice)

Reward = 1 when S=15, Reward = 0 for each time step

State Transition Probability P(s' | 0,a)

| | | |
|---|---|---|
| 2/3=P(0 \| 0,0) | | 1/3=P(4 \| 0,0) |
| 1/3=P(0 \| 0,1) | 1/3=P(1 \| 0,1) | 1/3=P(4 \| 0,1) |
| 1/3=P(0 \| 0,2) | 1/3=P(1 \| 0,2) | 1/3=P(4 \| 0,2) |
| 2/3=P(0 \| 0,3) | 1/3=P(1 \| 0,3) | |

FIGURE 5. Frozen Lake Environment

The following example illustrates the Value Iteration on a benchmark MDP known as the *Frozen Lake* Environment shown in Fig. 5. This figure shows a discrete state space $S = \{0, 1, 2, \ldots, 15\}$ as a 4 by 4 grid of squares representing locations on a frozen lake. Each grid element is classified as either a ”START” state (S), a ”FROZEN” state (F), a ”HOLE” state (H), or a ”GOAL” state (G). An episode starts with the agent in the S state. The agent selects one of 4 actions that move NORTH (3), EAST (2), SOUTH (1), or WEST (0). Because the ice is slippery, the next state the agent moves to after selecting its action is random. So for instance, if the agent selects action 3 (NORTH), then it moves to the grid element directly above it and the two states to the east and west with equal probability of $1/3$. Similar outcomes occur for the other selected actions, thereby defining the state transition kernel $p(s' \mid s, a)$. If the next state is a H (hole) or G (goal) state then the episode is over. The reward received by the agent is $1$ if the next state is the goal (G) state and is zero otherwise. So this is an MDP problem with delayed rewards in the sense that the agent only gets rewarded if it reaches the goal. There is no ”penalty” for falling in a hole and ending the episode early.

I’m going to use a Python script to implement the Value Iteration. The Frozen Lake environment is a toy environment for Reinforcement Learning algorithms discussed in chapter 6 and so we will embed our Value Iteration algorithm within a script implementing that the Frozen Lake environment as a Python class object

```
import random
import numpy as np
from frozen_lake import FrozenLakeEnv

env = FrozenLake(render_mode = "ansi")
```

The following script define a function implementing the value iteration for the state-action function, $Q$, and value function $V$.

```
def value_iteration(env, gamma, maxiter, tol):
     Q = np.zeros((env.observation_space.n,env.action_space.n))
     V = np.zeros(env.observation_space.n)
     for iter in range(maxiter):
          V_new = np.copy(V)
          for s in range(env.observation_space.n):
               for a in range(env.action_space.n):
                    Q[s][a] = sum([prob*(r+gamma*V_new[s_])
                                   for prob, s_, r, _ in env.P[s][a]])
                V[s] = max(Q[s,:])
          policy = np.zeros(env.observation_space.n)
          for s in range(env.obserbation_space.n):
               policy[s] = np.argmax(np.array(Q[s,:]))

          if (np.sum(np.fabs(V_new-V))<=tol):
               break
    return Q, V, policy
```

What this function returns is the optimal value function and the optimal policy. These are listed below as matrices representing the state space

$$V^* = \left[\begin{array}{cccc} 0.069 & 0.061 & 0.074 & 0.056 \\ 0.092 & & 0.112 & \\ 0.145 & 0.247 & 0.300 & \\ & 0.380 & 0.639 & \mathbf{G} \end{array}\right], \quad \pi^* = \left[\begin{array}{cccc} \text{W}(0) & \text{N}(3) & \text{W}(0) & \text{N}(3) \\ \text{W}(0) & & \text{W}(0) & \\ \text{N}(3) & \text{S}(1) & \text{N}(0) & \\ & \text{E}(2) & \text{S}(1) & \end{array}\right]$$

We can use the Frozen Lake Environment to ”simulate” the use of this optimal policy over many episodes and use then take the average of the total reward as the success rate of the policy.

The value or policy iteration both assume we already know the environment’s dynamics, $p(s'|s,a)$, and reward function $r(s,a,s')$. In many applications these functions are not known. Reinforcement learning is a set of algorithms that use the sequence of agent interactions with the environment to ”learn” an optimal action policy. Most of the RL algorithms rely on recursive algorithms based on the Bellman equation to learn an action policy. These algorithms use data from computer simulations modeling agent/environment interactions to drive the recursive learning algorithms. One of the main open research issues regarding RL involves the safe transfer of action policies learned in a virtual simulation environment to the real world environment.

```
success_rate = 0
for episode in range(num_episodes):
     s = env.reset()[0]
     done = False
     while not done:
          a = np.argmax(Q[s])
          s , reward, done, trunc,info = env.step(a)
          if done:
               success+rate += reward/num_episodes
               break
```

In this case we see a success rate of about 82%. So our policy does not always reach the goal, which is to be expected since the state transition kernel always has a finite chance of making a move that has the agent fall in a ”hole”.

## 8. Summary

This chapter reviewed fundamental results regarding the optimal control of dynamical systems over a finite horizon. One of this chapter’s main results was a variational approach for deriving necessary conditions for the optimal control, known as the Calculus of Variations. Our discussion of the Calculus of Variations follows Liberzon (2012), a recent textbook. The variational approach to optimal control generates *necessary* conditions that an optimal *open-loop* control law must follow. In practice, we can solve this problem numerically and such numerical solutions form a widely used approach to control known as model predictive control (receding horizon control) [Alessio and Bemporad (2009)]. Sufficient conditions for the optimal control can be obtained using *dynamic programming* [Bertsekas (1995)]. Our discussion of dynamic programming’s Bellman equation follows the more general treatment in Fleming and Rishel (1972) with the development of the stochastic HJB equation following [Dorato et al. (1994)]. Two important features of dynamic programming are first that it shows that state-feedback laws are the optimal control laws, but more importantly it is an extremely general framework that can be used for continuous-time, discrete-time, and discrete-event systems. We demonstrate this versatility by showing how the Bellman equation is used to form the HJB equation for deterministic and stochastic continuous-time systems and then using the same framework to obtain optimal controls for stochastic discrete-event systems (Markov Decision Processes) [Puterman (1994)]. This last application to MDP’s has been extremely popular in recent years due to the popularity of *Reinforcement Learning* [Sutton and Barto (2018)] in learning-based control systems. This chapter’s focus on *optimal* control shows that the problem can be formally developed for extremely general classes of dynamical systems. Enforcing optimality, however, has often led to control systems whose performance is extremely sensitive to uncertainty in the system dynamics. These optimal methods take advantage of our prior knowledge of the system dynamics and if that knowledge is not accurate then the resulting ”optimal” solutions may perform poorly in the ”real-life” system. This requires us to consider how we can recast these optimization problems to ensure some degree of robustness to model uncertainty. The following chapters investigate this question, first with regard to linear time-invariant systems and then for nonlinear affine systems.

CHAPTER 3

# Robust Linear Control - $\mathcal{H}_2/\mathcal{H}_\infty$ methods

This chapter discusses optimal control of LTI systems of the form

$$\begin{array}{rcllll} \dot{x}(t) & = & \mathbf{A}x & + \ \mathbf{B}_1 w & + & \mathbf{B}_2 u \\ z(t) & = & \mathbf{C}_1 x & & + & \mathbf{D}_{12} u \\ y(t) & = & \mathbf{C}_2 x & + \ \mathbf{D}_{21} w & & \end{array} \tag{18}$$

This system has two inputs; an exogenous disturbance, $w$, and a control input, $u$. The system also has two outputs; a virtual penalty signal, $z$, characterizing the system's performance and an observation, $y$, that is accessible by the controller. The exogenous disturbance, $w$, introduces a degree of uncertainty into the system and we want our control system's performance to be *robust* to that uncertainty. In addition to this, we will also assume that there is some uncertainty regarding the system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$. So again we want our controlled system's performance to also be *robust* to this model uncertainty. This chapter, therefore, is concerned with the robust optimal control of LTI systems.

Chapter 1 already showed that we can use an *observer-based* controller for an LTI system. An observer-based controller uses a Luenberger observer to estimate the system's states and then uses a state feedback control law on those state estimates to control the original plant. The resulting control system takes the form of the linear fractional transformation, $\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})$, shown in Fig. 1 in which the plant $\mathbf{P}$ and controller $\mathbf{K}$ have state space realizations

$$\mathbf{P} \overset{\text{s}}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{0} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right], \quad \mathbf{K} \overset{\text{s}}{=} \left[\begin{array}{c|c} \mathbf{A} + \mathbf{B}_2\mathbf{F} - \mathbf{L}\mathbf{C}_2 & \mathbf{L} \\ \hline \mathbf{F} & \mathbf{0} \end{array}\right]$$

where $\mathbf{F}$ are the state feedback gains and $\mathbf{L}$ are the observer gains. The control system shown in Fig. 1 is often called a *generalized regulator* because a number of real-life feedback systems can be put into this canonical form in which the controller $\mathbf{K}$ is selected to regulate the output $z$ with respect to bounded variations of the input $w$.

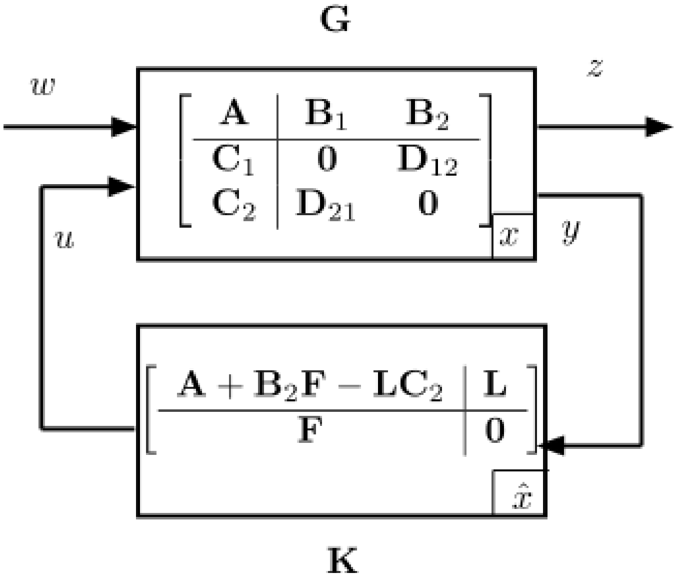

G
$w$
$z$
$\left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{0} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right]$
$x$
$u$
$y$
$\left[\begin{array}{c|c} \mathbf{A}+\mathbf{B}_2\mathbf{F}-\mathbf{L}\mathbf{C}_2 & \mathbf{L} \\ \hline \mathbf{F} & \mathbf{0} \end{array}\right]$
$\hat{x}$
K

FIGURE 1. LFT for Generalized Regulator

The basic optimal control problem of interest to us is

$$\begin{array}{ll} \text{minimize:} & \|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\| \\ \text{with respect to} & \mathbf{K} \\ \text{subject to:} & \mathcal{F}_\ell(\mathbf{P},\mathbf{K}) \text{ being internally stable} \end{array}$$

where $\|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\|$ is the induced gain of the closed loop LFT from input $w$ to output $z$. While this is formulated as an optimization problem, this chapter will show how one can ”fold” in prior bounds on model uncertainty to ensure the optimal controller’s performance is *robust* to this model uncertainty. In this case, robustness means that for plants within the uncertainty set, one can guarantee closed stability and a measure of control system performance. The optimization methods rely on extensions of traditional LQG controllers where instead of select controls that may be viewed as extending an optimization with respect to the $\mathcal{H}_2$ norm of the closed loop operator to the $\mathcal{H}_\infty$ norm. For this reason, the robust control methods presented in this chapter are referred to as $\mathcal{H}_\infty$ control.

The remainder of this chapter is organized as follows. We first examine LQG controllers as a special case of $\mathcal{H}_2$ controllers. The LQG controller is an observer-based controller (section 9, chapter 1) whose state feedback gains, **F**, are the LQR gains derived in the dynamic programming section 5 of chapter 2 and whose observer is a steady-state Kalman filter. We then demonstrate that LQG control systems lack robustness to model uncertainty. To address this robustness issue, We present sufficient conditions for a control system to have *robust stability* and *robust performance*. We then show how these conditions can be folded into a generalized regulator problem for unstructured multiplicative uncertainties and structured parameter uncertainties. Both of these formulations lead to a generalized regulator problem that seeks to minimize the $\mathcal{H}_\infty$ norm of the generalized regulator. We show how those controllers are synthesized and introduce MATLAB computational tools that can be used to find these $\mathcal{H}_\infty$ controllers.

## 1. Linear Quadratic Gaussian (LQG) and $\mathcal{H}_2$ Control

The linear quadratic Gaussian (LQG) control system is an observer-based control system whose state gain matrix **F** is the gain matrix used in an LQR control system and whose observer gain matrix **L** is the gain matrix for a *steady-state Kalman filter*. It is a special case of the $\mathcal{H}_2$ optimal controller that minimizes the $\mathcal{H}_2$ norm of a generalized regulator driven by white noise processes with an output consisting of the plant's state and applied control. The LQR gain matrix was already discussed in chapter 2. The following subsections develop the steady-state Kalman filter using duality arguments, derive the LQG controller, and show how it is related to the more general $\mathcal{H}_2$ controller.

**1.1. Steady-State Kalman Filter:** The steady state Kalman filter is a Luenberger observer that minimizes the mean squared state estimation error for an LTI system, **G**, with state equations

$$\begin{aligned}\dot{x} &= \mathbf{A}x + \mathbf{B}w \\ y &= \mathbf{C}x + v\end{aligned}$$

where $w$ and $v$ are i.i.d. normally distributed white noise processes with unit variance and zero mean. In particular, the filter may be seen as an input/output operator,

$\mathbf{F} : \mathcal{L}_{2\mathrm{e}} \to \mathcal{L}_{2\mathrm{e}}$ that maps the output measurement signal, $y \in \mathcal{L}_{2\mathrm{e}}$, from the plant $\mathbf{G}$ onto a state estimate $\widehat{x} \in \mathcal{L}_{2\mathrm{e}}$. The performance of this observer is measured by the mean squared estimation error (MSEE)

$$\lim_{T\to\infty} \mathbb{E}\left\{ \frac{1}{T} \int_0^T |x(\tau) - \widehat{x}(\tau)|^2 d\tau \right\}$$

Our objective is to find an observer, $\mathbf{F}$, that generates estimates $\widehat{x} = \mathbf{F}[y]$ that minimizes this MSEE.

We will solve this problem by augmenting the plant's state equations so it can be viewed as a two-port system with two inputs, the noise vectors $\begin{bmatrix} w \\ v \end{bmatrix}$ and the state estimate $\widehat{x}$, and two outputs, the state estimation error, $\widetilde{x} = x - \widehat{x}$, and the observed output $y$. The augmented plant's, $\mathbf{P}$, state equations are, therefore

$$\begin{aligned} \dot{x} &= \mathbf{A}x + \begin{bmatrix} \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \\ \widetilde{x} &= \mathbf{I}x - \mathbf{I}\widehat{x} \\ y &= \mathbf{C}x + \mathbf{I}v \end{aligned}$$

In packed matrix form the augmented plant's state space realization becomes

$$\mathbf{P} \overset{\mathrm{s}}{=} \left[ \begin{array}{c|c|c} \mathbf{A} & \begin{bmatrix} \mathbf{B} & \mathbf{0} \end{bmatrix} & \mathbf{0} \\ \hline \mathbf{I} & \begin{bmatrix} \mathbf{0} & \mathbf{0} \end{bmatrix} & -\mathbf{I} \\ \mathbf{C} & \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} & \mathbf{0} \end{array} \right]$$

The Kalman filter, $\mathbf{F} : \mathcal{L}_{2\mathrm{e}} \to \mathcal{L}_{2\mathrm{e}}$, is then a linear time invariant system that maps the observed output $y$ onto the state estimate $\widehat{x} = \mathbf{F}[y]$. We now interconnect the augmented plant, $\mathbf{P}$, and filter $\mathbf{F}$ to form the linear fractional transformation, $\mathcal{F}_\ell(\mathbf{P}, \mathbf{F})$, shown in Fig. 2. This LFT maps the noise inputs $\begin{bmatrix} w \\ v \end{bmatrix}$ onto the state estimation error $\widetilde{x}$ and our objective is to find a linear filtering system, $\mathbf{F}$, that minimizes the expected value of the squared estimation error. The resulting $\mathbf{F}$ is what we call the steady-state Kalman filter.

The Kalman filter equations are traditionally derived using stochastic arguments if we treat $w$ and $v$ as random processes [Kailath (1976)]. Rather than doing this,

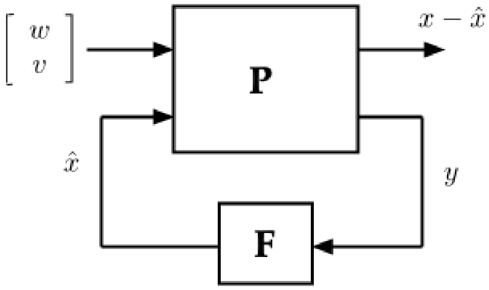


FIGURE 2. Augmenting Plant so its output is state estimation error

we will use duality arguments to derive the Kalman filter, since this approach allows us to make use of our earlier LQR control results. The adjoint of the augmented plant LFT, $\mathcal{F}_\ell(\mathbf{P}, \mathbf{F})$ is

$$[\mathcal{F}_\ell(\mathbf{P}, \mathbf{F})]^* = \mathcal{F}_\ell(\mathbf{P}^*, \mathbf{F}^*)$$

This adjoint is also an LFT as shown in Fig. 3. Since we already have a realization for $\mathbf{P}$ in packed matrix form, we can write down a realization for the adjoint $\mathbf{P}^*$

$$\begin{bmatrix} -\dot{p} \\ \hline \begin{bmatrix} w \\ v \end{bmatrix} \\ \widehat{x} \end{bmatrix} = \left[\begin{array}{c|cc} \mathbf{A}^T & \mathbf{I} & \mathbf{C}^T \\ \hline \begin{bmatrix} \mathbf{B}^T \\ \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \end{array}\right] \begin{bmatrix} p \\ \hline \widetilde{x} \\ y \end{bmatrix}$$

where $p$ is the adjoint system's state.

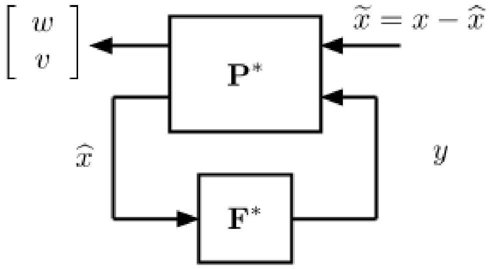

$\begin{bmatrix} w \\ v \end{bmatrix}$
$\widetilde{x} = x - \widehat{x}$
$\mathbf{P}^*$
$\widehat{x}$
$y$
$\mathbf{F}^*$

FIGURE 3. Adjoint of observer LFT

We substitute $\tau = -t$ to reverse the direction of time and the first two equations of the adjoint' realization become

$$\begin{aligned} \dot{p} &= \mathbf{A}^T p + \widetilde{x} + \mathbf{C}^T y \\ \begin{bmatrix} w \\ v \end{bmatrix} &= \begin{bmatrix} \mathbf{B}^T p \\ y \end{bmatrix} \end{aligned}$$

with noise input $\widetilde{x}$ and controllable input $y$. We will now consider the problem of finding a control input $y$ to the adjoint that minimizes the $\mathcal{L}_2$ norm of the adjoint's output. In other words seek an input, $y$, that minimizes

$$\lim_{T\to\infty} \mathbb{E}\left\{ \frac{1}{T} \int_0^T (w^T w + v^t v)dt \right\} = \lim_{T\to\infty} \mathbb{E}\left\{ \frac{1}{T} \int_0^T (p^T \mathbf{B}\mathbf{B}^T p + y^T y)dt \right\}$$

It should be apparent that this is the cost functional for a stochastic LQR problem and so we can deduce that the optimal $y$ that solves our problem has the form

$$y = -\mathbf{C}\mathbf{P}p$$

where $\mathbf{P}$ is a symmetric positive definite matrix that satisfies

$$0 = \mathbf{P}\mathbf{A}^T + \mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{C}^T\mathbf{C}\mathbf{P} + \mathbf{B}\mathbf{B}^T$$

Inserting this back into the equations for the adjoint system yields

$$\begin{aligned} -\dot{p} &= (-\mathbf{A}^T - \mathbf{C}^T\mathbf{C}\mathbf{P})p - \tilde{x} \\ y &= -\mathbf{C}\mathbf{P}p \end{aligned}$$

which is the state space realization for the adjoint system from $\widetilde{x}$ to $y$. If we then take the adjoint of this system, we obtain the realization

$$\begin{aligned} \widehat{x} &= (\mathbf{A} - \mathbf{P}\mathbf{C}^T\mathbf{C})\widehat{x} + \mathbf{P}\mathbf{C}^T y \\ &= \mathbf{A}\widehat{x} + \mathbf{P}\mathbf{C}^T(y - \mathbf{C}\widehat{x}) \end{aligned}$$

which shows the optimal observer gain $\mathbf{L} = \mathbf{P}\mathbf{C}^T$ where $\mathbf{P}$ satisfies the preceding algebraic Riccati equation. This observer is what we call the steady-state Kalman filter. The preceding discussion may be summarized in the following theorem.

THEOREM 8. *Consider the system*

$$\begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}w \\ y &= \mathbf{C}x + v \end{aligned}$$

*where $w$ and $v$ are zero mean, unit variance white noise processes. Then the following steady-state Kalman filter*

$$\dot{\widehat{x}} \quad = \quad \mathbf{A}\widehat{x} - \mathbf{L}_{\text{KF}}(y - \mathbf{C}\widehat{x})$$

*minimizes the expected $\mathcal{L}_2$ norm of the state estimation error $\widetilde{x} = x - \widehat{x}$*

$$\lim_{T\to\infty} \mathbb{E}\left\{ \frac{1}{T}\int_0^T (w^T(\tau)w(\tau) - v^T(\tau)v(\tau))d\tau \right\}$$

*where*

$$\mathbf{L}_{\text{KF}} = \mathbf{P}\mathbf{C}^T$$

*where $\mathbf{P}$ is a symmetric positive definite matrix satisfying the algebraic Riccati equation*

$$0 = \mathbf{P}\mathbf{A}^T + \mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{C}^T\mathbf{C}\mathbf{P} + \mathbf{B}\mathbf{B}^T$$

The preceding Kalman gain equations were developed assuming the process noise $w$ and measurement noise, $v$ were both independent white noise processes with unit variance. If $w$ and $v$ are white noise processes with covariance matrices $\mathbf{W}$ and $\mathbf{V}$, respectively then we can simply rescale $w$ and $v$ to show the steady state Kalman gains are

$$\mathbf{L}_{\text{KF}} = \mathbf{P}\mathbf{C}^T\mathbf{V}^{-1}$$

where $\mathbf{P}$ now satisfies the algebraic Riccati equation

$$\mathbf{0} = \mathbf{P}\mathbf{A}^T + \mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{C}^T\mathbf{V}^{-1}\mathbf{C}\mathbf{P} + \mathbf{B}\mathbf{W}\mathbf{B}^T$$

**Discrete time Kalman Filter:** The preceding discussion derived the equations for the continuous-time steady state Kalman filter. In practice, however, the Kalman filter is usually implemented with a computer as the *discrete Kalman Filter* (DKF) algorithm. The derivation of DKF is customarily done using stochastic arguments [Kailath (1976)] since these arguments are easier for discrete-time and this makes it easier to develop a time-varying Kalman filter. Rather than deriving these equations, I'll simply present them since I am more interested in how we "use" this

Kalman filter, rather than its formal derivation. We assume the system state equations are of the form

$$\begin{aligned} x(k) &= \mathbf{A}(k-1)x(k-1) + w(k-1) \\ y(k) &= \mathbf{C}(k)x(k) + v(k) \end{aligned}$$

where $w(k)$ is a zero mean white noise process representing process noise with a covariance $\mathbf{W}$ and $v(k)$ is a zero mean white noise process representing sensor measurement noise. with covariance $\mathbf{V}$. Note here that our system is time-varying so we need to present the equations for the time-varying version of the Kalman filter. The time-varying DKF is typically implemented in two-stages that I refer to as the *blooming* and *pruning* stages, respectively.

Immediately after time instant $k-1$, the information available to the filter is the current state estimate $\widehat{x}^+(k-1)$ and with error covariance $\mathbf{P}^+(k-1)$. The *blooming* stage takes that information and "propagates" it through the state transition matrix $\mathbf{A}(k)$ to just before the next time instant $k$. We refer to this as blooming because propagating through $\mathbf{A}(k)$ will typically increase the uncertainty in the state's true value. The bloomed state estimate and error covariances prior to time $k$ are denoted as $\widehat{x}^-(k)$ and $\mathbf{P}^-(k)$, respectively. One can readily show that these two quantities satisfy

$$\begin{aligned} \widehat{x}(k) &= \mathbf{A}(k-1)\widehat{x}^+(k-1) \\ \mathbf{P}^-(k) &= \mathbf{A}(k-1)\mathbf{P}^+(k-1)\mathbf{A}^T(k-1) + \mathbf{W} \end{aligned}$$

where $\widehat{x}^-(k)$ and $\mathbf{P}^-(k)$ represent the information available to the filter just prior to the updated provided by the $k$th measurement, $y(k)$.

The action of this update, $y(k)$, is to *prune* away the uncertainty that bloomed under propagation of the estimate through the state equations. The pruned state estimate and error covariance at time instant $k$ are denoted as $x^+(k)$ and $\mathbf{P}^+(k)$, respectively. These equations can be shown to satisfy

$$\begin{aligned} \widehat{x}^+(k) &= \widehat{x}^-(k) + \mathbf{L}(k)(y(k) - \mathbf{C}(k)\widehat{x}^-(k)) \\ \mathbf{P}^+(k) &= (\mathbf{I} - \mathbf{L}(k)\mathbf{C}(k))\mathbf{P}^-(k) \\ \mathbf{L}(k) &= \mathbf{P}^-(k)\mathbf{C}^T(k)(\mathbf{C}(k)\mathbf{P}^-(k)\mathbf{C}^T(k) + \mathbf{V})^{-1} \end{aligned}$$

where $\mathbf{L}(k)$ is called the *Kalman Gain*.

**1.2. LQG and $\mathcal{H}_2$ Controllers:** LQG (Linear Quadratic Gaussian) control systems are observer-based control systems where the state feedback gains are LQR control gains and whose observer gains are those of a Kalman filter. The LQG controller is a special case of the $\mathcal{H}_2$ optimal controller. This subsection shows that the LQG controller is *optimal* in the sense of minimizing the $\mathcal{H}_2$ norm of the LFT representing the closed loop system. We then discuss how it is a special case of the $\mathcal{H}_2$ controller and discuss a well-known example illustrating its lack of robustness to modeling error.

We start by considering the following plant

$$\begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}_1 w + \mathbf{B}_2 u \\ z &= \begin{bmatrix} \mathbf{C}_1 x \\ \mathbf{D}_{12} u \end{bmatrix} \\ y &= \mathbf{C}_2 x + \mathbf{D}_{21} v \end{aligned} \tag{19}$$

where $w$ and $v$ are independent white noise processes with covariance matrices $\mathbf{W}$ and $\mathbf{V}$, respectively. We assume that $(\mathbf{A}, \mathbf{C}_2)$ is detectable and $(\mathbf{A}, \mathbf{B}_2)$ is stabilizable. We further assume that $\mathbf{D}_{12}$ has full column rank and $\mathbf{D}_{21}$ has full row rank. In packed matrix form this system's realization is therefore

$$\mathbf{P} \overset{\text{s}}{=} \left[\begin{array}{c|cc} \mathbf{A} & \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} \end{bmatrix} & \mathbf{B}_2 \\ \hline \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_{12} \end{bmatrix} \\ \mathbf{C}_2 & \begin{bmatrix} \mathbf{0} & \mathbf{D}_{21} \end{bmatrix} & \mathbf{0} \end{array}\right]$$

The LQG controller, $\mathbf{K}_{\text{LQG}}$ is an LTI system with internal state $\widehat{x}$. The system takes the measurement $y$ as an input and generates the output $u$ according to the equation

$$u = \mathbf{F}\widehat{x}$$

where $\mathbf{F}$ is the LQR gain

$$\mathbf{F} = -(\mathbf{D}_{12}^T \mathbf{D}_{12})^{-1} \mathbf{B}_2^T \mathbf{S} \tag{20}$$

where $\mathbf{S} = \mathbf{S}^T > 0$ satisfies the LQR algebraic Riccati equation

$$0 = \mathbf{A}^T \mathbf{S} + \mathbf{S}\mathbf{A} + \mathbf{C}_1^T \mathbf{C}_1 + \mathbf{S}\mathbf{B}(\mathbf{D}_{12}^T \mathbf{D}_{12})^{-1} \mathbf{B}^T \mathbf{S} \tag{21}$$

The internal state, $\widehat{x}$, of the controller satisfies the following ODE

$$\begin{aligned}\dot{\widehat{x}} &= \mathbf{A}\widehat{x} + \mathbf{B}_2 u + \mathbf{L}(y - \mathbf{C}_2\widehat{x}) \\ &= (\mathbf{A} + \mathbf{B}_2\mathbf{F} - \mathbf{L}\mathbf{C}_2)\widehat{x} + \mathbf{L}y\end{aligned}$$

where

$$\mathbf{L} = -\mathbf{P}\mathbf{C}_2^T(\mathbf{D}_{21}\mathbf{V}\mathbf{D}_{21}^T)^{-1} \tag{22}$$

and $\mathbf{P}$ satisfies the Kalman algebraic Riccati equation

$$0 = \mathbf{P}\mathbf{A}^T + \mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{C}_2^T(\mathbf{D}_{21}\mathbf{V}\mathbf{D}_{21}^T)^{-1}\mathbf{C}_2\mathbf{P} + \mathbf{B}_1\mathbf{W}\mathbf{B}_1^T \tag{23}$$

These control gains are simply the LQR control gains that minimize the cost functional

$$J[u] = \int_0^\infty \left\{x^T\mathbf{C}_1^T\mathbf{C}_1 x + u^T\mathbf{D}_{12}^T\mathbf{D}_{12}u\right\} d\tau \tag{24}$$

Note that our original plant in equations (19) was augmented with the output

$$z = \left[\begin{array}{c}\mathbf{C}_1 x \\ \mathbf{D}_{12}u\end{array}\right]$$

From this we see that the cost functional in equation (24) may also be written as

$$J[u] = \int_0^\infty z^T z d\tau = \|z\|_{\mathcal{L}_2}^2$$

which is simply the energy in the virtual signal $z$.

The observer gains in equation (22) are the steady state Kalman filter gains that minimize the mean squared estimation error

$$\mathbb{E}\left\{\int_0^\infty \widetilde{x}^T\widetilde{x}d\tau\right\} = \mathbb{E}\left\{\|x - \widehat{x}\|_{\mathcal{L}_2}^2\right\}$$

for the controlled plant being driven by process noise with covariance matrix $\mathbf{W}$ and measurement noise with covariance $\mathbf{D}_{21}\mathbf{V}\mathbf{D}_{21}^T$.

Taken together we therefore see that the controlled system may be viewed as the LFT combination of the two-port plant in equation (19) with the LQG controller

$$\mathbf{K}_{\mathrm{LQG}} \overset{\mathrm{s}}{=} \left[\begin{array}{c|c}\mathbf{A} + \mathbf{B}_2\mathbf{F} - \mathbf{L}\mathbf{C}_2 & \mathbf{L} \\ \hline \mathbf{F} & \mathbf{0}\end{array}\right]$$

This LFT maps the noise inputs $\left[\begin{array}{c} w \\ v\end{array}\right]$ onto the energy signal $z$.

We now demonstrate that this LQG controller actually minimizes the cost functional

$$J_{\text{LQG}}[u] = \mathbb{E}\left\{\int_0^\infty z^T z d\tau\right\}$$

by introducing a similarity transformation on the state space

$$\begin{bmatrix} \widehat{x} \\ \widetilde{x} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} x \\ \widehat{x} \end{bmatrix}$$

The state estimation error, $\widetilde{x} = x - \widehat{x}$ satisfies

$$\begin{aligned} \dot{\widetilde{x}} &= \dot{x} - \dot{\widehat{x}} \\ &= \mathbf{A}(\widehat{x} + \widetilde{x}) + \mathbf{B}_1 + \mathbf{B}_2\mathbf{F}\widehat{x} \\ &\quad -(\mathbf{A} + \mathbf{B}_2\mathbf{F} - \mathbf{L}\mathbf{C}_2)\widehat{x} - \mathbf{L}(\mathbf{C}_2(\widehat{x} + \widetilde{x}) + \mathbf{D}_{21}v) \\ &= \mathbf{A}_L\widetilde{x} + \begin{bmatrix} \mathbf{B}_1 & -\mathbf{L}\mathbf{D}_{21} \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \end{aligned}$$

where $\mathbf{A}_L = \mathbf{A} - \mathbf{L}\mathbf{C}_2$. The state estimate equation (22) must now be rewritten in terms of these new states, $\widetilde{x}$ and $\widehat{x}$, to get

$$\begin{aligned} \dot{\widehat{x}} &= (\mathbf{A} + \mathbf{B}_2\mathbf{F} - \mathbf{L}\mathbf{C}_2)\widehat{x} + \mathbf{L}(\mathbf{C}_2(\widehat{x} + \widetilde{x}) + \mathbf{D}_{21}v) \\ &= (\mathbf{A} + \mathbf{B}_2\mathbf{F})\widehat{x} + \mathbf{L}\mathbf{C}_2\widetilde{x} + \mathbf{L}\mathbf{D}_{21}v \\ &= \mathbf{A}_F\widehat{x} + \mathbf{L}\mathbf{C}_2\widetilde{x} + \mathbf{L}\mathbf{D}_{21}v \qquad (25) \end{aligned}$$

where $\mathbf{A}_F = \mathbf{A} + \mathbf{B}_2\mathbf{F}$. Taken together we then get the closed-loop state equations

$$\begin{bmatrix} \dot{\widetilde{x}} \\ \dot{\widehat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_L & \mathbf{0} \\ \mathbf{L}\mathbf{C}_2 & \mathbf{A}_F \end{bmatrix} \begin{bmatrix} \widetilde{x} \\ \widehat{x} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 & -\mathbf{L}\mathbf{D}_{21} \\ \mathbf{0} & \mathbf{L}\mathbf{D}_{21} \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \qquad (26)$$

The system equation (26) forms an LTI system driven by the white noise process $\begin{bmatrix} w \\ v \end{bmatrix}$. Now consider the cost

$$\int_0^\infty \mathbb{E}\left\{\begin{bmatrix} \widetilde{x} \\ \widehat{x} \end{bmatrix} \begin{bmatrix} \widetilde{x}^T & \widehat{x}^T \end{bmatrix}\right\} d\tau = \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{bmatrix}$$

Note that $\mathbf{\Sigma}_{11}$ is the integrated mean squared error for the estimator. We know that the Kalman filter minimizes $\operatorname{trace}\mathbf{\Sigma}_{11}$. We also know that the estimate $\widehat{x}$ is a minimum mean squared estimate and so by the stochastic orthogonality principle we have $\mathbf{\Sigma}_{12} = \int_0^\infty \mathbb{E}\left\{\widetilde{x}\widehat{x}^T\right\} d\tau$ is zero. The random process $\widehat{x}$ is generated by equation (25) which is a linear stochastic differential equation driven by white noise processes $v$ and $\widetilde{x}$. This means we can use the stochastic version of the HJB equation to deduce that the LQR control gain $\mathbf{F}$ from equation (20) minimizes

$$J[u] = \mathbb{E}\left\{\int_0^\infty \left(\widehat{x}^T \mathbf{C}_1^T \mathbf{C}_1 \widehat{x} + u^T \mathbf{D}_{12}^T \mathbf{D}_{12} u\right) d\tau\right\}$$

where $u = \mathbf{F}\widehat{x}$. Because $x^T x = \text{trace}(xx^T)$ we can rewrite this functional as

$$\begin{aligned} J[u] &= \int_0^\infty \left[\text{trace}\left(\mathbf{C}_1^T \mathbb{E}\left\{\widehat{x}\widehat{x}^T\right\} \mathbf{C}_1\right) + \text{trace}\left(\mathbf{D}_{12}^T \mathbf{F}^T \mathbb{E}\left\{\widehat{x}\widehat{x}^T\right\} \mathbf{F}\mathbf{D}_{12}\right)\right] d\tau \\ &= \text{trace}\left(\mathbf{C}_1^T \mathbf{\Sigma}_{22} \mathbf{C}_1\right) + \text{trace}\left(\mathbf{D}_{12}^T \mathbf{F}^T \mathbf{\Sigma}_{22} \mathbf{F}\mathbf{D}_{12}\right) \end{aligned}$$

This last equation is our expression for the cost attained using an LQR controller

We can now look at what is done with the LQG controller. We again evaluate $J[u]$, but this time we get

$$\begin{aligned} J[u] &= \mathbb{E}\left\{\int_0^\infty \left(x^T \mathbf{C}_1^T \mathbf{C}_1 x + u^T \mathbf{D}_{12}^T \mathbf{D}_{12} u\right) d\tau\right\} \\ &= \mathbb{E}\left\{\|\mathbf{C}_1 x\|_{\mathcal{L}_2}^2 + \|\mathbf{D}_{12}\mathbf{F}\widehat{x}\|_{\mathbf{\mathcal{L}}_2}^2\right\} \\ &= \mathbf{C}\mathbb{E}\left\{\text{trace}\,(\widehat{x}+\widetilde{x})(\widehat{x}+\widetilde{x})^T\right\} \mathbf{C}_1 + \mathbf{D}_{12}^T \mathbf{F}^T \mathbf{\Sigma}_{22} \mathbf{F}\mathbf{D}_{12} \qquad (27) \\ &= \mathbf{C}_1^T \mathbf{\Sigma}_{11} \mathbf{C}_1 + \mathbf{C}_1^T \mathbf{\Sigma}_{22} \mathbf{C}_1 + \mathbf{D}_{12}^T \mathbf{F}^T \mathbf{\Sigma}_{22} \mathbf{F}\mathbf{D}_{12} \qquad (28) \end{aligned}$$

In equation (28) we see that the first is minimized by our choice of the Kalman observer gains and the second two terms are minimized by our choice for the LQR controller gains. We can therefore conclude that the quadratic cost functional is indeed minimized by the LQG's choice of gains.

Note we could have also written the LQG cost functional as

$$J[u] = \mathbb{E}\left\{\|z\|_{\mathcal{L}_2}^2\right\}$$

where $z$ is the virtual output signal in the plant equations (19). Since the input to this system is a white noise process, one can show that selecting a control that minimizes the output's $\mathcal{L}_2$ norm assuming the input is white noise is equivalent to minimizing the $\mathcal{H}_2$ norm of the closed-loop system's transfer function. For this reason we can see that LQG control is a special case of $\mathcal{H}_2$ optimal controller.

**1.3. Robustness of LQG Controllers:** LQG controllers are often used in practice due to the simplicity with which one can synthesize the controller. It is important to note, however, that these controllers are inherently "sensitive" to modeling error. LQG control systems, in other words, do not exhibit robust performance/stability with respect to model uncertainty.

This lack of robustness was illustrated in a well known example [Doyle (1978)]. That example considered a single-input single output system with state space realization

$$\mathbf{P} \overset{\mathrm{s}}{=} \left[\begin{array}{c|cc} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} \sqrt{\sigma} & 0 \\ \sqrt{\sigma} & 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \hline \begin{bmatrix} \sqrt{q} & \sqrt{q} \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 \end{bmatrix} & 0 \end{array}\right]$$

where $\sigma$ and $q$ are non-negative system parameters. It can be shown that the solutions to the LQR and Kalman gains are

$$\mathbf{F} = -\alpha \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{L} = -\beta \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

where $\alpha = 2+\sqrt{4+q}$ and $\beta = 2+\sqrt{4+\sigma}$. Suppose the controller for our system is a gain perturbed version of the LQG controller. In other words, the control input is $u = (1+\delta)\mathbf{K}_{\mathrm{LQG}}[y]$ where $\delta$ is the gain perturbation. It can be shown that the closed loop system matrix $\mathbf{A}_{\mathrm{cl}}$ becomes

$$\mathbf{A}_{\mathrm{cl}} = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 1 & -\delta\alpha & -\delta\alpha \\ \hline \beta & 0 & 1-\beta & 1 \\ \beta & 0 & -\beta-\alpha & 1-\alpha \end{array}\right]$$

The stability of this perturbed closed-loop system can be determined from the characteristic polynomial of $\mathbf{A}_{\mathrm{cl}}$. In particular, let us assume that this characteristic polynomial has the form

$$\det(s\mathbf{I} - \mathbf{A}_{\mathrm{cl}}) = s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0$$

and that when $\delta = 0$ all of the roots of the characteristic equation are stable. To ensure stability it is essential that $a_0 > 0$. But if we were to expand out the

$\det(s\mathbf{I} - \mathbf{A}_{\text{cl}})$ one would find that

$$a_0 = 1 - \delta\alpha\beta$$

So the perturbation that destabilizes the closed loop system would be

$$\delta_{\text{u}} > \frac{1}{\alpha\beta}$$

The constants $\alpha$ and $\beta$ are, essentially, the LQR and Kalman filter gains, respectively, so we can make the lower bound on the destabilizing perturbation, $\delta_{\text{u}}$, arbitrarily close to zero by making these gains large enough.

Note that a commonly used heuristic in reducing a system's sensitivity function is to increase the penalty matrix $\mathbf{Q}$ in the LQG cost functional, thereby giving a larger LQR gain. Another common heuristic is to assume a larger level of process noise than there actually is to improve the LQG control system's sensitivity to modeling error. Namely, we are treating the unknown modeling uncertainty as additional process noise. This tends to increase the size of the Kalman filter gains. So both strategies will increase the LQR gain, $\alpha$, and the Kalman gain, $\beta$ in the above example. In other words, we can make the controlled system's stability arbitrarily sensitive to modeling error by simply trying to improve performance by increasing the controller gains. For this example the LQG control system's stability is inherently not robust to arbitrarily small modeling error. This sensitivity of LQG and $\mathcal{H}_2$ control to model uncertainty was recognized very early on [Doyle (1978)] and stimulated a great deal of research into *robust optimal control methods*. The remainder of this chapter focuses on how $\mathcal{H}_\infty$ control systems were developed to address this robustness issue.

## 2. Multiplicative Model for Model Uncertainty

To develop a systematic framework for addressing robustness, we first need to introduce methods for modeling that uncertainty. We are, in particular, concerned with *modeling uncertainty*, namely when the actual system's plant, $\mathbf{G}$, is not exactly the same as the nominal plant, $\mathbf{G}_0$, that was used in designing the controller. Our problem is to determine bounds on how large the difference between the actual and nominal plant can be to ensure *robust stability* and *robust performance*.

While there are several frameworks for plant uncertainty (additive, multiplicative, coprime factor, structured, and unstructured), this section focuses on *unstructured multiplicative uncertainty models* since we will use it below in our study of $\mathcal{H}_\infty$ control. A multiplicative uncertainty model assumes the model uncertainty enters in a multiplicative manner (i.e. in series with the nominal plant) as shown on the left side of Fig. 4. In particular if we let $\mathbf{G}_0(s)$ denote the transfer function matrix of the *nominal plant*. Then the actual plant is an element of an *uncertainty set* that was characterize by the ordered pair $(\mathbf{G}_0, \mathbf{W}_\Delta)$. This model set is explicitly shown below

$$\mathbf{G}(s) \in \{(\mathbf{I} + \mathbf{W}_\Delta \mathbf{\Delta})\mathbf{G}_0 \, : \, \|\mathbf{\Delta}\|_{\mathcal{H}_\infty} \leq 1\} \tag{29}$$

The system $\mathbf{\Delta}(s)$ is a stable minimum phase rational transfer function (aka $\mathcal{RH}_\infty$) representing an unknown perturbation to the model. The system $\mathbf{W}_\Delta$ is a "weighting" system that is also stable and minimum phase rational transfer function in $\mathcal{RH}_\infty$. We assume the uncertainty $\mathbf{\Delta}(s)$, though unknown, has an $\mathcal{H}_\infty$ norm less than or equal to one. The system $\mathbf{W}_\Delta$ is a "weighting" system that is also stable and minimum phase rational transfer function in $\mathcal{RH}_\infty$.

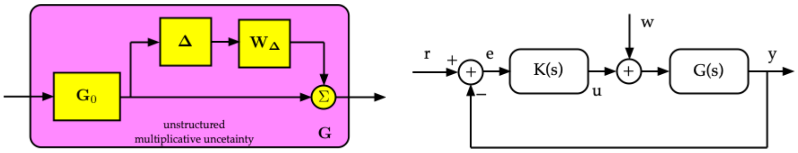


FIGURE 4. (left) Unstructured Multiplicative Uncertainty Model (right) one parameter control system with uncertain plant

We are interested in studying the robust stability and robust performance of the control loop on the right side of Fig. 4 where the plant $\mathbf{G}(s)$ is drawn from the uncertainty set in equation (29). Our uncertainty set is characterized by the ordered pair $(\mathbf{G}_0(s), \mathbf{W}_\Delta(s))$ where $\mathbf{G}_0$ is the nominal plant at the center of the set and $\mathbf{W}_\Delta$ characterizes the size of that set. By *robust stability* we mean that the closed loop system formed from any plant in the uncertainty set is *internally stable* (i.e. $u$ and $e$ remain bounded for all inputs, $r$ and $w$). By *robust performance*, we mean that the performance bound $\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} \leq 1$ for all closed loop sensitivity functions $\mathbf{S}$ formed from any plant $\mathbf{G}(s)$ in the uncertainty set with $\mathbf{W}_p$ being a known $\mathcal{RH}_\infty$ weighting system acting as a frequency-dependent specification on closed-loop performance.

**2.1. Robust Stability:** We want to obtain sufficient conditions for the robust stability of an uncertain one parameter control system with multiplicative uncertainty set $(\mathbf{G}_0, \mathbf{W}_\Delta)$. These conditions will be bounds on the size of the nominal loop function $\mathbf{L}_0(s) = \mathbf{G}_0\mathbf{K}$ where $\mathbf{K}$ is the controller. We will present a "naive" approach for deriving this bound.

Internal stability requires that the internal signals, $e$ and $u$, remain bounded for all bounded inputs, $r$ and $w$. This will be the case provided the transfer function matrix from $\begin{bmatrix} r \\ w \end{bmatrix}$ to $\begin{bmatrix} e \\ u \end{bmatrix}$ is BIBO stable.

$$\begin{bmatrix} e \\ u \end{bmatrix} = \begin{bmatrix} \mathbf{S}(s) & \mathbf{G}(s)\mathbf{S}(s) \\ \mathbf{K}(s)\mathbf{S}(s) & \mathbf{T}(s) \end{bmatrix} \begin{bmatrix} r \\ w \end{bmatrix}$$

where $\mathbf{G}$ is a multiplicative uncertainty of the nominal plant, $\mathbf{K}$ is the controller, and $\mathbf{L}(s) = \mathbf{G}(s)\mathbf{K}(s)$, $\mathbf{S}(s) = (\mathbf{I} + \mathbf{L}(s))^{-1}$, $\mathbf{T}(s) = \mathbf{L}(s)\mathbf{S}(s)$ are the loop function, sensitivity, and complementary sensitivity function, respectively, of the uncertain control system. We assume that the nominal closed loop system has internal stability. We can show this means

$$\det(\mathbf{I} + \mathbf{G}_0(j\omega)\mathbf{K}(j\omega)) \neq 0$$

for any real $\omega$. Let us assume there is a perturbation $\mathbf{\Delta}$ that makes the system *marginally stable*. This means there is some frequency $\omega_0$ for which

$$0 = \det\left\{\mathbf{I} + (\mathbf{I} + \mathbf{W}_\Delta\mathbf{\Delta})\mathbf{G}_0\mathbf{K}(j\omega_0)\right\} \tag{30}$$

We can rewrite this as

$$\begin{aligned} 0 &= \det\left\{\mathbf{I} + \mathbf{G}_0\mathbf{K}(j\omega_0) + \mathbf{W}_\Delta\mathbf{\Delta}\mathbf{G}_0\mathbf{K}(j\omega_0)\right\} \\ &= \det\left\{\left[(\mathbf{I} + \mathbf{W}_\Delta\mathbf{\Delta}\mathbf{G}_0\mathbf{K}(\mathbf{I} + \mathbf{G}_0\mathbf{K})^{-1}(\mathbf{I} + \mathbf{G}_0\mathbf{K})\right](j\omega_0)\right\} \\ &= \det\left\{\left[\mathbf{I} + \mathbf{W}_\Delta\mathbf{\Delta}\mathbf{L}_0(\mathbf{I} + \mathbf{L}_0)^{-1}\right](j\omega_0)\right\} \times \det\left\{[\mathbf{I} + \mathbf{L}_0](j\omega_0)\right\} \qquad (31) \end{aligned}$$

Since the nominal loop is internally stable, we know the second term on the right hand side of equation (31) is not zero. So the above product can only be zero if

$$0 = \det\left\{\left[\mathbf{I} + \mathbf{W}_\Delta\mathbf{\Delta}\mathbf{L}_0(\mathbf{I} + \mathbf{L}_0)^{-1}\right](j\omega_0)\right\} \tag{32}$$

We will use equation (32) to see how big $\mathbf{W}_\triangle\mathbf{\Delta}$ can be before we lose internal stability.

To get this bound we make use of the singular values of a matrix. Let $\Omega_{\mathbf{Q}}$ denote the set of all complex valued matrices, $\mathbf{R}$, such that $\det\{\mathbf{Q}+\mathbf{R}\} = 0$ for a known complex valued matrix, $\mathbf{Q}$. One can show that

$$\min_{\mathbf{R}\in\Omega_{\mathbf{Q}}} \overline{\sigma}(\mathbf{R}) = \underline{\sigma}(\mathbf{Q}) \tag{33}$$

where $\overline{\sigma}(\mathbf{R})$ and $\underline{\sigma}(\mathbf{Q})$ are the largest and smallest singular values of matrices $\mathbf{R}$ and $\mathbf{Q}$, respectively. We can think of $\mathbf{R}$ as a perturbation to the nonsingular matrix $\mathbf{Q}$. What equation (33) asserts is that the smallest $\mathbf{R}$, as measured by its *largest singular value*, that causes $\mathbf{R}+\mathbf{Q}$ to become singular is equal to the minimum singular value of $\mathbf{Q}$. In other words, this relation characterizes how *close* $\mathbf{Q}$ is to becoming singular.

We use equation (33) to see how close an internally stable nominal closed loop system is to being unstable. Let $\mathbf{R} = \mathbf{W}_p\mathbf{\Delta}\mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1}(j\omega_0)$ and we let $\mathbf{Q} = \mathbf{I}$. Equation (33) then shows

$$\min_{\mathbf{R}\in\Omega_{\mathbf{I}}} \overline{\sigma}\left\{\left[\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1}\right](j\omega_0)\right\} = \underline{\sigma}(\mathbf{I}) = 1 \tag{34}$$

So we can conclude that if

$$\overline{\sigma}\left\{\left[\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1}\right](j\omega_0)\right\} < 1 \tag{35}$$

for all $\omega$, then the perturbed closed loop system will be internally stable.

Equation (35) can be made more useful by invoking the sub-multiplicative property of maximum singular values. In particular this property ensures that for all $\omega$

$$\begin{aligned}
\overline{\sigma}\left\{\left[\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1}\right](j\omega)\right\} &\leq \overline{\sigma}\left\{\mathbf{W}_\triangle\mathbf{\Delta}(j\omega)\right\}\times\overline{\sigma}\left\{\left[\mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1}\right](j\omega)\right\} \\
&= \overline{\sigma}\left\{\mathbf{W}_\triangle\mathbf{\Delta}(j\omega_0)\right\}\times\overline{\sigma}\left\{\mathbf{T}_0(j\omega)\right\} \\
&\leq \overline{\sigma}\left\{\mathbf{W}_\triangle(j\omega)\right\}\times\overline{\sigma}\left\{\mathbf{T}_0(j\omega)\right\} \qquad (36)
\end{aligned}$$

where we used the fact that the complementary sensitivity is $\mathbf{T}_0 = \mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1}$ and we used the fact that $\overline{\sigma}(\mathbf{\Delta}(j\omega)) \leq 1$. Inserting this relation into equation (35) implies that if the following is true

$$\overline{\sigma}\left(\mathbf{W}_\triangle(j\omega)\right) < \frac{1}{\overline{\sigma}(\mathbf{T}_0(j\omega))} \tag{37}$$

then condition (35) must also hold. This last relation (37) places an upper bound on the weighting system, $\mathbf{W}_\Delta$, which is a frequency dependent level of uncertainty on the plant. This relation says that to ensure internal stability the perturbed system's uncertainty must be less than the reciprocal of the gain of the nominal system's complementary sensitivity function. This form of the condition is more useful to us because it is posed in terms of the *nominal* complementary sensitivity, $\mathbf{T}_0$.

The $\mathcal{H}_\infty$ gain of a transfer function matrix $\mathbf{G}(s)$, is

$$\|\mathbf{G}\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \overline{\sigma}\left(\mathbf{G}(j\omega)\right)$$

With this we see that our sufficient condition for internal stability becomes a bound on the $\mathcal{H}_\infty$ norm of the weighted complementary sensitivity function. Namely

$$\|\mathbf{W}_\Delta \mathbf{T}_0\|_{\mathcal{H}_\infty} < 1 \tag{38}$$

Since it is relatively easy to compute the $\mathcal{H}_\infty$ norm of a transfer function, this bound is relatively easy to verify. So if our selection of $\mathbf{K}$ ensures the above inequality holds for the *nominal* complementary sensitivity function then we are assured the perturbed (uncertain) closed loop system with uncertainty set $(\mathbf{G}_0, \mathbf{W}_\Delta)$ will also be internally stable. In other words, our multiplicatively perturbed system has *robust stability*.

The preceding argument relied on the condition

$$\det\left\{\left[\mathbf{I} + \mathbf{W}_\Delta \mathbf{\Delta} \mathbf{L}_0(\mathbf{I} + \mathbf{L}_0)^{-1}\right](j\omega)\right\} = 0$$

for some $\omega$. This requires $\mathbf{\Delta}$ be a rational function which may not be the case if our uncertainty arises from neglected nonlinear dynamics in the physical plant. So, in general, the preceding argument does not provide a compelling proof that the given bound is indeed sufficient for robust stability of real-life systems with uncertainty. Nonetheless this bound still holds for the more realistic case and we can prove that bound using a well known theorem from nonlinear systems analysis called the *small gain theorem*. We will examine the small gain theorem in chapter 4 when begin our study of nonlinear control systems. But for now we can provide the following theorem for robust stability of multiplicatively perturbed systems.

THEOREM 9. *Consider the multiplicatively perturbed uncertain control system in Fig. 4 with uncertainty set* $(\mathbf{G}_0, \mathbf{W}_\Delta)$*. Let* $\mathbf{K}$ *be an internally stabilizing controller for the nominal control system. If* $\|\mathbf{W}_\Delta \mathbf{T}_0\|_{\mathcal{H}_\infty} < 1$ *then the closed loop system is internally stable for all perturbed plants in the uncertainty set*

**2.2. Robust Performance:** Stability is a binary measure of control system performance. But in designing good control systems, it is also important that we determine a more quantitive measure of the performance level achieved by the system. As discussed in chapter 1, control system performance (such as tracking error) may be characterized by the induced gain of the weighted closed loop sensitivity, $\mathbf{W}_p\mathbf{S}$ . So obviously if our nominal closed system has achieved a desired level of performance

$$\|\mathbf{W}_p\mathbf{S}_0\|_{\mathcal{H}_\infty} \leq 1$$

then our multiplicatively perturbed system has *robust performance* if it too achieves the same performance level,

$$\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} \leq 1$$

Our objective is to find bounds on the nominal sensitivity functions that ensure robust performance for a control system with unstructured multiplicative uncertainties.

We again consider the closed loop system in Fig. 4 with an uncertainty set, $(\mathbf{W}_\Delta, \mathbf{G}_0)$. For a plant, $\mathbf{G}$, in this uncertainty set, we are interested in enforcing a desired level of tracking error performance with respect to an $\mathcal{RH}_\infty$ weighting system, $\mathbf{W}_p$. In other words, we want to ensure that

$$\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} < 1 \tag{39}$$

for any closed loop system whose plant lies in the uncertainty set, $(\mathbf{G}_0, \mathbf{W}_\Delta)$. As usual we assume the nominal closed loop system is internally stable so the given controller $\mathbf{K}$ ensures the nominal sensitivity function $\mathbf{S}_0 = (\mathbf{I}+\mathbf{L}_0)^{-1}$ is internally stable.

A useful relationship between the uncertain and nominal sensitivity functions can be derived as follows. Note that

$$\begin{aligned}\mathbf{S} &= (\mathbf{I}+(\mathbf{I}+\mathbf{W}_\triangle\mathbf{\Delta})\mathbf{G}_0\mathbf{K})^{-1}\\ &= (\mathbf{I}+\mathbf{G}_0\mathbf{K}+\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{G}_0\mathbf{K})^{-1}\end{aligned}$$

We factor the return difference, $\mathbf{I}+\mathbf{G}_0\mathbf{K}$, in the same way we did for our robust stability analysis to get

$$\begin{aligned}\mathbf{S} &= \left[(\mathbf{I}+\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1}(\mathbf{I}+\mathbf{L}_0)\right]^{-1}\\ &= (\mathbf{I}+\mathbf{L}_0)^{-1}(\mathbf{I}+\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0(\mathbf{I}+\mathbf{L}_0)^{-1})^{-1}\\ &= \mathbf{S}_0\,(\mathbf{I}+\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0\mathbf{S}_0)^{-1} \qquad (40)\end{aligned}$$

We will now use the preceding equation (40) to find bounds that ensure robust performance.

Let us start with a candidate bound on $\mathbf{S}_0$ and $\mathbf{T}_0$ and demonstrate that this bound enforces robust performance on the perturbed plant. Our candidate bound requires

$$1 > \overline{\sigma}(\mathbf{W}_p(j\omega))\overline{\sigma}(\mathbf{S}_0(j\omega)) + \overline{\sigma}(\mathbf{W}_\triangle(j\omega))\overline{\sigma}(\mathbf{T}_0(j\omega)) \qquad (41)$$

for all $\omega$. This bound in equation (41) can be rewritten as

$$\overline{\sigma}(\mathbf{W}_p(j\omega))\overline{\sigma}(\mathbf{S}_0(j\omega)) < 1 - \overline{\sigma}(\mathbf{W}(j\omega))\overline{\sigma}(\mathbf{T}_0(j\omega))$$

Because $\overline{\sigma}(\mathbf{\Delta}(j\omega)) < 1$ for all $\omega$ we can rewrite the above inequality as

$$\begin{aligned}\overline{\sigma}(\mathbf{W}_p(j\omega))\overline{\sigma}(\mathbf{S}_0(j\omega)) &< 1 - \overline{\sigma}(\mathbf{W}_\triangle(j\omega))\overline{\sigma}(\mathbf{T}_0(j\omega))\\ &< 1 - \overline{\sigma}(\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0\mathbf{S}_0)\\ &= \underline{\sigma}(\mathbf{I}+\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0\mathbf{S}_0(j\omega))\end{aligned}$$

for all $\omega$ This last inequality holds if and only if

$$\begin{aligned}1 &> \overline{\sigma}(\mathbf{W}_p(j\omega))\overline{\sigma}(\mathbf{S}_0(j\omega))\overline{\sigma}((\mathbf{I}+\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0\mathbf{S}_0(j\omega))^{-1})\\ &> \overline{\sigma}(\mathbf{W}_p(j\omega)\mathbf{S}_0(j\omega)(\mathbf{I}+\mathbf{W}_\triangle\mathbf{\Delta}\mathbf{L}_0\mathbf{S}_0(j\omega))^{-1})\\ &> \overline{\sigma}(\mathbf{W}_p\mathbf{S}(j\omega))\end{aligned}$$

where the last line comes from the relationship between $\mathbf{S}$ and $\mathbf{S}_0$ in equation (40). This last inequality is, of course, the performance specification in equation (39). So we have just established that the condition in equation (41) is sufficient to ensure robust performance.

Note that if this sufficient condition in equation (41) holds, then so too does

$$\begin{aligned} 1 &> \overline{\sigma}(\mathbf{W}_p(j\omega))\overline{\sigma}(\mathbf{S}_0(j\omega)) > \overline{\sigma}(\mathbf{W}_p\mathbf{S}_0(j\omega)) \\ 1 &> \overline{\sigma}(\mathbf{W}_\Delta(j\omega))\overline{\sigma}(\mathbf{T}_0(j\omega)) > \overline{\sigma}(\mathbf{W}\mathbf{T}_0(j\omega)) \end{aligned}$$

for all $\omega$. This is the same as an $\mathcal{H}_\infty$ bound on the weighted sensitivity functions

$$\|\mathbf{W}_p\mathbf{S}_0\|_{\mathcal{H}_\infty} < 1, \quad \|\mathbf{W}_\Delta\mathbf{T}_0\|_{\mathcal{H}_\infty} < 1 \tag{42}$$

The first inequality means that the ”nominal” system satisfies the performance specification. We refer to this as *nominal performance* (NP). The second inequality is the robust stability (RS) condition we derived in the preceding subsection. So we have just shown that if we enforce robust performance (RP) using equation (41) then we also ensure the performance of the nominal system (not surprising) and the robust stability of the perturbed system (again not surprising). In view of the above discussion it should be apparent that the robust performance condition in equation (41) may be cast, alternatively, as

$$\|\mathbf{W}_p\mathbf{S}_0\|_{\mathcal{H}_\infty} + \|\mathbf{W}_\Delta\mathbf{T}_0\|_{\mathcal{H}_\infty} < 1 \tag{43}$$

## 3. Generalized Regulator Problem

The main problem we want to address concerns the one-parameter control system on the left side of Fig 5. In particular, we wanted to design a controller, $\mathbf{K}$, so our control system has *robust performance* (i.e. $\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} < 1$) with respect to a performance specification, $\mathbf{W}_p$, on the sensitivity function, $\mathbf{S}$, for any plant in the uncertainty set, $(\mathbf{G}_0, \mathbf{W}_\Delta)$. We saw that if the controller was chosen so the nominal sensitivity functions satisfied equation (42) then our closed loop system would have robust performance and robust stability. The issue now is how does one go about finding such a controller, $\mathbf{K}$?

Our approach involves reframing the controller synthesis problem as an optimization problem posed with respect to a ”canonical” feedback system known as the *generalized regulator* shown on the right side of Fig¿ 5. We then pose the search for $\mathbf{K}$ as an optimization problem that minimizes the $\mathcal{H}_\infty$ norm of the generalized regulator with respect to controller $\mathbf{K}$ subject to internal stability. The utility of this approach is that any feedback control system can be recast as a generalized regulator, so that finding a single method for solving the generalized regulator problem will allow us to solve a wide range of control problems. This section reviews the generalized regulator, demonstrates how a one-parameter control system can be transformed into a generalized regulator, and then formally poses the generalized regulator problem.

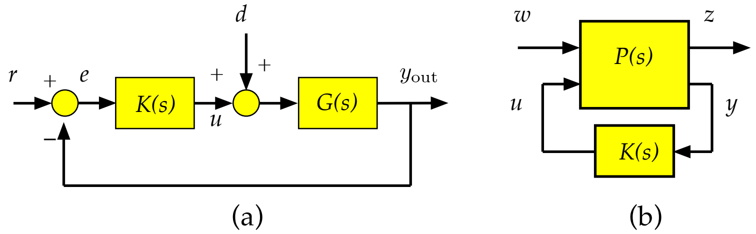


FIGURE 5. (a) traditional one parameter control system (b) Generalized regulator

The generalized regulator block diagram is shown on the right side of Fig. 5. The generalized regulator is the feedback interconnection of an *augmented plant*, $\mathbf{P}(s)$, and a controller $\mathbf{K}(s)$. The augmented plant's inputs are grouped into two categories; *disturbances*, $w$, and *controls*, $u$. The difference between these categories is that a disturbance is exogenous to the feedback system and the controls are generated internally by the controller. The augmented plant's outputs are also grouped into two categories; *penalties*, $z$, and *observations*, $y$. The difference between these output categories is that a penalty, $z$, is a virtual signal that may not actually be observable from outside the feedback system, whereas the observations, $y$, are accessible and are used by the controller to generate the control signal $u$.

Since the augmented plant, $\mathbf{P}(s)$, maps inputs ($w$ and $u$) onto outputs ($z$ and $y$), it will be convenient to partition the transfer function matrix into blocks

$$\mathbf{P}(s) = \left[\begin{array}{c|c} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \hline \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{array}\right]$$

that conform to the input/output signal categories. In particular, this means that we may express the closed loop system (with a slight abuse of notation) as

$$\begin{aligned} z &= \mathbf{P}_{11} w + \mathbf{P}_{12} u \\ y &= \mathbf{P}_{21} w + \mathbf{P}_{22} u \\ u &= \mathbf{K} y \end{aligned}$$

This interconnection is shown graphically in Fig. 5 as the generalized regulator. In particular, if $w$ is an exogenous disturbance and $z$ is a penalty signal whose size at time $t$ signifies how ”poorly” the system is performing at that time instant, then the control objective is to select $\mathbf{K}$ so the norm of $z$ due to $w$ is as small as possible. In other words, we want a controller that minimizes the control system's *sensitivity* to the disturbance. We know this can also be posed in terms of minimizing the induced gain of the generalized regulator.

To characterize this sensitivity, it will be necessary to get an explicit expression for the closed loop map from $w$ to $z$. Let $y$ and $u$ be treated as *internal signals* and note that

$$y = \mathbf{P}_{21} w + \mathbf{P}_{22} u = \mathbf{P}_{21} w + \mathbf{P}_{22} \mathbf{K} y$$

Solving for $y$ yields

$$y = (\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1} \mathbf{P}_{21} w$$

Insert this expression for $y$ back into our equation for $z$ gives the desired feedback map

$$z = \left[ \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21} \right] w \overset{\text{def}}{\equiv} \mathcal{F}_{\ell}(\mathbf{P}, \mathbf{K})$$

The expression for $\mathcal{F}_{\ell}(\mathbf{P}, \mathbf{K})$ is called a lower *linear fractional transformation* or LFT. LFT's are canonical representations for feedback maps and may be seen as a matrix generalization of a bilinear function.

We now want to see how a particular control system might be transformed into a generalized regulator. We consider the one-parameter control system in Fig. 5 and proceed to *pull out* the controller from this system and then redraw the block diagram so it conforms to the interconnection of an augmented plant, $\mathbf{P}$, and the controller $\mathbf{K}$ that we pulled out. Figure 6 shows the resulting change in the block diagram. In this case the generalized regulator's disturbance inputs are $w = \begin{bmatrix} d \\ r \end{bmatrix}$ and the control input is $u$. The penalty output $z = \begin{bmatrix} u \\ e \end{bmatrix}$ and the observation output is $y = e$ is the tracking error, $e$. We can identify the blocks of the augmented plant by simply tracking the signal flow from inputs to outputs in the block diagram in Fig. 6 to see that

$$\mathbf{P}(s) = \left[\begin{array}{c|c} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \hline \mathbf{P}_{21} & \mathbf{P}_{22} \end{array}\right] = \left[\begin{array}{cc|c} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ -\mathbf{G} & \mathbf{I} & -\mathbf{G} \\ \hline -\mathbf{G} & \mathbf{I} & -\mathbf{G} \end{array}\right]$$

thereby finding for an expression for the augmented plant in terms of the original control loops plant, $\mathbf{G}(s)$.

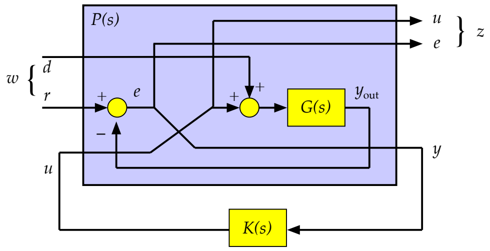


FIGURE 6. Generalized Regulator for unity gain feedback system on left side of Fig. 5

Our control objective is to minimize the sensitivity of the penalty output signal $z = \begin{bmatrix} u \\ e \end{bmatrix}$. Note that if this involves minimizing the $\mathcal{L}_2$ norm of $z$, then it corresponds to minimizing

$$\|z\|^2_{\mathcal{L}_2} = \int_0^\infty z^T z d\tau = \int_0^\infty (e^T e + u^T u) d\tau$$

which is the standard quadratic cost functional we studied in chapter 2. If we know $\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})$ is input/output stable then for any input signal $w$ with a bounded $\mathcal{L}_2$ norm we know $z$ will also have a bounded $\mathcal{L}_2$ norm. We also know

$$\|z\|_{\mathcal{L}_2} \leq \|\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})\|_{\mathcal{L}_2-\text{ind}} \|w\|_{\mathcal{L}_2}$$

So we can minimize the energy in the penalty signal, $z$, by simply minimizing the $\mathcal{L}_2$-induced gain of the LFT, $\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})$. Of course if this map is stable and rational, then

$$\|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\|_{\mathcal{L}_2-\text{ind}} = \sup_{\alpha>0}\sup_{\omega}\overline{\sigma}(\mathcal{F}_\ell(\mathbf{P}(\alpha+j\omega),\mathbf{K}(\alpha+j\omega)) \overset{\text{def}}{\equiv} \|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\|_{\mathcal{H}_\infty}$$

So the problem of finding a controller $\mathbf{K}$ that minimizes the energy in the penalty signal $z$ is recast as an optimization problem over the system space

$$\begin{array}{ll} \text{minimize:} & \|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\|_{\mathcal{H}_\infty} \\ \text{with respect to:} & \mathbf{K} \\ \text{subject to:} & \text{Internal Stability} \end{array}$$

The resulting controller is an $\mathcal{H}_\infty$ control law and provides the basis for developing control systems that are ”optimal” with respect to the penalty signal’s energy and ”robust” with respect to modeling uncertainty, thereby addressing the robustness issues of the classical LQG controller. We will look at two examples illustrating how real-life robust control problems can be posed as generalized regulator problems. We then discuss how this optimization problem is solved and introduce MATLAB toolkits often used in synthesizing $\mathcal{H}_\infty$ controllers.

## 4. Mixed Sensitivity Problem

The mixed sensitivity problem formulates a generalized regulator problem whose solution, $\mathbf{K}$, is the controller for a one parameter control system (Fig. 5) with a tracking error specification $\mathbf{W}_p$ and a plant with an unstructured multiplicative uncertainty, $(\mathbf{G}_0, \mathbf{W}_\Delta)$. For this one parameter control system we know the tracking performance requirement is

$$\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} < 1$$

for any sensitivity function, $\mathbf{S} = (\mathbf{I}+\mathbf{G}\mathbf{K})^{-1}$ for a plant in the uncertainty set $(\mathbf{G}_0, \mathbf{W}_\Delta)$. We know this condition will be satisfied if inequality (43) holds for the nominal sensitivities. In particular, if we design $\mathbf{K}$ so that

$$\left\| \begin{bmatrix} \mathbf{W}_p\mathbf{S}_0 \\ \mathbf{W}_\Delta\mathbf{T}_0 \end{bmatrix} \right\|_{\mathcal{H}_\infty} < \frac{1}{2}$$

then this implies inequality (43)

$$\|\mathbf{W}_p\mathbf{S}_0\|_{\mathcal{H}_\infty} + \|\mathbf{W}_\Delta\mathbf{T}_0\|_{\mathcal{H}_\infty} < 1$$

for any uncertain plant in the uncertainty set, thereby guaranteeing the robust performance of the uncertain closed loop system. Since $\mathbf{S}_0$ and $\mathbf{T}_0$ are both closed-loop maps for the one-parameter control systems, this suggests we should draw the generalized regulator so these two sensitivity functions map to the generalized regulator's penalty outputs. This approach to formulating the generalized regulator problem is called the *mixed sensitivity problem*.

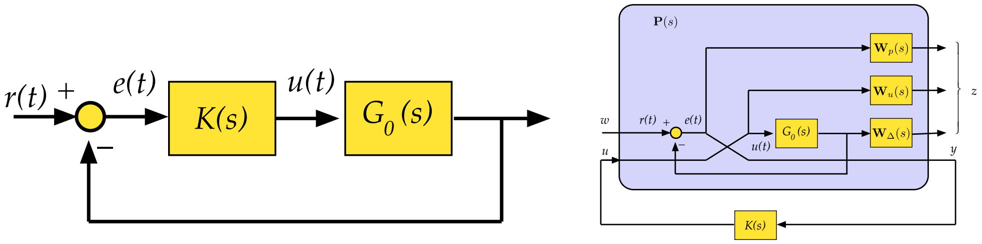


FIGURE 7. (left) nominal unity gain feedback system (right) Generalized Regulator for Mixed Sensitivity Problem

Let us consider the one-parameter control system for the nominal plant and redraw it as shown in Fig. 7. Our problem is then to find $\mathbf{K}$ such that

$$\left\| \begin{bmatrix} \mathbf{W}_p\mathbf{S}_0 \\ \mathbf{W}_u\mathbf{K}\mathbf{S}_0 \\ \mathbf{W}_\Delta\mathbf{T}_0 \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq 1$$

where $\mathbf{W}_u$ is another $\mathcal{RH}_\infty$ system that is a frequency dependent weight on the control effort, $u$. This is again a mixed-sensitivity problem but now we must also constrain the gain of an additional nominal sensitivity function $\mathbf{K}\mathbf{S}_0$ that characterizes the control effort used to enforce the nominal performance and robust stability requirements.

It is customary to solve the mixed sensitivity problem by first recasting the transfer function matrix $\begin{bmatrix} \mathbf{W}_p\mathbf{S}_0 \\ \mathbf{W}_u\mathbf{K}\mathbf{S}_0 \\ \mathbf{W}_\Delta\mathbf{T}_0 \end{bmatrix}$ as a generalized regulator, $\mathbf{F}_\ell(\mathbf{P},\mathbf{K})$. This is done by pulling out the controller from the nominal closed-loop system and weighting the internal signals $e$, $u$, and the plant's output with the $\mathcal{RH}_\infty$ weighting systems. The generalized regulator for the mixed sensitivity problem is shown in

Fig. 7. The augmented plant for this regulator can be shown to be

$$\mathbf{P}(s) = \left[\begin{array}{c|c} \mathbf{W}_p & -\mathbf{W}_p\mathbf{G}_0 \\ \mathbf{0} & \mathbf{W}_u \\ \mathbf{0} & \mathbf{W}_\Delta\mathbf{G}_0 \\ \hline \mathbf{I} & -\mathbf{G}_0 \end{array}\right]$$

It can be shown (tedious algebra) that the augmented plant $\mathbf{P}$ given above has the state space realization

$$\begin{aligned} \mathbf{P}(s) \quad &\overset{s}{=} \quad \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array}\right] \\ &= \quad \left[\begin{array}{cccc||c|c} \mathbf{A}_G & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_G \\ -\mathbf{B}_p\mathbf{C}_G & \mathbf{A}_p & \mathbf{0} & \mathbf{0} & \mathbf{B}_p & -\mathbf{B}_p\mathbf{D}_G \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_u & \mathbf{0} & \mathbf{0} & \mathbf{B}_u \\ \mathbf{B}_\Delta\mathbf{C}_G & \mathbf{0} & \mathbf{0} & \mathbf{A}_\Delta & \mathbf{0} & \mathbf{B}_\Delta\mathbf{D}_G \\ \hline\hline -\mathbf{D}_p\mathbf{C}_G & \mathbf{C}_p & \mathbf{0} & \mathbf{0} & \mathbf{D}_p & -\mathbf{D}_p\mathbf{D}_G \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_u & \mathbf{0} & \mathbf{0} & \mathbf{D}_u \\ \mathbf{D}_\Delta\mathbf{C}_G & \mathbf{0} & \mathbf{0} & \mathbf{C}_\Delta & \mathbf{0} & \mathbf{D}_\Delta\mathbf{D}_G \\ \hline -\mathbf{C}_G & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & -\mathbf{D}_G \end{array}\right] \end{aligned}$$

where

$$\begin{aligned} \mathbf{G}_0 \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A}_G & \mathbf{B}_G \\ \hline \mathbf{C}_G & \mathbf{D}_G \end{array}\right], \quad \mathbf{W}_p \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A}_p & \mathbf{B}_p \\ \hline \mathbf{C}_p & \mathbf{D}_p \end{array}\right], \\ \mathbf{W}_u \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A}_u & \mathbf{B}_u \\ \hline \mathbf{C}_u & \mathbf{D}_u \end{array}\right], \quad \mathbf{W}_\Delta \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A}_\Delta & \mathbf{B}_\Delta \\ \hline \mathbf{C}_\Delta & \mathbf{D}_\Delta \end{array}\right] \end{aligned}$$

With the preceding characterization of the mixed sensitivity problem's augmented plant, $\mathbf{P}$, we can now formalize the statement of the controller synthesis problem. Essentially, this is to find an *internally stabilizing* controller $\mathbf{K}$ that minimizes $\|\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})\|_{\mathcal{H}_\infty}$. The minimum that we achieve represents the *optimal* $\mathcal{H}_\infty$ controller for the mixed sensitivity problem.

## 5. Structured Uncertainty Problem

The mixed sensitivity problem assumes the plant had an unstructured multiplicative uncertainty. In many applications, however, we have some knowledge about the nature of the uncertainties. In a state-based model for instance, we may know the uncertainties are due to a specific model parameter. For such systems, it makes more sense to use a *structured* uncertainty model. Remember that $\mathcal{H}_\infty$ control is essentially an "old man's" strategy for hedging against risk. One designs a controller for the "worst-case" uncertainty. If we can structure the uncertainty, then it may be possible to hedge less.

Let us assume the *nominal* open-loop plant has a state-space realization of the form

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \left[\begin{array}{c|cc} \mathbf{A}_0 & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array}\right] \begin{bmatrix} x \\ w \\ u \end{bmatrix}$$

We assume our uncertainty in the actual process is a result of a *structured* and *additive* perturbation of the nominal system matrix, $\mathbf{A}_0$. In particular this means there are matrices $\mathbf{M} \in \mathbb{R}^{n\times\ell_1}$ and $\mathbf{N} \in \mathbb{R}^{n\times\ell_2}$ such that the perturbed system, $\mathbf{P}$, has the state space realization

$$\mathbf{P} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A}_0 - \mathbf{M}\boldsymbol{\Delta}\mathbf{N}^T & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array}\right]$$

where $\boldsymbol{\Delta} \in \mathbb{R}^{\ell_1\times\ell_2}$ is a real valued matrix whose matrix norm $|\boldsymbol{\Delta}| \leq 1$.

**Remark:** The uncertainty matrix $\boldsymbol{\Delta}$ is a real-valued matrix rather than a matrix of transfer functions. This means the uncertainty captures perturbations to the parameters of the nominal plant's state space realization. The unstructured model works well when the information we have regarding plant uncertainty comes from experimental measurements of the open-loop plant's frequency response. In many highly engineered systems, however, we have a state space realization developed from first principle modeling of the system. In the structured case the uncertainty matrix $\boldsymbol{\Delta}$ captures our uncertainties in the physical value of the coefficients in this first principle model.

We now rewrite the uncertain *open* loop augmented plant, $\mathbf{P}$, as an LFT in which the uncertainty matrix $\mathbf{\Delta}$ has been pulled out. In particular, pulling out $\mathbf{\Delta}$ means we treat it as a constant gain system that is coupled to the plant through the $\mathbf{M}$ and $\mathbf{N}$ matrices. This "pulling-out" procedure creates a new input and output for the nominal plant that we denote as $\tilde{w}$ and $\tilde{z}$, respectively. The new objective signal, $\tilde{z}$, for the augmented plant is

$$\tilde{z} = \begin{bmatrix} \mathbf{N}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x \\ \tilde{w} \\ w \\ u \end{bmatrix}$$

and the new disturbance signal, $\tilde{w}$, enters the augmented plant as

$$\dot{x} \quad = \quad \mathbf{A}_0 x + \mathbf{M}\tilde{w} + \mathbf{B}_1 w + \mathbf{B}_2 u$$

where

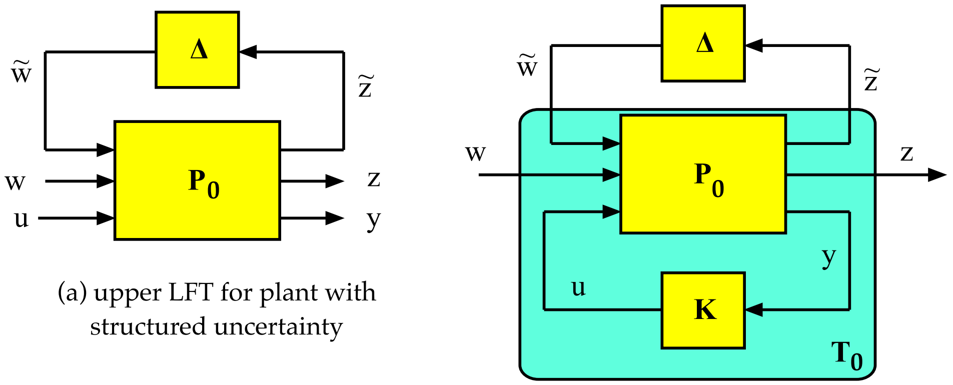
$$\tilde{w} = \mathbf{\Delta}\tilde{z}$$

So the LFT for our uncertain plant, $\mathbf{P}$, can be written as the feedback interconnection of a nominal plant, $\mathbf{P}_0$, with the uncertainty terms in $\mathbf{\Delta}$.

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \tilde{z} \\ z \\ y \end{bmatrix} &= \left[\begin{array}{c|ccc} \mathbf{A}_0 & \mathbf{M} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{N}^T & 0 & 0 & 0 \\ \mathbf{C}_1 & 0 & 0 & 0 \\ \mathbf{C}_2 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ \tilde{w} \\ w \\ u \end{bmatrix} \\ \tilde{w} &= \Delta\tilde{z} \end{aligned}$$

Note that the feedback block, $\mathbf{\Delta}$, reinjects the disturbance into the plant through $\tilde{w}$, rather than $u$. So this is not the *lower* LFT we used earlier in formulating the generalized regulator. In particular, it is an *upper* LFT as shown in Fig. 8(a).

We now attach a controller to the upper LFT representing the uncertain system. This results in the controlled system (generalized regulator) shown in Fig 8(b) with



(a) upper LFT for plant with structured uncertainty

(b) generalized regulator for plant with structure uncertainty

FIGURE 8. (left) upper LFT for open loop plant with structured uncertainty (right) generalized regulator for system with structured uncertainty

equations

$$
\begin{aligned}
\begin{bmatrix} \tilde{z} \\ z \\ y \end{bmatrix} &= \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} \end{bmatrix} \begin{bmatrix} \tilde{w} \\ w \\ u \end{bmatrix} \\
\tilde{w} &= \mathbf{\Delta}\tilde{z} \\
u &= \mathbf{K}[y]
\end{aligned}
$$

where we have conformally partitioned the nominal open loop plant $\mathbf{P}_0$ with respect to the three types of inputs ($\tilde{w}$, $w$, and $u$) and outputs ($\tilde{z}$, $z$, and $y$).

From Fig. 8(b), one can see that there is a two-port system $\mathbf{T}_0$ with inputs $\tilde{w}$ and $w$ and outputs $\tilde{z}$ and $z$ that can be written as a 2 by 2 block of systems

$$
\begin{aligned}
\mathbf{T}_0 &= \mathcal{F}_\ell(\mathbf{P}_0, \mathbf{K}) \\
&= \left[\begin{array}{c|c} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \hline \mathbf{T}_{21} & \mathbf{T}_{22} \end{array}\right] \\
&= \left[\begin{array}{c|c} \mathbf{P}_{11} - \mathbf{P}_{13}\mathbf{K}(\mathbf{I}+\mathbf{P}_{33}\mathbf{K})^{-1}\mathbf{P}_{31} & \mathbf{P}_{12} - \mathbf{P}_{13}\mathbf{K}(\mathbf{I}+\mathbf{P}_{33}\mathbf{K})^{-1}\mathbf{P}_{32} \\ \hline \mathbf{P}_{21} - \mathbf{P}_{23}\mathbf{K}(\mathbf{I}+\mathbf{P}_{33}\mathbf{K})^{-1}\mathbf{P}_{31} & \mathbf{P}_{22} - \mathbf{P}_{23}\mathbf{K}(\mathbf{I}+\mathbf{P}_{33}\mathbf{K})^{-1}\mathbf{P}_{32} \end{array}\right]
\end{aligned}
$$

If we then connect the $\mathbf{\Delta}$ block to $\mathbf{T}_0$ we obtain the following closed-loop map for the uncertain system

$$\begin{aligned} \mathbf{T}_{zw} &= \mathbf{T}_{22} - \mathbf{T}_{21}\mathbf{\Delta}(\mathbf{I} + \mathbf{T}_{11}\mathbf{\Delta})^{-1}\mathbf{T}_{12} \\ &:= \mathcal{F}_u(\mathbf{T}_0, \mathbf{\Delta}) \\ &= \mathcal{F}_u(\mathcal{F}_\ell(\mathbf{P}_0, \mathbf{K}), \mathbf{\Delta}) \end{aligned}$$

Our objective is then to find the controller $\mathbf{K}$ such that $\|\mathcal{F}_u(\mathbf{T}_0, \Delta)\|_{\mathcal{H}_\infty} \leq 1$ for all $|\mathbf{\Delta}| \leq 1$, since this controller would ensure the performance achieved by all uncertain closed-loop maps.

To find such a controller, let

$$\begin{bmatrix} \tilde{z} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \begin{bmatrix} \tilde{w} \\ w \end{bmatrix}$$

Note that if $\|\mathbf{T}_0\|_{\mathcal{H}_\infty} \leq 1$ then

$$\|\tilde{z}\|^2_{\mathcal{L}_2} - \|\tilde{w}\|^2_{\mathcal{L}_2} \leq \|w\|^2_{\mathcal{L}_2} - \|z\|^2_{\mathcal{L}_2}$$

Since $\tilde{w} = \mathbf{\Delta}\tilde{z}$ and $|\mathbf{\Delta}| < 1$, we can easily see that

$$\|\tilde{w}\|^2_{\mathcal{L}_2} \leq \|\tilde{z}\|^2_{\mathcal{L}_2}$$

and this, in turn implies that

$$\|z\|^2_{\mathcal{L}_2} \leq \|w\|^2_{\mathcal{L}_2}$$

This relationship, of course means that $\|\mathcal{F}_u(\mathbf{T}_0, \mathbf{\Delta})\|_{\mathcal{H}_\infty} \leq 1$. In other words, if we can design $\mathbf{K}$ so that $\|\mathbf{T}_0\|_{\mathcal{H}_\infty} \leq 1$, then for the set of uncertainty where $|\Delta| \leq 1$, we immediately know the uncertain closed-loop system $\mathcal{F}_u(\mathbf{T}, \mathbf{\Delta})$ also has an $\mathcal{H}_\infty$ norm less than one. In other words, the controlled system has robust performance. Again note that our robust performance problem has been cast in terms of a generalized regulator problem. Since $\mathbf{T}_0 = \mathcal{F}_\ell(\mathbf{P}_0, \mathbf{K})$, this means that to achieve robust performance with respect to *structured additive* perturbations of the state-space model, we need to find an internally stabilizing $\mathbf{K}$ such that $\|\mathcal{F}_\ell(\mathbf{P}_0, \mathbf{K})\|_{\mathcal{H}_\infty} \leq 1$. This is the same type of problem we discussed before with the unstructured uncertainty model. The difference rests with what the nominal plant $\mathbf{P}_0$ is.

## 6. Full Information $\mathcal{H}_\infty$ Problem

The preceding two sections showed how to formulate robust control problem in terms of minimizing the $\mathcal{H}_\infty$ norm of a generalized regulator. The next three sections examine how this minimization is done. The derivation of the $\mathcal{H}_\infty$ optimal controller is done in much the same way we established the optimality of the LQG controller. We first pose a problem in which all of the states are accessible to the controller and then show how the output problem can be seen as adding an observer to the controller that estimates the states from the observation. For the $\mathcal{H}_\infty$ control, this first step is called the *Full-Information* (FI) $\mathcal{H}_\infty$ generalized regulator problem. The second step involves finding a solution to an *output estimation* (OE) problem which when combined with the FI controller yields a solution to the *output feedback* $(OF)\mathcal{H}_\infty$ generalized regulator problem. This section confines its attention to the full-information problem. The OE and OF problems are covered in the next two sections.

Consider an augmented plant in the closed-loop map $\mathcal{F}_\ell(\mathbf{P}_{\text{FI}}, \mathbf{K})$ where $\mathbf{P}_{\text{FI}}$ is called the *full information* (FI) plant with the state space equations

$$\begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}_1 w + \mathbf{B}_2 u \\ z &= \mathbf{C}x + \mathbf{D}_{12} u \\ y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} x + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} w \end{aligned}$$

The following theorem provides conditions used in determine a controller that achieves a specified level, $\gamma$, of control system performance.

THEOREM 10. $\mathcal{H}_\infty$ **FI Controller:** *Consider closed loop map $\mathcal{F}_\ell(\mathbf{P}_{FI}, \mathbf{K})$ where $\mathbf{P}_{\text{FI}}$ is the full information augmented plant with state space realization*

$$\mathbf{P}_{\text{FI}} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C} & \mathbf{0} & \mathbf{D}_{12} \\ \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \end{array}\right] \tag{44}$$

*where $n$ is the state dimension, $m$ is the dimension of the measurement $y$, and $q$ is the dimension of the control input $u$. We assume that*

*(1)* $(\mathbf{A}, \mathbf{B}_2)$ *is stabilizable and* $(\mathbf{C}_2, \mathbf{A})$ *is detectable.*

*(2) Matrix* $\mathbf{D}_{12}$ *has full column rank.*

*(3) The following conditions hold for all real* $\omega$

$$\begin{aligned} \text{rank} \begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{12} \end{bmatrix} &= n + m \\ \text{rank} \begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_1 \\ \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} &= n + q \end{aligned}$$

*If there exists a symmetric positive semidefinite matrix* $\mathbf{X}$ *that satisfies the following algebraic Riccati equation*

$$\mathbf{X}\mathbf{A} + \mathbf{A}^T\mathbf{X} - \mathbf{X}\left(\mathbf{B}_2\mathbf{B}_2^T - \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T\right)\mathbf{X} + \mathbf{C}_1^T\mathbf{C}_1 = 0 \tag{45}$$

*then the control* $u = -\mathbf{B}_2^T\mathbf{X}x(t)$ *internally stabilizes the closed-loop system while enforcing the performance constraint* $\|\mathcal{F}_\ell(\mathbf{P}_{\mathrm{FI}}, \mathbf{K})\|_{\mathcal{H}_\infty} < \gamma$.

There are three assumptions in this theorem that need to be commented on.

- $(\mathbf{A}, \mathbf{B}_2)$ *is stabilizable and* $(\mathbf{C}_2, \mathbf{A})$ *is detectable*
  This assumption is necessary and sufficient for the existence of internally stabilizing controllers. These conditions essentially says the uncontrollable and unobservable poles of the system can be arbitrarily placed.
- *Matrix* $\mathbf{D}_{12}$ *has full column rank and matrix* $\mathbf{D}_{21}$ *has full row rank*
  This assumption is necessary for a well-posed control problem that does not allow the controller to exert arbitrarily large controls in achieving the tracking requirement. We refer to such unbounded controls as being *singular*. If these conditions are not satisfied in the original problem, then one can modify the system matrices so they do hold. In some cases, this is done through proper selection of weighting matrices.

- *The following conditions hold for all real $\omega$*

$$\begin{aligned} \text{rank}\begin{bmatrix} \mathbf{A}-j\omega\mathbf{I} & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{12} \end{bmatrix} &= n+m \\ \text{rank}\begin{bmatrix} \mathbf{A}-j\omega\mathbf{I} & \mathbf{B}_1 \\ \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \end{bmatrix} &= n+q \end{aligned}$$

  This condition is only sufficient for a solution to exist. In particular these conditions are needed to ensure that certain algebraic Riccati equations have stabilizing solutions. Essentially these conditions are equivalent to requiring that the plant has no poles on the imaginary axis. Details on why these conditions exist will be found in Zhou et al. (1996) which has a self-contained chapter on the algebraic Riccati equation.

**Proof:** A key lemma used in proving this theorem is the Kalman-Yakubovich-Popov (KYP) lemma. This classical result states a state-space realization has an $\mathcal{H}_\infty$ norm less than a specified constant $\gamma$ if and only if there exists a symmetric positive definite matrix that satisfies the so-called $\mathcal{H}_\infty$ algebraic Riccati equation. The KYP lemma is stated and proven in this chapter's appendix section .

Let us apply the control $u = -\mathbf{B}_2^T\mathbf{X}x$ to our closed loop system and obtain the following state space equations

$$\begin{aligned} \dot{x} &= (\mathbf{A}-\mathbf{B}_2\mathbf{B}_2^T)\mathbf{X}x + \mathbf{B}_1 w \\ z &= \begin{bmatrix} \mathbf{C}_1 \\ -\mathbf{D}_{12}\mathbf{B}_1^T\mathbf{X} \end{bmatrix} x \end{aligned}$$

for the closed-loop system. To verify that this system is asymptotically stable, we add and subtract $\mathbf{X}\mathbf{B}_2\mathbf{B}_2^T\mathbf{X}$ to the Riccati equation (45) and rearrange the resulting terms to obtain

$$\begin{aligned} \mathbf{0} &= \mathbf{X}\mathbf{A}+\mathbf{A}^T\mathbf{X}-\mathbf{X}(\mathbf{B}_2\mathbf{B}_2^T-\gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T)\mathbf{X}+\mathbf{C}_1^T\mathbf{C}_1 \\ &\quad +\mathbf{X}(\mathbf{B}_2\mathbf{B}_2^T-\mathbf{B}_2\mathbf{B}_2\mathbf{B}_2^T)\mathbf{X} \\ &= \mathbf{X}(\mathbf{A}-\mathbf{B}_2\mathbf{B}_2^T\mathbf{X})+(\mathbf{A}-\mathbf{B}_2\mathbf{B}_2^T\mathbf{X})^T\mathbf{X} \\ &\quad +\gamma^{-2}\mathbf{X}\mathbf{B}_1\mathbf{B}_1^T\mathbf{X}+\mathbf{X}\mathbf{B}_2\mathbf{B}_2^T\mathbf{X}+\mathbf{C}_1^T\mathbf{C}_1 \end{aligned}$$

Since $\mathbf{X} \geq 0$, we know from linear systems theory that every unstable mode of $\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{X}$ is unobservable through $\left[\begin{array}{ccc} -\gamma^{-1}\mathbf{X}\mathbf{B}_1 & \mathbf{X}\mathbf{B}_2 & \mathbf{C}_1^T \end{array}\right]^T$. Let $(\lambda, x)$ be an unstable mode of $\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{X}$, then we know that

$$\begin{aligned} (\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{X})x &= \lambda x \\ \mathbf{X}\mathbf{B}_1 x &= 0 \\ \mathbf{X}\mathbf{B}_2 x &= 0 \\ \mathbf{C}_1 x &= 0 \end{aligned}$$

We use these preceding relations to see that

$$(\mathbf{A} - (\mathbf{B}_2\mathbf{B}_2^T - \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T)\mathbf{X})x = (\mathbf{A} - (\mathbf{B}_2\mathbf{B}_2^T\mathbf{X})x = \lambda x$$

which implies that $(\lambda, x)$ is also a mode of $\mathbf{A} - (\mathbf{B}_2\mathbf{B}_2^T - \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T)\mathbf{X}$. The issue we have is that we know this matrix is Hurwitz, so $(\lambda, x)$ cannot be an unstable and unobservable. So we can conclude that all modes of $\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{X}$ are stable and observable. In other words, the control is stabilizing.

Since $\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{X}$ is Hurwitz, we know from the KYP lemma that $\|\mathcal{F}_\ell(\mathbf{P}_{FI}, \mathbf{K})\|_{\mathcal{H}_\infty} < \gamma$ if and only if there exists an $\mathbf{P}$ that satisfies the algebraic Riccati equation

$$\mathbf{X}(\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{P}) + (\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{P})^T\mathbf{X} + \gamma^{-2}\mathbf{X}\mathbf{B}_1\mathbf{B}_2^T\mathbf{X} + \mathbf{P}\mathbf{B}_2\mathbf{B}_2^T + \mathbf{C}_1^T\mathbf{C}_1 = \mathbf{0}$$

with $(\mathbf{A} - \mathbf{B}_2\mathbf{B}_2^T\mathbf{P} - \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T\mathbf{X})$ being Hurwitz. Clearly $\mathbf{X} = \mathbf{P}$ provides such a solution and so we can conclude that $\|\mathcal{F}_\ell(\mathbf{P}_{FI}, \mathbf{K})\|_{\mathcal{H}_\infty} < \gamma$. $\diamondsuit$

**Remark:** The controller $\mathbf{K}$ is a full-state feedback controller and $\mathbf{F}_\infty = -\mathbf{B}_2^T\mathbf{X}$ may be seen as a set of feedback gains.

**Remark:** The need for the theorem's first assumption regarding stabilizability and detectability is clearly necessary since these assumptions ensure the unstable poles of the system can be arbitrarily placed.

**Remark:** The theorem's second assumption on the rank condition for $\mathbf{D}_{12}$ is needed to ensure that the objective signal $z$ has enough of the control input so that penalizing the control effort does not allow unbounded controls. This then becomes a necessary condition for internal stability.

**Remark:** The rank conditions in the theorem's third assumption are required to ensure that a stabilizing symmetric positive definite solution of the $\mathcal{H}_\infty$ algebraic Riccati equation exists.

## 7. Output Estimation $\mathcal{H}_\infty$ Problems

This section considers two special generalized regulator problems known as the *disturbance feedforward* (DF) and *output estimation* (OE) problems. The DF problem has a state space realization of the form

$$\mathbf{P}_{DF} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{I} & \mathbf{0} \end{array}\right] \tag{46}$$

This DF plant's output takes the form of $y = \mathbf{C}_2 x + w$. The exogenous disturbance, $w$, is passed directly through to the plant's measured output, hence the name *disturbance feedforward*. The OE problem has the state space realization

$$\mathbf{P}_{OE} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{I} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right] \tag{47}$$

This problem takes its name from the fact that it is used to synthesize a state observer. The following theorem asserts that the OE and DF problems are algebraic duals of each other.

THEOREM 11. *The controller $\mathbf{K}_{DF}$ internally stabilizes the DF plant in equation (46) if and only if $\mathbf{K}_{DF}^T$ internally stabilizes an associated OE plant in equation (47).*

**Proof:** Note that

$$[\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})]^T = \mathcal{F}_\ell(\mathbf{P}^T, \mathbf{K}^T)$$

This implies that $\mathbf{K}$ internally stabilizes $\mathbf{P}$ if and only if $\mathbf{K}^T$ internally stabilizes $\mathbf{P}^T$. So assume $\mathbf{K}_{DF}$ internally stabilizes the DF plant

$$\mathbf{P}_{DF} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{I} & \mathbf{0} \end{array}\right]$$

From the previous statement this means that $\mathbf{K}_{DF}^T$ internally stabilizes the dual system $\mathbf{P}_{DF}^T$

$$\mathbf{P}_{DF}^T \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A}^T & \mathbf{C}_1^T & \mathbf{C}_2^T \\ \hline \mathbf{B}_1^T & \mathbf{D}_{11}^T & \mathbf{I} \\ \mathbf{B}_2^T & \mathbf{D}_{12}^T & \mathbf{0} \end{array}\right]$$

which has the same form as the OE plant in equation (47). So $\mathbf{K}_{DF}^T$ internally stabilizes an associated OE problem. It is in this sense that we say the DF and OE are algebraic duals of each other. $\diamondsuit$

Up to this point we have introduced three special regulator problems; FI, DF, and OE. The preceding theorem established the DF and OE problems are algebraic duals of each other. The following theorem asserts that the FI and DF problems are *equivalent* in the sense that

(1) A controller used to internally stabilize one problem can be used to internally stabilize the other problem.
(2) and both LFT's are the same.

This equivalence is formally stated and proven below.

THEOREM 12. *Assume* $\mathbf{A} - \mathbf{B}_1\mathbf{C}_2$ *is Hurwitz (i.e. the system's equilibrium is asymptotically stable), then* **(1)** $\mathbf{K}_{DF}$ *internally stabilizes* $\mathbf{P}_{DF}$ *in equation (46) if and only if* $\mathbf{K}_{DF}\begin{bmatrix} \mathbf{C}_2 & \mathbf{I} \end{bmatrix}$ *internally stabilizes* $\mathbf{P}_{FI}$ *in equation (44).*

**(2)** *If* $\mathbf{K}_{FI}$ *internally stabilizes* $\mathbf{P}_{FI}$ *then* $\mathbf{K}_{DF} = \mathbf{F}_\ell(\hat{\mathbf{P}}_{DF}, \mathbf{K}_{FI})$ *internally stabilizes* $\mathbf{P}_{DF}$ *where*

$$\hat{\mathbf{P}}_{DF} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} - \mathbf{B}_1\mathbf{C}_2 & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \begin{bmatrix} \mathbf{I} \\ -\mathbf{C}_2 \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \end{array}\right]$$

*and*

$$\mathbf{F}_\ell(\mathbf{P}_{FI}, \mathbf{K}_{FI}) = \mathbf{F}_\ell(\mathbf{P}_{DF}, \mathcal{F}_\ell(\hat{\mathbf{P}}_{DF}))$$

**Proof:** Note that

$$\mathbf{P}_{DF} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{I} \end{bmatrix} \mathbf{P}_{FI}$$

This is obtained through a straightforward computation. We may use this to redraw the LFT for the DF problem as shown in Fig. 9. In this drawing we shift the matrix $\begin{bmatrix} \mathbf{C}_2 & \mathbf{I} \end{bmatrix}$ from the "plant" side to the "controller" side of the LFT.

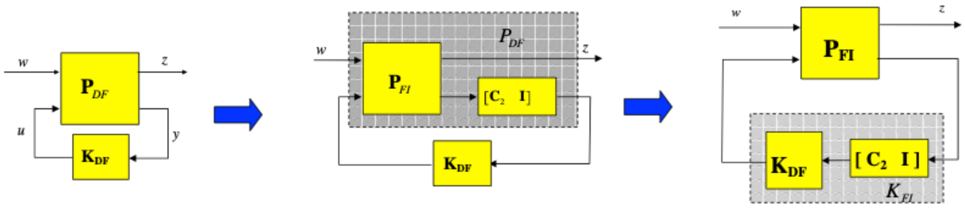


FIGURE 9. Transformation between DF and FI problems

From the far right hand side diagram in Fig. 9 we see that the controller in the LFT is now $\begin{bmatrix} \mathbf{C}_2 & \mathbf{I} \end{bmatrix} \mathbf{K}_{DF}$. This allows us to conclude that

$$\mathcal{F}_\ell(\mathbf{P}_{DF}, \mathbf{K}_{DF}) = \mathcal{F}_\ell(\mathbf{P}_{FI}, \begin{bmatrix} \mathbf{C}_2 & \mathbf{I} \end{bmatrix} \mathbf{K}_{DF})$$

which completes the proof for the theorem's first assertion.

We now turn to prove the second assertion. This proof is more involved but again involves redrawing the LFTs. Let us examine the star product of $\mathbf{P}_{DF}$ and $\hat{\mathbf{P}}_{DF}$. The star product is a common way of interconnecting to two-port plants to obtain another two-port plant as shown in Fig. 10. The star product is usually written as $\mathcal{C}_\ell(\mathbf{P}_{DF}, \hat{\mathbf{P}}_{DF})$.

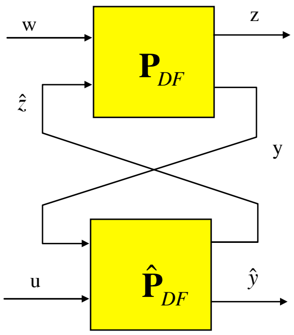


FIGURE 10. Star Product, $\mathcal{C}_\ell(\mathbf{P}_{DF}, \hat{\mathbf{P}}_{DF})$ of $\mathbf{P}_{DF}$ and $\hat{\mathbf{P}}_{DF}$

Let $x$ denote the state of $\mathbf{P}_{DF}$ and $\hat{x}$ denote the state of $\hat{\mathbf{P}}_{DF}$. The state equations for this star product may therefore be written as

$$\begin{aligned}
\frac{dx}{dt} &= \mathbf{A}x + \mathbf{B}_1 w + \mathbf{B}_2 \hat{z} \\
\frac{d\hat{x}}{dt} &= (\mathbf{A} - \mathbf{B}_1\mathbf{C}_2)\,\hat{x} + \mathbf{B}_1 y + \mathbf{B}_2 u \\
z &= \mathbf{C}_2 x + \mathbf{D}_{11} w + \mathbf{D}_{12}\hat{z} \\
\hat{z} &= u \\
y &= \mathbf{C}_2 x + w \\
\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} &= \begin{bmatrix} \hat{x} \\ -\mathbf{C}_2\hat{x} + y \end{bmatrix}
\end{aligned}$$

Let us define the error signal $e = x - \hat{x}$ and transform the above state equations from $(x, \hat{x})$ to $(x, e)$. The state equation for $e$ is

$$\begin{aligned}
\frac{de}{dt} &= \frac{dx}{dt} - \frac{d\hat{x}}{dt} \\
&= \mathbf{A}x + \mathbf{B}_1 w + \mathbf{B}_2 u - (\mathbf{A} - \mathbf{B}_1\mathbf{C}_2)\hat{x} - \mathbf{B}_1\mathbf{C}_2 x - \mathbf{B}_1 w - \mathbf{B}_2 u \\
&= (\mathbf{A} - \mathbf{B}_1\mathbf{C}_2)e
\end{aligned}$$

So the complete set of newly transformed state equations is

$$\begin{aligned}
\frac{dx}{dt} &= \mathbf{A}x + \mathbf{B}_1 w + \mathbf{B}_2 u \\
\frac{de}{dt} &= (\mathbf{A} - \mathbf{B}_1\mathbf{C}_2)e \\
z &= \mathbf{C}_1 x + \mathbf{D}_{11} w + \mathbf{D}_{12} u \\
\hat{y} &= \left[\begin{array}{c} \hat{x} \\ -\mathbf{C}_2\hat{x} + y \end{array}\right] = \left[\begin{array}{c} x - e \\ \mathbf{C}_2 e + w \end{array}\right]
\end{aligned}$$

By assumption, $\mathbf{A} - \mathbf{B}_1\mathbf{C}_2$ is Hurwitz so that $e \to 0$ as $t \to \infty$. We may therefore eliminate this state from the state space realization to obtain

$$\mathcal{C}_\ell(\mathbf{P}_{DF}, \hat{\mathbf{P}}_{DF}) \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{array}\right]$$

This is clearly the FI plant defined in equation (44). So if we now attach $\mathbf{K}_{FI}$ to the start product $\mathcal{C}_\ell(\mathbf{P}_{DF}, \hat{\mathbf{P}}_{DF})$ we obtain

$$\begin{aligned}
\mathcal{F}_\ell(\mathbf{P}_{FI}, \mathbf{K}_{FI}) &= \mathcal{F}_\ell\left(\mathcal{C}_\ell(\mathbf{P}_{DF}, \hat{\mathbf{P}}_{DF}), \mathbf{K}_{FI}\right) \\
&= \mathcal{F}_\ell\left(\mathbf{P}_{DF}, \mathcal{F}_\ell\left(\hat{\mathbf{P}}_{DF}, \mathbf{K}_{FI}\right)\right)
\end{aligned}$$

which completes the proof of the theorem's second assertion. $\diamondsuit$

We many now summarize the relationships between the three simplified problems (FI, DF, and OE). Since the DF and OE problems are dual, we can use a solution for the DF problem to obtain the OE controller. Moreover, because the FI and DF problems are equivalent, we can use a solution for the FI problem to obtain the DF controller. It should therefore be apparent that all three problems can be solved by determining the solution to the appropriate full-information (FI) problem. The next section shows that a simplified form of the full-information (FI) problem can be recast as a pair of FI and OE problems; namely a full-state feedback controller and an optimal $\mathcal{H}_\infty$ observer.

## 8. Simplified Output Feedback $\mathcal{H}_\infty$ Problem

A simplified version of the output feedback (OF) problem has the form

$$\mathbf{P}_{\text{OF}} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{0} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right]$$

where

$$\begin{aligned} \mathbf{C}_1^T\mathbf{D}_{12} = \mathbf{0}, \qquad \mathbf{B}_1\mathbf{D}_{21}^T = \mathbf{0} \\ \mathbf{D}_{21}\mathbf{D}_{21}^T = \mathbf{I}, \qquad \mathbf{D}_{12}^T\mathbf{D}_{12} = \mathbf{I} \end{aligned}$$

We seek an internally stabilizing control $\mathbf{K}$ that minimizes $\|\mathcal{F}_\ell(\mathbf{P}_{OF}, \mathbf{K})\|_{\mathcal{H}_\infty}$.

**Remark:** The simplified problem is posed this way because it is easier to establish the structure of the solution (state-feedback control and observer) with these assumptions in place. If these assumptions do not hold, it is always possible to transform the plant to the simplified form; though that transformation is a bit involved and beyond the scope of this class. See Zhou et al. (1996) for details on the relaxation of the simplified OF problem's assumptions. Note that we will later being using computational toolboxes in `MATLAB` to solve for the $\mathcal{H}_\infty$ controller. In these toolboxes, that transformation to the simplified form is done within the functions used to compute the controller.

The simplified $\mathcal{H}_\infty$ OF controller is obtained by decoupling the output feedback problem into a full information and output estimation problem. The resulting controller's state space realization is given in the following theorem.

THEOREM 13. *Consider the simplified output feedback problem with*

$$\mathbf{P}_{\text{OF}} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{0} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right]$$

*with*

$$\begin{aligned} \mathbf{C}_1^T\mathbf{D}_{12} = \mathbf{0}, \qquad \mathbf{B}_1\mathbf{D}_{21}^T = \mathbf{0} \\ \mathbf{D}_{21}\mathbf{D}_{21}^T = \mathbf{I}, \qquad \mathbf{D}_{12}^T\mathbf{D}_{12} = \mathbf{I} \end{aligned}$$

*and where $n$ is the state dimension, $m$ is the dimension of the measurement $y$, and $q$ is the dimension of the control input $u$. We further assume that*

*(1)* $(\mathbf{A}, \mathbf{B}_2)$ *is stabilizable and* $(\mathbf{C}_2, \mathbf{A})$ *is detectable.*

*(2) Matrix* $\mathbf{D}_{12}$ *has full column rank and* $\mathbf{D}_{21}$ *has full row rank.*

*(3) The following conditions hold for all real* $\omega$

$$\begin{aligned}
\operatorname{rank}\begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{12} \end{bmatrix} &= n + m \\
\operatorname{rank}\begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_1 \\ \mathbf{C}_2 & \mathbf{D}_{21} \end{bmatrix} &= n + q
\end{aligned}$$

*Then the simplified output feedback* $\mathcal{H}_\infty$ *problem's controller takes the form*

$$\mathbf{K}_{OF} \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A}_\infty & \mathbf{L}_\infty \\ \hline -\mathbf{F}_\infty & \mathbf{0} \end{array}\right]$$

*where*

$$\begin{aligned}
\mathbf{A}_\infty &= \mathbf{A} + \gamma^{-2}\mathbf{B}_1\mathbf{B}_2^T\mathbf{X} - \mathbf{Z}\mathbf{C}_2^T\mathbf{C}_2 - \mathbf{B}_2\mathbf{B}_2^T\mathbf{X} \\
\mathbf{L}_\infty &= -\mathbf{Z}\mathbf{C}_2^T \\
\mathbf{F}_\infty &= -\mathbf{B}_2^T\mathbf{X} \\
\mathbf{0} &= \mathbf{X}\mathbf{A} + \mathbf{A}^T\mathbf{X} - \mathbf{X}(\mathbf{B}_2\mathbf{B}_2^T - \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T)\mathbf{X} + \mathbf{C}_1^T\mathbf{C}_1 \\
\mathbf{0} &= \mathbf{Y}\mathbf{A}^T + \mathbf{A}\mathbf{Y} - \mathbf{Y}(\mathbf{C}_2^T\mathbf{C}_2 - \gamma^{-2}\mathbf{C}_1^T\mathbf{C}_2)\mathbf{Y} + \mathbf{B}_1\mathbf{B}_1^T \\
\mathbf{Z} &= \mathbf{Y}(\mathbf{I} - \gamma^{-2}\mathbf{X}\mathbf{Y})^{-1}
\end{aligned}$$

*when* $\rho(\mathbf{XY}) < 1$.

**Remark:** This theorem has the usual three conditions required for the existence of a stabilizing controller. The theorem also places additional restrictions on the augmented plant in terms of the orthogonality conditions,

$$\begin{aligned}
\mathbf{C}_1^T\mathbf{D}_{12} = \mathbf{0}, &\qquad \mathbf{B}_1\mathbf{D}_{21}^T = \mathbf{0} \\
\mathbf{D}_{21}\mathbf{D}_{21}^T = \mathbf{I}, &\qquad \mathbf{D}_{12}^T\mathbf{D}_{12} = \mathbf{I}
\end{aligned}$$

and that $\mathbf{D}_{11} = 0$ and $\mathbf{D}_{22} = 0$. These assumptions are made to simplify the proof. A key point that is not covered here is that any augmented plant can be transformed to this simplified form. See Zhou et al. (1996) for details on how this is done.

**Remark:** This controller still has the observer-based control structure of the $\mathcal{H}_2$ controller. This can be verified by rewriting the state equations as

$$\begin{aligned}\dot{\hat{x}} &= \mathbf{A}_\infty \hat{x} + \mathbf{L}_\infty y \\ &= \mathbf{A}\hat{x} + \mathbf{B}_1(\gamma^{-2}\mathbf{B}_1^T\mathbf{X}\hat{x}) + \mathbf{B}_2 u + \mathbf{L}_\infty(\mathbf{C}_2\hat{x} - y)\end{aligned}$$

with $u = \mathbf{F}_\infty \widehat{x}$. This has the general form of a full information feedback controller whose output is perturbed by the measurement output error $\mathbf{C}_2\hat{x} - y$ and another term $\gamma^{-2}\mathbf{B}_1\mathbf{X}\hat{x}$ which may be interpreted as a worst case disturbance for a related FI problem. Similarly $\mathbf{L}_\infty$ is the optimal filter gain for estimating the optimal FI control input $u$ in the presence of the worst case disturbance. We can think of this as an $\mathcal{H}_\infty$ filter.

**Proof:** In view of the preceding discussion, we make the following change of variables in the control loop.

$$\begin{aligned}\tilde{u} &= u + \mathbf{B}_2^T\mathbf{X}x \\ \tilde{w} &= w - \gamma^{-2}\mathbf{B}_1^T\mathbf{X}x\end{aligned}$$

These new variables, $\tilde{u}$ and $\tilde{w}$, represent the disturbances away from the optimal control $u^* = -\mathbf{B}_2^T\mathbf{X}x$ and the worst case disturbance $w^* = \gamma^{-2}\mathbf{B}_1^T\mathbf{X}x$

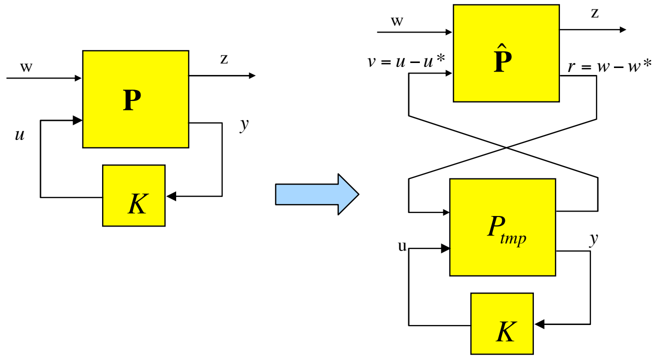


FIGURE 11. Loop transformation for reducing OF problem

We can then restructure the closed loop system using these new internal variables as shown in Fig. 11. This transformation shows that we can view the original plant $\mathbf{P}$ as the star product of $\hat{\mathbf{P}}$ and $\mathbf{P}_{\text{tmp}}$ where

$$\begin{aligned}
\hat{\mathbf{P}} &\overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A}+\mathbf{B}_2\mathbf{F}_\infty & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1+\mathbf{D}_{12}\mathbf{F}_\infty & \mathbf{0} & \mathbf{D}_{12} \\ -\gamma^{-2}\mathbf{B}_1^T\mathbf{X} & \mathbf{I} & \mathbf{0} \end{array}\right] \\
\mathbf{P}_{\text{tmp}} &\overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A}+\gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T\mathbf{X} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline -\mathbf{F}_\infty & \mathbf{0} & \mathbf{I} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array}\right]
\end{aligned}$$

From this figure it is apparent that

$$\mathcal{F}_\ell(\mathbf{P}_{OF},\mathbf{K}) = \mathbf{F}_\ell\left(\hat{\mathbf{P}},\mathbf{F}_\ell(\mathbf{P}_{\text{tmp}},\mathbf{K})\right)$$

and we see that the objective signal generated by $\hat{\mathbf{P}}$ is $z = (\mathbf{C}_1+\mathbf{D}_{12}\mathbf{F}_\infty)x+\mathbf{D}_{12}u$.

We can show that $\hat{\mathbf{P}}$ is inner which means that

$$\|\mathcal{F}_\ell(\mathbf{P},\mathbf{K})\|_{\mathcal{H}_\infty} < \gamma \Leftrightarrow \|\mathcal{F}_\ell(\mathbf{P}_{\text{tmp}},\mathbf{K})\|_{\mathcal{H}_\infty} < \gamma$$

Note, however, that $\mathbf{P}_{\text{tmp}}$ is an output estimation (OE) system. So we can immediately write down its controller $\mathbf{K}$.

$$\mathbf{K} \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A}_{\text{tmp}}+\mathbf{L}_\infty\mathbf{C}_2+\mathbf{B}_2\mathbf{F}_\infty & \mathbf{L}_\infty \\ \hline -\mathbf{F}_\infty & \mathbf{0} \end{array}\right]$$

where $\mathbf{A}_{\text{tmp}} = \mathbf{A}+\gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T\mathbf{X}$, $\mathbf{L}_\infty = -\mathbf{Z}\mathbf{C}_2^T$, and $\mathbf{Z}$ satisfies the algebraic Riccati equation

$$\mathbf{0} = \mathbf{A}_{\text{tmp}}\mathbf{Z}+\mathbf{Z}\mathbf{A}_{\text{tmp}}^T-\mathbf{Z}(\mathbf{C}_2^T\mathbf{C}_2-\gamma^{-2}\mathbf{F}_\infty^T\mathbf{F}_\infty)\mathbf{Z}+\mathbf{B}_1\mathbf{B}_1^T$$

This Riccati equation is cumbersome to work with because $\mathbf{F}_\infty = -\mathbf{B}_2^T\mathbf{X}$ is included in it and because $\mathbf{X}$ is a solution to the FI Riccati equation. In other words, unlike the case for our $\mathcal{H}_2$ problem, the two Riccati equations are coupled. We can still introduce a type of separation for the $\mathcal{H}_\infty$ OF controller by introducing the new matrix $\mathbf{Y}$ that satisfies

$$\mathbf{Y} = \mathbf{Z}(\mathbf{I}+\gamma^{-2}\mathbf{X}\mathbf{Z})^{-1}$$

It can be shown through a straightforward computation that $\mathbf{Y}$ satisfies the OE algebraic Riccati equation

$$\mathbf{0} = \mathbf{A}\mathbf{Y}+\mathbf{Y}\mathbf{A}^T+\mathbf{B}_1\mathbf{B}_1^T-\mathbf{Y}(\mathbf{C}_2^T\mathbf{C}_2-\gamma^{-2}\mathbf{C}_1^T\mathbf{C}_1)\mathbf{Y}$$

Note that if $\mathbf{Y}$ is invertible, then we can show that $\mathbf{Z}$ satisfies the Riccati equation for the OF problem.

There is, however, a small snag here because we are dealing with inverses. We must ensure that they actually exist. In particular, we need to recognize that only if $\rho(\mathbf{XY}) < 1$, then the matrix $\mathbf{I} - \gamma^{-2}\mathbf{XY}$ will be nonsingular. If this condition does not hold then we cannot define $\mathbf{Z}$ according to the above equation. This means that we indeed have a separation principle for the $\mathcal{H}_\infty$ controller, but only if we can ensure that $\rho(\mathbf{XY}) < 1$. This then completes the proof. $\diamondsuit$.

## 9. Computational Tools for $\mathcal{H}_\infty$ Controller Synthesis

There are a several methods that can be used to find an internally stabilizing controller that minimizes the $\mathcal{H}_\infty$ norm of a generalized regulator. The output feedback theorem 13 characterized the FI controller gains and the estimator gains in terms of solutions to two $\mathcal{H}_\infty$ algebraic Riccati equations. But this was for the simplified problem. So computing a solution to a real-life problem requires us first to 1) find the augmented plant, 2) transform the resulting regulator to its simplified form, 3) solve the Riccati equations to find the gains, 4) transform our solution back to the original problem (not the simplified problem), and 5) verify that the solution actually works well. This is usually too tedious to do by hand and so computational tools have been built to automate much of the computation needed in finding an $\mathcal{H}_\infty$ controller. This section reviews the functions in `MATLAB` (version 2018a) and uses them to solve mixed sensitivity problems and a well known structured uncertainty problem known as the ACC (American Control Conference) benchmark problem [Wie and Bernstein (1992)].

**9.1. Mixed Sensitivity Problem - simple example:** This subsection describes a simple mixed sensitivity problem that is used primarily to introduce the MATLAB functions used to determine the controller. Let us consider a mixed sensitivity problem associated with the feedback system shown in Fig. 5(a) where

$$\mathbf{G}_0(s) = \begin{bmatrix} \frac{4}{s+4} & 0 \\ \frac{4s(3s+16)}{(s+4)(s+8)} & \frac{8(s-200)}{s+8} \end{bmatrix}$$

The specifications for the mixed sensitivity problem are defined with respect the following weighting systems

$$\mathbf{W}_p(s) = \frac{s+100}{2s}\mathbf{I}, \quad \mathbf{W}_u(s) = 0.0001\mathbf{I}, \quad \mathbf{W}_\Delta(s) = \frac{10^{-3}s+1}{2(10^{-6}s+1)}\mathbf{I}$$

where $\mathbf{I}$ is a 2 by 2 identity matrix. The problem is to find an internally stabilizing controller $\mathbf{K}$ such that

$$\left\| \begin{bmatrix} \mathbf{W}_p\mathbf{S}_0 \\ \mathbf{W}_u\mathbf{K}\mathbf{S}_0 \\ \mathbf{W}_\Delta\mathbf{T}_0 \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq 2$$

since we know this will ensure the robust performance condition

$$\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} \leq 1$$

is satisfied for all uncertain plants that satisfy the unstructured multiplicative uncertainty model,

$$\mathbf{G} \in \{(\mathbf{I} + \mathbf{W}_\Delta(s)\boldsymbol{\Delta}(s))\mathbf{G}_0(s) \,:\, \|\boldsymbol{\Delta}\|_{\mathcal{H}_\infty} \leq 1\}$$

We will use the MATLAB functions provided in its Robust Control Toolbox to solve this problem. An early reference for these functions is provided in Balas et al. (2008). But MATLAB has updated the toolbox since the publication of Balas et al. (2008) and so I'll be using the toolbox functions as they appear in version 2018a of MATLAB. The documentation for these functions can be found through MATLAB's on-line reference.

MATLAB is useful because it provides a number of tools that automate the computationally tedious steps of the formulating the mixed sensitivity problem, as well as providing a function that transforms the generalized regulator to its simplified form and then carrying the computations outlined in theorem 13.

MATLAB creates data objects that represent the LTI dynamical systems. This makes it relatively easy to write the script declaring the systems in our example problem. For instance the following script will create the data objects representing the system $\mathbf{G}_0$, $\mathbf{W}_p$, $\mathbf{W}_u$ and $\mathbf{W}_\Delta$.

```
 s = zpk('s');
G0 = [4/(s+4) 0;...
    4*s*(3*s+16)/((s+4)*(s+8)) 8*(s-200)/(s+8)];
```

```
Wp = (s+100)/(2*s)*eye(2,2);
Wu = 1.e-10*eye(2,2);
Wd = (s*1e-3+1)/(2*(s*1e-6+1))*eye(2,2);
```

MATLAB’s robust control tool box creates the the augmented plant, $\mathbf{P}$, with the following function call

```
P = augw(G0,Wp,Wu,Wd);
```

The computation of the controller and the associated sensitivity function is done by the following

```
gamlower = 2; gamupper = 1.e10;
[K,CL,GAM,info] = hinfsyn(P,...
         ’GMIN’,gamlower,’GMAX’,gamupper,...
         ’DISPLAY’,’on’,’METHOD’,’ric’);
```

The main function is `hinfsyn` which computes $\mathcal{H}_\infty$ controller, `K`, for the augmented plant, `P` with the minimal cost of the controller being `GAM` and where `CL` is closed-loop map $\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})$.

The computation that `hinfsyn` is a bit different than what is outlined in theorem 13. This theorem only provides a sufficient condition for a controller to meet the requirement $\|\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})\|_{\mathcal{H}_\infty} < \gamma$. In this regard, the theorem describes an *algorithmic oracle* which asserts that *there is a controller* $\mathbf{K}$ *which meets the* $\gamma$ *constraint* and tells us one such controller that meets the condition. The parameter $\gamma$ is something we have to choose in theorem 13.

The MATLAB function `hinfsyn` embeds this oracle test in theorem 13 in a bisection search for the smallest $\gamma$ for which an internally stabilizing controller exists. It does this by first specifying an upper bound, `GMAX`, and a lower bound, `GMIN` over which we want to search for the minimum $\gamma$. In our case, we want that minimum value to be 2. So in the function call above, we specify the lower bound as 2 and the upper bound as something big ($10^{10}$). The next option given in the `hinfsyn` function call is `DISPLAY` which we set to `’on’` so the function tells us what it is doing. The last option in the command is `METHOD` which specifies the method used to find the controller. Since theorem 13 characterizes the control in terms of a solution to two Riccati equations, this option flag is therefore set to `'ric'`.

**Remark:** The Riccati method for synthesizing $\mathcal{H}_\infty$ controller was used because it more clearly shows the relationship between $\mathcal{H}_\infty$ control and the LQG controller. Another well known method for finding the controller is based on the use of *linear matrix inequalities* (LMI) Gahinet and Apkarian (1994). An LMI is an affine matrix-valued function of the form

$$\mathbf{F}(x) = \mathbf{F}_0 + \sum_{i=1}^{m} x_i \mathbf{F}_i > 0$$

where $x \in \mathbb{R}^m$ is a decision vector and $\mathbf{F}_i = \mathbf{F}_i^T \in \mathbb{R}^{n \times n}$ are symmetric matrices for $i = 0, 1, 2 \ldots, n$. An important aspect of the LMI is that it defines a *convex set* so that the problem of finding a vector $x$ given the matrix $\{\mathbf{F}_i\}_{i=0}^n$ such that $\mathbf{F}(x) > 0$ is a convex optimization problem. Since we now have polynomial time algorithms for solving such optimization problems, if one can reformulate your problem into an LMI, then we can verify if the inequality is satisfied even when the problem is very large. The basis for using LMI's to solve the $\mathcal{H}_\infty$ control problem rests with the *Schur complement* theorem which establishes the equivalence between a quadratic form consisting of matrices and an LMI. One can select the solution method in `hinfsyn` by setting the keyword `METHOD` to `'lmi'`. In reality, the computational cost of the LMI method may not be that different from solving a Riccati equation using the invariant embedding method. One potential benefit of using LMI methods is that they allow one to specify constraints on the poles of the controller which can help constrain the transient response of the system.

For the output generated by the preceding commands is

```
>> mixed_problem

  [a b1;c2 d21] does not have full row rank at s=0
```

which indicates that we could not find a controller because one of the assumptions in theorem 13 is not satisfied. It is easy to verify that the stabilizability/detectability condition on the augmented plant is satisfied. We can also show that the second condition is satisfied on the control weight $\mathbf{W}_u$. The third condition, however, requires that the augmented plant have no poles on the $j\omega$-axis and this is precisely what the error message returned by the function tells us. Re-inspecting the problem, we see that this condition is violated by the performance weight

$$\mathbf{W}_p(s) = \frac{s+100}{2s}\mathbf{I}$$

which clearly has a pole at the origin. We propose fixing this problem by modifying the performance weight to

$$\mathbf{W}_p(s) = \frac{s+100}{2s+\epsilon}\mathbf{I}$$

where $\epsilon$ is something small, i.e. $\epsilon = 0.0001$. This modification moves the poles of $\mathbf{W}_p(s)$ off of the imaginary axis so the conditions allowing the Riccati equations to have a stabilizing solution will be satisfied. If we now re-run the script after redeclaring

```
Wp = (s+100)/(2*s+0.0001)*eye(2,2);
```

then we get the following output

```
>> mixed_problem_updated
Test bounds:      2.0000 <  gamma  <=  67108864.0000

  gamma    hamx_eig  xinf_eig  hamy_eig   yinf_eig   nrho_xy   p/f
6.711e+07   3.8e+00 -1.6e-03#  5.0e-05   -1.6e-28    0.0000    f
Gamma max, 67108864.0000, is too small !!
Test bounds:      2.0000 <  gamma  <=      2.0067

  gamma    hamx_eig  xinf_eig  hamy_eig   yinf_eig   nrho_xy   p/f
    2.007   4.0e+00 -5.7e-08   1.1e-05    0.0e+00    0.0000    p
    2.003   4.0e+00 -5.3e-08   1.1e-05    0.0e+00    0.0000    p

 Gamma value achieved:     2.0033
```

The first 5 lines of the output says that the function readjusted the upper bound for the $\gamma$ interval from $[2, 10^{10}]$ to $[2, 2.0067]$. The next few lines of output describe the steps taken by the bisection search, eventually finding that value for $\gamma$ of 2.0033 that satisfies the termination conditions of the bisection search. The controller computed by the function is in the data object `K` and the closed-loop map is in `CL`.

Let us now see how well the controller generated by the mixed sensitivity approach works. The easiest way of evaluating the controller's performance is to check the step response of the nominal closed-loop system $\mathcal{F}_\ell(\mathbf{P}, \mathbf{K})$ since these represent the weighted tracking errors.

```
 t = 0:.001:0.08;
 [y,t] = step(CL,t);

 figure(10);
 for i=1:2;
     for j=1:2;
        subplot(2,2,2*(i-1)+j)
plot(t,y(:,i,j),'linewidth',2);
    end;
end;
```

The resulting step response is shown in Fig. 12(a) which shows what we would expect for this system.

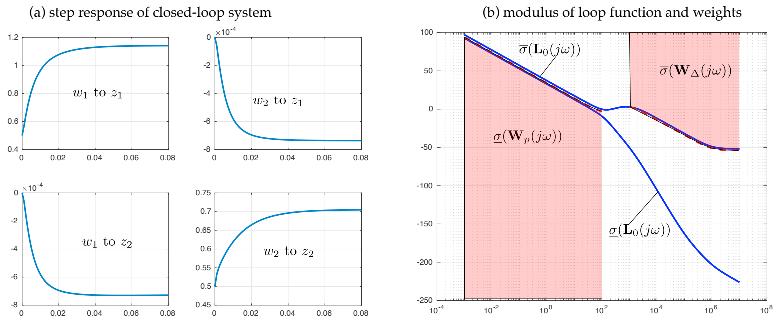


FIGURE 12. (a) step responses for solution to first mixed sensitivity problem (b) loop function and weights used to check robust performance condition

The step responses, however, do not say much about how robust this system is to modeling uncertainty. In particular, we recall that we can use bounds on the singular values of the nominal loop function $\mathbf{L}_0(s) = \mathbf{G}_0(s)\mathbf{K}(s)$ to determine how well we meet the robust performance requirement that $\|\mathbf{W}_p\mathbf{S}\|_{\mathcal{H}_\infty} \leq 1$. In particular, we know that if

$$\begin{aligned} &\underline{\sigma}(\mathbf{L}_0(j\omega)) > \overline{\sigma}(\mathbf{W}_p(j\omega)) && \text{for } \omega \text{ where } \underline{\sigma}(\mathbf{L}_0(j\omega)) > 1 \\ &\overline{\sigma}(\mathbf{L}_0(j\omega)) < \underline{\sigma}(\mathbf{W}_\Delta^{-1}(j\omega)) && \text{for } \omega \text{ where } \overline{\sigma}(\mathbf{L}_0(j\omega)) < 1 \end{aligned}$$

The first condition is the low-frequency tracking requirement and the second condition represents the robust stability requirement. These conditions represent natural extensions of the loopshaping concepts we derived earlier for uncertain scalar systems. The difference is that now the loop gains and gains of the weighting systems are characterized in terms of their minimum and maximum singular values since they are MIMO systems. Fig. 12(b) shows the gain magnitude of the nominal loop's singular values (solid blue lines) against the singular values of the weights $\mathbf{W}_p$ and $\mathbf{W}_\Delta^{-1}$. This plot clearly shows that the loop function constraints ensuring robust performance are satisfied, so indeed the controller should solve this particular robust performance problem.

Note that the low frequency region is given by $\omega$ less than $100$ rad/sec and the high frequency region is given by $\omega$ greater than $2000$ rad/sec. As can be seen the low frequency performance constraint is satisfied exactly, but the high frequency robust stability constraint is satisfied with some margin to spare. In other words over the high frequency region $\overline{\sigma}(\mathbf{L}_0(j\omega))$ is about 20 dB below the $\underline{\sigma}(\mathbf{W}_\Delta^{-1}(j\omega))$. This suggests that our system has robust stability, but it also suggests that we may be able to improve overall system performance on the low frequency end without compromising the robust stability condition.

**9.2. HIMAT Design Problem:** This subsection presents another example in which the $\mathcal{H}_\infty$ mixed sensitivity problem is used to enforce robust performance. The control problem posed here is drawn from a real-life system known as the HIMAT vehicle. This vehicle was a scaled remotely piloted vehicle (RPV) of an advanced fighter that was flight tested in the late 1970's. The actual HIMAT vehicle is currently on display in the Smithsonian National Aerospace Museum in Washington D.C.. A picture of that exhibit is shown in Fig. 13.

This problem considers the longitudinal dynamics of the airplane. These dynamics are assumed to be uncoupled from the lateral-directional dynamics. The state vector consists of the vehicle's rigid body variables:

(1) $\delta v$ - perturbations along the velocity vector,
(2) $\alpha$ - angle of attack,
(3) $q$ - pitch rate,

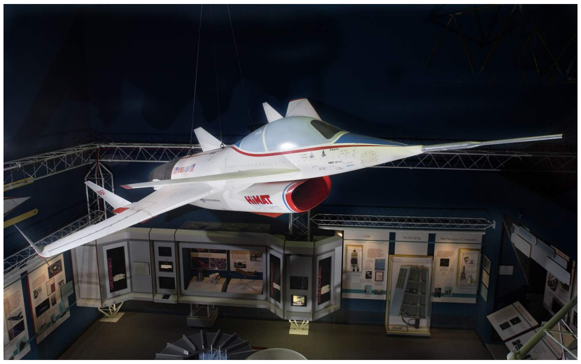


FIGURE 13. HIMAT vehicle at Smithsonian National Aerospace Museum

(4) $\theta$ - pitch angle

The control inputs are the elevon ($\delta_e$) and the canard $\delta_c$ angles. The variables to be measured are the angle of attack $\alpha$ and the pitch angle $\theta$. A linearized state-space realization (based on MATLAB example) for the plane's longitudinal dynamics is

$$\mathbf{G}_0 \overset{s}{=} \left[\begin{array}{cccccc|cc} -.027 & -36.6 & -18.9 & -32 & 3.25 & -.76 & 0 & 0 \\ 0 & -1.9 & .98 & 0 & -.17 & 0 & 0 & 0 \\ .012 & 11.7 & -2.6 & 0 & -31.6 & 22.4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -30 & 0 & 30 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30 & 0 & 30 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

An unstructured multiplicative uncertainty model for this aircraft is

$$\mathbf{G} = \mathbf{G}_0(\mathbf{I} + \mathbf{W}_{\boldsymbol{\Delta}}\boldsymbol{\Delta})$$

where the uncertainty's weighting system is

$$\mathbf{W}_\Delta(s) = \begin{bmatrix} \frac{20(s+25)}{s+1000} & 0 \\ 0 & \frac{20(s+25)}{s+1000} \end{bmatrix}$$

Such an uncertainty bound is assumed to have been obtained from exhaustive empirical testing of the aircraft.

The performance of the closed loop system will be evaluated by the output sensitivity function. In particular, we assume the following performance specification

$$\overline{\sigma}(\mathbf{S}(j\omega))\mathbf{W}_p(j\omega) \leq 1$$

for all $\omega$ and the performance weighting function is

$$\mathbf{W}_p(s) = \begin{bmatrix} \frac{0.5(s+\beta)}{s+1.5} & 0 \\ 0 & \frac{0.5(s+\beta)}{s+1.5} \end{bmatrix}$$

where $\beta$ is a constant that we'll specify in a second. Such a performance weight indicates that at low frequencies (frequencies below $\beta$) the closed loop system should reject disturbances at the output by a factor of $20\beta/3$. At high frequencies, the performance gets less stringent. The parameter $\beta$, therefore, characterizes some important qualitative properties of the closed loop system. The larger $\beta$, the better this system's performance will be. How large can we make $\beta$ and still preserve robust performance? To answer this question, we begin by writing a `MATLAB` script to declare the dynamical systems for the plant and weighting systems.

```
A = [
-2.27e-02  -3.66e+01  -1.89e+01  -3.20e+01   3.25e+00  -7.6e-01;
0  -1.9e+00   9.8e-01  0  -1.7e-01  0;
1.2e-02   1.17e+01  -2.6e+00   0  -3.16e+01   2.24e+01;
0            0   1.0000e+00            0            0            0;
0            0            0            0  -3.0000e+01            0;
0            0            0            0            0  -3.0000e+01];
B = [0     0;      0     0;
      0     0;      0     0;
     30     0;   0    30];
C = [0     1     0     0     0     0;
      0     0     0     1     0     0];
D = [0     0;
      0     0];
G0 = ss(A,B,C,D);

s     = zpk(’s’); % Laplace variable s
beta =10;
Wp = 0.5*(s+beta)/(s+1.5);
Wu = 0.0001*eye(2,2);
Wd = 20*(s+25)/(s+1000);
```

We then use `augw` and `hinfsyn` to determine the controller using the following commands

```
P = augw(G0,Wp,Wu,Wd);
gamlower = 0;gamupper = 1.e10;
[K1,CL1,GAM1,info] = hinfsyn(P,...
      GMIN’,gamlower,’GMAX’,gamupper,...
       ’DISPLAY’,’on’,’METHOD’,’ric’);

if (GAM1>2)
    tstring = sprintf(’GAM1= %f2.1, NO robust performance\n’,GAM1);
    disp(tstring);
end;
```

We’ve set the upper and lower limits on the $\gamma$-interval to be $0$ and $10^{10}$ respectively. In calling this, the function $\mathrm{hinfsyn}$ will return with the smallest value of $gamma$ for which we can find a stabilizing controller. If $\gamma \leq 2$, then we should satisfy the robust performance constraints, otherwise the system is not robustly stable. The output generated by the above script is as follows

```
>> himat_problem
Resetting value of Gamma min based on D_11, D_12, D_21 terms

Test bounds:      0.5000 <  gamma  <=      1.1304

  gamma    hamx_eig  xinf_eig  hamy_eig   yinf_eig   nrho_xy   p/f
    1.130   2.2e-02   1.1e-10   2.5e-01   -5.9e-17    0.0288    p
    0.815   2.2e-02   1.1e-10   2.5e-01   -2.1e-17    0.1055    p
    0.658   2.2e-02   1.1e-10   2.5e-01   -7.1e-19    0.5366    p
    0.579   2.2e-02 -1.8e+05#  2.5e-01   -1.7e-16   37.4705#   f
    0.618   2.2e-02   1.1e-10   2.5e-01   -3.2e-17    1.3724#   f
    0.638   2.2e-02   1.1e-10   2.5e-01   -2.0e-16    0.8009    p
    0.628   2.2e-02   1.1e-10   2.5e-01   -1.5e-18    1.0243#   f

 Gamma value achieved:     0.6379
```

which shows that the minimum $\gamma$ for which we were able to obtain a stabilizing controller when $\beta = 10$ is $0.6379$. Since this is less than $2$ the system has robust performance.

We check this design by plotting the step response and plotting the gain magnitude of the loop function using the script

```
L1 = G0*K1;
I = eye(size(L1));
```

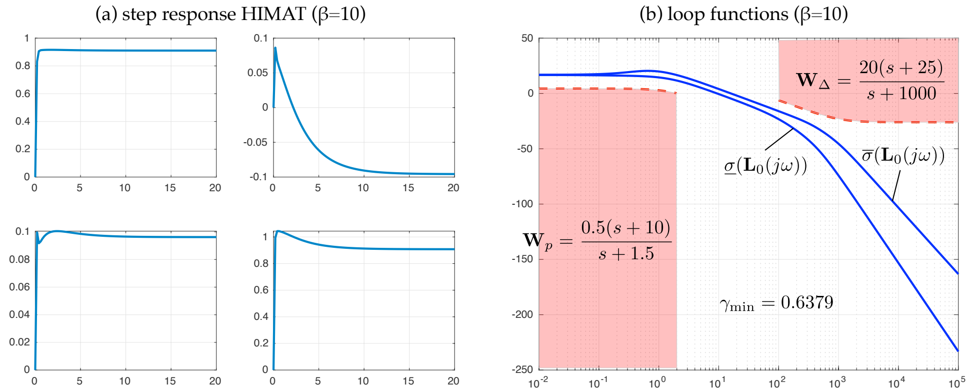
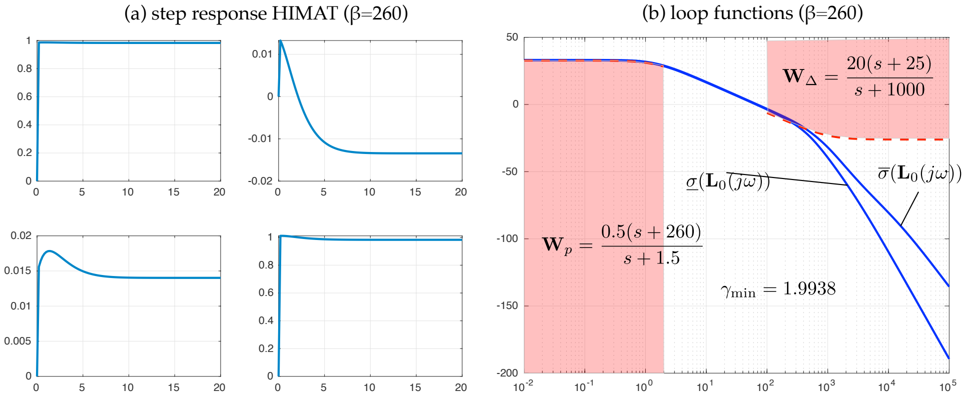


FIGURE 14. (a) step response of HIMAT system when $\beta = 10$ (b) loop function of HIMAT when $\beta = 10$

```
S1 = feedback(I,L1); % S=inv(I+L1);
T1 = I-S1;
figure;
[y,t] = step(T1,t);
     for i=1:2;
         for j=1:2;
             subplot(2,2,2*(i-1)+j)
             plot(t,y(:,i,j),’linewidth’,2);
             grid on;
         end;
     end;
figure;
w = logspace(-2,5,1000);
[svL,w1]= sigma(L1,w);
[svWp,w2]=sigma(Wp/2,w(w<2));
[svWdi,w3] = sigma(1/Wd,w(w>100));
 semilogx(w1,20*log10(svL),’b’,’linewidth’,2);
 hold on;
 semilogx(w2,20*log10(svWp),’r--’,’linewidth’,2);
 semilogx(w3,20*log10(svWdi),’r--’,’linewidth’,2);
 grid on;
```

The step response is shown in Fig. 14 where the response $y_2$ due to $w_2$ has a settling time of about 5 seconds. Because $\gamma$ is much less than 2, we expect the robust performance constraint to be satisfied with some margin to spare. The singular values of the loop function, $\mathbf{G}_0\mathbf{K}$, the weights $\mathbf{W}_p/2$ and $\mathbf{W}_\Delta^{-1}$ are sown in Fig. 14(b). This plot clearly shows the low and high frequency constraints are satisfied with some margin to spare, thereby implying robust performance.

Note that a step response time of about 5 seconds is a bit slow for a fighter. So we try improving this by increasiing $\beta$, thereby increasing the low frequency gain. If we rerun with $\beta = 260$ we get a much faster system. Fig. 15 shows the step response, which if magnified indicates a rise time of 5 msec. We also see from the singular value plots of the loop function and its weights in Fig. 15, that the bounds are much more closely adhered to than they were when $\beta$ was only 10. This suggests we are at the very limits of the breaking the robust performance constraints.


(a) step response HIMAT (β=260)

(b) loop functions (β=260)

FIGURE 15. (a) step response of HIMAT system when $\beta = 260$ (b) loop function of HIMAT when $\beta = 260$

**9.3. ACC Benchmark Problem:** This subsection presents an example of $\mathcal{H}_\infty$ controller synthesis for a system with *structured* parametric uncertainties as described in section 5. This example consider the two mass spring system shown in Fig. 16. This system may be seen as a generic model of an uncertain dynamical system with rigid body and one vibrational mode. It is assumed that the nominal system has $m_1 = m_2 = 1$ with a spring constant $k = 1$. A control force and disturbance act on body 1 and the position of body 2 is measured, thereby resulting in a non-collocated control problem. This system can be represented in state space form as

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k/m_1 & k/m_1 & 0 & 0 \\ k/m_2 & -k/m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} (u+w) \\
y &= x_2 \\
z &= x_2
\end{aligned}$$

where $x_1$ and $x_2$ are the positions of body 1 and 2, respectively; $x_3$ and $x_4$ are the velocities of body 1 and 2 respectively; $u$ is the control input acting on body 1; $y = x_2$ is the measured output available to the controller; $w$ is the plant disturbance acting on body 1; and $z = y$ is our objective signal - namely the output to be controlled. This particular system is a benchmark problem that was presented at the American Control Conference (ACC) several years ago; so we refer to it as the ACC benchmark problem [Wie and Bernstein (1992)].

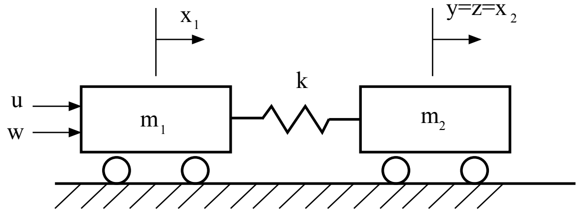


FIGURE 16. ACC Benchmark Problem

The uncertainty in this system arises from the fact that the spring constant is uncertain. In particular, we assume $k \in [0.5, 1.5]$. If we then let $\mathbf{T}(s)$ denote the closed-loop transfer function from $w$ to $z$ for this "uncertain" system, then the objective is to find the feedback controller $\mathbf{K}$ such that $\|\mathbf{T}\|_{\mathcal{H}_\infty} < \gamma$ with $\gamma$ being as small as possible. What is different about this problem from the mixed sensitivity problem in the preceding examples is that the uncertainty is *state-based* and so we follow the approach laid out in section 5 to formulate the generalized regulator problem we need to solve.

Following section 5, we start by identifying a nominal open-loop plant's state space realization

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \left[\begin{array}{c|cc} \mathbf{A}_0 & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array}\right] \begin{bmatrix} x \\ w \\ u \end{bmatrix}$$

where, for this system, the matrices are

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}, \qquad \mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \left[\begin{array}{c|c} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \hline \mathbf{D}_{21} & \mathbf{D}_{22} \end{array}\right] = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array}\right]$$

The parametric uncertainty is only on the spring constant and it only impacts the system matrix. So we perturb the system matrix as

$$\begin{aligned} \mathbf{A} &= \mathbf{A}_0 - \mathbf{M}\Delta\mathbf{N}^T \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -0.5 \\ 0.5 \end{bmatrix} \Delta \begin{bmatrix} -1 & 1 & 0 & 0 \end{bmatrix} \end{aligned}$$

where $|\mathbf{\Delta}| \leq 1$. We can then rewrite the uncertain open loop plant, $\mathbf{P}$, as an upper LFT,

$$\mathbf{P} = \mathcal{F}_u(\mathbf{P}_0, \mathbf{\Delta})$$

formed from a nominal system, $\mathbf{P}_0$ with state equations

$$\begin{bmatrix} \dot{x} \\ \tilde{z} \\ z \\ y \end{bmatrix} = \left[\begin{array}{c|ccc} \mathbf{A}_0 & \mathbf{M} & \mathbf{B} & \mathbf{B} \\ \hline \mathbf{N}^T & 0 & 0 & 0 \\ \mathbf{C} & 0 & 0 & 0 \\ \mathbf{C} & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ \tilde{w} \\ w \\ u \end{bmatrix}$$

connected with the feedback system

$$\tilde{w} = \Delta\tilde{z}$$

Section 5 showed that the $\mathcal{H}_\infty$ norm of the lower LFT, $\mathcal{F}_\ell(\mathbf{P},\mathbf{K})$ formed from the uncertain plant $\mathbf{P}$ and the controller $\mathbf{K}$ is equal to the $\mathcal{H}_\infty$ norm of $\mathbf{T}_0 = \mathbf{F}_\ell(\mathbf{P}_0,\mathbf{K})$. So this last generalized regulator is what we want to use in finding $\mathbf{K}$.

The augmented plant, $\mathbf{P}_0$, in the ACC benchmark problem has the inputs $\begin{bmatrix}\tilde{w}\\ w\end{bmatrix}$ and $u$ with outputs $\begin{bmatrix}\tilde{z}\\ z\end{bmatrix}$ and $y$. So the state space realization for this plant is

$$\mathbf{P} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A}_0 & \begin{bmatrix}\mathbf{M} & \mathbf{B}\end{bmatrix} & \mathbf{B} \\ \hline \begin{bmatrix}\mathbf{N}^T\\ \mathbf{C}\end{bmatrix} & \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} & \begin{bmatrix}0\\ \epsilon\end{bmatrix} \\ \mathbf{C} & \begin{bmatrix}0 & \epsilon\end{bmatrix} & 0 \end{array}\right]$$

where $\epsilon$ is a small variable $(0.001)$ used to ensure there is a penalty on the control effort. The following script can now be used to compute the $\mathcal{H}_\infty$ controller

```
A0 = [ 0 0 1 0; 0 0 0 1; -1 1 0 0; 1 -1 0 0];
M  = [ 0; 0; .5; -.5];
N   = [ -1; 1; 0 0];
B   = [0 ; 0 ; 1 ; 0 ];
C   = [0 1 0 0];
D   = [0 0; 0 0];

eps = 1.e-3;
A      = A0;
B1    = [M B];
B2    = B;
C1    = [N’ ; C];
C2    = C;
D11 = [0 0 ; 0 0];
D12 = [ 0 ; eps];
D21 = [0 eps];
D22  = 0;

P = ss(A,[B1 B2],[C1;C2],[D11 D12;D21 D22]);

gamlower = 0; gamupper = 1.e10;
nmeas=1;ncont=1;
[Kss,Tss,GAM,info] = hinfsyn(P,nmeas,ncont,...
                  ’GMIN’,gamlower,’GMAX’,gamupper,...
                  ’DISPLAY’,’on’,’METHOD’,’ric’);
```

The output from this script for $k = 1$ is a controller whose state space realization is

$$\mathbf{K} \overset{s}{=} \left[\begin{array}{rrrr|r} 6.885 & -4361 & .997 & -1.062 & 4351 \\ .01207 & -40.02 & 0 & 0.9902 & 39.97 \\ -9035 & -32600 & -223.9 & -79870 & 2176 \\ 1.413 & -773.3 & -.0005 & -0.1883 & 771.2 \\ \hline -9045 & -30440 & -224.1 & -79930 & 0 \end{array}\right]$$

The impulse response for the controlled system was computed with $\Delta = .5$, $1$, and $1.5$. The results shown in Fig. 17 show that indeed this controller is indeed robustly internally stabilizing the entire range of uncertain plants.

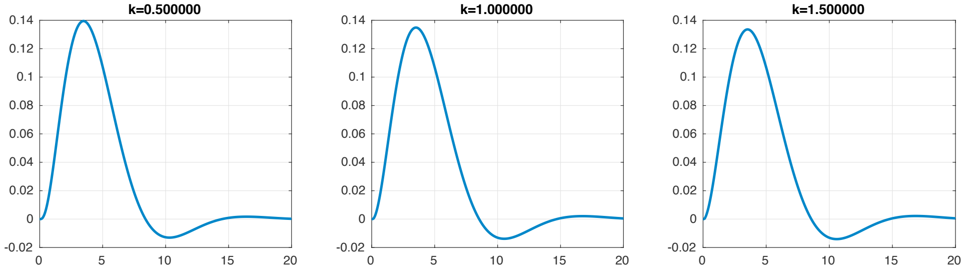


FIGURE 17. Impulse Response for $\mathcal{H}_\infty$ controlled ACC Benchmark Problem

It is worthwhile to at least compare the difference of this to simply using an LQG design for the nominal system and then seeing how it behaves when $k$ is perturbed. The controller seeks to minimize

$$J[u] = \int_0^\infty \left\{ x^T \mathbf{Q} x + u^T \mathbf{R} u \right\} dt$$

where $\mathbf{Q} = \mathbf{I}_{4\times 4}$ and $\mathbf{R} = 1$. assuming a process noise covariance, $\mathbf{W} = \mathbf{I}_{4\times 4}$ and $\mathbf{V} = 1$. The following script was used to compute the impulse response for the LQG controlled system.

```
Q = eye(4,4); R=1;
W= eye(4,4); V=1;
A0 = [0 0 1 0; 0 0 0 1; -1 1 0 0; 1 -1 0 0];
B = [0; 0; 1 ; 0];
C = [0 1 0 0];
```

```
 D = 0;

 QWV = [W  zeros(size(B)); zeros(size(C)) V];
 QXU = [Q  zeros(size(B)); zeros(size(C)) R];
 Klqg = lqg(ss(A0,B,C,D),QXU,QWV);
for del=-1:1:1;

   A   = A0 + M*del*N';
   Gss = ss(A,B,C,0);
    Tlqg = feedback(Gss,-Klqg);
    pole(Tlqg)


   figure(11);
   subplot(1,3,del+2);
   t = 0:.01:20;
   [y,t]=impulse(Tlqg,t)
   plot(t,y,'linewidth',2);

   tstring = sprintf('k=%f', 1+.5*del);
   title(tstring);
   grid on;
end
```

The output of this script is shown in Fig. 18. Note that for $k = 0.5$, the system is unstable and for $k = 1.5$ the system is highly oscillatory. In other words, the LQG controller is definitely not robust to the parametric variations in the spring constant, whereas the $\mathcal{H}_\infty$ controller certainly is.

It may be possible to adjust the matrices $Q$, $R$, $W$, and $V$ in such a way to make the response when $k = 1$ approach that of the $\mathcal{H}_\infty$ controlled system in Fig. 17, but that adjustment is not obvious. This is in fact one of the chief issues one might have with regard to the LQG controller formulation is the "arbitrary" nature of selecting weighting matrices. In the context of the $\mathcal{H}_\infty$ formulation, the basis for selecting the weighting systems is much clearer.

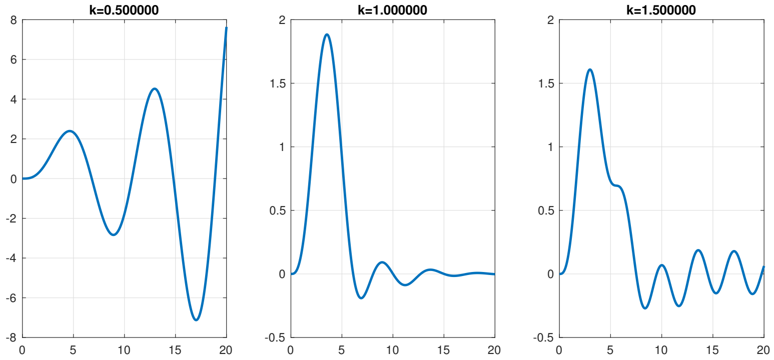


FIGURE 18. Impulse Response for LQG Controlled ACC Benchmark Problem

## 10. Summary

This chapter discussed methods used to design robust optimal controllers for continuous time LTI systems. We posed the design problem in terms of minimizing the $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm of the closed-loop transfer function for a *generalized regulator*. The generalized regulator is a ”canonical” form of a feedback control system that can be used to represent a wide range of actual control systems. We saw that the $\mathcal{H}_2$ optimal controller was essentially the LQG controller consisting of an LQR gain matrix with a steady-state Kalman filter. We demonstrated that this ”optimal” control law is not robust to model uncertainty [Doyle (1978)] and so formulated ways of characterizing model uncertainty in a way that can then be used in designing optimal controllers whose performance and stability are ”robust” to bounded model uncertainty. We showed that this robust optimal control problem could be posed in terms of minimizing the $\mathcal{H}_\infty$ norm of a suitably chosen generalized regulator. We presented computational tools from MATLAB that can be used to synthesize the controller and provided several example illustrating its use. Our discussion of the LQG and $\mathcal{H}_2$ controllers was strongly influenced by [Dorato et al. (1994)] and [Green and Limebeer (2012)]. The formulation of the generalized regulator will be found in Green and Limebeer (2012), Zhou et al. (1996), or Sanchez-Pena and Sznaier (1998). Our development of the $\mathcal{H}_\infty$ controller using the FI, OE, and OF subproblems follows the development in Sanchez-Pena and Sznaier (1998).

## 11. Appendix: KYP Lemma

The bounded real and Kalman-Yakubovich-Popov (KYP) lemmas provide necessary and sufficient conditions $\|\mathbf{G}\|_{\mathcal{H}_\infty} < \ gamma$. In the bounded real lemma, this test is a condition on the eigenvalues of a specially formed matrix. This particular test can then be used as an algorithmic oracle who declares whether or not the norm of $\mathbf{G}$ is less than a specified $\gamma$. This oracle is the basis for an efficient algorithmic approach for estimating the $\mathcal{H}_\infty$ norm of a rational transfer function matrix, $\mathbf{G}(s)$, through a bisection search strategy.

The KYP lemma recasts the bounded real condition in terms of an algebraic Riccati equation (ARE). This condition is more useful in synthesizing $\mathcal{H}_\infty$ controllers for the generalized regulator problem. Because the KYP lemma develops an ARE characterization of the $\mathcal{H}_\infty$ controller, it provides a basis for more clearly seeing the similarities and differences between $\mathcal{H}_2$ (LQG) controllers and the more robust $\mathcal{H}_\infty$ controllers.

THEOREM 14. **[Bounded Real Lemma]** *Let* $\gamma > 0$ *and assume* $\mathbf{G} \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]$. *Assume* $\mathbf{A}$ *has no eigenvalues on the* $j\omega$ *axis. Then* $\|\mathbf{G}\|_{\mathcal{H}_\infty} < \gamma$ *if and only if* $\overline{\sigma}(\mathbf{D}) < \gamma$ *and the matrix*

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ -\mathbf{C}^T(\mathbf{I} + \mathbf{D}\mathbf{R}^{-1}\mathbf{D}^T)\mathbf{C} & -(\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C})^T \end{bmatrix} \tag{48}$$

*where* $\mathbf{R} = \gamma^2 I - \mathbf{D}^T\mathbf{D}$ *has no eigenvalues on the* $j\omega$*-axis.*

**Proof:** note that

$$\|\mathbf{D}\| = \overline{\sigma}(\mathbf{G}(j\infty)) \leq \sup_\omega(\overline{\sigma}(\mathbf{G}(j\omega))) < \gamma$$

This implies that if $\|\mathbf{D}\| = \overline{\sigma}(\mathbf{D}) > \gamma$, then it is impossible for $\|\mathbf{G}\|_{\mathcal{H}_\infty}$ to be less than $\gamma$.

So let us assume $\|\mathbf{D}\| < \gamma$ for the remainder of this proof. We now recall that singular values measure how close a non-singular matrix is to being singular. In particular $\sup_\omega \overline{\sigma}(\mathbf{G}(j\omega)) < \gamma$ if and only if

$$\sup_\omega \overline{\sigma}(\mathbf{G}^*(j\omega)\mathbf{G}(j\omega)) < \gamma^2\mathbf{I}$$

which holds if and only if the matrix $\gamma^2\mathbf{I} - \mathbf{G}^*(j\omega)\mathbf{G}(j\omega)$ is nonsingular for all $\omega$.

Now consider the dynamical system with transfer function matrix

$$\mathbf{\Phi}(s) = (\gamma^2\mathbf{I} - \mathbf{G}^*(s)\mathbf{G}(s))^{-1}$$

In view of our preceding condition $\|\mathbf{G}\|_{\mathcal{H}_\infty} < \gamma$ if and only if $\Phi(s)$ has no poles on the imaginary axis. For if such a pole exists, say at $j\omega_0$, then

$$\mathbf{\Phi}^{-1}(j\omega_0) = 0 = \gamma^2\mathbf{I} - \mathbf{G}^*(j\omega_0)\mathbf{G}(j\omega_0)$$

which contradicts our assumption that $\mathbf{\Phi}^{-1}(j\omega)$ is nonsingular for all $\omega$.

The next thing we note is that $\mathbf{\Phi}(s)$ has the state space realization

$$\mathbf{\Phi}(s) \overset{s}{=} \left[\begin{array}{c|c} \mathbf{H} & \left[\begin{array}{c} \mathbf{B}\mathbf{R}^{-1} \\ -\mathbf{C}^T\mathbf{D}\mathbf{R}^{-1} \end{array}\right] \\ \hline \left[\begin{array}{cc} \mathbf{R}^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{R}^{-1}\mathbf{B}^T \end{array}\right] & \mathbf{R}^{-1} \end{array}\right]$$

where $\mathbf{H}$ is the matrix in equation (48). This matrix is often called a *Hamiltonian matrix*.

So assume that $\mathbf{H}$ has an eigenvalue on the imaginary axis (say at $j\omega_0$), then there exists $x_0 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \neq 0$ such that $(j\omega_0\mathbf{I} - \mathbf{H})x_0 = 0$. If this eigenvalue corresponds to a controllable/observable mode of $\mathbf{\Phi}(s)$, then $\mathbf{\Phi}(s)$ has a pole on the imaginary axis and $\|\mathbf{G}\|_{\mathcal{H}_\infty}$ cannot be less than $\gamma$.

So if $\|\mathbf{G}\|_{\mathcal{H}_\infty} < \gamma$, then $j\omega_0$ must be either an uncontrollable or unobservable mode of $\mathbf{\Phi}(s)$. We will show that this case cannot occur. So let's assume $j\omega_0$ is an unobservable mode of $\mathbf{\Phi}(s)$. The Popov-Bellman Hautus (PBH) observability test requires the matrix

$$\left[\begin{array}{c|c} \lambda\mathbf{I} - \mathbf{H} & \left[\begin{array}{cc} \mathbf{R}^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{R}^{-1}\mathbf{B}^T \end{array}\right] \end{array}\right]$$

have full row rank for all $\lambda$ if the system is observable. So if $j\omega_0$ is an unobservable mode, however, then there exists $x_0 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \neq 0$ such that

$$\left[\begin{array}{c|c} \lambda\mathbf{I} - \mathbf{H} & \left[\begin{array}{cc} \mathbf{R}^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{R}^{-1}\mathbf{B}^T \end{array}\right] \end{array}\right] x_0 = 0$$

which can only happen if

$$\begin{aligned} \mathbf{H}x_0 &= j\omega_0 x_0 \\ 0 &= \left[ \begin{array}{cc} \mathbf{R}^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{R}^{-1}\mathbf{B}^T \end{array} \right] x_0 \end{aligned}$$

These equations, however, may be expanded out to

$$\begin{aligned} (j\omega_0\mathbf{I} - \mathbf{A})x_1 &= 0 \\ (j\omega_0\mathbf{I} + \mathbf{A}^T)x_2 &= -\mathbf{C}^T\mathbf{C}x_1 \\ \mathbf{D}^T\mathbf{C}x_1 + \mathbf{B}^T x_2 &= 0 \end{aligned}$$

Since $\mathbf{A}$ is assumed to have no eigenvalues on the imaginary axis, we know that $(j\omega_0\mathbf{I} - \mathbf{A})x_1 = 0$ implies $x_1 = 0$. Inserting $x_1 = 0$ into the second equation yields, $(j\omega_0\mathbf{I} + \mathbf{A}^T)x_2 = 0$. Again $\mathbf{A}^T$ has no eigenvalues on the imaginary axis and so $x_2$ must be zero. This contradicts, however, our earlier assertion that $x_0 \neq 0$.

Similarly, we can treat the case where we assume $j\omega_0$ is an uncontrollable mode of $\mathbf{\Phi}(s)$. The application of the PBH test for controllability generates a contradiction and we must therefore conclude that $\mathbf{\Phi}(s)$ can have no poles on the $j\omega$ axis if and only if the Hamiltonian matrix $\mathbf{H}$ has no eigenvalues on the imaginary axis and this completes the proof. $\diamondsuit$

THEOREM 15. ***[KYP Lemma]*** *Suppose* $\mathbf{G}(s) = \mathbf{D} + \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1})\mathbf{B}$ *with* $\mathbf{A}$ *being Hurwitz. Let* $\mathbf{R} = \gamma^2\mathbf{I} - \mathbf{D}^T\mathbf{D}$*, then* $\|\mathbf{G}\|_{\mathcal{H}_\infty} < \gamma$ *if and only if* $\|\mathbf{D}\| < \gamma$ *and there exists* $\mathbf{P} = \mathbf{P}^T \geq 0$ *satisfying the algebraic Riccati equation*

$$\mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C}) + (\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C})^T\mathbf{P} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{C}^T(\mathbf{I} - \mathbf{D}\mathbf{R}^{-1}\mathbf{D}^T)\mathbf{C} = \mathbf{0}$$

*such that* $\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}(\mathbf{D}^T\mathbf{D}^T\mathbf{C} + \mathbf{B}^T\mathbf{P})$ *is Hurwitz.*

**Proof:** We prove this lemma by showing that the solution to the Riccati equation is equivalent to the condition in the bounded real lemma that requires the Hamiltonian matrix

$$\mathbf{H} = \left[ \begin{array}{cc} \mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ -\mathbf{C}^T(\mathbf{I} + \mathbf{D}\mathbf{R}^{-1}\mathbf{D}^T)\mathbf{C} & -(\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C})^T \end{array} \right] = \left[ \begin{array}{cc} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{array} \right]$$

has no eigenvalues on the $j\omega$ axis. Once this is established then the KYP lemma follows from the bounded real lemma.

Let us first assume the existence of an internally stabilizing solution to the Riccati equation in the Bounded Real Lemma. In other words, $\mathbf{P}$, satisfies the Riccati equation where

$$\overline{\mathbf{A}} = \mathbf{A} + \mathbf{B}\mathbf{R}^{-1}(\mathbf{D}^T\mathbf{C} + \mathbf{B}^T\mathbf{P})$$

is Hurwitz. A similarity transformation shows that

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} \end{bmatrix} \mathbf{H} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{P} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{A}} & \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ \mathbf{0} & -\overline{\mathbf{A}}^T \end{bmatrix}$$

Because $\overline{\mathbf{A}}$ is Hurwitz, we know $-\overline{\mathbf{A}}^T$ will be the system matrix for an unstable system, which means that $\mathbf{H}$ can have no eigenvalues on the imaginary axis.

Since $\mathbf{H}$ is Hamiltonian, one can show the eigenvalues are symmetric with respect to the imaginary axis. By assumption, none of these eigenvalues are on the $j\omega$ axis so $\mathbf{H}$ has exactly $n$ eigenvalues with strictly negative real parts. Let $\mathbf{\Lambda}$ be a Hurwitz matrix whose eigenvalues are the stable eigenvalues of $\mathbf{H}$. Then we can find matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ such that

$$\mathbf{H} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} \mathbf{\Lambda}, \tag{49}$$

The matrix $\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}$ is a $2n \times n$ matrix with full column rank in which $n$ is the dimension of $\mathbf{\Lambda}$.

This matrix $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}$ satisfies the equation $\mathbf{X}^T\mathbf{J}\mathbf{X}$ where $\mathbf{J} = \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$. So by the property of Hamiltonians that $\mathbf{J}\mathbf{H}$ is symmetric we see that the righthand side of the above equation is zero. So we can see that

$$0 = (\mathbf{X}^T\mathbf{J}\mathbf{X})\mathbf{\Lambda} + \mathbf{\Lambda}^T(\mathbf{X}^T\mathbf{J}\mathbf{X})$$

By assumption the eigenvalues of $\mathbf{\Lambda}$ all have negative real parts, so the preceding linear matrix equation implies that $\mathbf{X}^T\mathbf{J}\mathbf{X} = 0$. If we expand this last equation out, we see this is equivalent to $\mathbf{X}_1^T\mathbf{X}_2 = \mathbf{X}_2^T\mathbf{X}_1$. We will need to use this identity below.

**$\mathbf{X}_1$ is nonsingular:** We will now prove that $\mathbf{X}_1$ is nonsingular. We will do this through contradiction so assume that $\mathbf{X}_1$ is singular. This means there exists $z \neq 0$

such that $\mathbf{X}_1 z = 0$. From equation 49, we can expand the first block to see that

$$(\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C})\mathbf{X}_1 + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{X}_2 = \mathbf{X}_1\mathbf{\Lambda} \tag{50}$$

Premultiplying by $\mathbf{X}_2^T$ yields,

$$\mathbf{X}_2^T\left(\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C}\right)\mathbf{X}_1 + \mathbf{X}_2^T\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{X}_2 = \mathbf{X}_2^T\mathbf{X}_1\mathbf{\Lambda} = \mathbf{X}_1^T\mathbf{X}_2\mathbf{\Lambda}$$

The last equality holds because $\mathbf{X}_1^T\mathbf{X}_2 = \mathbf{X}_2^T\mathbf{X}_1$. We now pre and post-multiply the last equation by $z$ to obtain

$$\mathbf{B}_2^T\mathbf{X}_2 z = 0$$

Post-multiplying equation 50 by $z$ and using the preceding equation implies that

$$\mathbf{X}_1\mathbf{\Lambda} z = 0$$

We can repeat this argument to show that

$$\mathbf{X}_1 z = 0 \quad \Rightarrow \quad \left[\begin{array}{c} \mathbf{X}_1 \\ \mathbf{B}^T\mathbf{X}_2 \end{array}\right]\mathbf{\Lambda}^k z = 0$$

for $k = 0, 1, 2, \cdots$. This last relationship implies that $\left(\mathbf{\Lambda}, \left[\begin{array}{c} \mathbf{X}_1 \\ \mathbf{B}^T\mathbf{X}_2 \end{array}\right]\right)$ is not observable. Consequently by the Popov-Bellman-Hautus (PBH) observability test, there exits $y \neq 0$ and $\lambda$ such that

$$\left[\begin{array}{c} \mathbf{\Lambda} - \lambda I \\ \mathbf{X}_1 \\ \mathbf{B}^T\mathbf{X}_2 \end{array}\right] y = 0 \tag{51}$$

Note that $\mathrm{Re}(\lambda) < 0$ because $\mathbf{\Lambda}$ is asymptotically stable.

We know from the second block of equation 49 that

$$-\mathbf{C}^T(\mathbf{I} + \mathbf{D}\mathbf{R}^{-1}\mathbf{D}^T)\mathbf{C}\mathbf{X}_1 - (\mathbf{A}^T + \mathbf{C}^T\mathbf{D}\mathbf{R}^{-1}\mathbf{B}^T)\mathbf{X}_2 = \mathbf{X}_2\mathbf{\Lambda}$$

Multiplying by $y$ we get $-\mathbf{A}^T\mathbf{X}_2 y = \lambda\mathbf{X}_2 y$, which we may rearrange to obtain $(\lambda I + \mathbf{A}^T)\mathbf{X}_2 y = 0$. Because $\mathbf{A}$ is asymptotically stable and $\mathrm{Re}(\lambda) < 0$, we can see that $\lambda I + \mathbf{A}^T$ is nonsingular and therefore $\mathbf{X}_2 y = 0$. From equation 51, we know that $\mathbf{X}_1 y = 0$. We've just shown that $\mathbf{X}_2 y = 0$. These two equations contradict the full rank property of $\left[\begin{array}{c} \mathbf{X}_1 \\ \mathbf{X}_2 \end{array}\right]$. From this contraction we must conclude that $\mathbf{X}_1$ is nonsingular.

**Verify the Riccati Equation** Let $\mathbf{P} = \mathbf{X}_2\mathbf{X}_1^{-1}$. From our earlier intermediate result, we know that $\mathbf{X}_1\mathbf{X}_2 = \mathbf{X}_2^T\mathbf{X}_1$. So we can see that $\mathbf{X}_2\mathbf{X}_1^{-1} = (\mathbf{X}_1^T)^{-1}\mathbf{X}_2^T$, which means that $\mathbf{P}$ is symmetric.

Our intermediate result also showed that $\mathbf{X}^T\mathbf{J}\mathbf{X} = 0$, so we can conclude that

$$\mathbf{X}^T\mathbf{J}\mathbf{H}\mathbf{X} = \mathbf{X}^T\mathbf{J}\mathbf{X}\mathbf{\Lambda} = 0$$

We can therefore see that

$$\begin{aligned} 0 &= (\mathbf{X}_1^T)^{-1}\mathbf{X}^T\mathbf{J}\mathbf{H}\mathbf{X}\mathbf{X}_1^{-1} \\ &= \begin{bmatrix} \mathbf{P} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{P} \end{bmatrix} \\ &= \mathbf{P}\mathbf{H}_{11} - \mathbf{H}_{21} + \mathbf{P}\mathbf{H}_{12}\mathbf{P} - \mathbf{H}_{22}\mathbf{P} \end{aligned}$$

Because of the Hamiltonian property, we know that $\mathbf{H}_{22} = -\mathbf{H}_{11}^T$, so the last equation can be rewritten as

$$0 = \mathbf{P}\mathbf{H}_{11} + \mathbf{H}_{11}^T\mathbf{P} + \mathbf{P}\mathbf{H}_{12}\mathbf{P} - \mathbf{H}_{21}$$

Substituting in for the Hamiltonian block matrices, we obtain the algebraic Riccati equation,

$$\mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C}) + (\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{D}^T\mathbf{C})^T\mathbf{P} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{C}^T(\mathbf{I} - \mathbf{D}\mathbf{R}^{-1}\mathbf{D}^T)\mathbf{C} = 0$$

So our choice for $\mathbf{P}$ clearly satisfies the $\mathcal{H}_\infty$ Riccati equation.

**The solution is stablizing:** We now establish that the solution to the Riccati equation is stabilizing. In other words, we will show that $\mathbf{A}+\mathbf{B}\mathbf{R}^{-1}(\mathbf{D}^T\mathbf{C}+\mathbf{B}^T\mathbf{P})$ is asymptotically stable. A straightforward computation shows that

$$\begin{aligned} \mathbf{H}_{11} + \mathbf{H}_{12}\mathbf{P} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{H}\mathbf{X}\mathbf{X}_1^{-1} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{X}\mathbf{\Lambda}\mathbf{X}_1^{-1} \\ &= \mathbf{X}_1\mathbf{\Lambda}\mathbf{X}_1^{-1} \end{aligned}$$

which implies that

$$\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}(\mathbf{D}^T\mathbf{C} + \mathbf{B}^T\mathbf{P}) = \mathbf{X}_1\mathbf{\Lambda}\mathbf{X}_1^{-1}$$

Since $\mathbf{\Lambda}$ is a stable matrix and $\mathbf{X}_1$ is a nonsingular matrix, we know that $\mathbf{A} + \mathbf{B}\mathbf{R}^{-1}(\mathbf{D}^T\mathbf{C} + \mathbf{B}^T\mathbf{P})$ is asymptotically stable.

**The solution is symmetric:** Let $\mathbf{P} = \mathbf{X}_2\mathbf{X}_1^{-1}$. From our earlier intermediate result, we know that $\mathbf{X}_1\mathbf{X}_2 = \mathbf{X}_2^T\mathbf{X}_1$. So we can see that $\mathbf{X}_2\mathbf{X}_1^{-1} = (\mathbf{X}_1^T)^{-1}\mathbf{X}_2^T$, which means that $\mathbf{P}$ is symmetric.

**The solution is positive definite:** To show that $\mathbf{P} \geq 0$, rewrite the Riccati equation as

$$\mathbf{PA} + \mathbf{A}^T\mathbf{P} + (\mathbf{D}^T\mathbf{C} + \mathbf{B}^T\mathbf{P})^T\mathbf{R}^{-1}(\mathbf{D}^T\mathbf{C} + \mathbf{B}^T\mathbf{P}) + \mathbf{C}^T\mathbf{C} = 0$$

This equation, along with the fact that $\mathbf{A}$ is stable and $\mathbf{R}^{-1} > 0$ implies that $\mathbf{P} \geq 0$ since $\mathbf{P}$ would then be the observability gramian of $\left(\mathbf{A}, \begin{bmatrix} \mathbf{C} \\ \mathbf{R}^{-1/2}(\mathbf{D}^T\mathbf{C} + \mathbf{B}^T\mathbf{P}) \end{bmatrix}\right)$.
$\diamondsuit$

CHAPTER 4

# Stability Concepts for Nonlinear Control Systems

Chapters 1 and 3 confined their attention to the design of stabilizing control laws for linear time-invariant systems. But most systems are not linear, they are nonlinear. While we can still view the system as a causal operator between linear signal spaces, that operator may not satisfy the principle of superposition. In many cases, using a linear approximation to this nonlinear operator based on a Taylor series expansion about a specified equilibrium point may be used to develop controllers that stabilize that equilibrium. This is particularly true when the equilibrium is hyperbolic (i.e. the linearization's $\mathbf{A}$ matrix has no eigenvalues with zero real parts). But even in this case, the neighborhood about which the linearization is useful may be too small for practical applications. We therefore need to move beyond the methods in chapters 1 and 3 and develop practical frameworks for controlling nonlinear systems. We will focus on the *stabilization* of a nonlinear system about a desired operating point. To develop such controllers, this chapter introduces several stability concepts that are used in designing nonlinear control systems.

The remainder of this chapter is organized around four stability concepts; Lyapunov stability, input-to-state stability (ISS), $\mathcal{L}_p$-stability, and passivity. These four concepts provide the fundamental tools that have been previously used for nonlinear control. We review the fundamentals of Lyapunov stability, which was covered in detail in the linear systems theory class. So the most relevant new concepts will concern input-to-state stability, $\mathcal{L}_p$ stability, and passivity. This chapter first presents sufficient conditions certifying these stability concepts. But we will also need to examine the relationship between these stability concepts. Of great importance to us will be whether these stability properties are preserved under system interconnections; in particular feedback interconnections and cascade interconnections.

## 1. Lyapunov Stability

Consider a dynamical system characterized by a differential equation

$$\dot{x}(t) = f(x(t))$$

where $f : D \to \mathbb{R}^n$ is locally Lipschitz on an open connected domain, $D \subset \mathbb{R}^n$. For this system, we say a state $x^* \in D$ is an *equilibrium point* if $f(x^*) = 0$. If the equilibrium $x^* \neq 0$, then one can introduce a change of coordinates so that in the new coordinate frame the system's equilibrium is at the origin. Stating many of the results regarding Lyapunov stability are simplified if we don't need to carry along $x^*$ notationally through the derivation. So it is customary to assume the origin is the equilibrium point and we will do so in this chapter unless otherwise stated.

We say the equilibrium $x^* = 0$ is *stable in the sense of Lyapunov* if for all $\epsilon > 0$ there exists $\delta > 0$ such that if $|x(0)| < \delta$ then $|x(t)| < \epsilon$ for all $t \geq 0$. We say the equilibrium is *unstable* if it is not Lyapunov stable. We say the equilibrium is asymptotically stable if it is stable and $x(t) \to 0$ as $t \to \infty$ for all $x(0)$ in a neighborhood of the origin.

### 1.1. Basic Lyapunov Stability Theorems:

A sufficient condition for an equilibrium to be stable or asymptotically stable is the existence of a positive definite function known as the *Lyapunov function*. This is stated in the following theorem without proof (we prove this in linear systems theory course)

THEOREM 16. ***Lyapunov's Direct Method:*** *Let $0$ be the equilibrium point for the system $\dot{x}(t) = f(x(t))$ where $f : D \to \mathbb{R}^n$ is locally Lipschitz on the connected open set $D \subset \mathbb{R}^n$. If there exists a $C^1$ function $V : D \to \mathbb{R}$ such that*

- *$V$ is positive definite ($V(0) = 0$ and $V(x) > 0$ for all $x \in D - \{0\}$)*
- *$\dot{V}(x) = \dfrac{\partial V(x)}{\partial x} f(x)$ is negative semidefinite ($\dot{V}(x) \leq 0$ for all $x \in D$)*

*then $x = 0$ is stable in the sense of Lyapunov. Furthermore if one can show that $\dot{V}(x)$ is negative definite ($\dot{V}(0) = 0$ and $\dot{V}(x) < 0$ for all $x \in D - \{0\}$), then the equilibrium is asymptotically stable.*

A $C^1$ function, $V : D \to \mathbb{R}^n$ that satisfies the conditions in theorem 16 is called a *Lyapunov function*. We can think of it as a *certificate* for Lyapunov stability since the existence of such a function is sufficient to "certify" that the origin is Lyapunov stable.

As defined above, Lyapunov stability is a *local* property of the equilibrium since it only holds for $x(0)$ in a neighborhood of the origin. If we can assure that this property holds for all $x(0)$ then the property is said to be *global*. Note that even if there is a positive definite $V$ for which $\dot{V} < 0$ for all $x \in \mathbb{R}^n$, this will not necessarily certify the equilibrium is "globally" asymptotically stable. The following example shows why this might occur

Consider the dynamical system

$$\begin{aligned}
\dot{x}_1 &= f_1(x_1, x_2) = -\frac{6x_1}{(1+x_1^2)^2} + 2x_2 \\
\dot{x}_2 &= f_2(x_1, x_2) = -2\frac{x_1 + x_2}{(1+x_1^2)^2}
\end{aligned}$$

One can readily verify that a "candidate" Lyapunov function

$$V(x) = \frac{x_1^2}{1+x_1^2} + x_2^2$$

has $\dot{V} < 0$ for all $x \in \mathbb{R}^n$, and so the origin is asymptotically stable for $x(0)$ in a "neighborhood" of the origin, but not all of $\mathbb{R}^2$. To see this consider a hyperbola in $\mathbb{R}^2$

$$x_2 = \frac{2}{x_1 - \sqrt{2}}$$

One can show that the ratio of the two vectors fields along the hyperbola, $\frac{f_2}{f_1}$ will always be greater than the slope of the hyperbola's tangents. So if $x(0)$ starts in the "divergent" region shown in Fig. 1, we can see that there is no way for a point in this divergent region to cross the hyperbola and so the equilibrium is not globally asymptotically stable.

The reason for this issue can be readily seen if we return to our definition. Lyapunov stability requires that for any $\epsilon > 0$ there exists a $\delta > 0$ such that $|x(0)| < \delta$ implies $|x(t)| < \epsilon$ for all $t \geq 0$. The problem is that $\delta$ is a function of $\epsilon$ and the particular pathology that might take place is that as $\epsilon \to 0$, we might find $\delta(\epsilon) \to \overline{\delta} < \infty$ converging to a constant that is finite. The resulting neighborhood

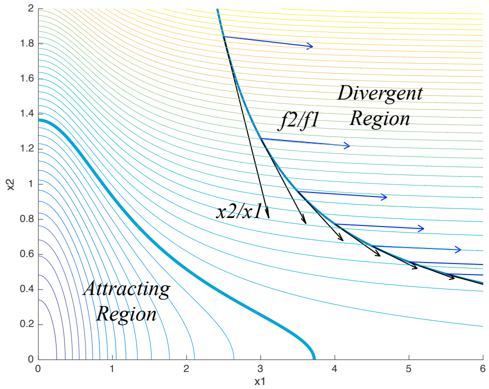


FIGURE 1. Example where $\dot{V}(x) < 0$ for all $x$ but the origin is not globally asymptotically stable

$N_{\overline{\delta}}(0)$ is an *inner approximation* to the set of all states that converge asymptotically to the equilibrium. That set is called the equilibrium's region of attraction (RoA). This observation also suggests a simple way out of our dilemma. In particular, if we select a candidate Lyapunov function that is positive definite and *radially unbounded* in the sense that $V(x) \to \infty$ as $|x| \to \infty$, then we can certify the origin is globally asymptotically stable. This is the assertion in the following theorem which is again presented without proof.

THEOREM 17. **Barbashin-Krasovskii Theorem:** *Let $x = 0$ be an equilibrium for $\dot{x} = f(x)$ where $f$ is locally Lipschitz. Let $V : \mathbb{R}^n \to \mathbb{R}$ be a radially unbounded positive definite $C^1$ function such that $\dot{V}(x) < 0$ for all $x \neq 0$. Then the origin of the system is globally asymptotically stable.*

As mentioned above, it is customary to "linearize" a nonlinear system about its equilibrium point and use that linearization as a basis for control system design. The justification for this approach is known as *Lyapunov's Indirect method.*

THEOREM 18. **Lyapunov's Indirect Method:**. *Let $\dot{x} = \mathbf{A}x$ be the Taylor jet linearization of the nonlinear system $\dot{x} = f(x)$ about the equilibrium point $x^* = 0$. Let $\{\lambda_i\}_{i=1}^n$ denote the eigenvalues of matrix* $\mathbf{A}$. *If* $\mathrm{Re}(\lambda_i) < 0$ *for all $i = 1, \ldots, n$, then the nonlinear system's equilibrium is asymptotically stable. If there exists at least one $i$ such that* $\mathrm{Re}(\lambda_i) > 0$ *then the origin is unstable.*

Proving the stability part of this theorem is interesting enough to discuss. Finding Lyapunov functions is, in general, difficult to do. So what we often do is select a Lyapunov function for a system that is close to the one we're interested in and use that as a candidate Lyapunov function. We know the Taylor linearization of the nonlinear system is ”close” to the nonlinear system because

$$f(x) = \mathbf{A}x + g(x)$$

where $g(x)$ is a little-o remainder term, i.e. $\lim\limits_{x\to 0} \dfrac{|g(x)|}{|x|} = 0$, then let us try using a Lyapunov function for the linear system $\dot{x} = \mathbf{A}x$ as a candidate Lyapunov function for $\dot{x} = f(x)$. A Lyapunov function for $\dot{x} = \mathbf{A}x$ will be $V(x) = x^T\mathbf{P}x$ where $\mathbf{P} = \mathbf{P}^T > 0$ satisfies the Lyapunov equation

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} = 0 \tag{52}$$

for some $\mathbf{Q} = \mathbf{Q}^T > 0$. We now compute $\dot{V}$ with respect to the nonlinear system's $f(x) = \mathbf{A}x + g(x)$ to get

$$\begin{aligned} \dot{V} &= x^T\mathbf{P}f(x) + f^T(x)\mathbf{P}x \\ &= x^T(\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P})x + 2x^T\mathbf{P}g(x) \\ &= -x^T\mathbf{Q}x + 2x^T\mathbf{P}g(x) \end{aligned}$$

The first term is negative definite and the second term is indefinite. But we know that $g$ is little-o so there exists $r > 0$ such that for any $\gamma > 0$ we have

$$|g(x)| < \gamma |x|$$

when $|x| < r$. This means that for $x$ with $|x| < r$ we have

$$\begin{aligned} \dot{V} &< -x^T\mathbf{Q}x + 2\gamma \,\|\mathbf{P}\|\, |x|^2 \\ &< -(\underline{\lambda}(\mathbf{Q}) - 2\gamma \,\|\mathbf{P}\|)|x|^2 \end{aligned}$$

where $\underline{\lambda}(\mathbf{Q})$ is the minimum eigenvalue of $\mathbf{Q}$. So if we choose $\gamma < \dfrac{\underline{\lambda}(\mathbf{Q})}{2\,\|\mathbf{P}\|}$ then we can certify $\dot{V} < 0$ for $|x| < r$ which establishes the asymptotic stability of the origin provided $\mathbf{A}$ is Hurwitz.

**Remark:** We can summarize the three main findings of the Indirect Method below

- If the equilibrium of the linearization is asymptotically stable, then the origin of the original nonlinear system is also locally asymptotically stable.
- If the linearization has any eigenvalue with a positive real part, then the origin of the nonlinear system is unstable.
- If the linearization of the nonlinear system has eigenvalues with nonpositive real parts and there is at least one eigenvalue with a zero real part, then nothing can be concluded about the asymptotic stability of the equilibrium.

. There is, therefore, a hole in our linearization's ability to deduce the asymptotic stability properties of nonlinear systems. In particular, we require the linearization not have a center eigensubspace.

**1.2. Advanced Lyapunov Stability Theorems:** The direct method only established the equilibrium's asymptotic stability if $\dot{V}$ was negative definite. It is often easier to certify that $\dot{V}$ is negative semidefinite, in which case we cannot use the direct method. The *invariance principle* provides a tool that allows us to infer that trajectories of the system are "attracted" to an invariant set when $\dot{V}$ is only negative semidefinite. Given the system $\dot{x} = f(x)$, where $f : D \to \mathbb{R}^n$ is Lipschitz, we say a set $M \subset D$ is attracting if for any $x(0) \notin M$, we have $x(t) \to M$ as $t \to \infty$. The condition we need to ensure this occurs is that the trajectories of $\dot{x} = f(x)$ are confined to a compact invariant set. Compactness is a useful topological property in which a subset $K \subset \mathbb{R}^n$ is compact if it is closed and bounded. The set is forward invariant if for any $x(0) \in S$ we have $x(t) \in S$ for all $t \geq 0$ under the state equation $\dot{x} = f(x)$. The following theorem, known as the *Invariance Principle*, formally states this result

THEOREM 19. **Invariance Principle:** *Consider the system $\dot{x} = f(x)$ where $f : D \to \mathbb{R}^n$ is locally Lipschitz on $D \subset \mathbb{R}^n$. Let $K \subset D$ be a compact invariant set with respect to $f$ and $V : D \to \mathbb{R}$ be a $C^1$ function such that $\dot{V}(x) \leq 0$ on $K$. Let $M$ be the largest forward invariant set in $E = \{x \in K : \dot{V}(x) = 0\}$, then $M$ is attracting for all trajectories starting in $K$.*

Note that $V$ need not be positive definite since the trajectories are confined to a compact set. This theorem also relaxes the requirement that $\dot{V}$ is negative definite. So it greatly relaxes the sufficient conditions found in Lyapunov's direct method. When the invariant set of interest is the equilibrium, then this theorem essentially proves asymptotic stability of the equilibrium

THEOREM 20. ***Asymptotic Stability - invariance theorem:*** *Let $x = 0$ be an equilibrium point for $\dot{x} = f(x)$ where $f : D \to \mathbb{R}^n$ is locally Lipschitz on $D \subset \mathbb{R}^n$. Let $V : D \to \mathbb{R}$ be a $C^1$ positive definite function on $D$ containing $x = 0$ such that $\dot{V}(x) \leq 0$ on $D$. If the origin, $\{0\}$, is the largest invariant set in the set $\{x \in D : \dot{V}(x) = 0\}$, then the origin is asymptotically stable.*

**Example:** Consider the system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -g(x_1) - h(x_2) \end{aligned}$$

where $g$ and $h$ are Lipschitz such that $g(0) = h(0) = 0$, $yg(y) > 0$ and $yh(y) > 0$. We can think of this as a mechanical system in which $x_2$ is velocity, $x_1$ is position. The equilibrium for this system is clearly the origin and we consider a candidate Lyapunov function

$$V(x) = \int_0^{x_1} g(y)dy + \frac{1}{2}x_2^2$$

where the first term on the left may be seen as potential energy and the second term is kinetic energy. $V$ is clearly positive definite and its directional derivative is

$$\begin{aligned} \dot{V}(x) &= g(x_1)x_2 + x_2(-g(x_1) - h(x_2)) \\ &= -x_2 h(x_2) \leq 0 \end{aligned}$$

So $\dot{V}$ is only negative semidefinite. We will take set $E$ to be

$$E = \{x : \dot{V}(x) = 0\} = \{x : x_2 = 0\}$$

If we let $x(t)$ be any trajectory starting in $E$, then this implies that $\dot{x}_1(0) = 0 = x_2(0) = 0$ which means $x_1(t)$ is constant for all time. But if $x_1(t) \neq 0$, then

$$\dot{x}_2(t) = -g(x_1(t)) - h(x_2(t)) \neq 0$$

which would force $x(t)$ to leave the set $E$, unless $x_1(0)$ were also zero. So we can conclude that the origin is the largest invariant set in $E$ and so the origin must be asymptotically stable.

**Example:**. Consider the Lyapunov equation

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{C}^T\mathbf{C}$$

associated with the linear system

$$\begin{aligned} \dot{x} &= \mathbf{A}x(t) \\ y &= \mathbf{C}x(t) \end{aligned}$$

where $\mathbf{C}$ does not have full rank. This means $x^T\mathbf{C}^T\mathbf{C}x$ is only positive semidefinite. Let $V(x) = x^T\mathbf{P}x$ where $\mathbf{P} = \mathbf{P}^T > 0$. Note that $\dot{V}(x) = -x^T\mathbf{C}^T\mathbf{C} \leq 0$ So $\dot{V}$ is only negative semidefinite and we cannot use the direct method to infer stability.

We can, however, try using the invariance principle. Consider the set where $\dot{V} = 0$ and note that is also the set

$$E = \{x \,:\, \mathbf{C}x = 0\}$$

This system's output would then be

$$y(t) = \mathbf{C}x(t) = \mathbf{C}e^{\mathbf{A}t}x_0$$

which would be identically zero (i.e. remains in $E$) if and only if $(\mathbf{A}, \mathbf{C})$ is an observable pair and $x_0 = 0$. So the only trajectory that can remain in $E$ for all time is the one that starts in the origin. We can therefore conclude that under the additional assumption that $(\mathbf{A}, \mathbf{C})$ is an observable pair, the largest invariant set in $E$ is the origin and so by the invariance principle is attracting. In other words, the origin is asymptotically stable.

The preceding results have only been sufficient conditions for Lyapunov stability. When the system is linear one can prove that the existence of a Lyapunov function is also necessary for stability. Results of this type are known as *converse theorems*. They are important in the design of control systems, because if we require our control system to be stable, it would have to have a Lyapunov function and that condition often helps us to determine what the controller should be. In addition to linear systems, it is possible to establish converse theorems for certain classes of nonlinear systems. The following theorem establishes a converse theorem for systems (linear or nonlinear) whose origin have *uniform asymptotic stability* (UAS).

In stating this theorem we need to consider class $\mathcal{K}$ and $\mathcal{KL}$ functions. In particular a continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is class $\mathcal{K}$ if and only if $\alpha(0) = 0$ and $\alpha$ is strictly increasing. We say a function is class $\mathcal{K}_\infty$ if it is class $\mathcal{K}$ and radially unbounded. A continuous function $\beta : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is class $\mathcal{KL}$ if

- for any fixed $s$, $\beta(r, s)$ is class $\mathcal{K}$,
- for any fixed $r$, $\beta(r, s)$ is a decreasing function such that $\beta(r, s) \to 0$ as $s \to \infty$.

Basic properties of class $\mathcal{K}$ and class $\mathcal{KL}$ functions are itemized below without formal proof.

- if $\alpha \in \mathcal{K}$ over $[0, a)$, then $\alpha^{-1} \in \mathcal{K}$ over $[0, \alpha(a))$.
- If $\alpha \in \mathcal{K}_\infty$ then $\alpha^{-1} \in \mathcal{K}_\infty$.
- If $\alpha_1, \alpha_2 \in \mathcal{K}$, then $\alpha_1 \circ \alpha_2 \in \mathcal{K}$
- If $\alpha_1, \alpha_2 \in \mathcal{K}$ and $\beta \in \mathcal{KL}$ then $\alpha_1(\beta(\alpha_2(r), s)) \in \mathcal{KL}$.

One particularly useful result is the class $\mathcal{K}$ Comparison Principle. This principle states that if $\alpha \in \mathcal{K}$, then the differential equation, $\dot{y} = -\alpha(y)$, with initial condition $y(0) = y_0$ has a unique solution $y(t) = \sigma(y_0, t)$ where $\sigma$ is class $\mathcal{KL}$.

With this background on class $\mathcal{K}$ and $\mathcal{K}L$ functions, we can now state the UAS converse theorem. The proof of this result is somewhat sophisticated and so the proof is not presented below.

THEOREM 21. ***UAS - Converse Theorem:*** *Let $x(t) = 0$ be the equilibrium of $\dot{x} = f(t, x)$ where $f$ is continuously differentiable on $D = \{x \in \mathbb{R}^n : |x| < r\}$ and the Jacobian matrix of $f$ is uniformly bounded in t. Let $\beta : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a class $\mathcal{KL}$ function and assume there exists $r_0 > 0$ such that for any$|x(t_0)| < r$ we have such that*

$$|x(t)| \leq \beta(|x(t_0)|, t - t_0)$$

*for all* $t \geq t_0 > 0$*. Then there is a continuously differentiable function* $V : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ *and class* $\mathcal{K}$ *functions* $\underline{\alpha}$*,* $\overline{\alpha}$*,* $\alpha$*, and* $\omega$ *such that*

$$\begin{aligned} \underline{\alpha}(|x|) \leq V(t,x) &\leq \overline{\alpha}(|x|) \\ \dot{V}(x) &\leq -\alpha(|x|) \\ \left\| \frac{\partial V(x)}{\partial x} \right\| &\leq \omega(|x|) \end{aligned}$$

## 2. Input-to-State Stability (ISS)

Lyapunov stability is a property of unforced (i.e. homogeneous) dynamical systems. Real-life systems, of course, are subject to forcing from the external environment and if we want to regulate their sensitivity to that environment we need a control input. So we will need to consider extensions of the Lyapunov stability concept to forced (i.e. inhomogeneous) dynamical systems. This section introduces two related state-based stability concepts for forced systems; uniform ultimate boundedness and input-to-state stability.

**2.1. Uniform Ultimate Boundedness:** Consider a dynamical system with a state-dependent forcing term,

$$\dot{x}(t) = f(x) + g(t,x) \overset{\text{def}}{\equiv} F(t,x)$$

with $f(0) = 0$ and $g(t,0) \neq 0$. So the unforced system, $\dot{x} = f(x)$, has an equilibrium at the origin, but when we perturb it with a state-dependent disturbance $g(t,x)$, that equilibrium disappears. Since there is no longer an ”equilibrium” for the perturbed system, $\dot{x} = F(t,x)$, the state will not asymptotically approach the origin. Rather the best we can hope for is that the state remains bounded in a sufficiently small neighborhood of the origin. Systems that exhibit this property are said to be *uniformly ultimately bounded* or UUB.

Formally we define UUB as follows. The system $\dot{x} = F(t,x)$ is said to be UUB if there exists $c > 0$ such that for all $a < c$ there are real positive constants $b$ and $T$ such that

$$\text{if } |x(t_0)| < a \text{ then } |x(t)| \leq b \text{ for all } t \geq t_0 + T \tag{53}$$

Fig.2 is a graphical interpretation of this concept. The figure shows that any trajectory originating in a ball $N_c(0)$ will enter and remain within a closed ball $N_b(0)$

within a finite time $T$. The radius, $b$, of that target neighborhood is independent of the initial time, so the property holds in a "uniform" manner. So all trajectories are ultimately convergent to a uniformly bounded set about the origin. The size of that neighborhood, $b$, is called the *ultimate bound*.

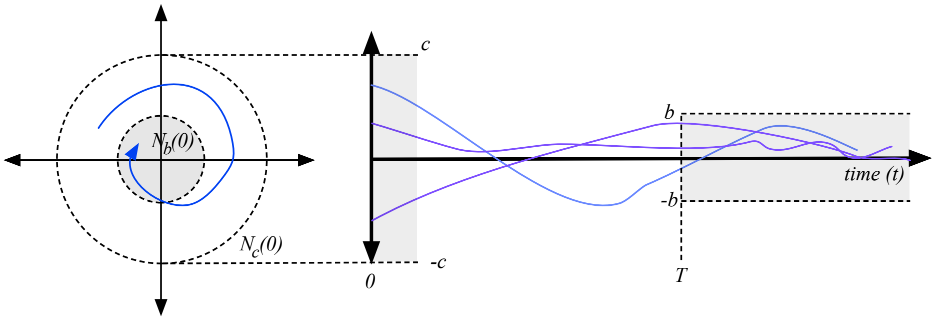


FIGURE 2. Uniform Ultimate Boundedness

The following theorem characterizes how the ultimate bound, $b$, varies as a function of the non-vanishing perturbation, $g(t,x)$. It is useful because it provides a Lyapunov-like sufficient condition for UUB that we will use later when we study input-to-state stability (ISS).

THEOREM 22. **UUB Theorem:** *Consider the system $\dot{x} = F(t,x)$ where $F : [0,\infty) \times D \rightarrow \mathbb{R}^n$ is piecewise continuous in $t$ and locally Lipschitz in $x$. Let $V : [0,\infty) \times D \rightarrow \mathbb{R}$ be $C^1$ such that for all $t$,*

$$\begin{aligned} \underline{\alpha}(|x|) &\leq V(t,x) \leq \overline{\alpha}(|x|) \\ \dot{V} &\leq -\alpha(|x|), \quad \textit{for all } |x| \geq \mu > 0 \end{aligned}$$

*where $\mu > 0$ and the functions $\underline{\alpha}, \overline{\alpha}$, and $\alpha$ are class $\mathcal{K}$. Then there exists a class $\mathcal{KL}$ function $\beta$ and a finite time $T > 0$ such that solutions for the system $\dot{x}(t) = F(t,x)$ satisfy*

$$\begin{aligned} |x(t)| &\leq \beta(|x(t_0)|, t - t_0), \quad \textit{for all } t_0 \leq t < T \\ x(t) &\in \left\{ x \in \mathbb{R}^n \,:\, |x| \leq \underline{\alpha}^{-1}(\overline{\alpha}(\mu)) \right\}, \quad \textit{for all } t > T \end{aligned}$$

**Proof:**. This theorem's proof relies on the properties of class $\mathcal{K}$ and class $\mathcal{KL}$ functions that we itemized above. We already know from the theorem's assumptions that $\dot{V}$ is negative for $x$ outside the closed ball $B_\mu(0) = \{x : |x| \leq \mu\}$. For notational convenience let $\eta = \overline{\alpha}(\mu)$ and define the set

$$\Omega_{t,\eta} = \{x : V(t,x) < \eta\}$$

Note that $B_\mu(0) \subset \Omega_{t,\eta}$ so that $\dot{V}$ is negative outside of $\Omega_{t,\eta}$. For any state outside of $\Omega_{t,\eta}$ we have

$$\dot{V} \leq -\alpha(|x|) \leq -\alpha(\overline{\alpha}^{-1}(V))$$

Since $\alpha \circ \overline{\alpha}$ is class $\mathcal{K}$ we know from the class $\mathcal{K}$ comparison principle that there exists a class $\mathcal{KL}$ function, $\sigma$, such that

$$V(t, x(t) \leq \sigma(V(t_0, x(t_0)), t - t_0)$$

for all $x(t)$ outside of $\Omega_{t,\eta}$. Since $\sigma$ is class $\mathcal{KL}$ we know $V(t, x(t))$ is decreasing until it eventually enters the set $\Omega_{t,\eta}$. Let $t_0 + T$ be the time when this occurs.

From the assumed bounds on $V(t,x)$ and $\dot{V}$, we can deduce that for $t_0 \leq t < t_0 + T$ that

$$\underline{\alpha}(|x|) \leq V(t,x) \leq \sigma(\overline{\alpha}(|x(t_0)|), t - t_0)$$

Applying $\underline{\alpha}^{-1}$ to both sides of the inequality yields

$$|x(t)| \leq \underline{\alpha}^{-1}(\sigma(\overline{\alpha}(|x(t_0)|), t - t_0))$$

for all $t_0 \leq t_0 + T$. So from above listed properties for comparison functions we can say the right hand side of the above inequality is class $\mathcal{KL}$.

Finally, we know that once $x(t)$ enters $\Omega_{t,\eta}$ it remains there because $\dot{V} < 0$ for all $x$ outside of the set. For $t > t_0 + T$, we know $V(t, x(t)) < \eta$ and using our assumed bounds on $V$ we get

$$|x| \leq \underline{\alpha}(V(t,x)) \leq \underline{\alpha}^{-1}(\eta) \leq \underline{\alpha}^{-1}(\overline{\alpha}(\mu))$$

which is the theorem's ultimate bound. ♢

**2.2. ISS-Lyapunov Functions:** The UUB concept shows how Lyapunov-like functions can be used to characterize whether the long-term behavior of a system is ultimately bounded. This idea was formalized into a stability concept for forced systems known as *input-to-state* stability or ISS. In particular, we now consider the system

$$\dot{x}(t) = f(x(t), w(t))$$

where $w$ is an exogenous disturbance and $f(0,0) = 0$. So the origin of the unforced system is an equilibrium point. We assume that $w$ is a bounded piecewise continuous function of time. This system is said to be *input-to-state stable* or ISS if there exists a class $\mathcal{KL}$ function $\beta$ and a class $\mathcal{K}_\infty$ function $\gamma$ such that for any initial state $x(0) = x_0 \in \mathbb{R}^n$ the corresponding state trajectory, $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, for any $w \in \mathcal{L}_\infty$ satisfies the inequality

$$|x(t)| \leq \beta(|x_0|, t) + \gamma(\|w\|_{\mathcal{L}_\infty}) \tag{54}$$

for all $t \geq 0$.

Since $\beta > 0$ and $\gamma > 0$, one can readily see that

$$\max(\beta, \gamma) \leq \beta + \gamma \leq 2\max(\beta, \gamma)$$

So an alternative and equivalent characterization of ISS is that the state trajectory satisfy

$$|x(t)| \leq \max\left(\beta(|x_0|, t), \gamma(\|w\|_{\mathcal{L}_\infty})\right) \tag{55}$$

Through these lectures we will use both ISS definitions interchangeably.

Fig. 3 provides a graphical view of what the condition in equation (55) means. In particular, it says there are two terms; one that bounds the initial transient decay of the system by a class $\mathcal{KL}$ function and an ultimate bound that is a function of the amplitude of the disturbance, $w$. To be ISS means that any trajectory lies below the maximum of these two bounding functions.

There is a strong similarity between the ISS and UUB concepts. This suggests that one can probably develop a Lyapunov-like certificate for ISS as is stated and proven below.

THEOREM 23. **ISS Lyapunov Function:** *If there exists a $C^1$ function $V : \mathbb{R}^n \to \mathbb{R}$ with class $\mathcal{K}_\infty$ functions $\underline{\alpha}, \overline{\alpha}$, and $\alpha$ and class $\mathcal{K}$ function $\rho$ such that*

$$\begin{aligned} \underline{\alpha}(|x|) &\leq V(x) \leq \overline{\alpha}(|x|) \\ \dot{V} &\leq -\alpha(|x|), \quad \textit{for } |x| > \rho(\|w\|_{\mathcal{L}_\infty}) \end{aligned}$$

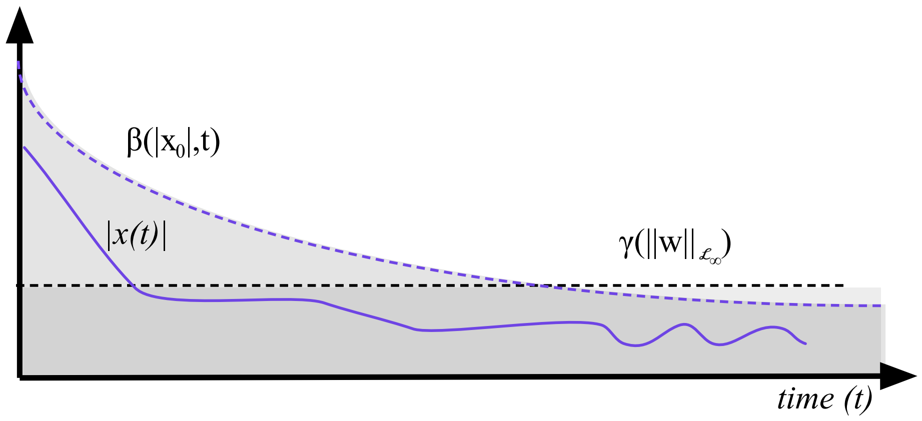


FIGURE 3. Input-to-State Stability (ISS)

*then the system is $ISS$. A function $V$ that satisfies the above conditions is called an* ISS-Lyapunov *function.*

**Proof:** The UUB theorem implies there exists a $\mathcal{KL}$ function $\beta$ and $T \geq 0$ such that

$$|x(t)| \leq \beta(|x_0|, t)$$

for $0 \leq t < T$ and the ultimate bound is

$$|x(t)| \leq \underline{\alpha}^{-1}(\overline{\alpha}(r))$$

for $T > t$ where $r = \rho(\|w\|_{\mathcal{L}_\infty})$. Note that $\gamma = \underline{\alpha}^{-1} \circ \overline{\alpha} \circ \rho$ is class $\mathcal{K}$ by our earlier facts about comparison functions and so

$$|x(t)| \leq \max(\beta(|x_0|, t), \gamma(\|w\|_{\mathcal{L}_\infty})) \leq \beta(|x_0|, t) + \gamma(\|w\|_{\mathcal{L}_\infty})$$

which establishes the system is ISS. ♢

There is a useful alternative test in which the Lyapunov-like function $V$ satisfies a dissipative inequality. In some applications, this second characterization of the ISS-Lyapunov function is easier to certify.

THEOREM 24. **ISS Lyapunov Function - dissipative form:**. *Consider the system $\dot{x} = f(x, w)$ where $f$ is Lipschitz and $w \in \mathcal{L}_\infty$. A $C^1$ function $V : \mathbb{R}^n \to \mathbb{R}$ is an ISS-Lyapunov function for this system if and only if there exist $\mathcal{K}_\infty$ functions*

$\underline{\alpha}, \overline{\alpha}, \alpha$ *and class* $\mathcal{K}$ *function* $\gamma$ *such that*

$$\begin{aligned} \underline{\alpha}(|x|) &\leq V(x) \leq \overline{\alpha}(|x|) \\ \dot{V} &\leq -\alpha(|x|) + \gamma(|w|) \end{aligned}$$

*for all* $x \in \mathbb{R}^n$ *and all* $w \in \mathbb{R}^m$.

## 3. $\mathcal{L}_p$ Stability:

Lyapunov stability and input-to-state stability are all defined with respect to an equilibrium at the origin. Lyapunov stability is concerned with whether the state of the unforced system can be kept arbitrarily close to the origin. ISS is concerned with measuring how far away from the origin the state can get when the system is forced by an exogenous disturbance. We also know, however, that we can define stability in terms of the input/output behavior of the system, without any reference to the system's internal states. In particular, we say a forced system is input/output stable if all bounded inputs to the system result in a bounded output. Bounded, in this case, means that the input and output signals have finite norms and so with this stability concept, the system is viewed as an operator (potentially nonlinear) between the input and output signal spaces. This section formally develops the notion of $\mathcal{L}_p$ stability, an input/output stability concept where the input/output signal spaces have been augmented with an $\mathcal{L}_p$ norm.

Rather than think of the system, $\mathbf{G}$, as a map between two $\mathcal{L}_p$ spaces, we define it as an operator between two *extended* $\mathcal{L}_p$ spaces. In particular, $\mathcal{L}_{\rm pe}$ is the space of all functions, $w$, such that the truncation of $w$ for any finite time $T$

$$w_T(t) = \begin{cases} w(t) & \text{for } t \leq T \\ 0 & \text{otherwise} \end{cases}$$

is in $\mathcal{L}_p$. We say this space is "extended" because it contains all signals in $\mathcal{L}_p$ as well as unbounded signals whose truncations to a finite time interval $[0, T)$ are bounded. For such systems, we say $\mathbf{G}$ is $\mathcal{L}_p$ stable if and only if there exists a class $\mathcal{K}$ function, $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ and a non-negative constant, $\beta$, such that

$$\|\mathbf{G}[w]_T\|_{\mathcal{L}_p} \leq \alpha(\|w_T\|_{\mathcal{L}_p}) + \beta \tag{56}$$

for any $w \in \mathcal{L}_{\rm pe}$ and $T \geq 0$. We refer to the constant $\beta$ as a *bias* and $\alpha$ as a gain function.

Note that in many cases we can bound the action of the gain function, $\alpha$ by a linear function of the inputs norm. This allows us to refine our stability concept to that of *finite-gain $\mathcal{L}_p$ stability*. In particular, we say that $\mathbf{G} : \mathcal{L}_{\rm pe} \to \mathcal{L}_{\rm pe}$ is finite-gain $\mathcal{L}_p$ stable if there exists a $\gamma > 0$ such that

$$\|\mathbf{G}[w]_T\|_{\mathcal{L}_p} \leq \gamma \|w_T\|_{\mathcal{L}_p} + \beta$$

Note that this definition can be refined further by introducing the notion of the operator's $\mathcal{L}_p$-induced gain

$$\|\mathbf{G}\|_{\mathcal{L}_p} \stackrel{\rm def}{=} \inf \left\{ \gamma \,:\, \|(\mathbf{G}[w])_T\|_{\mathcal{L}_p} \leq \gamma \|w_T\|_{\mathcal{L}_p} + \beta, \quad \text{for all } w \in \mathcal{L}_p \text{ and } T \geq 0 \right\}$$

There are formulae that can be used to estimate the $\mathcal{L}_2$ and $\mathcal{L}_\infty$ induced gain of LTI systems. If the operator is a nonlinear map, we can use the Hamilton-Jacobi inequality (HJI) to bound the operator's $\mathcal{L}_2$-induced gain.

THEOREM 25. **Hamilton-Jacobi Inequality:** *Consider the time-invariant system*

$$\begin{aligned} \dot{x} &= f(x) + g(x)w \\ y &= h(x) \end{aligned}$$

*with $f(0) = 0$ and $h(0) = 0$. Let $\gamma$ be a positive constant and suppose there exists a $C^1$ positive semi-definite function $V : \mathbb{R}^n \to \mathbb{R}$ such that*

$$\frac{\partial V}{\partial x} f(x) + \frac{1}{2\gamma^2} \frac{\partial V}{\partial x} g(x) g^T(x) \left( \frac{\partial V}{\partial x} \right)^T + \frac{1}{2} h^T(x) h(x) \leq 0 \tag{57}$$

*Then the system is finite gain $\mathcal{L}_2$ stable with a gain less than or equal to $\gamma$.*

**Proof:** We prove this using a completing the square argument on $\dot{V}$. This means

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x} f + \frac{\partial V}{\partial x} g w \\ &= -\frac{1}{2}\gamma^2 \left| w - \frac{1}{\gamma^2} g^T \left[ \frac{\partial V}{\partial x} \right]^T \right|^2 + \frac{\partial V}{\partial x} f + \frac{1}{2\gamma^2} \frac{\partial V}{\partial x} g(x) g^T(x) \left[ \frac{\partial V}{\partial x} \right]^T + \frac{1}{2}\gamma^2 |w|^2 \end{aligned}$$

The theorem's assumption means that

$$\frac{\partial V}{\partial x} f + \frac{1}{2\gamma^2} \frac{\partial V}{\partial x} g g^T \left[ \frac{\partial V}{\partial x} \right]^T < -\frac{1}{2} h^T(x) h(x)$$

Inserting this into our expression for $\dot{V}$ and using the fact that $y = h(x)$ gives

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x} f + \frac{\partial V}{\partial x} g w \\ &\leq \frac{1}{2}\gamma^2 |w|^2 - \frac{1}{2}|y|^2 - \frac{1}{2}\gamma \left| u - \frac{1}{\gamma^2} g^T \left[\frac{\partial V}{\partial x}\right]^T \right|^2 \\ &\leq \frac{1}{2}\gamma^2 |w|^2 - \frac{1}{2}|y|^2 \end{aligned}$$

Note that this allows us to infer that

$$V(x(T)) - V(x_0) \leq \frac{1}{2}\gamma^2 \int_0^T |w(t)|^2 dt - \frac{1}{2}\int_0^T |y(t)|^2 dt$$

Since $V(x(T)) \geq 0$ this implies

$$\int_0^T |y(t)|^2 dt \leq \gamma^2 \int_0^T |w(t)|^2 dt + 2V(x_0)$$

Taking the square root of both sides and using the fact that $\sqrt{a^2 + b^2} \leq a + b$ when $a, b \geq 0$, we can conclude

$$\|y_T\|_{\mathcal{L}_2} \leq \gamma \|w_T\|_{\mathcal{L}_2} + \sqrt{2V(x_0)}$$

which means the system is finite gain $\mathcal{L}_2$ stable. $\diamondsuit$

## 4. Dissipative and Passive Systems

Dissipativity and Passivity are stability-like concepts for input/output systems. These concepts rely on the physical intuition that systems which dissipate ”energy” are inherently stable. This is useful since we know what the kinetic and potential energy functions are for a mechanical/electrical systems and so it is relatively easy to determine a useful Lyapunov-like ”storage function” to certify if a system is passive.

To provide a concrete interpretation of passivity, let us consider the electrical network shown on the left side of Fig. 4 where $u : \mathbb{R} \rightarrow \mathbb{R}$ is an applied voltage and $y : \mathbb{R} \rightarrow \mathbb{R}$ is the current injected into the circuit. We view this circuit as a dynamical system whose input is $u$ (the applied voltage) and whose output is $y$ (the injected current). The *instantaneous power*, $p : \mathbb{R} \rightarrow \mathbb{R}$ injected into the network is given by $p(t) = u(t)y(t)$. If $p(t) \geq 0$, then the convention is that energy is being *absorbed* by or *delivered to* the network. If $p(t) < 0$,then the network is acting as an energy source and is *delivering* power to the attached source. The circuit is said to be *passive* if the rate of change of the energy stored within the system is less than the power delivered to the network.

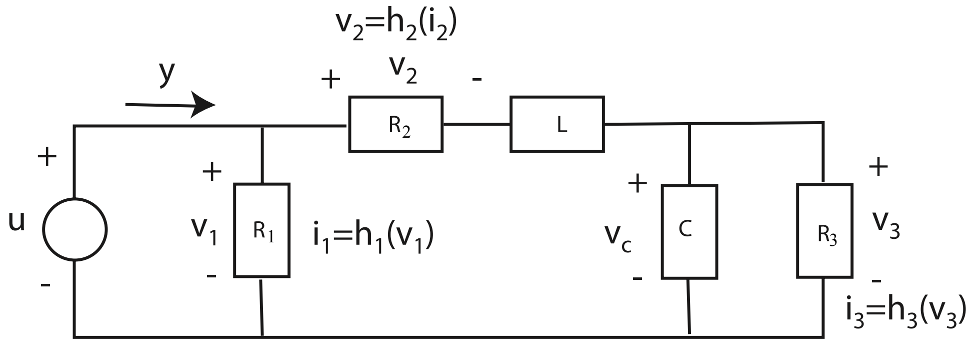


FIGURE 4. RLC Circuit

The specific RLC circuit in Fig. 4 provides a more concrete example of the passivity concept. In this circuit the current through the inductor is $i_2$ and the voltage across the capacitor is $v_c$. Since the inductor and capacitor are the only energy storage devices in the circuit, we treat $i_2$ and $v_c$ as state variables that we denote as $x_1$ and $x_2$, respectively. With these conventions the state equations for the circuit become

$$\begin{aligned} L\dot{x}_1 &= u - h_2(x_1) - x_2 \\ C\dot{x}_2 &= x_1 - h_3(x_2) \\ y &= h_1(u) + x_1 \end{aligned}$$

where $h_1 : \mathbb{R} \to \mathbb{R}$ is a nonlinear admittance function for the current through the nonlinear resistor $R_1$, $h_2$ is the nonlinear impedance function for the voltage across the resistor $R_2$, and $h_3$ is the nonlinear admittance function for the current across resistor $R_3$.

The electrical energy stored within this circuit is

$$V(x) = \frac{1}{2}Lx_1^2 + \frac{1}{2}Cx_2^2$$

namely, the energy stored in both the inductor and the capacitor. Our preceding "heuristic" notion of passivity requires the total energy injected into the system by the source be greater than what is actually stored in the system. That injected energy is the integral of the instantaneous power and so passivity requires

$$\int_0^t u(s)y(s)ds \geq V(x(t)) - V(x(0))$$

The energy $V : \mathbb{R}^2 \rightarrow \mathbb{R}$ that we just defined above is called a *storage function* and it represents how much energy is stored in the system.

If $V$ is a $C^1$ function, then this above relation is equivalent to $\dot{V}(x(t)) \leq u(t)y(t)$ thereby providing a "differential" characterization of system passivity. For the particular network in Fig. 4 we can see that

$$\begin{array}{rcl}\dot{V} & = & Lx_1\dot{x}_1 + Cx_2\dot{x}_2 \\ & = & x_1(u - h_2(x_1) - x_2) + x_2(x_1 - h_3(x_2)) \\ & = & x_1(u - h_2(x_2)) - x_2h_3(x_2) \\ & = & (x_1 + h_1(u))u - (uh_1(u) + x_1h_2(x_1) + x_2h_3(x_2)) \\ & = & uy - \text{positive definite term} \leq uy\end{array}$$

This last inequality implies that $\dot{V} \leq uy$ where $uy$ is the instantaneous power injected into the circuit from the source. Based on the preceding heuristic notion of passivity we can therefore assert that this is a passive circuit.

The notion of "passivity" seen in the example can be generalized as the notion of *dissipativity*. Consider a dynamical system of the form

$$\begin{array}{rcl}\dot{x} & = & f(x,u) \\ y & = & h(x,u)\end{array} \tag{58}$$

where $f(0,0) = 0$, $h(0,0) = 0$ with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and output $y \in \mathbb{R}^p$. Associated with this system we define a function $r : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ called the *supply rate*. The system is *dissipative* with respect to supply rate $r(u,y)$ if there exists a function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ with $V(x) \geq 0$ such that

$$V(x(t)) - V(x(0)) \leq \int_0^t r(u(s), y(s))ds$$

for all $t \geq 0$. The function $V$ is called a *storage function* and if $V$ is smooth enough then we can recast the preceding integral as a *dissipative inequality* of the form $\dot{V}(x(t)) \leq r(u(t), y(t))$. Passivity is a special case of dissipativity in which the supply rate is $r(u,y) = u^Ty$. In our following discussion we'll confine our attention to passivity with the understanding that all of our results also apply to dissipative systems with minor tweaks.

## 5. Relationship Between Stability Concepts

We have introduced a number of stability concepts that are often used in the design of nonlinear controllers. It is important for us to understand how these stability concepts are related to each other. This section states (without proof) several theorems relating Lyapunov stability to ISS, $\mathcal{L}_p$ stability, and passivity. The first theorem below asserts that a system with an exponentially (Lyapunov) stable equilibrium is in fact finite-gain $\mathcal{L}_p$ stable.

THEOREM 26. **Exponential Stability implies $\mathcal{L}_p$ stability:** *Consider the state-based input-output system $\dot{x}(t) = f(x, w)$ with $y(t) = h(x, w)$ where the origin is an exponentially stable equilibrium of the unforced system $\dot{x} = f(x, 0)$. Assume there exist positive constants $L, r, r_2, \eta_1$, and $\eta_2$ such that*

$$\begin{aligned} |f(x,w) - f(x,0)| &\leq L[w] \\ |h(x,w)| &\leq \eta_1|x| + \eta_2|w| \end{aligned}$$

*for all $|x| < r$ and $|w| < r_w$. If there exists a $C^1$ function $V : \mathbb{R}^n \to \mathbb{R}$ and non-negative constants $c_1, c_2, c_3$ and $c_4$ such that*

$$\begin{aligned} c_1|x|^2 &\leq V(x) \leq c_2|x|^2 \\ \dot{V}(x,0) &\leq -c_3|x|^2 \\ \left\| \frac{\partial V}{\partial x} \right\| &\leq c_4|x| \end{aligned}$$

*then the system is finite gain $\mathcal{L}_p$-stable.*

We can also relate input-to-state stability and $\mathcal{L}_p$ stability. This is done in the following theorem which is also stated without proof.

THEOREM 27. **ISS implies $\mathcal{L}_\infty$ stability:** *Consider the input-output system*

$$\begin{aligned} \dot{x} &= f(t,x,w), \quad x(0) = x_0 \\ y(t) &= h(t,x,w) \end{aligned}$$

*where the origin, $x = 0$, is an exponentially stable equilibrium of $\dot{x} = f(t, x, 0)$. Let $f$ be piecewise continuous in $t$ and locally Lipschitz in $x$ and $w$. Let $h$ be piecewise continuous in $t$ and continuous in $x$ and $w$. Suppose $\dot{x} = f(t, x, w)$ is locally ISS and assume there exist class $\mathcal{K}$ functions $\alpha_1, \alpha_2$ and a non-negative constant $\eta_3$ such that*

$$|h(t, x, w)| \leq \alpha_1(|x|) + \alpha_2(|w|) + \eta_3$$

*Then there is a constant $k_1 > 0$ such that for all initial conditions with $|x_0| < k_1$, the system is $\mathcal{L}_\infty$-stable.*

One of the basic issues we must address is the relationship between passivity and Lyapunov stability. In general, passivity does not imply Lyapunov stability. The reason for this is that the storage function $V$ is only required to be *positive semi-definite*, not positive definite. This means that in the presence of an unobservable part of the system, one can still have the origin unstable, and yet the system will be passive. For passivity to imply Lyapunov stability, we need to impose detectability conditions that limit the ability of the unobservable parts of the system becoming unstable.

We now examine when passivity implies Lyapunov stability. As discussed earlier, because the storage function is only positive semidefinite, we need to enforce detectability assumptions that ensure the unobservable part of the system remains stable. These detectability conditions are known as zero-state detectability (ZSD) and zero-state observability, both of which are defined in the following paragraph.

Now consider the system $G$ with zero input so that $\dot{x} = f(x, 0)$ and $y = h(x, 0)$ and let $Z \subset \mathbb{R}^n$ be the largest invariant set contained in $\{x \in \mathbb{R}^n : h(x, 0) = 0\}$. We say that $G$ is *zero-state detectable* if $x = 0$ is asymptotically stable whenever $x_0 \in Z$. In other words, any other perturbation of the origin confined to $Z$ will asymptotically converge to the origin. If the $Z = \{0\}$, then we say $G$ is zero state observable (ZSO). The ZSD property is essential for the stability of a passive system's equilibrium. This fact is stated in the following theorem. The proof relies on the invariance principle.

THEOREM 28. **(Passivity and Stability)** *Let the system $G$ in equations (58) be passive with a $C^1$ storage function $V$.*

- *If $V$ is positive definite then the origin of $G$ when $u = 0$ is Lyapunov stable.*
- *If $G$ is ZSD and $V$ is positive semidefinite, then the equilibrium $0$ of $G$ with $u = 0$ is Lyapunov stable.*
- *Furthermore if there is no direct throughput of $u$ in $y = h(x, u)$, then the feedback $u = -ky$ for any $k > 0$ ensures the origin is asymptotically stable if and only if $G$ is ZSD.*

There are a couple of things to remark about theorem 28. The first is that the ZSD property plays a pivotal role in assuring that passivity implies Lyapunov stability. The second thing to notice is that if we have the ZSD property, then we can only assure the origin is asymptotically stable if we apply output feedback $-ky$ about the system. So feedback is critical in establishing asymptotic stability of passive systems, but moreover, this feedback is high-gain feedback that only uses the system output. This has the potential of greatly simplifying the feedback controller design.

**Example:** Consider the following system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -ax_1^3 - kx_2 + u \\ y &= x_2 \end{aligned}$$

where $a, k > 0$. Let us consider the following storage function

$$V(x) = \frac{1}{4}ax_1^4 + \frac{1}{2}x_2^2$$

This is clearly PD, so let us compute its directional derivative

$$\begin{aligned} \dot{V} &= ax_1^3x_2 + x_2(-ax_1^3 - kx_2 + u) \\ &= -ky^2 + uy \leq uy \end{aligned}$$

So this system is *passive* and because $V > 0$, we know the origin is Lyapunov stable. We can actually go one step further and assert that it is asymptotically stable in the following way. Note that $\dot{V} = (u - ky)y$ where we can think of $-ky$ as a feedback control input. Since $k > 0$, this would mean the origin is asymptotically stable provided it is ZSD. In particular, this system is ZSO so if we consider the set of states when $y(t) \equiv 0$ for all time, we can readily see this is true if and only if $x_2(t) \equiv 0$ for all time. But this would mean that $\dot{x}_2 = -ax_1^3$ when $u = 0$, which would mean $x_1(t) \equiv 0$ for all time, and so this system is ZSO because the only state that generates an zero output trajectory ($y(t) = 0$) is the the state trajectory $x(t) = 0$.

## 6. Stability of Feedback Interconnections:

One of the most useful aspects of an input-output system is that they can be interconnected. The output of one system can be used as the input to another system to form a larger system with more desirable properties. This section reviews results that determine when a feedback interconnection satisfies an input-output stability concept such as ISS, $\mathcal{L}_p$ stability, or passivity. These results are important for they provide the foundation for constructive approaches to nonlinear control system design [Sepulchre et al. (2012)].

We first investigate input-to-state stability (ISS) of a feedback interconnection of two ISS systems [Jiang et al. (1994)]. Fig. 5 shows the interconnection of two ISS systems

$$
\begin{array}{rcll}
\dot{x}_1 & = & f_1(x_1, x_2), & x_1(0) = x_{10} \\
\dot{x}_2 & = & f_2(x_1, x_2, u), & x_2(0) = x_{20}
\end{array}
\tag{59}
$$

where $x_2$ is the input to the first system with vector field $f_1$ and $(x_1, u)$ are the inputs to the second system with vector field $f_2$. The initial conditions for the first and second system are $x_{10}$ and $x_{20}$, respectively. The following theorem presented without proof is due to Jiang et al. (1994). It asserts that the feedback interconnect of two ISS stable systems is also ISS provided the composition of their gain functions $\gamma_1 \circ \gamma_2$ is contractive.

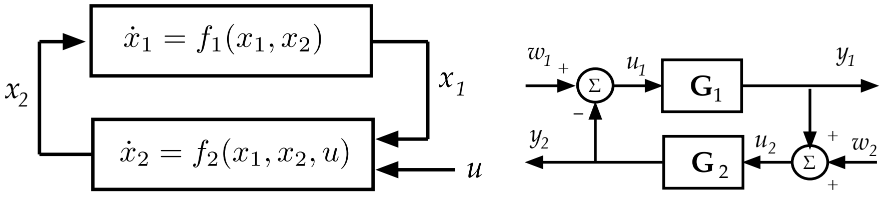


FIGURE 5. (left) Feedback Interconnection of ISS Systems (right) Feedback Interconnection of $\mathcal{L}_p$ stable systems

THEOREM 29. **(ISS Small Gain Theorem)** *Consider the interconnected system in equation (59) where $f_1(0,0) = 0$ and $f_2(0,0,0) = 0$. Assume that the first system is ISS with respect to input $x_2$ so that when $x_2 \in \mathcal{L}_\infty$, there exist class $\mathcal{KL}$ function $\beta_1$ and class $\mathcal{K}$ function $\gamma_1$ such that*

$$|x_1(t)| \leq \max\left\{\beta_1(|x_{10}|, t), \gamma_1(\|x_2\|_{\mathcal{L}_\infty})\right\}$$

*Assume that the second system is ISS with respect to inputs $x_1$ and $u$ so that for any $x_1, u \in \mathcal{L}_\infty$ there exist class $\mathcal{KL}$ function $\beta_2$ and class $\mathcal{K}$ functions $\gamma_2$ and $\gamma_u$ such that*

$$|x_2(t)| \leq \max\left\{\beta_2(|x_{20}|, t), \gamma_2(\|x_1\|_{\mathcal{L}_\infty}, \gamma_u(\|u\|_{\mathcal{L}_\infty})\right\}$$

*If for all $r > 0$, we can verify that $\gamma_1(\gamma_2(r)) < r$, then the interconnected system is ISS with respect to input $u$.*

**Example:** Let us consider the following system

$$\begin{aligned}\dot{x}_1 &= -x_1^3 + x_1 x_2 \\ \dot{x}_2 &= a x_1^2 - x_2 + u\end{aligned}$$

where $a$ is a real parameter. We can regard this as the feedback interconnection of two scalar systems. For the upper system

$$\dot{x}_1 = f_1(x_1, x_2) = -x_1^3 + x_1 x_2$$

we view $x_1$ as the state and $x_2$ as the input. Consider the ISS-certificate

$$V(x_1) = \frac{1}{2}x_1^2$$

and its directional derivative with respect to the upper system is

$$\dot{V} = \frac{\partial V}{\partial x_2} f_1(x_1, x_2) \leq -|x_1|^4 + |x_1|^2 |x_2|$$

So choose $0 < \theta < 1$ and redistribute the negative definite term to obtain

$$\dot{V} \leq -(1-\theta)|x_1|^4 - \theta|x_1|^4 + |x_1|^2|x_2|$$

For $x_1$ such that

$$(1-\theta)|x_1|^2 \geq |x_2|$$

we can see that

$$\dot{V} \leq -\alpha(|x_1(t)|)$$

where $\alpha(r) = \theta r^4$, which is clearly a class $\mathcal{K}_\infty$ function. So $\dot{V} \leq -\alpha(|x|)$ when $|x_1| \geq \rho(|x_2|) = \sqrt{\frac{|x_2|}{1-\theta}}$ which shows that $f_1$ is ISS with respect to $x_2$.

For the lower system

$$\dot{x}_2 = f_2(x_1, x_2, u) = ax_1^2 - x_2 + u$$

we let $x_2$ be the state and $(x_1, u)$ be the inputs. The candidate ISS-certificate will be $V(x_2) = \frac{1}{2}x_2^2$ whose directional derivative with respect to $f_2$ is

$$\dot{V} = \frac{\partial V}{\partial x_2} f_2(x_1, x_2, u) \leq |x_2|(|a||x_1|^2 - |x_2| + |u|)$$

Again select $0 < \theta < 1$ and redistribute the negative definite term to rewrite $\dot{V}$ as

$$\dot{V} = -(1-\theta)|x_2|^2 - \theta|x_2|^2 + (a|x_1|^2 + |u|)|x_2|$$

So if $x_2$ satisfies

$$(1-\theta)|x_2| \geq |a||x_1|^2 + |u|$$

then we can conclude $\dot{V} \leq -\alpha(|x_2|)$ where $\alpha(r) = \theta r^2$ is also $\mathcal{K}_\infty$. In particular this means that if

$$|x_2| \geq \max\{\gamma_2(|x_1|), \gamma_u(|u|)\}$$

where $\gamma_2(r) = \frac{2|a|r^2}{1-\theta}$ and $\gamma_u(r) = \frac{2r}{1-\theta}$, then $\dot{V} \leq -\alpha(|x_2|)$. This is sufficient to show that the lower system is ISS with respect to $x_1$ and $u$.

So we now have the gains for the two ISS systems. The gain of the upper system is $\gamma_1(r) = \sqrt{\frac{r}{1-\theta}}$. The gain for the lower system is $\gamma_2 = \frac{2|a|r^2}{1-\theta}$. We now check the small gain condition in theorem 29 to obtain

$$\gamma_2(\gamma_1(r)) = \frac{2|a|}{1-\theta}\frac{r}{1-\theta} = \frac{2|a|r}{(1-\theta)^2}$$

which if $|a| < \frac{1}{2}$ shows that $\gamma_2(\gamma_1(r)) < r$ and so for this range of $a$ the small gain condition is satisfied and we can conclude the full system is ISS.

We now turn to a similar small gain result that holds for the feedback interconnection of a pair of $\mathcal{L}_p$ stable systems. The system under consideration is shown in Fig. 5 where there are two systems $G_1 : \mathcal{L}_{pe} \to \mathcal{L}_{pe}$ and $G_2 : \mathcal{L}_{pe} \to \mathcal{L}_{pe}$. The $\mathcal{L}_p$ small gain theorem is similar to the ISS small gain theorem in that the $\mathcal{L}_p$ stability of the interconnected system is guaranteed if the product of the $\mathcal{L}_p$ gains of the subsystems is less than one. The proof of this theorem uses a different technique than was used for the ISS small gain theorem. The proof relies on an application of the Banach contraction mapping principle

THEOREM 30. **Contraction Mapping Principle:** *Let $X$ be a Banach space and let $\mathbf{G} : X \to X$ be a contraction mapping. This means there is $0 \leq \gamma < 1$ such that*

$$\|\mathbf{G}[x] - \mathbf{G}[y]\| \leq \gamma \|x - y\|$$

*Then there exists a unique element $x^* \in X$ such that $x^* = \mathbf{G}[x^*]$.*

One may prove the contraction mapping principle by showing that the recursive equation

$$x_{k+1} = \mathbf{G}[x_k]$$

generates a Cauchy sequence of functions when $0 < \gamma < 1$. In Banach spaces all Cauchy sequences converge to a unique element of the space, so the result follows easily from standard methods in real analysis [Rudin (1964)]. The $\mathcal{L}_p$ small gain uses this fact to establish the existence of an internal signal for the loop which has to belong to $\mathcal{L}_p$.

THEOREM 31. **($\mathcal{L}_p$ Small Gain Theorem)** *Consider the interconnection shown in Fig. 5 of two systems $G_1 : \mathcal{L}_{pe} \to \mathcal{L}_{pe}$ and $G_2 : \mathcal{L}_{pe} \to \mathcal{L}_{pe}$ where both subsystems are finite gain $\mathcal{L}_p$ stable. This means, therefore, that there exist non-negative constants $\gamma_1$, $\beta_1$, $\gamma_2$, and $\beta_2$ such that*

$$\begin{aligned} \|y_{1T}\|_{\mathcal{L}_p} &\leq \gamma_1 \|u_{1T}\|_{\mathcal{L}_p} + \beta_1 \\ \|y_{1T}\|_{\mathcal{L}_p} &\leq \gamma_2 \|u_{2T}\|_{\mathcal{L}_p} + \beta_2 \end{aligned}$$

*for any $T > 0$. Then the interconnected system is finite gain $\mathcal{L}_p$-stable if $\gamma_1\gamma_2 < 1$.*

**Proof:** Consider the operator $S_2 : \mathcal{L}_{pe} \to \mathcal{L}_{pe}$ defined by the equation

$$S_2[u_{2T}] = w_{2T} + (G_1[w_{1T} + (G_2[u_{2T}])_T])_T$$

Consider two $\mathcal{L}_p$ signals $u_{2T}$ and $\hat{u}_{2T}$ and examine the $\mathcal{L}_p$ norm of the difference between $S_2[u_{2T}]$ and $S_2[\hat{u}_{2T}]$. This consideration yields,

$$\begin{aligned} \|S_2[u_{2T}] - S_2[\hat{u}_{2T}]\|_{\mathcal{L}_p} &= \|G_1[w_{1T} + (G_2[u_{2T}])_T] - G_1[w_{1T} + (G_2[\hat{u}_{2T}])_T]\|_{\mathcal{L}_p} \\ &\leq \gamma_1 \|G_2[u_{2T}] - G_2[\hat{u}_{2T}]\|_{\mathcal{L}_p} + \beta_1 \\ &\leq \gamma_1\gamma_2 \|u_{2T} - \hat{u}_{2T}\|_{\mathcal{L}_p} + \beta_1 + \beta_2 \end{aligned}$$

By assumption $\gamma_1\gamma_2 < 1$, so that $S_2$ is a contraction mapping and we can infer the existence of a unique $\mathcal{L}_{pe}$ function $u_2$ such that $u_2 = S_2[u_2]$. A similar argument establishes the existence of a unique $u_1 \in \mathcal{L}_{pe}$ that satisfies

$$u_1 = S_1[u_1] = w_1 + G_2[w_2 + G_1[u_1]]$$

Since the loop in Fig. 5 is well-posed (i.e. the internal signals, $u_1$ and $u_2$, exist in $\mathcal{L}_{pe}$), we can now look at the $\mathcal{L}_p$ norm of the internal signals. Since $G_1$ and $G_2$ are finite gain $\mathcal{L}_p$ stable, we can see for $u_1$ that

$$\begin{aligned} \|u_{1T}\|_{\mathcal{L}_p} &\leq \|w_{1T}\|_{\mathcal{L}_p} + \|(G_2[u_{2T}])_T\|_{\mathcal{L}_p} \\ &\leq \|w_{1T}\|_{\mathcal{L}_p} + \gamma_2\|u_{2T}\|_{\mathcal{L}_p} + \beta_2 \\ &\leq \|w_{1T}\|_{\mathcal{L}_p} + \gamma_2(\|w_{2T}\|_{\mathcal{L}_p} + \gamma_1\|u_{1T}\|_{\mathcal{L}_p} + \beta_1) + \beta_2 \\ &= \gamma_1\gamma_2\|w_{1T}\|_{\mathcal{L}_p} + (\|u_{1T}\|_{\mathcal{L}_p} + \gamma_2\|w_{2T}\|_{\mathcal{L}_p} + \beta_2 + \gamma_2\beta_1) \end{aligned}$$

solving for $\|u_{1T}\|_{\mathcal{L}_p}$ in the above inequality yields,

$$\|u_{1T}\|_{\mathcal{L}_p} \leq \frac{1}{1-\gamma_1\gamma_2}\left(\|w_{1t}\|_{\mathcal{L}_p} + \gamma_2\|w_{2T}\|_{\mathcal{L}_p} + \beta_2 + \gamma_2\beta_1\right)$$

and a similar argument shows that

$$\|u_{2T}\|_{\mathcal{L}_p} \leq \frac{1}{1-\gamma_1\gamma_2}\left(\|w_{2T}\|_{\mathcal{L}_p} + \gamma_1\|w_{1T}\|_{\mathcal{L}_p} + \beta_1 + \gamma_2\beta_2\right)$$

which means the $\mathcal{L}_p$ norm of both signals is finite and so the interconnected system is finite gain $\mathcal{L}_p$ stable. $\diamondsuit$

An important application of the small gain theorem is seen in the study of robust stability conditions for LTI systems. Recall that from chapter 3 we developed a robust stability condition for a multiplicatively perturbed one-parameter feedback system that relied on the perturbation $\boldsymbol{\Delta}$ being in $\mathcal{RH}_\infty$. Clearly in most physical applications that uncertainty may arise from neglected dynamics that are most probably not linear, so the proof could not be used. But the small gain theorem does not require $\mathbf{\Delta}$ to be linear. Working in this way we can re-derive the robust stability condition from chapter 3 by simply requiring that the composition of the uncertainty and the closed loop map is contractive.

**Passivity Theorem for Feedback Interconnections:** We now turn to study the passivity of feedback interconnects. This result differs significantly from the prior small-gain results in that it asserts that any feedback interconnect of passive systems is again passive. In particular, we see that passivity is preserved under feedback interconnections. This is useful in studying large complex networked dynamical systems [Moylan and Hill (1978)] that consist solely of feedback interactions. It suggests that if you have a large-scale passive system, then the feedback interconnection of the large-scale system with another passive system does not destroy the passivity of the larger system. This result, therefore, provides a modular way to build passive networked systems of arbitrarily large scale.

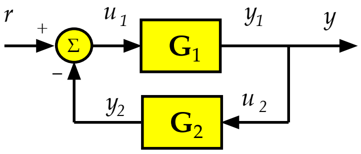


FIGURE 6. Feedback Interconnection of Passive Systems

THEOREM 32. **(Passivity of Feedback Interaconnects)** *Consider the feedback connection shown in Fig. 6 of two systems $G_1$ and $G_2$ with state equations,*

$$\begin{aligned} \dot{x}_i &= f_i(x_i, e_i) \\ y_i &= h_i(x_i, e_i) \end{aligned}$$

*for $i = 1, 2$ where $e_1 = u_1 - y_2$ and $e_2 = u_2 + y_1$. If systems $G_1$ and $G_2$ are passive, then the feedback connection is also passive.*

**Proof:** Let $V_1$ and $V_2$ be storage functions for $G_1$ and $G_2$, respectively. Since these systems are passive, we know

$$\dot{V}_i \leq e_i^T y_i$$

Let $V = V_1 + V_2$ and from our feedback connections we see that

$$\begin{aligned} e_1^T y_1 + e_2^T y_2 &= (u_1 - y_2)^T y_2 + (u_2 + y_1)^T y_2 \\ &= u_1^T y_1 + u_2^T y_2 \end{aligned}$$

This implies that

$$u^T y = u_1^T y_1 + u_2^T y_2 \geq \dot{V}_1 + \dot{V}_2 = \dot{V}$$

and so the feedback system is also passive. ♢

One of the reasons we are interested in the feedback connection being passive is that it can be used to determine if the connection is $\mathcal{L}_2$-stable. The following theorem shows how we can use theorem 32 to establish the $\mathcal{L}_2$ stability of two interconnected passive systems.

THEOREM 33. **($\mathcal{L}_2$-stability of Passive Feedback Systems)** *Consider the feedback connection in Fig. 6 where $G_1$ and $G_2$ are two passive systems with storage functions $V_1$ and $V_2$, respectively such that*

$$e_i^T y_i \geq \dot{V}_i + \epsilon_i e_i^T e_i + \delta_i y_i^T y_i$$

*for $i = 1, 2$. Then the feedback system is finite gain $\mathcal{L}_2$ stable if $\epsilon_1 + \delta_2 > 0$ and $\epsilon_2 + \delta_1 > 0$.*

**Proof:** Let $V = V_1 + V_2$ and note that we can rewrite $\dot{V}$ as

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 \\ &\leq -y^T \left[ \begin{array}{cc} (\epsilon_2 + \delta_1) I & 0 \\ 0 & (\epsilon_1 + \delta_2) I \end{array} \right] y - u^T \left[ \begin{array}{cc} \epsilon_1 I & 0 \\ 0 & \epsilon_2 I \end{array} \right] + u^T \left[ \begin{array}{cc} I & 2\epsilon_1 I \\ -2\epsilon_2 I & I \end{array} \right] \end{aligned}$$

Let $a = \min\{\epsilon_2 + \delta_1, \epsilon_1 + \delta_2\}$, $b = \left\| \left[ \begin{array}{cc} I & 2\epsilon_1 I \\ -2\epsilon_1 I & I \end{array} \right] \right\| \geq 0$, and $c = \left\| \left[ \begin{array}{cc} \epsilon_1 I & 0 \\ 0 & \epsilon_2 I \end{array} \right] \right\| \geq 0$, then

$$\begin{aligned} \dot{V} &\leq -a|y|^2 + b|u||y| + c|u|^2 \\ &= -\frac{1}{2a}(b|u| - a|y|)^2 + \frac{b^2}{2a}|u|^2 - \frac{a}{2}|y|^2 + c|u|^2 \\ &\leq \frac{k^2}{2a}|u|^2 - \frac{a}{2}|y|^2 \end{aligned}$$

where $k^2 = b^2 + 2ac$. Integrating over $[0, T]$ and using the vector hat $V(x) \geq 0$, we obtain

$$\|y_T\|_{\mathcal{L}_2} \leq \frac{k}{a}\|u_T\|_{\mathcal{L}_2} + \sqrt{2V(x(0)/a}$$

which establishes that the feedback interconnection is finite gain $\mathcal{L}_2$ stable with a gain less than $\frac{k}{a}$ and a bias of $\sqrt{2V(x(0))/a}$. ♢

**Example:** Consider the feedback connection

$$\begin{aligned} G_1 &: \begin{cases} \dot{x} = f(x) + g(x)e_1 \\ y_1 = h(x) \end{cases} \\ G_2 &: \; y_2 = ke_2 \end{aligned}$$

with $k > 0$. Suppose a positive definite $V : \mathbb{R}^n \to \mathbb{R}$ exists such that

$$\frac{\partial V}{\partial x} f(x) \leq 0$$

and

$$\frac{\partial V}{\partial x} g(x) = h^T(x)$$

This means that the directional derivative of $V$ is

$$\dot{V} = \frac{\partial V}{\partial x} f(x) + \frac{\partial V}{\partial x} g(x)e_1 \leq y_1^T e_1$$

which means that $G_1$ is passive. The second system is memoryless and so it too is passive. Now note that

$$e_2^T y_2 = ke_2^T e_2 = \gamma k e_2^T e_2 + \frac{1-\gamma}{k} y_2^T y_2$$

So the conditions in theorem 33 are satisfied with $\epsilon_1 = \delta_1 = 0$, $\epsilon_2 = \gamma k$ and $\delta_2 = \frac{1-\gamma}{k}$. So by the preceding theorem, this means the entire interconnected system is finite gain $\mathcal{L}_2$ stable.

We now consider the feedback interconnection of two passive systems when the external input $u = 0$. In this case we want to know whether the resulting interconnection is asymptotically stable. The following theorem provides such conditions.

THEOREM 34. *Consider the feedback connection of two time-invariant dynamical systems where $u = 0$. The origin is asymptotically stable if*

- *both feedback components are strictly passive or*
- *both feedback components are output strictly passive and zero state observable.*

**Proof:** Let $V_1$ and $V_2$ be the storage functions for $G_1$ and $G_2$, respectively. Let $V(x) = V_1(x_1) + V_2(x_2)$ be a candidate Lyapunov function for the closed loop system. In the first case,

$$\dot{V} \leq u^T y - \psi_1(x_1) - \psi_2(x_2) = -\psi_1(x_1) - \psi_2(x_2)$$

with $u = 0$ where $\psi_1$ and $\psi_2$ are positive definite functions. This is sufficient to establish that the origin is asymptotically stable. In the second case,

$$\dot{V} \leq -y_1^T \rho_1(y_1) - y_2^T \rho_2(y_2)$$

where $y_i^T \rho_i(y_i) > 0$ for $i = 1, 2$ and all $y_i \neq 0$. Here $\dot{V}$ is only negative semi-definite and $\dot{V} = 0$ implies $y = 0$. To use the Invariance principle, we need to show that $y(t) = 0$ for all $t$ implies $x(t) = 0$. Note that $y_2(t) = 0$ implies $e_1(t) = 0$ and the zero-state observability of $G_1$ implies that if $y_1(t) = 0$, then $x_1(t) = 0$. A similar argument applies for $G_2$ and so the origin must be asymptotically stable by the Invariance principle. $\diamondsuit$

The proof uses the idea that the sum of the storage functions for the feedback components can be used as a candidate Lyapunov function for the feedback connection. The preceding analysis is restrictive in the sense that for $\dot{V} = \dot{V}_1 + \dot{V}_2 < 0$, we require both $\dot{V}_1 \leq 0$ and $\dot{V}_2 \leq 0$. This is not necessary. One term, $\dot{V}_1$, for instance could be positive as long as $\dot{V}_2$ is sufficiently negative that the sum of both is negative. This idea is exploited in the following examples.

**Example:** Consider the feedback connection

$$G_1 : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -ax_1^3 - kx_2 + e_1 \\ y_1 = x_2 \end{cases} \qquad G_2 : \begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = -bx_3 - x_4^3 + e_2 \\ y_2 = x_4 \end{cases}$$

where $a$,$b$, and $k$ are positive constants. Let $V_1 = \frac{a}{4}x_1^4 + \frac{1}{2}x_2^2$ as the storage function for $H_1$. we obtain

$$\dot{V}_1 \quad = \quad ax_1^3 x_2 - ax_1^3 x_2 - kx_2^2 + x_2 e_1 = -ky_1^2 + y_1 e_1$$

So $H_1$ is output strictly passive. With $e_1 = 0$, we have

$$y_1(t) \equiv 0 \Leftrightarrow x_2(t) \equiv 0 \Rightarrow x_1(t) \equiv 0$$

which shows that $H_1$ is zero-state observable. Using $V_2 = \frac{b}{2}x_3^2 + \frac{1}{2}x_4^2$ as the storage function for $H_2$, we obtain

$$\dot{V}_2 = bx_3x_4 - bx_3x_4 - x_4^4 + x_4e_2 = -y_2^4 + y_2e_2$$

So $H_2$ is strictly output passive and with $e)2$, we have

$$y_2(t) \equiv 0 \Leftrightarrow x_4(t) \equiv 0 \Rightarrow x_3(t) \equiv 0$$

which shows $H_2$ is zero-state observable. Thus by the second case in the above theorem, and the fact that $V_1$ and $V_2$ are radially unbounded, we conclude that the origin is globally asymptotically stable.

## 7. Stability of Cascade Interconnections

Based on the definitions for the various input-output stability concepts, it should be apparent that the parallel composition of two stable systems will preserve that stability. At first glance, one might also think that the cascade (series) connection of any two stable systems will also preserve stability, but this is not always true as can be demonstrated in the following example.

**Example:** Consider the cascaded system where the *driving system* $G_1$ in Fig. 7 has the state space realization

$$\begin{array}{rcl} \dot{\xi}_1 & = & \xi_2 \\ \dot{\xi}_2 & = & -\gamma^2\xi_1 - 2\gamma\xi_2 + u \\ y_1 & = & \xi_2 \end{array} \tag{60}$$

and the *driven system*, $G_2$, has the state space realization

$$\begin{array}{rcl} \dot{\eta} & = & -\frac{1}{2}(1 - y_1)\eta^3 \\ y & = & \eta \end{array} \tag{61}$$

Note that the driving system is a linear system and where $\gamma > 0$. The question is whether the origin of this cascaded system is asymptotically stable.

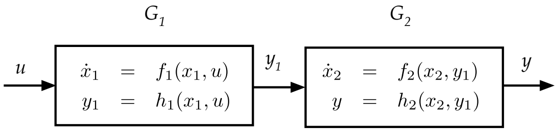


FIGURE 7. Cascade Connection of Two Input-Output Systems

The above system is a cascaded system in which the driving system when $u = 0$ is a linear system of the form

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\gamma^2 & -2\gamma \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

The state transition matrix for this linear system is

$$\Phi(t) = \begin{bmatrix} (1+\gamma t)e^{-\gamma t} & te^{-\gamma t} \\ -\gamma^2 te^{-\gamma t} & (1-\gamma t)e^{-\gamma t} \end{bmatrix}$$

Note that the $(2, 1)$ element has a $\gamma^2$ term so that for large enough $\gamma$, this term may have an extremely large peak. In particular if we let the initial condition for $G_1$ be $\xi_1(0) = 1$ and $\xi_2(0) = 0$, then $\xi_2(t) = -\gamma^2 te^{-\gamma t}$ whose plot is shown on the left side of Fig. 8 for $\gamma = 10$. Note that we see a large negative excursion in $\xi_2$ before it returns to $0$. This phenomena is referred to as *peaking* [Sussmann and Kokotovic (1991)].

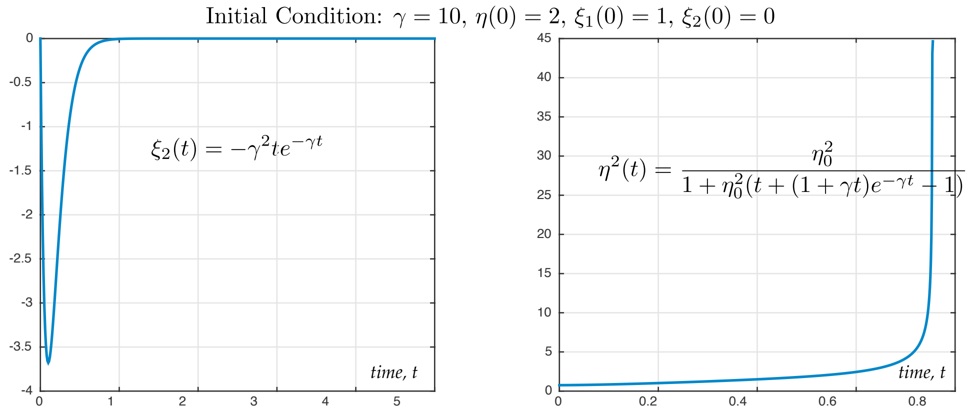


FIGURE 8. Peaking in the driving system triggers a finite escape in the driven system- (left) The state of the driving system, $\xi$, over $5$ seconds - (right) The state of the driven system showing a finite escape at $0.755$ sec.

While $\xi_2$ is negative, we see that $\eta(t)$ will be increasing. In fact if we insert our closed form expression for $\xi_2(t)$ into the differential equation for $G_2$, we get

$$\dot{\eta} = -\frac{1}{2}(1 - \gamma^2 te^{-\gamma t})\eta^3$$

This ODE is separable and we can therefore integrate it to obtain the following

$$\eta^2(t) = \frac{\eta_0^2}{1 + \eta_0^2(t + (1 + \gamma t)e^{-\gamma t} - 1)}$$

Note that the denominator may go to zero for a finite $t$. This would mean that $\eta(t)$ becomes unbounded at a finite time. The specific parameters chosen for Fig. 8 exhibit this finite escape time at $t \approx 0.755$. The right hand pane of the figure shows this finite escape.

What this example shows is that even though both cascaded systems are asymptotically stable when there is no input, the cascade combination of these "stable" systems is unstable. Note that both of these systems are passive, and so clearly, the cascade combination of the two passive systems may not necessarily lead to a stable system. The question is whether the other stability concepts we've introduced (ISS and $\mathcal{L}_p$-stability) also suffer from the same problem. It is relatively easy to show that cascades of ISS or $\mathcal{L}_p$ stable systems will preserve the underlying stability concept. This is one important way in which these other stability concepts differ from passivity.

Finally, it is important to say something about why such cascaded systems are of interest to us. In particular, one important way of synthesizing controllers for nonlinear systems is through a feedback linearization process to be introduced in chapter 5. This linearization automatically generates a cascade of linear systems. Clearly, for this case the stability or stabilizability of such cascades will be an important theme in the development of nonlinear control systems. The fact that the cascade of two $\mathcal{L}_p$-stable system will again be $\mathcal{L}_p$ is formalized in the following theorem.

THEOREM 35. *Consider the cascade connection of a driving system $G_1 : \mathcal{L}_{pe} \rightarrow \mathcal{L}_{pe}$ and a driven system $G_2 : \mathcal{L}_{pe} \rightarrow \mathcal{L}_p$. If $G_1$ and $G_2$ are both finite gain $\mathcal{L}_p$ stable then the cascaded system $G_2G_1$ is also finite gain $\mathcal{L}_p$ stable.*

**Proof:** Since $G_1$ and $G_2$ are both finite gain $\mathcal{L}_p$ stable there exist positive constants $\gamma_1$, $\gamma_2$, $\beta_1$, and $\beta_2$ such that for all $T > 0$

$$\begin{aligned} \|y_{1T}\|_{\mathcal{L}_p} &\leq \gamma_1\|u_{1T}\|_{\mathcal{L}_p} + \beta_1 \\ \|y_{2T}\|_{\mathcal{L}_p} &\leq \gamma_2\|u_{2T}\|_{\mathcal{L}_p} + \beta_2 \end{aligned}$$

Since the output of system $G_1$ is driving the input to system $G_2$, we can rewrite the second inequality as

$$\begin{aligned} \|y_{2T}\|_{\mathcal{L}_p} &\leq \gamma_2 \left(\gamma_1 \|u_{1T}\|_{\mathcal{L}_p} + \beta_1\right) + \beta_2 \\ &= \gamma_2\gamma_1 \|u_{1T}\|_{\mathcal{L}_p} + (\gamma_2\beta_1 + \beta_2) \end{aligned}$$

which shows that the cascaded system is finite-gain $\mathcal{L}_p$ stable with a gain of $\gamma_2\gamma_1$ and a bias of $\gamma_2 + \beta_1 + \beta_2$. $\diamondsuit$

A similar result an be established with the cascade in Fig. 7 consists of two input-to-state stable (ISS) systems. This result is stated in the following theorem without formal proof.

THEOREM 36. **(Cascade of ISS Systems)** *Consider the cascaded*

$$\begin{aligned} \dot{x} &= f(x, z) \\ \dot{z} &= g(z, u) \end{aligned}$$

*where $f(0,0) = 0$, $g(0,0) = 0$ with $f$ and $g$ being locally Lipschitz. Suppose that the upper (driven) system is ISS with respect to input $z$. Suppose that the lower (driving) system is ISS with respect to input $u$. Then the cascaded system is ISS with respect to input $u$.*

Let us return to our peaking system and re-examine the driving and driven system with regard to $\mathcal{L}_p$-stability, ISS stability, and passivity. The simulation result in Fig. 8 show that the driven system in equation (61) is not $\mathcal{L}_p$ and is not ISS. This is true because the input to the driven system is $\xi$ and the simulation establishes the existence of an input $\xi$ with finite $\mathcal{L}_p$ norm such that the driven system's output becomes unbounded. So clearly we cannot apply theorems 35 or 36. In fact, this example shows that ensuring the two systems are ISS or $\mathcal{L}_p$ stable are critical conditions that when violated can lead to unstable cascades.

If we examine the driving and driven system, with regard, to passivity, we see that the driving system in equation (60) is a linear system whose $A$ matrix is Hurwitz. This is sufficient to establish that the driving linear system is strictly passive. If we examine the driven system, let us consider the storage function

$$V(\eta) = \frac{1}{2}\eta^2$$

and if we compute its directional derivative with respect to the driven system, we obtain

$$\dot{V} = \eta(-\frac{1}{2}(1-u)\eta^3) = -\frac{1}{2}(1-u)\eta^4 = -\frac{1}{2}\eta^4 + \frac{1}{2}u\eta^4$$

If we define the supply rate function $r(u,\eta) = \frac{1}{2}u\eta^4$, then clearly

$$r(u,\eta) \leq \dot{V} + \psi(\eta)$$

where $\psi(r) = \frac{1}{2}r^4$ is positive definite. We can therefore conclude that the driven system is also strictly dissipative. In this regard, both systems are strictly dissipative and so both are asymptotically stable when the input is zero. However, our simulation results show that the cascade is not asymptotically stable and so establishing the dissipative nature of each subsystem is not sufficient to assure the stable or dissipative nature of the whole.

## 8. Computational Methods for Stability Certificates

Lyapunov's direct method certifies the asymptotic stability of a system's equilibrium by checking if there exists a function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ that is positive definite, $V(x) > 0$, with negative definite directional derivative, $-\dot{V}(x) > 0$. The direct method, however, provides little guidance on how to "find" such certificates. One method for finding a Lyapunov function is to start with a function that is already known to be a stability certificate for a closely related system, introducing a parameterization of that function, and then searching for the parameters which establish this "candidate" function is indeed a Lyapunov function. That search can be conducted computationally as part of an optimization problem that seeks to minimize some measure of the parameter's "cost" over a feasible set for which $V > 0$ and $-\dot{V} > 0$. This means that we have transformed the analysis problem into a computational problem. In recent years, the advances in numerical methods for convex optimization now make it possible to numerically find such Lyapunov functions.

One of the main roadblocks we face in developing such a computational approach is that the problem of deciding whether a multi-variate function, $V$, is positive semidefinite is undecidable. If we restrict our attention to $V$ that are polynomial, then the search becomes NP-hard. So at the outset, our problem of certifying whether a candidate Lyapunov function is a stability certificate appears to be computationally intractable. One may get around this issue by relaxing the Lyapunov conditions to a criterion that is only *sufficient* for positivity and yet is computationally easy to verify. The particular relaxation we consider search for certificates, $V$, that are *sum-of-squares* or SOS polynomials.

Let $\mathbb{R}[x]$ denote the set of all polynomials with indeterminate variables $x = \{x_1, \ldots, x_n\}$ with real valued coefficients. If a polynomial $V \in \mathbb{R}[x]$ is positive semidefinite, then an obvious necessary condition is that its degree is even. A simple sufficient condition for $V$ to be positive semidefinite, therefore, is the existence of an SOS decomposition of the form

$$V(x) = \sum_i v_i^2(x)$$

where $v_i \in \mathbb{R}[x]$ for all $i = 1, 2, \ldots, m$. If we can find an SOS decomposition, then one can conclude that $V$ is positive semidefinite. The obvious questions are 1) how conservative is this SOS decomposition and 2) how easy is it to find such decomposition? The first question is known as Hilbert's 17th problem Reznick (2000). In particular, it can be shown that the SOS and non-negative polynomials are equivalent for polynomials of one variable, quadratic polynomials and quartic polynomials in two variables Parrilo (2003).

To answer the second question regarding finding SOS decompositions, let us consider a polynomial $V \in \mathbb{R}[x]$ of degree $2d$ and let us assume it can be written as a quadratic form in all monomials of degree less than equal $d$ given by the different products of the $x$ variables. In particular, this means we can write

$$V(x) = v^T \mathbf{Q} v, \quad v^T = \left[1, x_1, x_2, \ldots, x_n, x_1 x_2, \ldots, x_n^d\right] \tag{62}$$

where $\mathbf{Q}$ is a constant matrix. The length of the monomial vector, $v$, is $\begin{pmatrix} n+d \\ d \end{pmatrix}$. If the matrix $\mathbf{Q}$ is positive semidefinite, then $V(x)$ has an SOS decomposition and so is nonnegative. Note that the matrix $\mathbf{Q}$ is not unique and so $\mathbf{Q}$ may be PSD for some representations and not for others. By expanding out the right hand side of equation (62) and matching coefficients of $x$, one can readily show that the set of matrices that satisfy equation (62) will form an affine variety of a linear subspace (in the space of symmetric matrices). If the intersection of this affine subspace with the positive semidefinite matrix cone is nonempty, then the function $V$ is guaranteed to be SOS and so is also nonnegative.

As an example, consider a function $V$ of the form

$$V(x,y) = 2x^4 + 2x^3y - x^2y^2 + 5y^4 \tag{63}$$

If we take $v^T = \left[x^2, y^2, xy\right]$, then $V$ may be written as a quadratic form,

$$\begin{aligned} V(x,y) &= \begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix} \\ &= q_{11}x^4 + q_{22}y^4 + (q_{33} + 2q_{12})x^2y^2 + 2q_{13}x^3y + 2q_{23}xy^3 \end{aligned}$$

If we then equate coefficients, we obtain the following system of linear equations

$$\begin{bmatrix} 2 \\ 5 \\ -1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{22} \\ q_{33} \\ q_{12} \\ q_{13} \\ q_{23} \end{bmatrix}$$

The set of all solutions to this system of linear inequalities can be readily shown to be

$$\begin{bmatrix} q_{11} & q_{22} & q_{33} & q_{12} & q_{13} & q_{23} \end{bmatrix} = \begin{bmatrix} 2 & 5 & -1-2\lambda & \lambda & 1 & 0 \end{bmatrix}$$

where $\lambda \in \mathbb{R}$ is any real value and so our expression for $V$ take the form,

$$V(x,y) = v^T \begin{bmatrix} 2 & \lambda & 1 \\ \lambda & 5 & 0 \\ 1 & 0 & -1-2\lambda \end{bmatrix} v = v^T\mathbf{Q}(\lambda)v = v^T(\mathbf{Q}_0 + \lambda\mathbf{Q}_1)v \tag{64}$$

where $\mathbf{Q}_0 = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix}$ and $\mathbf{Q}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}$. To see if $V$ has an SOS decomposition, we need to find $\lambda$ such that $\mathbf{Q}(\lambda) = \mathbf{Q}_0 + \lambda\mathbf{Q}_1$ is a positive semidefinite matrix. Note that this takes the form of a nonstrict linear matrix inequality or LMI.

The "standard form" for a "strict" linear matrix inequality (LMI) is an affine matrix-valued function of the form,

$$\mathbf{Q}(\lambda) = \mathbf{Q}_0 + \sum_{i=1}^{m} \lambda_i \mathbf{Q}_i > 0$$

where $\lambda \in \mathbb{R}^m$ are decision variables and $\mathbf{Q}_i = \mathbf{Q}_i^T \in \mathbb{R}^{n \times n}$ are symmetric matrices for $i = 1, 2, \ldots, m$. The LMI feasibility problem is given symmetric matrices, $\{\mathbf{Q}_i\}_{i=1}^m$, determine where there exists a vector $\lambda \in \mathbb{R}^m$ such that the LMI $\mathbf{Q}(\lambda) > 0$. We've stated the strict version of this problem. The nonstrict version requires us to verify that $\mathbf{Q}(\lambda) \geq 0$ which is actually the form of the problem we gave in our example.

The LMI feasibility problem is one of those matrix problems which are computationally tractable. This problem is efficiently solved using "interior-point" techniques that revolutionized the solution of linear programs back in the mid 1980's Adler et al. (1989). The development of interior-point solvers for strict LMI problems appeared in the early 1990's Gahinet et al. (1994). These solvers are recursive algorithms with polynomial time-complexity. Surprisingly, the number of recursions is relatively constant with respect to the number of problem decision variables, which makes these methods extremely efficient. Algorithms that solve the nonstrict LMI problems are sometimes called semidefinite programs Vandenberghe and Boyd (1996). Freely available SDP solvers such as SDPT3 began to appear around 2000 Toh et al. (1999).

One of the main issues in using such SDP solvers is that their user interfaces are not in a form that is easy to use directly. This has led to the development of a number of toolkits that essentially translate LMI expressions that are in the form of matrix inequalities, into the standard form that the solvers then work with. One of the first widely used toolkits that was developed specifically for SOS programming was SOSTOOLS Prajna et al. (2002). The interface for SOSTOOLS can be somewhat clumsy to work with and so a more recent interface toolkit known as YALMIP Lofberg (2004) has been gaining widespread acceptance across the community. The examples that we show below use YALMIP as the interface to the SDP solver.

We will now use YALMIP to see if the polynomial in equation (63) has an SOS decomposition. Recall that this involves finding a real $\lambda$ such that $\mathbf{Q}(\lambda)$ in equation (64) is positive semidefinite. We start by declaring the state variables and forming the polynomial we want to check,

```
x = sdpvar(1,1); y = sdpvar(1,1);
V = (2*x^4)+(2*x^3*y)-(x^2*y^2)+(5*y^4);
```

We then form the vector of monomials, $v$, in equation (62) and then construct the quadratic form, $v^T\mathbf{Q}v$. The command `monolist` constructs a list of all monomials with degree less than $2$. For this problem that means

$$v^T = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 \end{bmatrix} \tag{65}$$

You could have also specified a specific list of monomials.

```
v          = monolist([x y],degree(V)/2);
Q         = sdpvar(length(v));
V_sos = v'*Q*v;
```

We then form the set of SOS constraints that are passed on to the solver. These constraints require $\mathbf{Q}$ to be PSD and the coefficients of the SOS polynomial to match the coefficients of the specified $V$. Once this is done we can call the SOS solver that computes the SOS decomposition of $V$ (if it exists). Since we did not formally declare any SOS-type constraint, we use the solver `optimize` which returns the desired answer (if it exists) in the matrix $\mathbf{Q}$.

```
F = [coefficients(V-V_sos,[x y])==0, Q>=0];
sol=optimize(F);
if sol.problem==0
value(Q);
end
```

The diagnostics from `optimize` are contained in the structure `sol` and if member `sol.problem` is zero, then the SDP solver was able to find a positive semidefinite matrix $\mathbf{Q}$ that satisfied the problem's constraints. For this particular example that matrix is

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & -1.4476 \\ 0 & 0 & 0 & 1 & 1.8952 & 0 \\ 0 & 0 & 0 & -1.4476 & 0 & 5 \end{bmatrix}$$

which is defined with respect to the monomial ordering in equation (65).

The original ordering we used in defining our problem in equation (64) had a monomial ordering of $v^T = [x^2, y^2, xy]$. If we extract out these rows and columns of the $\mathbf{Q}$ computed using YALMIP, then we obtain

$$\mathbf{Q}(\lambda) = \begin{bmatrix} 2 & -1.4476 & 1 \\ -1.4476 & 5 & 0 \\ 1 & 0 & 1.8952 \end{bmatrix} = \begin{bmatrix} 2 & \lambda & 0 \\ \lambda & 5 & 0 \\ 1 & 0 & -1-2\lambda \end{bmatrix}$$

YALMIP asserted that for $\lambda = -1.4476$, this $\mathbf{Q}(\lambda)$ is positive semidefinite. This observation is verified by computing the eigenvalues of $\mathbf{Q}(-1.4476)$ to find they ( $0.9633$, $3.2922$, and $5.6398$) are all nonnegative. The SOS decomposition can be obtained by taking the square root of $\mathbf{L}^T\mathbf{L} = \mathbf{Q}$, to obtain

$$\mathbf{L} = \begin{bmatrix} 1.2927 & -0.4202 & 0.3903 \\ -0.4202 & 2.1957 & 0.0467 \\ 0.3903 & 0.0467 & 1.3193 \end{bmatrix} \Rightarrow V(x) \begin{aligned}[t] &= \sum_{i=1}^{3} v_i(x) \\ &= (1.2927x^2 - 0.4202y^2 - 0.3903xy)^2 \\ &\quad +(-0.4202x^2 + 2.1957y^2 + 0.0467xy)^2 \\ &\quad +(0.3903x^2 + 0.0467y^2 + 1.3193xy)^2 \\ &= 2x^4 + 2x^3y - x^2y^2 + 5y^4 \end{aligned}$$

which verifies that $V$ is SOS.

The preceding discussion steps through with YALMIP what we did in forming the LMI $\mathbf{Q}(\lambda) = \mathbf{Q}_0 + \lambda \mathbf{Q}_1$. YALMIP also provides a more direct way of doing this through the command `sos` that streamlines the task of forming an SOS constraint and then using the command `solvesos` to compute the decomposition and actually find the $\mathbf{Q}$ matrices. Alternatively, one could use the command `sosd` to just return the SOS decomposition.

```
 x = sdpvar(1,1); y = sdpvar(1,1);
V = (2*x^4)+(2*x^3*y)-(x^2*y^2)+(5*y^4);
F = sos(V);
[sol,u,Q,res] = solvesos(F);
if sol.problem==0
    sdisplay(u{1})
    value(Q{1})
    v = sosd(F);
    sdisplay(v)
end;
```

This returns a slightly different decomposition than we obtained doing the long way, but it still forms an SOS decomposition for $V$, merely emphasizing the fact that these decompositions are not unique.

With the preceding introduction to using YALMIP in finding SOS decompositions, we now proceed to show how to use it in finding Lyapunov functions for nonlinear dynamical systems. Consider the linear dynamical system

$$
\begin{aligned}
\dot{x}_1 &= -ax_1 + x_2 \\
\dot{x}_2 &= x_1 - x_2
\end{aligned}
$$

We can check the stability of this system by looking at the eigenvalues of $A = \begin{bmatrix} -a & 1 \\ 1 & -1 \end{bmatrix}$. These eigenvalues will all have nonpositive real parts for $a \geq 1$. At $a = 1$, it has a zero eigenvalue and for $a < 1$, the system is unstable. We will use this to check the calculation made by YALMIP. As before, our YALMIP script starts by cleaning up the workspace and declaring the state variables,

```
clear all;
yalmip(’clear’);
sdpvar x1 x2;
x = [x1 ; x2];
```

We then declare the vector field with $a = 2$,

```
a = 2;
f = [  -a*x1+x2; x1-x2];
```

We form the first SOS constraint that requires $V \geq 0$. This constraint uses `sos` to form the SOS constraint

```
P = sdpvar(length(x));
V = x’*P*x;
F = [P>=0]+[sos(V)];
```

We then form the second SOS constraint that requires $-\dot{V} > 0$. This constraint uses `jacobian` to symbolically compute the Jacobian of $V$. Note that the actual constraint we are checking to be SOS is $-\dot{V} - \epsilon(x_1^2 + x_2^2) \geq 0$. The second part of this inequality forces $-\dot{V}$ to be strictly positive definite since the SDP solver only works with nonstrict inequality constraints.

```
negVdot = -jacobian(V,x)*f;
eps = 0.1;
F = F + [sos(negVdot-eps*(x’*eye(2,2)*x))];
```

We then use `solvesos` to compute the SOS decomposition of these constraints. The function returns the solution status `sol`, a vector `u` of the monomials, and the symmetric matrix $\mathbf{Q}$ associated with those monomials. The returned vector `u` and `Q` are data structures that contain two members; one for the first sos constraint on $V$ and another for the second SOS constraint on $\dot{V}$. We're interested in the first one. In particular if `sol.problem` equals 0, then SDPT3 found a feasible solution and we can then display it

```
[sol,u,Q] = solvesos(F);
if sol.problem == 0
    disp('Constraints are SOS');
    sdisplay(u{1}'*Q{1}*u{1})
else
    disp('Constraints FAILED');
end
```

The Lyapunov function returned from this has the form

$$\begin{aligned}
V(x_1,x_2) &= 5.2779x_1^2 + 6.7222x_2^2 + 2.8886x_1x_2 \\
&= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 5.2779 & 1.443 \\ 1.443 & 6.722 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
&= x^T\mathbf{P}x
\end{aligned}$$

We can readily check to see that $\mathbf{P}$ is indeed positive definite and symmetric with real eigenvalues $4.3852$ and $7.6148$. We can also verify that it satisfies the Lyapunov equation

$$\begin{aligned}
\mathbf{A}^T\mathbf{P} + \mathbf{PA} &= \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 5.2779 & 1.443 \\ 1.443 & 6.722 \end{bmatrix} + \begin{bmatrix} 5.2779 & 1.443 \\ 1.443 & 6.722 \end{bmatrix} \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} \\
&= \begin{bmatrix} -24 & 7.6671 \\ 7.6671 & -7.6671 \end{bmatrix}
\end{aligned}$$

which has eigenvalues $-27.0352$ and $-4.6320$ and so is negative definite as expected.

To double check our answer, let us see what happens if we let $a = 1$, so that the system has a zero eigenvalue. In this case, running the same script yields `problem.sol=1`, which implies that SDPT3 could not find an SOS decomposition. In particular, we fail to find a positive definite $V$ for this problem. If we relaxed the requirement for $-\dot{V} > 0$ and simply required it to be $\dot{V} \geq 0$, we would be able to get a solution. We can relax this restriction by simply changing the SOS constraint on $\dot{V}$ to

```
F =  F + [sos(negVdot)];
```

With this change, SDPT3 does find a solution, but the resulting $V(x)$ matrix is now

$$V(x) = 5.7114x_1^2 + 5.7114x_2^2 + 1.6364x_1x_2 = x^T \begin{bmatrix} 5.7114 & 1.6364 \\ 1.6364 & 5.7114 \end{bmatrix} x$$

which is positive definite. But now when we look at the Lyapunov equation we see that,

$$\begin{aligned} \mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} &= \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 5.1174 & 0.8182 \\ 0.8182 & 5.7114 \end{bmatrix} + \begin{bmatrix} 5.1174 & 0.8182 \\ 0.8182 & 5.7114 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} -9.7864 & 9.7864 \\ 9.7864 & -9.7864 \end{bmatrix} \end{aligned}$$

which has eigenvalues $-19.5729$ and $0$. So this did not yield an asymptotically stable system, as expected. The reason why our SOS decomposition failed was because we were forcing $-\dot{V}$ to be positive definite, not just positive semidefinite.

## 9. Summary

This chapter reviewed advanced stability concepts used in characterizing the behavior of systems that are driven by external inputs. In particular, this chapter reviewed results regarding Lyapunov stability, input-to-state stability, $\mathcal{L}_p$-stability, and passivity. The basic concepts from Lyapunov stability that were covered in my Linear Systems Theory course were reviewed here and we introduced advanced results regarding global Lyapunov stability and the Invariance Theorem. We developed the notion of Input-to-state stability as a formal extension of uniform ultimate boundedness. Our results on $\mathcal{L}_p$ stability were also covered in Linear systems theory with advanced topics on the Hamilton-Jacobi-Isaacs Inequality. This chapter also reviewed basic results on the relationship between Lyapunov stability and passivity. Of particular importance in this chapter were results regarding whether these stability properties were preserved under feedback and cascade connections. In particular, many of these stability concepts are preserved under feedback connections if the interconnected systems satisfy a small gain condition or if the systems were passive. Cascades of passive systems, however, do not usually preserve passivity as was demonstrated using the peaking example. This example will be important in our later study of constructive nonlinear control schemes. Much of this material was drawn directly from Khalil (2002). Some of the results on passive systems were drawn from Sepulchre et al. (2012). The chapter closed by demonstrating how SOS-programming can be used to search for Lyapunov functions. These methods are also useful in establishing safety certificates and regions of attraction. The discussion presented here is drawn largely from Parrilo (2003) and Lofberg (2004).

CHAPTER 5

# Constructive Nonlinear Control Systems

**Constructive nonlinear control** is a sophisticated approach to nonlinear control that was pioneered by Koktovic and his students at UCSB [Sepulchre et al. (2012)]. It is called ”constructive” because the controllers are synthesized in a recursive manner from control Lyapunov functions (CLF) [Sontag (1989)] that are constructed in a step by step manner using integrator backstepping [Krstić and Kokotović (1996)] on a base scalar system. That base scalar system is obtained through the normal form of an affine nonlinear system that has been feedback linearized [Isidori (1995, 1999)]. Controller synthesis can be based directly on the CLF using well known ISS formulations, or it can be based on the feedback passivation of the cascaded chain of integrators in the normal form. The chapter concludes by illustrating this constructive synthesis on a well known benchmark known as the TORA problem [Wan et al. (1996)].

## 1. Input-Ouptut Feedback Linearization

Traditionally, linearization of a nonlinear system is done through a Taylor series approximation as seen in Lyapunov's indirect method. This approximation is useful in a local neighborhood of the equilibrium provided it has no center eigensubspace. One issue with this approach is that the size of the ”neighborhood” may be too small to be of practical value. A different approach to linearization works with nonlinear systems that are *affine* in the control

$$\begin{aligned} \dot{x}(t) &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

In this case, we can linearize the system through a transformation on the input $u$ such that with respect to the new input, $v$, the system's input/output map appears to be linear. This approach is known as *input-output feedback linearization*. It greatly enlarges the size of the region in which the linearization holds, but it does tend to be sensitive to modeling uncertainty and so has a robustness issue.

Consider a nonlinear scalar input/output system whose input $u$ enters the state equation in an "affine" manner

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\ y(t) &= h(x(t)) \end{aligned}$$

with $f(0) = 0$ so that $0$ is an equilibrium point of the unforced (i.e. $u = 0$) system. For convenience, we introduce the following notation for the Lie derivative of $h$ with respect to $f$,

$$L_f h(x) \stackrel{\text{def}}{=} \frac{\partial h(x)}{\partial x} f(x)$$

Lie derivatives are sometimes called *directional derivatives* since they describe the rate of change in $h$ along the trajectories generated by the vector field $f(x)$. This notation can be iterated upon so that

$$L_f^k h(x) \stackrel{\text{def}}{=} \frac{\partial L_f^{k-1} h(x)}{\partial x} f(x)$$

with $L_f^0(h(x)) \stackrel{\text{def}}{=} h(x)$. With this notation, let us compute the time derivative of the output,

$$\dot{y}(t) = L_f h(x(t)) + L_g h(x(t))u(t)$$

Note that if $L_g h(x) \neq 0$, then we can introduce an input variable transformation of the form

$$u(t) = \frac{1}{L_g h(x(t))} \left( v(t) - L_f h(x(t)) \right)$$

which would give the following *linear differential equation* relating the new input, $v$, to the output $y$

$$\dot{y}(t) = v(t)$$

If $L_g h(x) = 0$, then we simply differentiate again and continue doing so until $u$ appears in the expression. This would mean that there is an integer $r > 0$ such that

$$\frac{d^r y(t)}{dt^r} = y^{(r)}(t) = L_f^r h(x(t)) + L_g L_f^{r-1} h(x(t)) u(t)$$

and where $L_g L_f^{r-1} h(x) \neq 0$. One could then use the feedback transformation

$$u(t) = \frac{1}{L_g L_f^{r-1} h(x(t))} (v(t) - L_f^r h(x(t)))$$

to obtain the *linear* input-output map

$$y^{(r)}(t) = v(t)$$

This is a linear system consisting of a chain of $r$ integrators. The value of $r$ for which $L_g L_f^{r-1} h(x) \neq 0$ is called the *relative degree* of the system.

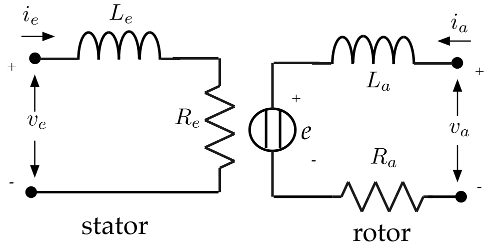


FIGURE 1. Field-controlled DC Motor

**Example:** Let us now present an example illustrating how this input-output feedback linearization works. We consider a field-controlled DC motor whose physical layout is shown in Fig. 1. The state equations associated with the electrical part of this system are

$$\begin{aligned} v_e &= L_e \frac{di_e}{dt} + R_e i_e \\ v_a &= L_a \frac{di_a}{dt} + R_a i_a + e \end{aligned}$$

where $e$ is the back EMF generated by the motor spinning at angular rate $\omega$. This EMF is proportional to the product of the stator current and the angular rate in which $c$ is the proportionality constant.

$$e = ci_e\omega$$

The motor torque is $T = \theta i_e i_a$ where $i_a$ is the current in the rotor and this defines the mechanical part of the motor. If we let $v_e$ be the control input $u$, the output $y = \omega$ and the states are $x_1 = i_e$ (stator current), $x_2 = i_a$ rotor current, and $x_3 = \omega$. With these variable assignments we get the following system of state equations

$$\begin{aligned}\dot{x}_1 &= -ax_1 + u \\ \dot{x}_2 &= -bx_2 + \rho - cx_1x_3 \\ \dot{x}_3 &= \theta x_1 x_2\end{aligned}$$

where $a = R_e/L_e$, $b = R_a/L_a$, and $\rho = v_a/L_a$. The open loop system has an equilibrium at $x_1 = 0$, $x_2 = \rho/b$, and a constant shaft speed setpoint of $\omega_0$. The operating point for this system is therefore taken to be $x^* = \begin{bmatrix} 0 & \rho/b & \omega_0 \end{bmatrix}^T$.

Our system equation is in the form of $\dot{x} = f(x) + g(x)u$ with $y = h(x)$ in which

$$h(x) = x_3, \quad f(x) = \begin{bmatrix} -ax_1 \\ -bx_2 + \rho - cx_1x_3 \\ \theta x_1 x_2 \end{bmatrix}, \quad g(x) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

To find the I/O feedback linearizing control we take the output $y$ and begin differentiating by time until $u$ appears. The first derivative of $y$ yields,

$$\dot{y} = L_f h(x) + L_g h(x) u = \dot{x}_3 = \theta x_1 x_2$$

Since $u$ does not appear in this expression, we know $L_g h(x) = 0$. We must therefore differentiate one more time to get

$$\begin{aligned}\ddot{y} &= L_f^2 h(x) + L_g L_f h(x) u = \theta x_1 \dot{x}_2 + \theta \dot{x}_1 x_2 \\ &= \theta x_1(-bx_2 + \rho - cx_1x_3) + \theta(-ax_1 + u)x_2\end{aligned}$$

Since $u$ appears in $\ddot{y}$, we know the relative degree of this system is $r = 2$. We now determine the I/O linearizing control $u$. From the prior discussion we know this linearizing control will be $u = \frac{1}{L_g L_f h(x)}\left(v - L_f^2 h(x)\right)$; so we will need to compute the Lie derivatives $L_f^2 h(x)$ and $L_g L_f h(x)$.

We now compute the iterated Lie derivatives of $h$ with respect to $f$ and $g$. These iterated Lie derivatives are

$$\begin{aligned}
L_f^0 h &= h(x) = x_3 \\
L_f h = \frac{\partial h}{\partial x} f(x) &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -ax_1 \\ -bx_2 + \rho - cx_1 x_3 \\ \theta x_1 x_2 \end{bmatrix} = \theta x_1 x_2 \\
L_g L_f h = \frac{\partial L_f h}{\partial x} g(x) &= \frac{\partial \theta x_1 x_2}{\partial x} g(x) = \begin{bmatrix} \theta x_2 & \theta x_1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \theta x_2 \\
L_f^2 h(x) = \frac{\partial L_f h}{\partial x} f(x) &= \frac{\partial \theta x_1 x_2}{\partial x} f(x) = \begin{bmatrix} \theta x_2 & \theta x_1 & 0 \end{bmatrix} \begin{bmatrix} -ax_1 \\ -bx_2 + \rho - cx_1 x_3 \\ \theta x_1 x_2 \end{bmatrix} \\
&= -(a+b)\theta x_1 x_2 + \rho\theta x_1 - c\theta x_1^2 x_3
\end{aligned}$$

We therefore see that the I/O linearizing control is

$$\begin{aligned}
u &= \frac{1}{L_g L_f h(x)}(v - L_f^2 h(x)) \\
&= \frac{v - (-(a+b)\theta x_1 x_2 + \rho\theta x_1 - c\theta x_1^2 x_3)}{\theta x_2} \\
&= \boxed{(a+b)x_1 + c\frac{x_1^2 x_3}{x_2} - \rho\frac{x_1}{x_2} + \frac{v}{\theta x_2}}
\end{aligned}$$

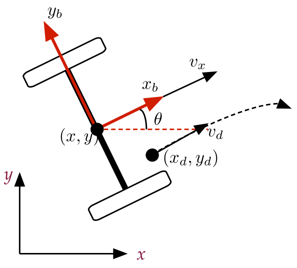
We can verify the correctness of our control by substituting back into the $\ddot{y}$ equation and seeing if this reduces to a chain of two integrators. In particular, this computation yields

$$\begin{aligned}
\ddot{y} &= \theta x_1 \dot{x}_2 + \theta \dot{x}_1 x_2 \\
&= \theta x_1(-bx_2 + \rho - cx_1x_3) + \theta(-ax_1 + u)x_2 \\
&= -\theta(a+b)x_1 x - 2 + \theta\rho x_2 - c\theta x_1^2 x_3 \\
&\quad + \left[\theta(a+b)x_1x_2 + c\theta x_1^2 x_3 + \theta\rho x_1 + v\right] \\
&= v
\end{aligned}$$

which indeed verifies that the proposed control linearizes the input-output map from $v$ to $y$.

It is important to note that we have a system with three states, but that only two are actually observable at the output. This raises a question with regard to the ”stability” of the third state variable. The following section addresses this issue by introducing the *normal form* for a scalar affine system which shows more clearly how the chain of integrators is related to the unobservable dynamical states.

s **Example - Tracking Control of a 2D Mobile Robot:** We now present a tracking example that will be used later with other robust nonlinear control methods. This section compares the performance of tracking controllers based on Taylor Jet and I/O Feedback linearizations of the plant. The “plant” is a two-wheeled robotic vehicle shown in Fig. 2. Let $F$ denote the force applied by both wheels along the body's $x$-axis and let $T$ denote the torque developed by these wheels about the vehicle's center of mass which is located at point $(x, y)$ in the plane. We introduce the control vector $u(t) = \begin{bmatrix} F(t) \\ T(t) \end{bmatrix}$. The state variables of this system are the plant's center of mass, $x$ and $y$, the angle of the body with respect to an inertial reference, $\theta$, the velocity of the vehicle in the direction of the body's $x$-axis, $v_x$, and the angular rate, $\omega$, of that body angle. With these conventions, the system to be regulated and its equations of motion are shown in the middle pane of Fig. 2.



$$
\begin{aligned}
\dot{x} &= v_x \cos\theta \\
\dot{y} &= v_x \sin\theta \\
\dot{\theta} &= \omega \\
\dot{v}_x &= F \\
\dot{\omega} &= T
\end{aligned}
$$

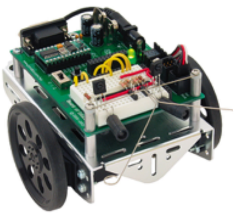


FIGURE 2. Two-wheeled Robot - (left) vehicle geometry - (middle) equations of motion - (right) picture of system

We will examine two methods for determining the regulating controller, both of which involve *linearizing* the original system equations. The first approach uses a Taylor jet linearization for the system $\dot{x} = f(x,u)$. This linearization has the form

$$
\begin{aligned}
\dot{x} &= \left[\frac{\partial f}{\partial x}(x^*,0)\right](x - x^*) + \left[\frac{\partial f}{\partial u}(x^*,0)\right] u \\
&= \mathbf{A}(x - x^*) + \mathbf{B}u
\end{aligned}
\tag{66}
$$

where $x^*$ is the operating point about which the Taylor jet is constructed. Since $\mathbf{A}$ and $\mathbf{B}$ are real-valued matrices, this is a system that is commonly studied in linear systems theory. This suggests that if one were to design a state feedback controller matrix, $\mathbf{K}$, such that the control signal $u = \mathbf{K}(x - x^*)$ asymptotically stabilizes the linearized system about this equilibrium point, then we should achieve adequate regulation of the nonlinear system.

The first step in developing such a state feedback control is to find the linearization in equation (66) for our two-wheeled cart. We start by introducing the new tracking variables $z_1 = x - x_d$, $z_2 = y - y_d$, $z_3 = \theta - \theta_d$, $z_4 = v_x - v_d$, and $z_5 = \omega$ where $(x_d(t), y_d(t))$ is the trajectory we want our vehicle to track in the plane, $\theta_d(t)$ is the direction of the desired trajectory's velocity vector, and $v_d(t)$ is the magnitude of that desired velocity vector.

With this change of variables our system equations in Fig. 2 become

$$\begin{aligned}\begin{bmatrix}\dot z_1\\ \dot z_2\\ \dot z_3\\ \dot z_4\\ \dot z_5\end{bmatrix} &= \begin{bmatrix}(z_4+v_d)\cos(z_3+\theta_d)-\dot x_d\\ (z_4+v_d)\sin(z_3+\theta_d)-\dot y_d\\ z_5-\dot\theta_d\\ -\dot v_d\\ 0\end{bmatrix} + \begin{bmatrix}0&0\\0&0\\0&0\\1&0\\0&1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}\\ &= F(x)+G(x)u\end{aligned}$$

Computing the Jacobian matrix for $F$ yields the following linearized system equation

$$\begin{aligned}\begin{bmatrix}\dot z_1\\ \dot z_2\\ \dot z_3\\ \dot z_4\\ \dot z_5\end{bmatrix} &= \begin{bmatrix}0&0&-v_d\sin(\theta_d)&\cos(\theta_d)&0\\ 0&0&v_d\cos(\theta_d)&\sin(\theta_d)&0\\ 0&0&0&0&1\\ 0&0&0&0&0\\ 0&0&0&0&0\end{bmatrix}\begin{bmatrix}z_1\\ z_2\\ z_3\\ z_4\\ z_5\end{bmatrix} + \begin{bmatrix}0&0\\0&0\\0&0\\1&0\\0&1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}\\ &= \mathbf{A}z+\mathbf{B}u\end{aligned}$$

We can then use any one of a number of methods to design stabilizing controllers for this system. In particular, we'll compute the linear quadratic regulator (LQR) that finds the state gains, $K$, such that the controller $u = Kz$ minimizes the cost functional

$$J[u] = \int_0^\infty (z^Tz + u^Tu)d\tau$$

This controller was simulated in the following MATLAB script (Fig. 3) with the desired trajectory $(x_d(t), y_d(t))$ being defined by the following equations

$$\begin{aligned}\dot x_d(t) &= 50\sin\left(\frac{2\pi t}{50}\right), \quad x_d(0)=0\\ \dot y_d(t) &= 50\cos\left(\frac{4\pi t}{50}\right), \quad y_d(0)=0\end{aligned}$$

The LQR control was recomputed at each time instant using the desired reference trajectory states. The resulting vehicle trajectory for a vehicle initially at rest at position $(x_0, y_0) = (50, 0)$ is shown on the left hand side of

Fig. 3. We indeed obtain tracking of the desired reference trajectory, though the vehicle's initial transient shows some significant oscillation while it is picking up speed.

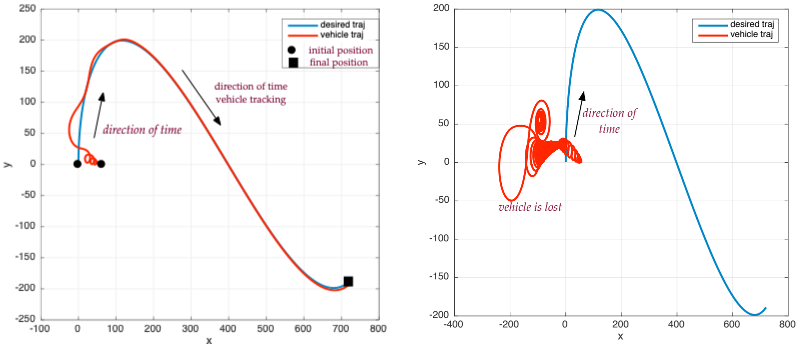


FIGURE 3. (left) trajectories for linearized control with $(x_0, y_0) = (50, 0)$ - (right) trajectories with $(x_0, y_0) = (55, 0)$.

An important limitation of the preceding linearization approach is that the topological equivalence is only *local* (i.e. in a neighborhood of the equilibrium point). This suggests that if we were to start the vehicle further away from the desired reference trajectory then our control strategy might fail. This indeed is the case for our system. In particular, if we change the initial condition to $(x_0, y_0) = (55, 0)$, then we obtain the system trajectory shown on the right side of Fig 3. In this case, we see the vehicle simply spins around close to its starting position while it is trying to gather sufficient speed to catch up to the desired state trajectory. In this case the vehicle was never able to track the desired reference trajectory.

The "local" nature of our control is an important limitation of the linearization approach used above. One way of overcoming this limitation is to base our controller on a *feedback linearization* of the plant, since we know this allows a much larger operating region. In the feedback linearization approach we will find it convenient to introduce a change of control variables in which

$$\begin{aligned} u_1(t) &= F(t) = \int_0^t v_1(s)ds \\ u_2(t) &= T(t) = v_2(t) \end{aligned}$$

The original control, $u_1 = F$, is then treated as another system state, thereby extending the state vector of the original system. With this change of control variable we obtain the following state equations for our cart,

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \\ v_x \\ \omega \\ F \end{bmatrix} = \begin{bmatrix} v_x \cos\theta \\ v_x \sin\theta \\ \omega \\ F \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

We now introduce a state transformation which is obtained by taking the derivatives of the tracking error. This means that the first three states are

$$z_1 = x - x_d, \quad z_2 = \dot{x} - \dot{x}_d, \quad z_3 = \ddot{x} - \ddot{x}_d$$

and the second three states are obtained from the derivatives of the $y$ component,

$$z_4 = y - y_d, \quad z_5 = \dot{y} - \dot{y}_d, \quad z_6 = \ddot{y} - \ddot{y}_d$$

The differential equations for these components are then readily computed as

$$\begin{aligned} \dot{z}_1 &= v_x \cos\theta - \dot{x}_d = \boxed{z_2} \\ \dot{z}_2 &= F\cos\theta - v_x\omega\sin\theta - \ddot{x}_d = \boxed{z_3} \\ \dot{z}_3 &= \boxed{v_1\cos\theta - v_x v_2 \sin\theta - (2F\omega\sin\theta + v_x\omega^2\cos\theta) - \dddot{x}_d} \\ \dot{z}_4 &= v_x \sin\theta - \dot{y}_d = \boxed{z_5} \\ \dot{z}_5 &= F\sin\theta + v_x\omega\cos\theta - \ddot{y}_d = \boxed{z_6} \\ \dot{z}_6 &= \boxed{v_1\sin\theta + v_x v_2 \cos\theta + (2F\omega\cos\theta - v_x\omega^2\sin\theta) - \dddot{y}_d} \end{aligned}$$

These equations have the form of two chains of integrators driven by the inputs into states $z_3$ and $z_6$. This means we can rewrite the above differential equations in the following form,

$$
\begin{aligned}
\left[\begin{array}{c} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \hline \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \end{array}\right]
&= \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]
\left[\begin{array}{c} z_1 \\ z_2 \\ z_3 \\ \hline z_4 \\ z_5 \\ z_6 \end{array}\right]
+ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array}\right]
\left[\begin{array}{c} -(2F\omega\sin\theta + v_x\omega^2\cos\theta) - \dddot{x}_d \\ 2F\omega\cos\theta - v_x\omega^2\sin\theta - \dddot{y}_d \end{array}\right] \\
&\quad + \left[\begin{array}{cc} \cos\theta & -v_x\sin\theta \\ \sin\theta & v_x\cos\theta \end{array}\right]
\left[\begin{array}{c} v_1 \\ v_2 \end{array}\right] \\
\dot{z} &= \mathbf{A}z + \mathbf{E}(\alpha + \rho v)
\end{aligned}
$$

where $\mathbf{A}$ is the linear matrix representing the chain of integrators, $\alpha$ and $\rho$ are matrices whose components are functions of the original system states, $F$, $\omega$, $\theta$, and $v_x$. Note that if we select the control $v$ to have the form

$$
v = \rho^{-1}\left(-\alpha + \left[\begin{array}{c} \dddot{x}_d \\ \dddot{y}_d \end{array}\right] + \mathbf{K}z\right) \tag{67}
$$

Then the resulting state equation is given by

$$
\dot{z} = (\mathbf{A} + \mathbf{E}\mathbf{K})z \tag{68}
$$

where

$$
\begin{aligned}
\mathbf{E}^T &= \left[\begin{array}{cccccc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \\
\mathbf{K} &= \left[\begin{array}{cccccc} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \end{array}\right]
\end{aligned}
$$

The important thing to note here is that equation (68) is a *linear* differential equation and so if we can select $\mathbf{K}$ so that $\mathbf{A}+\mathbf{E}\mathbf{K}$ is a Hurwitz matrix, then we would have globally stabilized our vehicular system using the control in equation (67).

We can indeed check this out using a MATLAB simulation. The results are shown in Fig. 4 for a vehicle at rest with initial positions $(x_0, y_0) = (50, 0)$ and $(x_0, y_0) = (250, 250)$. These simulation results show convergence to the desired trajectory in which the tracking error appears to be a monotone decreasing function of time. The gains were not chosen, in this case to be optimal, they were simply chosen to place all of the system's closed poles at $(-1, 0)$. It would have been relatively easy to obtain better performance by simply increasing these gain values. By showing the response from both initial conditions, we demonstrate that the feedback linearized control is indeed "global" in a manner that is far superior to the "local" linear controller.

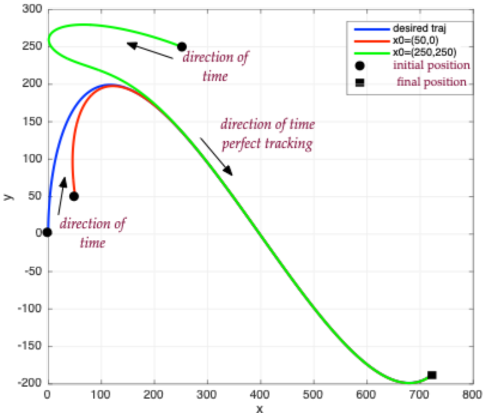


FIGURE 4. Feedback Linearized Controller

## 2. Normal Form for Scalar Affine Systems

The feedback transformation used in the preceding section can be seen as introducing a nonlinear change of coordinates where the original states, $x$, are transformed into a new state whose components are the output $y$ and its derivatives. The resulting realization is called the *normal form* for the scalar affine system and it is often used in developing robust stabilizing controls. One advantage of the normal form is that it clearly shows what parts of the original system are *unobservable* from the output which is useful in trying to develop constructive or passivity-based controllers.

Let us consider a scalar input-output system of the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

where about a point $x_0 \in \mathbb{R}^n$ the system has a relative degree of $r < n$. What we showed above is that if we simply differentiate the output $r$ times, we can then introduce a specific feedback linearizing transformation $u = k(x,v)$ such that the input-output map from $v$ to $y$ satisfies the $r$th order differential equation $y^{(r)} = v$, which is a linear map. This linear system has $r$ states, $z_i$ for $i = 1, \ldots, r$, defined as

$$\begin{aligned} z_1 &= y \\ z_2 &= \dot{y} \\ \vdots & \quad \vdots \\ z_r &= y^{(r-1)} \end{aligned}$$

and we will find it convenient to think of the map from the nonlinear system states $x$ into $z_i$ for $i = 1, \ldots, r$ as coordinate transformations. In other words, there will be smooth functions $T_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, r$ such that

$$z_1 = T_1(x), z_2 = T_2(x), \ldots z_r = T_r(x)$$

Let us now assume we can find $n - r$ other functions, $T_k$ for $k = r + 1, \ldots, n$ such that $T : \mathbb{R}^n \to \mathbb{R}^n$

$$z = T(x) = \begin{bmatrix} T_1(x) \\ \vdots \\ T_n(x) \end{bmatrix}$$

is a *diffeomorphism* in a neighborhood of $x_0$. This means that $T$ is $C^1$ and $T^{-1}$ is $C^1$ and that any orbits of the $x$-system can be mapped smoothly into the orbits of the $z$-system and vice versa. In other words, it would mean that $T$ is a *nonlinear similarity transformation* between the $x$-system states and the $z$-system states that leaves the topological properties of the orbits unchanged. Simply stated this means that assessing stability of the $x$-system can be done by looking at the $z$-system and vice versa.

Let us further assume that in addition to $T$ being a local diffeomorphism that we can choose $T_{r+1}, \ldots, T_n$ so that

$$\frac{\partial T_k(x)}{\partial x} g(x_0) := L_g T_k(x_0) = 0 \tag{69}$$

for all $k = r+1, \ldots, n$. In fact, we can usually establish that this can be done (i.e. $T$ is both a diffeomorphism and satisfies the conditions in equation (69). In particular these conditions define a set of partial differential equations that can be used to find the $T_k$. Such PDEs also require some boundary conditions. In general, since we usually study stability with respect to the system's origin, this would mean that $x_0 = 0$ and so we would also usually want $T(x_0) = 0$ as well.

Provided we can find such a $T$, then we can write the dynamics of the $z$-system as

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= z_3 \\
\vdots\ & \quad \vdots \\
\dot{z}_{r-1} &= z_r \\
\dot{z}_r &= L_f^r h(x) + L_g L_f^{r-1} h(x) u
\end{aligned}$$

Since $T$ is a diffeomorphism there exists $T^{-1}(z) = x$ so that

$$\dot{z}_r = L_f^r h(T^{-1}(z)) + L_g L_f^{r-1} h(T^{-1}(z)) u$$

which shows that $\dot{z}_r$ is a function of the $z$ state and for convenient we take

$$a(z) = L_g L_f^{r-1} h(T^{-1}(z)), \quad b(z) = L_f^r(T^{-1}(z))$$

so that the $z_r$ state equation is written as

$$\dot{z}_r = b(z) + a(z) u$$

The state equation for $z_{r+1}$ is then written as

$$\begin{aligned}
\dot{z}_{r+1} &= \frac{\partial T_{r+1}(x)}{\partial x}(f(x)+g(x)u) \\
&= L_f T_{r+1}(x) + L_g T_{r+1}(x)u \\
&= L_f T_{r+1}(T^{-1}(z)) \\
&= q_{r+1}(z)
\end{aligned}$$

where we used that orthogonality condition in equation (69) and the fact that $x = T^{-1}(z)$. This shows that $\dot{z}_{r+1}$ is also a function of the system state $z$. We can repeat this for the remaining states $z_k$ for $k = r+2, \ldots, n$ to see that

$$\begin{aligned}
\dot{z}_{r+2} &= q_{r+2}(z) \\
\vdots \quad & \quad \vdots \\
\dot{z}_n &= q_n(z)
\end{aligned}$$

If we then choose the control transformation

$$u = \frac{1}{a(z)}(v - b(z))$$

we decouple the states $z_1, \ldots, z_r$ from the other states $z_{r+1}, \ldots, z_n$ and we get the output map

$$\begin{aligned}
\dot{z}_i &= z_{i+1}, \quad (i = 1, \ldots, r-1) \\
\dot{z}_r &= v \\
y &= z_1
\end{aligned}$$

It is more convenient to rewrite the state equations in a manner that separates out components $z_1, \ldots, z_r$ from $z_{r+1}, \ldots, z_n$. In particular, define

$$z = \begin{bmatrix} \xi \\ \eta \end{bmatrix}, \quad \text{where} \quad \xi = \begin{bmatrix} z_1 \\ \vdots \\ z_r \end{bmatrix}, \quad \text{and} \quad \eta = \begin{bmatrix} z_{r+1} \\ \vdots \\ z_n \end{bmatrix}$$

With this notation we can rewrite the $z$-system equations as

$$\begin{aligned}\dot{\xi} &= \mathbf{A}_c\xi + \mathbf{B}_c\left(a(\eta,\xi)u + b(\eta,\xi)\right)\\ \dot{\eta} &= q(\eta,\xi)\\ y &= \mathbf{C}_c\xi\end{aligned}$$

where

$$\mathbf{A}_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \vdots & 0 \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} 0\\ \vdots\\ 0\\ 1 \end{bmatrix}, \quad \mathbf{C}_c = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

The preceding realization is called the *normal form* for the scalar input/output system $\dot{x} = f(x) + g(x)u$ of relative degree $r$.

The normal form of an affine system provides the usual starting point in the design of the constructive nonlinear controllers discussed in following sections. The reason for this is because it separates out that part of the nonlinear system whose algebraic structure is nearly linear from that part which is not. Since the linear part is easily controlled, this allows us to reduce the dimensionality of the harder nonlinear part. So the normal form provides a way to reduce the nonlinear control problem's size. Since it provides an important starting point for design, we'll find it convenient to present a couple of examples showing how to convert an affine system to its normal form.

**Example 1:** Consider the system

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2 - x_3\\ \dot{x}_2 &= -x_1x_3 - x_2 + u\\ \dot{x}_3 &= -x_1 + u\\ y &= x_3\end{aligned}$$

This is an affine system where

$$f(x) = \begin{bmatrix} -x_1 + x_2 - x_3 \\ -x_1 x_3 - x_2 \\ -x_1 \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad h(x) = x_3$$

We want to put this system in normal form about the origin, $x_0 = 0$.

We first find the system's relative degree by differentiating the output $y = x_3$,

$$\dot{y} = \dot{x}_3 = -x_1 + u$$

Since $u$ appears after one differentiation of the output the relative degree $r = 1$. This means that the first coordinate transformation is

$$z_1 = T_1(x) = x_3$$

We now need to find $T_2(x)$ and $T_3(x)$ such that

$$T(x) = \begin{bmatrix} T_1(x) \\ T_2(x) \\ T_3(x) \end{bmatrix} \text{ is a diffeomorphism about } x_0 = 0$$

A sufficient condition for such a $T$ is that the Jacobian

$$\frac{\partial T}{\partial x} = \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} \\ \frac{\partial T_2}{\partial x_1} & \frac{\partial T_2}{\partial x_2} & \frac{\partial T_2}{\partial x_3} \\ \frac{\partial T_3}{\partial x_1} & \frac{\partial T_3}{\partial x_2} & \frac{\partial T_3}{\partial x_3} \end{bmatrix}$$

is nonsingular at $x_0$. If this is the case then the inverse function theorem says there is a smooth inverse $T^{-1}(x)$ in the neighborhood of $x_0$ (i.e. $T$ is a diffeomorphism).

We also would like $T_2$ and $T_3$ to satisfy an orthogonality condition with respect to $g(x)$. This means that

$$\begin{aligned} 0 &= \frac{\partial T_2}{\partial x} g(x) = \left[ \begin{array}{ccc} \frac{\partial T_2}{\partial x_1} & \frac{\partial T_2}{\partial x_2} & \frac{\partial T_2}{\partial x_3} \end{array} \right] \left[ \begin{array}{c} 0 \\ 1 \\ 1 \end{array} \right] \\ &= \frac{\partial T_2}{\partial x_2} + \frac{\partial T_2}{\partial x_3} \end{aligned}$$

and similarly for $T_3$,

$$\begin{aligned} 0 &= \frac{\partial T_3}{\partial x} g(x) = \left[ \begin{array}{ccc} \frac{\partial T_3}{\partial x_1} & \frac{\partial T_3}{\partial x_2} & \frac{\partial T_3}{\partial x_3} \end{array} \right] \left[ \begin{array}{c} 0 \\ 1 \\ 1 \end{array} \right] \\ &= \frac{\partial T_3}{\partial x_2} + \frac{\partial T_3}{\partial x_3} \end{aligned}$$

The last condition we want is that $T$ maps $x_0 = 0$ (i.e. the origin) back to the origin in $z$=coordinates. In other words, we want $T(0) = 0$ since this means the equilibrium about which we are interested is preserved. This is done because generic certificates for Lyapunov or input-to-state stability are usually defined with respect to the origin.

Note that if we let

$$\begin{aligned} z_2 &= T_2(x) = x_1 \\ z_3 &= T_3(x) = x_2 - x_3 \end{aligned}$$

then we have the transformation,

$$T(x) = \left[ \begin{array}{c} x_3 \\ x_1 \\ x_2 - x_3 \end{array} \right] = \left[ \begin{array}{c} z_1 \\ z_2 \\ z_3 \end{array} \right]$$

whose Jacobian is

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} = \left[ \begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & -1 \end{array} \right]$$

This Jacobian is clearly nonsingular which means by the inverse function theorem that $T$ is a local diffeomorphism in a neighborhood around the origin.

We can also check and see that if the orthogonality condition is satisfied,

$$\begin{aligned}\frac{\partial T_2}{\partial x} &= \left.\frac{\partial T_2}{\partial x_2}\right|_{x=0} + \left.\frac{\partial T_2}{\partial x_3}\right|_{x=0} = 0 + 0 = 0\\ \frac{\partial T_3}{\partial x} &= \left.\frac{\partial T_3}{\partial x_3}\right|_{x=0} + \left.\frac{\partial T_3}{\partial x_3}\right|_{x=0} = 1 - 1 = 0\end{aligned}$$

So the orthogonality condition is satisfied and $T$ is a local diffeomophism. We need the inverse map $T^{-1}$ which is

$$T^{-1}(z) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} z_2 \\ z_1 + z_3 \\ z_1 \end{bmatrix}$$

So the normal form is defined with respect to the variables $\xi = z_1$ and $\eta = \begin{bmatrix} z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$. The dynamical equations for these variables are

$$\begin{aligned}\dot{\xi} &= \dot{x}_3 = -x_1 + u\\ &= \eta_1 + u\\ \dot{\eta}_1 &= \dot{x}_1 = -x_1 + x_2 - x_3\\ &= -z_2 + z_1 + z_3 - z_1\\ &= -z_2 + z_3\\ &= -\eta_1 + \eta_2\\ \dot{\eta}_2 &= \dot{z}_3 = \dot{x}_2 - \dot{x}_3\\ &= -x_1x_3 - x_2 + u - (-x_1 + u)\\ &= x_1 - x_2 - x_1x_3\\ &= z_2 - (z_1 + z_3) - z_2z_1\\ &= -z_1 + z_2 - z_3 - z_1z_2\\ &= -\xi + \eta_1 - \eta_2 - \xi\eta_1\end{aligned}$$

Combining these equations we see that the normal form for this affine system is

$$\begin{aligned}\dot{\xi} &= \eta_1 + u\\ \dot{\eta}_1 &= -\eta_1 + \eta_2\\ \dot{\eta}_2 &= -\xi + \eta_1 - \eta_2 - \xi\eta_1\end{aligned}$$

**Example 2:** The preceding example's PDE's were relatively easy to solve because they were linear PDEs. Let us consider a more interesting example where the PDE's are no longer linear. Consider the system

$$\begin{aligned}\dot{x}_1 &= -x_1 + \frac{2+x_3^2}{1+x_3^2}u\\ \dot{x}_2 &= x_3\\ \dot{x}_3 &= x_1x_3 + u\\ y &= x_2\end{aligned}$$

This system is affine in $u$ where

$$f(x) = \begin{bmatrix} -x_1 \\ x_3 \\ x_1x_3 \end{bmatrix}, \quad g(x) = \begin{bmatrix} \frac{2+x_3^2}{1+x_3^2} \\ 0 \\ 1 \end{bmatrix}, \quad h(x) = x_2$$

We want to put this in normal form about $x_0 = 0$.

We first determine the system's relative degree. Taking the first derivative of the output $y$ gives

$$\dot{y} = \dot{x}_2 = x_3$$

There is no $u$ in this equation, so we differentiate one more time,

$$\ddot{y} = \dot{x}_3 = x_1x_3 + u$$

The control, $u$, appears and so the relative degree $r = 2$. This means that the first two coordinate transformations are

$$\begin{aligned} z_1 &= T_1(x) = x_2 = h(x) \\ z_2 &= T_2(x) = x_3 = L_f h(x) \end{aligned}$$

We now need to find $T_3(x)$ so that

$$T(x) = \begin{bmatrix} T_1(x) \\ T_2(x) \\ T_3(x) \end{bmatrix} \text{ is a local diffeomorphism about } x_0 = 0$$

In other words we want the Jacobian of $T$ to be nonsingular at $0$ with the additional requirement that $T(0) = 0$ and

$$L_g T_3(x) = \frac{\partial T_3}{\partial x} g(x) = 0$$

The last condition (orthogonality) gives

$$\begin{aligned} 0 &= L_g T_3(x) = \begin{bmatrix} \frac{\partial T_3}{\partial x_1} & \frac{\partial T_3}{\partial x_2} & \frac{\partial T_3}{\partial x_3} \end{bmatrix} \begin{bmatrix} \frac{2+x_3^2}{1+x_3^2} \\ 0 \\ 1 \end{bmatrix} \\ &= \frac{\partial T_3}{\partial x_1} \frac{2 + x_3^2}{1 + x_3^2} + \frac{\partial T_3}{\partial x_3} \end{aligned}$$

Note that this is the PDE we need to solve to find $T_3$, but it is definitely not a linear PDE.

We can solve this PDE using the separation of variables technique. In particular we assume

$$T_3(x) = \ln\left(P(x_1)Q(x_3)\right)$$

where we now need to find the functions $P$ and $Q$. For convenience let $P'(x_1) = \frac{dP}{dx_1}$ and $Q'(x_3) = \frac{dQ}{dx_3}$. The partial derivatives for $T_3$ can therefore be written as

$$\begin{aligned}\frac{\partial T_3}{\partial x_1} &= \frac{1}{PQ}P'Q = \frac{P'}{P} \\ \frac{\partial T_3}{\partial x_3} &= \frac{Q'}{Q}\end{aligned}$$

If we insert this into the orthogonality PDE we get

$$\frac{Q'(x_3)}{Q(x_3)}\frac{1+x_3^2}{2+x_3^2} = -\frac{P'(x_1)}{P(x_1)} = K \quad \text{(constant)}$$

This holds since the PDE relationship must hold for all $x_1$ and $x_3$. This approach turns the PDE into a pair of ODE's

$$\frac{dP}{P} = -Kdx_1, \quad \Rightarrow \quad \ln P(x_1) = -Kx_1 + K_2$$

where $K_2$ is a constant of integration and

$$\frac{dQ}{Q} = K\left(1 + \frac{1}{1+x_3^2}\right)dx_3, \quad \Rightarrow \ln Q(x_3) = Kx_3 + K\tan^{-1}x_3 + K_3$$

and $K_3$ is another constant of integration.

We choose $K = 1$ and $K_2 = K_3 = 0$ so that

$$\begin{aligned}T_3(x) &= \ln P(x_1) + \ln Q(x_3) \\ &= -x_1 + x_3 + \tan^{-1}(x_3)\end{aligned}$$

Note that for this choice $T_3(0) = 0$ (i.e. the origin is preserved as the equilibrium in the $z$-system) and

$$\begin{aligned}\left.\frac{\partial T_3}{\partial x}\right|_0 &= \begin{bmatrix} -1 & 0 & 1+\frac{1}{1+x_3^2}\end{bmatrix}_{x=0} \\ &= \begin{bmatrix} -1 & 0 & 2\end{bmatrix}\end{aligned}$$

So the Jacobian of $T$ at the origin is

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 2\end{bmatrix}$$

Which means the Jacobian of $T$ about the origin is nonsingular and so by the inverse function theorem $T$ is a local diffeomorphism about the origin.

So the coordinate transformation $T$ is

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = T(x) = \begin{bmatrix} T_1(x) \\ T_2(x) \\ T_3(x) \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ -x_1 + x_3 + \tan^{-1} x_3 \end{bmatrix}$$

We now try to find the inverse transform is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = T^{-1}(z) = \begin{bmatrix} T_1^{-1}(z) \\ T_2^{-1}(z) \\ T_3^{-1}(z) \end{bmatrix}$$

Clearly

$$\begin{aligned} x_2 &= T_2^{-1}(z) = z_1 \\ x_3 &= T_3^{-1}(z) = z_2 \end{aligned}$$

We also know

$$\begin{aligned} z_3 &= -x_1 + x_3 + \tan^{-1} x_3 \\ &= -x_1 + z_2 + \tan^{-1} z_2 \end{aligned}$$

which implies

$$x_1 = -z_3 + z_2 + \tan^{-1} z_2$$

and so the inverse transform is

$$T^{-1}(z) = \begin{bmatrix} -z_3 + z_2 + \tan^{-1} z_2 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Now we write the normal form in terms of the variables $\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ and $\eta = z_3$. The dynamical equations are

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \dot{z}_2 = \dot{x}_3 = x_1x_3 + u \\ &= (-z_3 + z_2 + \tan^{-1}(z_2))z_2 + u \\ &= (-\eta + \xi_2 + \tan^{-1}(\xi_2))\xi_2 + u \end{aligned}$$

Finally,

$$\begin{aligned} \dot{\eta} &= \dot{z}_3 = -\dot{x}_1 + \dot{x}_3 + \frac{1}{1+x_3^2}\dot{x}_3 \\ &= x_1 - \left(1 + \frac{1}{1+x_3^2}\right) u + x_1x_3 + u + \frac{1}{1+x_3^2}(x_1x_3 + u) \\ &= x_1 + x_1x_3\left(1 + \frac{1}{1+x_3^2}\right) \\ &= x_1\left(1 + \frac{2+x_3^2}{1+x_3^2}x_3\right) \\ &= \left(-z_3 + z_2 + \tan^{-1} z_2\right)\left(1 + \frac{2+z_2^2}{1+z_2^2}z_2\right) \\ &= \left(-\eta + \xi_2 + \tan^{-1} \xi_2\right)\left(1 + \frac{2+\xi_2^2}{1+\xi_2^2}\xi_2\right) \\ y &= x_2 = \xi_1 \end{aligned}$$

Putting it all together gives us

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \left(-\eta + \xi_2 + \tan^{-1} \xi_2\right)\xi_2 + u \\ \dot{\eta} &= (-\eta + \xi_2 + \tan^{-1}(\xi_2))\left(1 + \frac{2+\xi_2^2}{1+\xi_2^2}\xi_2\right) \\ y &= \xi_1 \end{aligned}$$

The normal form is usually where one starts in using constructive nonlinear methods to design controllers. The normal form allows us to use our feedback linearizing transformation to decouple the $z$ states directly obtained by differentiating the output from the $z$ states used to fill out the diffeomorphism $T$'s final $n - r$ transformations. Note that that linear states $\xi$ are decoupled from the nonlinear states $\eta$, but the nonlinear states are driven by $\xi$. A major part of our future work will be to determine under what conditions the nonlinear part (also called the zero dynamics) is still stable. Another useful part of this is that since the linear control part is easily solved, it means that the harder nonlinear part is smaller (lower dimensionality). So the normal form provides a good basis for reducing the complexity of the nonlinear controller synthesis problem. In the next section we take closer look at the $\eta$-dynamics for this nonlinear part.

## 3. Zero Dynamics and Peaking

The normal form provides considerable insight into the structure of a scalar affine system. In particular, if we choose the feedback control

$$u = -\frac{1}{a(\eta,\xi)}\left(b(\eta,\xi) - v\right)$$

then the subsystem $\dot{\eta} = q(\eta,\xi)$ is disconnected from the $\xi$ states and are therefore unobservable from the output. In other words, the introduction of the I/O linearizing control law decomposes the system states into observable and unobservable states. This decomposition is shown more clearly in Fig. 5 where the right side shows the system before the feedback linearizing transformation and the left side shows the system after the feedback transformation.

Note that under the I/O feedback transformation the upper system on the left of Fig. 5 is a linear system. If we select $v = 0$ and initialize all $\xi_i = 0$ for $i = 1, \ldots, r$, then the output $y(t)$ will be identically zero for all future time. Since $\eta$ is unobservable from $y$, we can therefore conclude that no

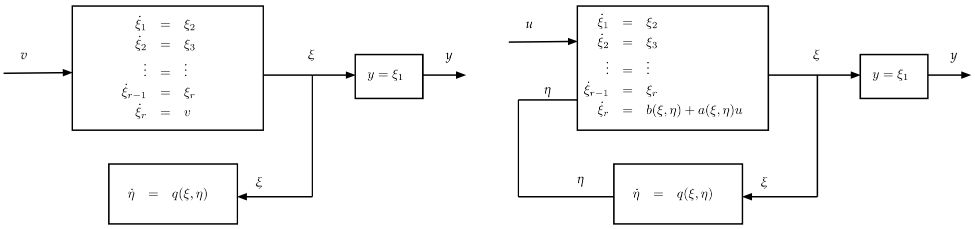


FIGURE 5. (right) normal form of scalar affine system (left) normal form after I/O feedback linearization

matter what the lower system is doing, $y$ will still be zero. It is customary, therefore, to refer to the dynamical system

$$\dot{\eta} = q(0, \eta)$$

as the *zero-dynamics* of the affine system.

Let us now consider an example illustrating how one computes the normal form for a system and then go ahead and find its zero dynamics. Let us consider the input-output system

$$\begin{aligned} \dot{x} &= \begin{bmatrix} x_3 - x_2^3 \\ -x_2 \\ x_1^2 - x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} u \\ y &= x_1 \end{aligned}$$

We begin by differentiating the output to determine the system's relative degree.

Differentiating the output yields,

$$\begin{aligned} \dot{y} &= x_3 - x_2^3 \\ \ddot{y} &= x_1^2 - x_3 + 3x_2^3 + (1 + 3x_2^3)u \end{aligned}$$

Since $u$ appears in the output after the second differentiation, we know the system has a relative degree $r = 2$. To determine the zero dynamics of this system we must first transform it to its normal form.

To obtain the normal form, we first take $z_1 = x_1 = y$ and take

$$z_2 = \dot{z}_1 = \dot{y} = x_3 - x_2^3$$

So the first two coordinate transformations giving the observable states are

$$\begin{aligned} T_1(x) &= x_1 = \xi_1 \\ T_2(x) &= x_3 - x_2^3 = \xi_2 \end{aligned}$$

To determine the last coordinate transformation we solve for $T_3$ and require tht

$$\begin{aligned} 0 = L_g T_3(x) &= \frac{\partial T_3}{\partial x} g(x) \\ &= \begin{bmatrix} \frac{\partial T_3}{\partial x_1} & \frac{\partial T_3}{\partial x_2} & \frac{\partial T_3}{\partial x_3} \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \end{aligned}$$

This last relation implies that $T_3$ must satisfy the following partial differential equation

$$\frac{\partial T_3}{\partial x_2} = \frac{\partial T_3}{\partial x_3}$$

This is separable and one solution would be

$$T_3(x) = x_2 + x_3$$

So the local normal form coordinates for this example are

$$\begin{aligned} \xi_1 &= z_1 = x_1 \\ \xi_2 &= z_2 = x_3 - x_2^3 \\ \eta &= z_3 = x_2 + x_3 \end{aligned}$$

The normal form equations (using $T^{-1}$ to be put everything in terms of $\eta$ or $\xi$) will therefore be

$$\begin{aligned}
\dot{\xi}_1 &= \dot{x}_1 = x_3 - x_2^3 \\
&= \xi_2 \\
\dot{\xi}_2 &= \dot{x}_3 - 3x_2^2\dot{x}_2 \\
&= x_1^2 - x_3 + u - 3x_2^2(-x_2 - u) \\
&= \left(x_1^2 + 3x_2^3 - x_3\right) + (1 + 3x_2^3)u \\
&= b(\xi,\eta) + a(\xi,\eta)u \\
\dot{\eta} &= \dot{x}_2 + \dot{x}_3 \\
&= -x_2 - u + x_1^2 - x_3 + u \\
&= x_1^2 - x_2 - x_3 \\
&= \xi_1^2 - \eta
\end{aligned}$$

The zero dynamics at $\xi = 0$ are given by

$$\dot{\eta} = -\eta$$

so this system is minimum phase.

In studying the stability of this affine system, we must not only ensure $\xi \to 0$, but we must also make sure that $\eta \to 0$ also. We select the control input $v$ to ensure the asymptotic stability of the $\xi$ states. If we also want $\eta \to 0$, the temptation might be to suppose that the zero-dynamics are also "stable". Unfortunately, this may not always be true. From linear systems we know that the cascade of stable linear systems will still be stable. But the following section examines a particular nonlinear system where this is not the case.

**Peaking Phenomenon:** Let us consider the cascade of two systems in which the driving system is an asymptotically stable system of the form

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\gamma^2 & -2\gamma \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

with $\gamma > 0$. The driven system has the state space realization

$$\dot{\eta} = -\frac{1}{2}(1+\xi_2)\eta^3$$

Note that when $\xi_2 = 0$, then the driven system's origin is also asymptotically stable. In other words, both the origin of both systems are "asymptotically stable" in the sense of Lyapunov (i.e. the state asymptotically goes to zero without external forcing). Fig. 6 shows the block diagram for this system

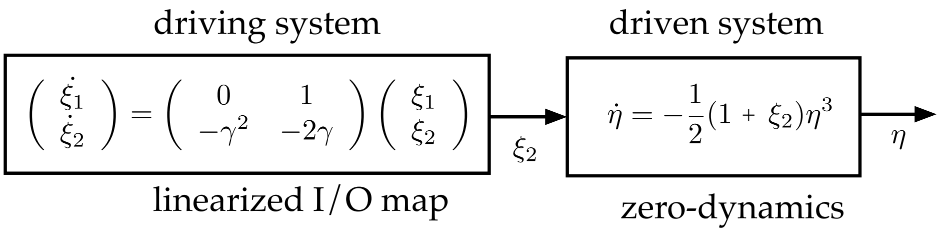

driving system
$\begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\gamma^2 & -2\gamma \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}$
linearized I/O map
$\xi_2$
driven system
$\dot{\eta} = -\frac{1}{2}(1+\xi_2)\eta^3$
zero-dynamics
$\eta$

FIGURE 6. System exhibiting Peaking Phenomenon

In this example, we can think of the driving system as the linearized I/O map after we've applied a control $v$ to stabilize that map. The driven system, of course, refers to the zero-dynamics. We know the driving system is asymptotically stable. The question is whether the driven system is also asymptotically stable when it is driven by the driving system?

To answer this question first note that the state transition matrix for the driving system is

$$\Phi(t) = \begin{bmatrix} (1+\gamma t)e^{-\gamma t} & te^{-\gamma t} \\ -\gamma^2 te^{-\gamma t} & (1-\gamma t)e^{-\gamma t} \end{bmatrix}$$

Note that the $(2,1)$ element has a $\gamma^2$ term so that for large enough $\gamma$, this term may have an extremely large peak. In particular, if we let $\xi_1(0) = 1$, $\xi_2(0) = 0$ with $\gamma = 10$, then the second state $\xi_2(t) = -\gamma te^{-\gamma t}$ has the behavior shown on the left of Fig. 7. There is a large negative excursion in $\xi_2$ before it asymptotically goes to zero. We refer to this phenomena as *peaking*.

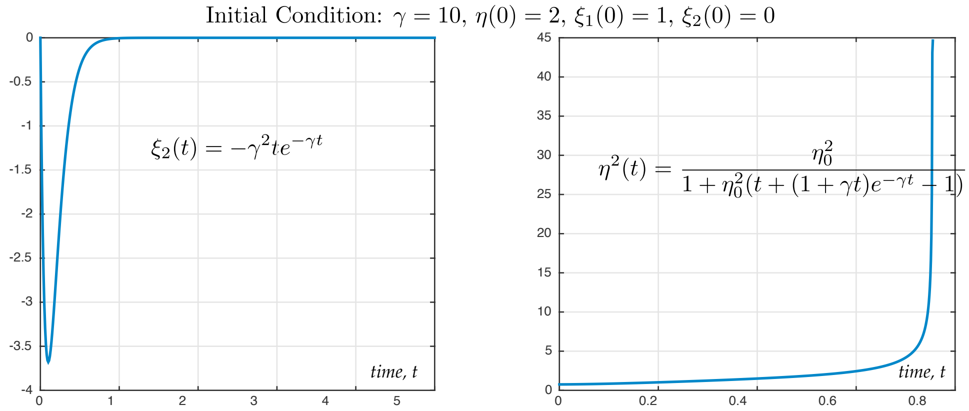

Initial Condition: $\gamma = 10$, $\eta(0) = 2$, $\xi_1(0) = 1$, $\xi_2(0) = 0$

FIGURE 7. Peaking in the driving system (left) triggers a finite escape in the driven system (right)

While $\xi_2$ is negative we see that $\eta(t)$ will be increasing. In fact, if we insert our closed form expression for $\xi_2(t)$ into the driven system's differential equation we get

$$\dot{\eta} = -\frac{1}{2}(1 + \gamma^2 t e^{-\gamma t})\eta^3$$

This ODE is separable and can therefore be integrated to get

$$\eta^2(t) = \frac{\eta^2(0)}{1 + \eta^2(0)(t + (1+\gamma t)e^{-\gamma t} - 1)}$$

Note that the denominator in the above equation may go to zero for finite $t$. So $\eta(t)$ may become unbounded even though both the origin of the driving and driven systems are asymptotically stable. In other words, the cascade is not asymptotically stable.

This particular example shows that it is not enough for the zero-dynamics to be asymptotically stable to ensure the stability of the entire system. In particular, we will need both systems to be *input-to-state stable* or ISS. From our results in chapter 4 we know that cascades of ISS stable systems are still ISS.

There is another potential issue with this feedback linearization strategy. In particular, it relies on the fact that we know $a(\xi, \eta)$ and $b(\xi, \eta)$ exactly so they can be cancelled out and replaced with a linear dynamic. In real-life, of course, such cancellations are never exact and this means that the I/O feedback linearized system may be more accurately written as

$$\begin{aligned}\dot{\xi}_1 &= \xi_2\\ \dot{\xi}_2 &= \xi_3\\ \vdots &= \vdots\\ \dot{\xi}_{r-1} &= \xi_r\\ \dot{\xi}_r &= v + \phi^T(\xi,\eta)\theta(t)\\ \dot{\eta} &= q(\xi,\eta)\end{aligned}$$

where $\phi(\xi,\eta)$ is a vector of known monomials and $\theta(t)$ is a vector of time-varying unknown parameters. In this case, the selection of $v$ must be done in a way that assures the robust stabilization of the upper $\xi$ system in the presence of the uncertain $\theta(t)$. The next chapter will investigate methods for the robust stabilization of such nonlinear systems.

The preceding discussion showed that while feedback linearization provides a powerful new linearization method for designing feedback control laws, it has the potential to be sensitive to peaking and inexact cancellation of the original system's nonlinearities. To deal with this issue we will develop a *constructive* approach to nonlinear control based on the robust stabilization of a scalar system and then showing how one can "backstep" that control through the chain of integrators seen in the system's normal form.

## 4. Control Lyapunov Functions

Control Lyapunov functions are used in our constructive approach to synthesizing nonlinear controllers. A control Lyapunov function (CLF) for a system

$$\dot{x}(t) = f(x(t), u(t))$$

is a $C^1$ positive definite, radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}$ such that when $x \neq 0$ we have

$$\inf_{u \in U} \frac{\partial V(x)}{\partial x} f(x,u) < 0 \tag{70}$$

where $U$ is a convex set of admissible values of the control variable, $u$. In other words, a CLF is a candidate Lyapunov function whose directional derivative can be forced to be negative definite by the choice of the control values. Note that if $f$ is continuous and there exists a continuous state feedback, $u = k(x)$, such that the origin is globally asymptotically stable then the standard converse theorems imply that a CLF exists. If $f$ is affine in the control variable, then one can show that the existence of a CLF is sufficeint for stabilizability via continuous state feedback. We can therefore conclude that just as the existence of a Lyapunov function is necessary and sufficient for the stability of a state-based system without inputs, so to is the existence of a CLF necessary and sufficient for the stabilizability of controlled systems in equation (70).

**Example:** Consider the second order system

$$\begin{aligned} \dot{x}_1 &= -x_1^3 + x_2\phi(x_1,x_2) \\ \dot{x}_2 &= u + \psi(x_1,x_2) \end{aligned}$$

where $\phi$ and $\psi$ are continuous functions and $u$ takes values in $U = \mathbb{R}$. The function $V(x_1,x_2) = \frac{1}{2}(x_1^2 + x_2^2)$ satisfies

$$\begin{aligned} \inf_{u \in U} \frac{\partial V(x)}{\partial x} f(x,u) &= \int_{u \in U} \left(-x_1^4 + x_1x_2\phi(x_1,x_2) + x_2u + x_2\psi(x_1,x_2)\right) \\ &= \begin{cases} -x_1^4 & \text{if } x_2 = 0 \\ -\infty & \text{if } x_2 \neq 0 \end{cases} \end{aligned}$$

So we can conclude $\frac{1}{2}|x|^2$ is a CLF of the system. This system is stabilizable since the control input

$$u(x_1,x_2) = -x_2 - \psi(x_1,x_2) - x_1\phi(x_1,x_2)$$

renders $\dot{V}(x) < 0$.

**Example:** Suppose there exists a diffeomorphism, $\xi = \Phi(x)$ with $\Phi(0) = 0$ which transforms the system in equation (70) into

$$\dot{\xi} = \mathbf{A}_c\xi + \mathbf{B}_c \left(a(\xi)u + b(\xi)\right)$$

where $(\mathbf{A}, \mathbf{B})$ is controllable and the functions $a$ and $b$ are continuous with $a$ being nonsingular for all $\xi$.. Let $\mathbf{P}$ be a symmetric positive definite matrix satisfying

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{B}^T\mathbf{P} + \mathbf{I} = 0$$

Then the function, $V(x) = \Phi(x)\mathbf{P}\Phi(x) = \xi^T\mathbf{P}\xi$ satisfies

$$\begin{aligned}
\inf_{u\in U} \frac{\partial V(x)}{\partial x} f(x,u) &= \inf_{uinU} \left[\xi^T \left(\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A}\right)\xi + 2\xi^T\mathbf{P}\mathbf{B}(a(\xi)u + b(\xi))\right] \\
&= \inf_{u\in U} \left[-\xi^T\xi + \xi^T\mathbf{P}\mathbf{B}\left(\mathbf{B}^T\mathbf{P}\xi + 2(a(\xi)u + b(\xi))\right)\right] \\
&= \begin{cases} -\xi^T\xi & \text{if } \xi^T\mathbf{P}\mathbf{B} = 0 \\ -\infty & \text{if } \xi^T\mathbf{P}\mathbf{B} \neq 0 \end{cases}
\end{aligned}$$

So this $V$ is a CLF for the system. As in the prior eample, we can also conclude that this system is globally asymptotically stabilizable. Note that we have come to this conclusion without having to actually construct a feedback linearizing control. The existence of a CLF allows us to deduce stabilizability without actually having to assume anything about the structure of controller.

## 5. Robust Stabilization of Scalar Nonlinear Systems

One method for the robust stabilization of nonlinear systems in normal form is the *constructive method*. The constructive method starts by robustly stabilizing a "scalar" subsystem in the original system and then backsteps that control through a chain of integrators to get a robust stabilizing control for the entire system. We say it is constructive because of its use of the backstepping strategy. This section presents the basic result (nonlinear damping theorem) used to robustly stabilize a scalar system. The next section presents backstepping strategies and illustrates their use on the 2D cart tracking example we considered earlier.

Let us consider a scalar system of the form

$$\dot{x} = f(x) + g(x)\left[u + \phi(x)^T w(t)\right]$$

where $u$ is a scalar control input, $w \in \mathbb{R}^p$ is a disturbance input, $x \in \mathbb{R}^n$, is the system state, and $\phi(x)$ is a $p$-vector of *known* smooth nonlinear functions. This system has a set of external disturbances, $w$, that are injected through a set of nonlinear functions $\phi_i(x)$. The objective is to find a control law $u$ that renders the system input-to-state stable.

We start by assuming there exists a controller $k_0(x)$ that uniformly asymptotically stabilizes (UAS) the origin when the disturbance $w = 0$. Because the controlled system is UAS, we can use one of the converse theorems to assert there is a $C^1$ positive definite function $V : \mathbb{R}^n \to \mathbb{R}$ and a positive definite $W : \mathbb{R}^n \to \mathbb{R}$ such that

$$\frac{\partial V}{\partial x}\left[f(x) + g(x)k_0(x)\right] \leq -W(x)$$

As we did when we studied the robust stability of LTI systems, we investigate how a perturbation of the nominally controlled system impacts the stability of the closed-loop system. So let us compute the Lie derivative of $V$ with respect to the perturbed system $\dot{x} = f + g(u + \phi^T w)$ to get

$$\dot{V} = \frac{\partial V}{\partial x}\left(f(x) + g(x)u\right) + \frac{\partial V}{\partial x}\phi^T(x)w$$

We now introduce a control $u$ of the form

$$u = k_0(x) - k\frac{\partial V}{\partial x}g(x)|\phi(x)|^2$$

where $k > 0$ and $k_0(x)$ is the stabilizing control for the undisturbed system. If we insert this $u$ into the above expression for $\dot{V}$, we get

$$\begin{aligned}\dot{V} &= \frac{\partial V}{\partial x}\left(f(x) + g(x)k_0(x)\right) - k\left(\frac{\partial V}{\partial x}g(x)\right)^2 |\phi(x)|^2 + \frac{\partial V}{\partial x}g(x)\phi^T(x)w \\ &\leq -W(x) - k\left(\frac{\partial V}{\partial x}g(x)\right)^2 |\phi(x)|^2 + \left|\frac{\partial V}{\partial x}g(x)\right| |\phi(x)| \, \|w\|_{\mathcal{L}_\infty}\end{aligned}$$

We now complete the square associated with the last two terms to get

$$\begin{aligned}\dot{V} &\leq -W(x) - k\left(\left|\frac{\partial V}{\partial x}g(x)\right| \phi(x)| - \frac{\|w\|_{\mathcal{L}_\infty}}{2k}\right)^2 + \frac{\|w\|^2_{\mathcal{L}_\infty}}{4k} \\ &\leq -W(x) + \frac{\|w\|^2_{\mathcal{L}_\infty}}{4k}\end{aligned}$$

We can therefore conclude that $\dot{V}$ is negative when $W(x) > \frac{\|w\|^2_{\mathcal{L}_\infty}}{4k}$. It can be shown that there exists a class $\mathcal{K}_\infty$ function $\gamma$ such that $\gamma(|x|) \leq W(x)$ when $W$ is positive definite. Since a class $\mathcal{K}$ function is invertible and its inverse is class $\mathcal{K}$, we can conclude that

$$|x| \geq \gamma^{-1}\left(\tfrac{\|w\|^2_{\mathcal{L}_\infty}}{4k}\right) \text{ implies that } \dot{V}(x) < 0$$

From the ISS theorem, we know this means that the controlled system is ISS with respect to $w$ and that $V$ is an ISS-Lyapunov function for the system.

The preceding control strategy is sometimes referred to as a *nonlinear damping* control since the control augments the Lyapunov stabilizing control $k_0(x)$ whose additional damping dominates the disturbance. This particular approach to robustly stabilizing nonlinear systems is useful enough to formalize as a theorem.

THEOREM 37. **(Nonlinear Damping Theorem)** *Consider a system whose state $x \in \mathbb{R}^n$ and control $u \in \mathbb{R}$ satisfy*

$$\dot{x} = f(x) + g(x)\left[u + \phi^T(x)w(t)\right]$$

*where $\phi(x)$ is a $(p \times 1)$ vector of known smooth nonlinear functions and $w(t)$ is a $p$-vector of disturbance inputs that are in $\mathcal{L}_\infty$. If there is a positive* *definite $C^1$ function $V : \mathbb{R}^n \to \mathbb{R}$ and positive definite $W : \mathbb{R}^n \to \mathbb{R}$ such that*

$$\dot{V} = \frac{\partial V(x)}{\partial x}\left[f(x) + g(x)k_0(x)\right] \leq -W(x)$$

*for some smooth $k_0 : \mathbb{R}^n \to \mathbb{R}$, then the control input*

$$u = k_0(x) - k\frac{\partial V}{\partial x}g(x)|\phi(x)|^2$$

*for any $k > 0$ renders the closed-loop system ISS with respect to disturbance input $w$.*

Let us introduce an example to illustrate how one might use these theorems. Consider the 2-wheeled robot example in Fig. 2 that we want to track a reference trajectory

$$\begin{aligned} \dot{x}_r &= v_r \cos\theta_r \\ \dot{y}_r &= v_r \sin\theta_r \\ \dot{\theta}_r &= \omega_r \end{aligned}$$

where we've specified $\theta_r(t)$ and $\omega_r(t)$.

We consider a *disturbed* version of this system in which there is a disturbance, $\overline{w}$, on the vehicle's turning rate, $\dot{\theta}$. The resulting equations of motion for this system therefore are

$$\begin{aligned} \dot{x} &= v_x \cos\theta \\ \dot{y} &= v_x \sin\theta \\ \dot{\theta} &= \omega + (\theta_r - \theta)^2\overline{w} \end{aligned}$$

where $v_x$ and $\omega$ are treated as controls. The disturbance $\overline{w}$ enters the system through the turning rate $\omega$-equation and is proportional to the squared error between the body angle and the reference trajectory.

Rather than working directly with physical variables, we transform these variables to measure the tracking error with respect to body coordinates.

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}$$

The error dynamics for the systems can therefore be shown to be

$$\begin{aligned}
\dot{e}_1 &= [-\sin\theta(x_r - x) + \cos\theta(y_r - y)]\dot{\theta} + \cos\theta\dot{x}_r - \cos\theta\dot{x} + \sin\theta\dot{y}_r - \sin\theta\dot{y} \\
&= e_2(\omega + e_3^2\overline{w}) + v_r(\cos\theta\cos\theta_r + \sin\theta\sin\theta_r) - \cos\theta(v_x\cos\theta) - \sin\theta(v_x\sin\theta) \\
&= e_2\omega + v_r\cos e_3 - v_x + e_2 e_3^2\overline{w} \\
\dot{e}_2 &= [-\cos\theta(x_r - x) - \sin\theta(y_r - y)]\,\dot{\theta} - \sin\theta\dot{x}_r + \sin\theta\dot{x} + \cos\theta\dot{y}_r - \cos\theta\dot{y} \\
&= -e_1(\omega + e_3^2\overline{w}) - \sin\theta(v_r\cos\theta_r) + \cos\theta v_r\sin\theta_r + \sin\theta(v_x\cos\theta) - \cos\theta(v_x\sin\theta) \\
&= -e_1\omega + v_r\sin e_3 - e_1 e_3^2\overline{w} \\
\dot{e}_3 &= \dot{\theta}_r - \dot{\theta} = \omega_r - \omega - e_3^2\overline{w}
\end{aligned}$$

Putting these equations in a form that highlights the affine nature of the system yields,

$$\begin{aligned}
\dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} &= \begin{bmatrix} v_r\cos e_3 + e_2\omega - v_x + e_2 e_3^2\overline{w} \\ v_r\sin e_3 - e_1\omega - e_1 e_3^2\overline{w} \\ \omega_r - \omega \end{bmatrix} \\
&= \begin{bmatrix} v_r\cos e_3 \\ v_r\sin e_3 \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \left( \begin{bmatrix} v_x \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ e_3^2 \end{bmatrix} \overline{w} \right) \\
&= f(e) + g_1(e)v_x + g_2(e)(\omega + e_3^2\overline{\omega})
\end{aligned}$$

where $u = \begin{bmatrix} v_x \\ \omega \end{bmatrix}$ is the control vector.

The preceding equations of motion are in the form needed to apply the nonlinear damping theorem. To start, however, we first need to identify a control that renders the origin of the error system asymptotically stable in the absence of a disturbance. Consider the following control

$$u = \begin{bmatrix} k_{01}(e) \\ k_{02}(e) \end{bmatrix} = \begin{bmatrix} v_x \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix}$$

where $k_1$, $k_2$, and $k_3$ are positive control gains. This control can be shown to asymptotically stabilize the origin through the following candidate Lyapunov function

$$V(e) = \frac{k_2}{2} e_1^2 + \frac{k_2}{2} e_2^2 + (1 - \cos e_3)$$

in which $|e_3| < \frac{\pi}{2}$. Computing the Lie derivative of $V$ with respect to the undisturbed (i.e. $\overline{w} = 0$) vector field yields,

$$\begin{aligned} \dot{V} &= k_2 e_1 \dot{e}_1 + k_2 e_2 \dot{e}_2 + \sin e_3 (\dot{e}_3) \\ &= k_2 e_1 (\omega e_2 - k_1 e_1) + k_2 e_2 (-\omega e_1 + v_r \sin e_3) + \sin e_3 (-k_2 v_r e_2 - k_3 v_r \sin e_3) \\ &= -k_1 k_2 e_1^2 - k_3 v_r \sin^2 e_3 \end{aligned}$$

Notice that $\dot{V} = 0$ for all $x$ where $e_1 = e_3 = 0$, but $e_2$ can be anything. So we can only use the direct Lyapunov theorem to infer Lyapunov stability because $\dot{V}$ is only negative semi-definite. It is still possible to deduce asymptotic stability, but we will need to use the *invariance principle*.

To use this theorem in our preceding example, we first note that under the proposed controls that the system equations are

$$\dot{e} = \begin{bmatrix} \omega e_2 - k_1 e_1 \\ -\omega e_1 + v_r \sin e_3 \\ -k_2 v_r e_2 - k_3 v_r \sin e_3 \end{bmatrix}$$

We note that the set

$$E = \left\{ e \in \mathbb{R}^3 \,:\, e_1 = e_3 = 0 \right\}$$

Let us consider a point $e \in E$ such that $e_2(0) \neq 0$. Then one may see that

$$\dot{e}_1(0) = e_2(0)\omega \text{ and } \dot{e}_3(0) = -k_2 v_r e_2(0)$$

since $v_r > 0$, we see that $\dot{e}_3 \neq 0$ and so the state trajectory will not remain in $E$. So the largest invariant set in $E$ is the origin and so by the invariance theorem we know that the origin of this controlled system is asymptotically stable.

Note that the system is not exactly what we described in the nonlinear damping theorem for the control is $u$ is a vector, rather than a scalar. In reviewing the proof of this theorem, however, it should be apparent that because our disturbance only enters a single component of $u$ that the original proof can be slightly modified to account for this extra control. As a result, a slight modification of the nonlinear damping theorem allows us to robustly stabilize the disturbed system using the velocity control $v_x = k_{01}(e)$ and the turning control

$$\omega = k_{02}(e) - k\frac{\partial V}{\partial e} g_2(e)|\phi(e)|^2$$

Since

$$\frac{\partial V}{\partial e} g_2(e) \quad = \quad \left[\begin{array}{ccc} k_2 e_1 & k_2 e_2 & \sin e_3 \end{array}\right] \left[\begin{array}{c} e_2 \\ -e_1 \\ -1 \end{array}\right] = -\sin e_3$$

the turning control can therefore be written as

$$\omega = k_{02}(e) + k e_3^4 \sin e_3$$

Let us now simulate the proposed control laws and see how well they perform on the 2-d robot tracking problem. For completeness the script used in this simulation is shown below. The stabilizing gains $k_1$, $k_2$, and $k_3$ are all chosen to be $1$ and the gain, $k$, on the nonlinear damping term is either $0$ or $1$ (i.e. damping is either "off" or "on"). The disturbance we use in this simulation is a worst-case simulation that appears as a constant torque of value $-1$. The reference trajectory was generated by velocity and angle rates that followed $v_r(t) = 0.1\cos(t) + 0.2$ and $\omega_r(t) = 0.5\sin(t)$.

We ran two simulations. The first simulation was run over the time interval $[0, 2.75]$ with the nonlinear damping gain $k = 0$. This simulation, therefore, corresponds to the controller that is known to stabilize the system when there is not disturbance, but does not guarantee stability when there is a disturbance. If the disturbance $\overline{w} = 0$, the origin of the error system is indeed asymptotically stable. If, however, we let $\overline{w} = -1$, then the simulation results in Fig. 8 exhibit a finite escape time at about $t = 2.75$ seconds. The left hand plots on Fig. 8 indeed show that this finite escape of $e_3$ and show commanded turning rate $\omega$ beginning to spin the vehicle around. The right hand plot reference trajectory and vehicle trajectory in the position space, $\mathbb{R}^2$. This plot shows that eventually the vehicle stops making forward progress because the system was not stable and therefore began spinning around.

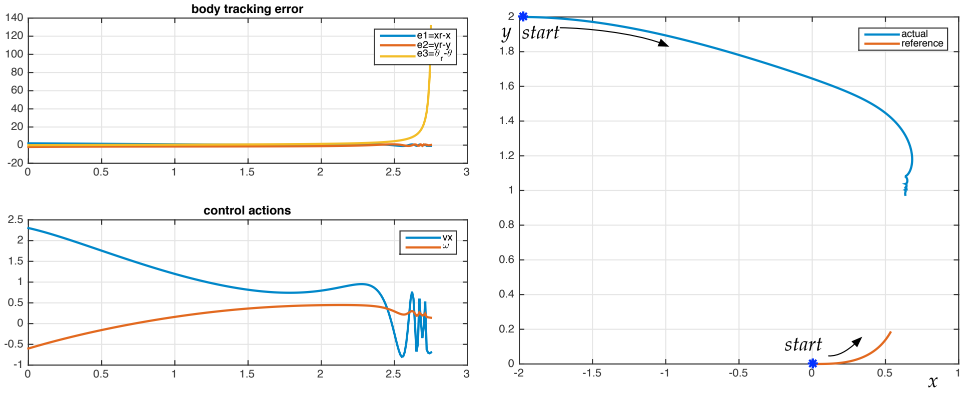


FIGURE 8. Simulation Results for Disturbed Vehicle Tracking Problem using the Lyapunov stabilizing control $v_x = k_{01}(e)$ and $\omega = k_{02}(e)$.

The whole point of the preceding discussion was to show that we can stabilize this system in the presence of the constant disturbance by adding a nonlinear damping term. In this case, we simulate the system with the nonlinear damping gain $k = 1$. The simulation was run over the time interval $[0, 50]$ because we were able to avoid the finite escape that was seen in the simulation from Fig. 8. The left hand plots of Fig. 9 indeed show that the tracking errors $e$ converge asymptotically to the origin and that the control effort remains bounded as the system tracks the sinusoidal reference trajectory. The right hand plot of Fig. 9 shows that the vehicle's actual trajectory in position space $\mathbb{R}^2$ indeed asymptotically converges and tracks the reference trajectory for the system. This simulation example therefore demonstrates that the nonlinear damping theorem achieves robust stabilization in the presence of disturbances.

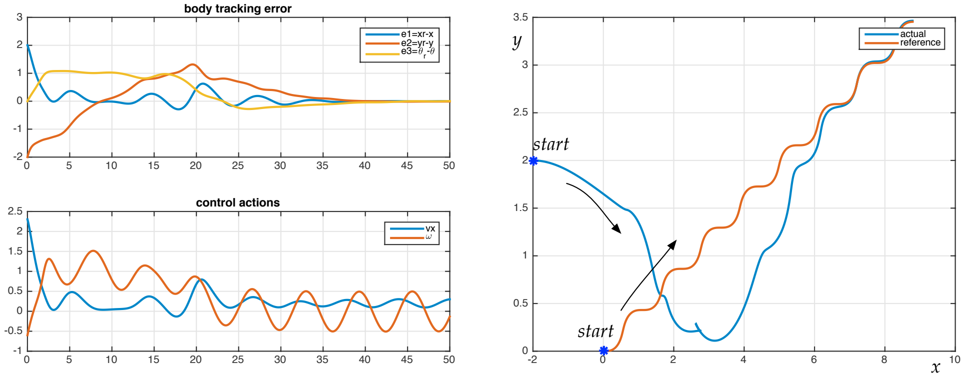


FIGURE 9. Simulation Results for Disturbed Vehicle Tracking Problem using the Robustly stabilizing control $v_x = k_{01}(e)$ and $\omega = k_{02}(e) + ke_3^4 \sin(e_3)$.

## 6. Backstepping Control Strategies

The preceding section assumed that the 2-DOF vehicle could be controlled by directly specifying the vehicle's velocity, $v_x$, and its turning rate $\omega$. In reality, however, these control inputs are forces and torques and this means that the real system equations are of the form

$$\begin{aligned}
\dot{x} &= v_x \cos\theta \\
\dot{y} &= v_x \sin\theta \\
\dot{\theta} &= \omega + (\theta_r - \theta)^2 \overline{w} \\
\dot{v} &= F \\
\dot{\omega} &= T
\end{aligned}$$

where the actual control inputs are the applied force, $F$, and the torque $T$. In this case we cannot use the control to directly dominate system nonlinearities and uncertainties.

For such systems, we can employ a stabilization strategy known as *backstepping*. Under backstepping, we consider the system

$$\begin{aligned} \dot{z} &= f(z) + g(z)\xi \\ \dot{\xi} &= u \end{aligned} \tag{71}$$

where $u$ is the control and $f(0) = 0$. We're interested in finding a state feedback control that stabilizes the origin. We approach this problem by viewing it as a *cascade* of two systems in which the driving system is an integrator with state $\xi$ and the driven system is the affine nonlinear system $\dot{z} = f(z) + g(z)\xi$. The idea behind backstepping is to treat the state variable $\xi$ in the driven system as a *virtual control*. We determine a virtual control to stabilize the driven system and then use the Lyapunov function for that stabilized system to develop a composite Lyapunov function for the entire system which serves as the basis for synthesizing the full system's controller.

Note that the preceding discussion revolved about asymptotic stabilization of a system without external disturbances. But the backstepping construction can also be extended to disturbed systems. In this case, however, we use the backstepping procedure to develop an ISS Lyapunov function from which one can synthesize a robustly stabilizing controller. In our 2-DOF cart system, we will use this robust form of backstepping. But before doing that we consider the somewhat simpler case in which backstepping is used to simply stabilize the system in equation (71).

Let us therefore consider the system in equation (71). Suppose there is a control $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ that stabilizes the upper system about the origin. In other words the origin of the system

$$\dot{z} = f(z) + g(z)\phi(z)$$

is asymptotically stable. We refer to $\phi$ as the *virtual control* for the upper system. Suppose further that for this virtual control we know that the positive definite $C^1$ function $V : \mathbb{R}^n \to \mathbb{R}$ is a Lyapunov function that satisfies

$$\frac{\partial V(z)}{\partial z}\left(f(z) + g(z)\phi(z)\right) \leq -W(z)$$

where $W : \mathbb{R}^n \to \mathbb{R}$ is positive definite.

We want to find a way of generating the virtual control $\phi(z)$ from the lower system's control $u$ such that the origin of the entire cascade is asymptotically stable. To do this we take rewrite the original system (71) as

$$\begin{aligned}\dot{z} &= [f(z) + g(z)\phi(z)] + g(z)\,[\xi - \phi(z)] \\ \dot{\xi} &= u\end{aligned}$$

This equation treats $\xi - \phi(z)$ as the *error* between the actual input, $\xi$, and the virtual control, $\phi(z)$, that we already know stabilizes the upper system.

Let us now introduce a change of variables to explicitly characterize this control error

$$y = \xi - \phi(z) \tag{72}$$

and in terms of this new variable the original system (71) becomes

$$\begin{aligned}\dot{z} &= [f(z) + g(z)\phi(z)] + g(z)y \\ \dot{y} &= u - \dot{\phi}(z) \\ &= u - \frac{\partial \phi(z)}{\partial z}\,[f(z) + g(z)\xi]\end{aligned}$$

Let us now consider the following composite candidate Lyapunov function that is obtained by augmenting our original $V(z)$ with the backstepping variable $y$

$$V_c(z, \xi) = V(z) + \frac{1}{2}y^2 = V(z) + \frac{1}{2}(\xi - \phi(z))^2$$

The directional derivative of $V_c$ is

$$\begin{aligned}\dot{V}_c &= \frac{\partial V}{\partial z}\left[f(z)+g(z)\phi(z)\right]+\frac{\partial V}{\partial z}g(z)y+yv \\ &= -W(z)+\frac{\partial V}{\partial z}g(z)y+yv\end{aligned}$$

where $v = u - \dot{\phi}(z)$. The variable $v$ may be thought of as a control variable transformation. If we take the new control, $v$, to be

$$v = -\frac{\partial V}{\partial z}g(z) - ky$$

where $k > 0$ then inserting this into our expression for $\dot{V}_c$ yields,

$$\dot{V}_c \leq -W(z) - ky^2$$

which implies that under this particular control, the origin $(z, y) = (0, 0)$ for the cascaded system is asymptotically stable. Remember that we need to transform $v$ back to the orignal control variable $u$,

$$\begin{aligned}u &= v+\dot{\phi}(z) \\ &= -\frac{\partial V}{\partial z}g(z)-k(\xi-\phi(z))+\frac{\partial \phi(z)}{\partial z}\left(f(z)+g(z)\xi\right)\end{aligned}$$

This result is important enough to formalize as a theorem

THEOREM 38. **(Backstepping Theorem)** *Consider the system*

$$\begin{aligned}\dot{z} &= f(z)+g(z)\xi \\ \dot{\xi} &= u\end{aligned}$$

*Let $\phi : \mathbb{R}^n \to \mathbb{R}$ with $\phi(0) = 0$ stabilze the origin of $\dot{z} = f(z) + g(z)\phi(z)$. Let $V : \mathbb{R}^n \to \mathbb{R}$ be a $C^1$ positive definite function such that*

$$\frac{\partial V}{\partial z}\left[f(z)+g(z)\phi(z)\right] \leq -W(z)$$

*for some positive definite $W : \mathbb{R}^n \to \mathbb{R}$. Then the feedback law*

$$u = \frac{\partial \phi(z)}{\partial z}\left[f(z)+g(z)\xi\right]-\frac{\partial V(z)}{\partial z}g(z)-k(\xi-\phi(z))$$

*for $k > 0$ stabilizes the origin of the original system.*

The backstepping procedure is illustrated in the following example. Consider the system

$$\begin{aligned}\dot{z} &= z^2 - z^3 + \xi \\ \dot{\xi} &= u\end{aligned}$$

The upper system is

$$\dot{z} = z^2 - z^3 + \xi$$

and we consider the virtual control

$$\phi(z) = -z^2 - z$$

and use the Lyapunov function $V(z) = \frac{1}{2}z^2$ to show that the origin of the upper system is asymptotically stable.

The backstepping change of variables is

$$y = \xi - \phi(z) = \xi + z + z^2$$

which transforms the full system to

$$\begin{aligned}\dot{z} &= -z - z^3 + y \\ \dot{y} &= u + (1+2z)(-z - z^3 + y)\end{aligned}$$

So we can directly use backstepping theorem, or simply use

$$V_c(z,y) = \frac{1}{2}z^2 + \frac{1}{2}y^2$$

as the control Lyapunov function. To find the control, we compute

$$\begin{aligned}\dot{V}_c &= z\dot{z} + y\dot{y} \\ &= z(-z - z^3 + y) + y\big(u + (1+2z)(-z - z^3 + y)\big) \\ &= -z^2 - z^4 \\ &\quad +y(u + z + (1+2z)(-z - z^3 + y))\end{aligned}$$

Clearly just select $u$ to cancel out the last two terms and introduce a $y$ feedback,

$$u = -ky - z - (1+2z)(-z - z^3 + y)$$

which would simplify $\dot{V}_c$ to

$$\dot{V}_c = -z^2 - z^4 - y^2 < 0$$

thereby certifying the asymptotic stability of the full system.

We now apply the backstepping theorem to our earlier 2-DOF robotic system. In our earlier work we derived a “robust stabilizing control”, but because our backstepping theorem was developed for an asymptotically stabilizing control, we will start from there. In that case the asymptotically stabilizing control law assuming $v_x$ and $\omega$ are the control variables will be

$$\left[\begin{array}{c} v_x \\ \omega \end{array}\right] = \left[\begin{array}{c} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{array}\right]$$

This control had the associated Lyapunov function

$$V(e) = \frac{k_2}{2} e_1^2 + \frac{k_2}{2} e_2^2 + (1 - \cos e_3)$$

The system we wish to control, however, has $v_x$ and $\omega$ as state variables determined by the state equations $\dot{v}_x = F$ and $\dot{\omega} = T$ where $F$ and $T$ are the control inputs. So following the backstepping theorem, we treat our earlier control as a ”virtual control

$$\phi(e) = \left[\begin{array}{c} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{array}\right]$$

for the augmented system

$$\begin{array}{rcl} \dot{e} & = & f(e) + g(e)\xi \\ \dot{\xi} & = & u \end{array}$$

where

$$f(e) = \left[\begin{array}{c} v_r \cos e_3 \\ v_r \sin e_3 \\ \omega_r \end{array}\right], \quad g(e) = \left[\begin{array}{cc} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{array}\right], \quad \xi = \left[\begin{array}{c} v_x \\ \omega \end{array}\right], \quad u = \left[\begin{array}{c} F \\ T \end{array}\right]$$

From the backstepping theorem, we then know that the stabilizing control will be

$$u = \frac{\partial \phi(e)}{\partial e}\left[f(e) + g(e)\xi\right] - \left[\frac{\partial V(e)}{\partial e} g(e)\right]^T - k_b(\xi - \phi(e))$$

with $k_b > 0$.

We simulated this backstepping control with the following script. This nearly identical to the earlier script we used with the main exception being the formulation of the backstepping control signal, $u$, and the extra two states in the system equations.

This simulation tracked a "slower" reference trajectory and the results are shown in Fig. 10. The plots on the left side show the time history for the body tracking error and the control effort. As expected errors asymptotically converge to zero. Though if the reference trajectory were oscillating faster then the cart actually has some trouble tracking with zero error. The right side shows the vehicle and reference position in $\mathbb{R}^2$, which also shows that we are tracking the reference.

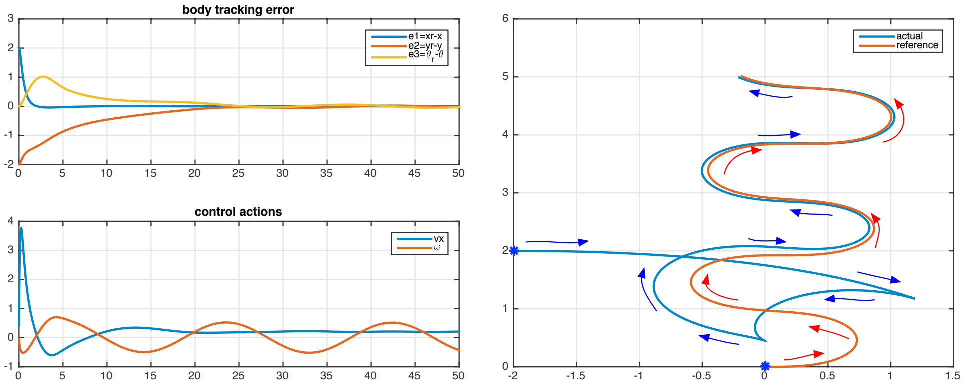


FIGURE 10. simulation results for vehicle tracking using a backstepping control

The preceding simulation was for a backstepping control based on a control that asymptotically stabilized the base system. It should be noted that we could have also developed a backstepping theorem for the disturbed system with very little additional effort. The only difference being that we would be working with ISS-Lyapunov functions and the nonlinear damping theorem. The associated backstepping under uncertainty result is stated below in the following theorem.

THEOREM 39. **(Backstepping under Uncertainty)** *Consider the system*

$$\dot{z} = f(z) + g(z)u + F_0(z)w_0$$

*where $u$ is a scalar control and $F_0(x)$ is a matrix of known smooth nonlinearities and $w_0$ is a vector of uniformly bounded disturbances. Suppose there exists a feedback control $u = \phi(z)$, $C^1$ positive definite functions $V : \mathbb{R}^n \to \mathbb{R}$ and $W : \mathbb{R}^n \to \mathbb{R}$, and positive constant $b > 0$ such that*

$$\frac{\partial V(z)}{\partial z}\left[f(z) + g(z)\phi(z) + F_0(z)w_0(t)\right] \leq -W(z) + b$$

*Then the control*

$$\begin{aligned} u &= c[\xi - \phi(z)] + \frac{\partial \phi(z)}{\partial z}\left[f(z) + g(z)\xi\right] - \frac{\partial V(z)}{\partial z} g(z) \\ &\quad -k(\xi - \phi(z))\left[|F_1(z,\xi)|^2 + \left|\frac{\partial \phi(z)}{\partial z} F_0(z)\right|^2\right] \end{aligned}$$

*will ensure that $z$ and $\xi$ are globally uniformly bounded in the augmented system*

$$\begin{aligned} \dot{z}(t) &= f(z) + g(z)\xi + F_0(z)w_0(t) \\ \dot{\xi}(t) &= u + F_1(z,\xi)w_1(t) \end{aligned}$$

*where $w_1$ is a uniformly bounded disturbance and $F_1(z,\xi)$ is a matrix of known nonlinearities.*

**Proof:** We use the backstepping variable

$$y = \xi - \phi(z)$$

to rewrite the entire system as

$$\begin{aligned} \dot{z} &= f(z) + g(z)\left[\phi(z) + y\right] + F_0(z)w_0 \\ \dot{y} &= u + F_1(z,\xi)w_1(t) - \frac{\partial \phi(z)}{\partial z}\left[f(z) + g(z)\xi + F_0(z)w_0\right] \end{aligned}$$

We then consider the composite candidate ISS-Lyapunov function

$$V_c(z,\xi) = V(z) + \frac{1}{2}(\xi - \phi(z))^2 = V(z) + \frac{1}{2}y^2$$

and compute its directional derivative to get

$$\begin{aligned}
\dot{V}_c &= \frac{\partial V(z)}{\partial z}(f(z) + g(z)\phi(z) + F_0(z)w_0) + \frac{\partial V(z)}{\partial z}g(z)y \\
&\quad + y\left[u + F_1(z,\xi)w_1 - \frac{\partial \phi(z)}{\partial z}(f(z) + g(z)\xi + F_0(z)w_0)\right] \\
&= \frac{\partial V}{\partial z}(f(z) + g(z)\phi(z) + F_0(z)w_0) + y\left[u - \frac{\partial \phi(z)}{\partial z}(f(z) + g(z)\xi) + \frac{\partial V}{\partial z}g(z)\right] \\
&\quad + y\left[F_1(z,\xi)w_1 - \frac{\partial \phi(z)}{\partial z}F_0(z)w_0\right] \\
&\leq -W(z) + b - cy^2 - ky^2\left[|F_1(z,\xi)|^2 + \left|\frac{\partial \phi}{\partial z}F_0(z)\right|^2\right] \\
&\quad + |y|\,|F_1(z,\xi)|\|w_1\|_{\mathcal{L}_\infty} + |y|\left|\frac{\partial \phi}{\partial z}F_0(z)\right|\|w_0\|_{\mathcal{L}_\infty} \\
&\leq -W(z) - cy^2 + b + \frac{\|w_0\|^2_{\mathcal{L}_\infty}}{4k} + \frac{\|w_1\|^2_{\mathcal{L}_\infty}}{4k}
\end{aligned}$$

which shows that $V_c$ is an ISS-Lyapunov function for the combined system. $\diamondsuit$

Backstepping, as described above, pertains to a single integrator driving another system. The method can be extended to longer chains of integrators and even linear systems in a relatively straightforward manner. This is done by exploiting the inherent modularity in the approach. Consider, for example, the system

$$\begin{aligned}
\dot{z}_1 &= z_1^2 - z_1^3 + z_2 \\
\dot{z}_2 &= \xi \\
\dot{\xi} &= u
\end{aligned}$$

The driven system is a second order system that can be easily stabilized through a virtual control

$$\phi_1(z_1) = -z_1 2 - z_1$$

that cancels the quadratic nonlinearity and adds an additional negative linear term. In this manner, one can assure the origin is exponential and so we can use the Lyapunov function $V_1(z_1) = z_1^2/2$ whose directional derivative,

$$\dot{V}_1 = z_1 \dot{z}_1 = -z_1^2 - z_1^4 < 0$$

clearly certifies the origin of the $z_1$ system is stable if $z_2$ equals $\phi(z_1)$.

To develop the backstepping control for this system driven by a chain of two integrators, we first use our earlier backstepping procedure to develop a backstepping control for the top two equations,

$$\begin{aligned} \dot{z}_1 &= z_1^2 - z_1^3 + z_2 \\ \dot{z}_2 &= \xi \end{aligned}$$

where the virtual control used for $\xi$ is

$$\phi_2(z_1, z_2) = -z_1 + (-1 + 2z_1)(z_1^2 - z_1^3 + z_2) - (z_2 + z_1 + z_1^2)$$

The Lyapunov function for this combined system is

$$V_2(z_1, z_2) = \frac{1}{2} z_1^2 + \frac{1}{2}(z_2 + z_1 + z_1^2)^2$$

We now perform a second backstep by first introducing the following change of variables

$$y = \xi - \phi_2(z_1, z_2)$$

to obtain

$$\begin{aligned} \dot{z}_1 &= z_1^2 - z_1^3 + z_2 \\ \dot{z}_2 &= \phi_2(z_1, z_2) + y \\ \dot{y} &= u - \frac{\partial \phi_2}{\partial z_1}(z_1^2 - z_1^3 + z_2) - \frac{\partial \phi_2}{\partial z_2}(\phi_2 + y) \end{aligned}$$

The control Lyapunov function constructed from $V_2$ is

$$V_c(z_1, z_2, y) = V_2(z_1, z_2) + \frac{1}{2} y^2$$

whose directional derivative is

$$\begin{aligned}\dot{V}_c &= \frac{\partial V_2}{\partial z_1}(z_1^2 - z_1^3 + z_2) + \frac{\partial V}{\partial z_2}(y + \phi_2) \\ &\quad +y\left[u - \frac{\partial \phi_2}{\partial z_1}(z_1^2 - z_1^3 + z_2) - \frac{\partial \phi_2}{\partial z_2}(y + \phi_2)\right] \\ &= -z_1^2 - z_1^4 - (z_2 + z_1 + z_1^2)^2 \\ &\quad +y\left[\frac{\partial V_2}{\partial z_2} - \frac{\partial \phi_2}{\partial z_1}(z_1^2 - z_1^3 + z_2) - \frac{\partial \phi_2}{\partial z_2}(y + \phi_2) + u\right]\end{aligned}$$

We then take $u$ to cancel out the undesirable terms in the last line to obtain

$$u = -\frac{\partial V_2}{\partial z_2} + \frac{\partial \phi_2}{\partial z_1}(z_1^2 - z_1^3 + z_2) + \frac{\partial \phi_2}{\partial z_2}(y + \phi_2) - y$$

This forces $\dot{V}_c < 0$ and thereby stabilizes the origin of the entire cascade.

The preceding discussion assumed the diving lower systems were integrator chains, but one can also simply assume that they are linear systems. In particular, assume the full system has the form

$$\begin{aligned}\dot{z} &= f(z) + g(z)y \\ \dot{\xi} &= \mathbf{A}\xi + bu \\ y &= c^T\xi\end{aligned}$$

We assume that the driving system is globally stable when $\xi = 0$ with a Lyapunov function $V$ such that

$$\dot{V} = \frac{\partial V}{\partial z} f(z) \leq -W(z) < 0$$

for some positive definite $W$. The problem is to find a linear control that stabilizes the driving linear system without destabilizing the driven nonlinear system. Note that the stability of the lower linear system does not always guarantee the cascaded full system is stable. An example of this was seen in the peaking phenomenon.

Let us assume $(\mathbf{A}, b)$ is stabilizable so there exists a matrix $\mathbf{K}$ such that $\mathbf{A} + b\mathbf{K}$ is Hurwitz. This means there exist matrices $\mathbf{P} = \mathbf{P}^T > 0$ and

$\mathbf{Q} = \mathbf{Q}^T > 0$ such that

$$(\mathbf{A} + b\mathbf{K})^T\mathbf{P} + \mathbf{P}(\mathbf{A} + b\mathbf{K}) = -\mathbf{Q}$$

We then consider the backstepping control Lyapunov for the entire cascade,

$$V_c(z, \xi) = V(z) + \xi^T\mathbf{P}\xi$$

and consider a control of the form

$$u = \mathbf{K}\xi + v$$

Note that we are using the stabilizing linear control for the lower system with an additional term $v$ that is intended to stabilize the entire cascade. To determine this $v$, let us example the directional derivative of $V_c$,

$$\begin{aligned}
\dot{V}_c(z, \xi) &= \dot{V}(z) + \dot{\xi}^T\mathbf{P}\xi + \xi\mathbf{P}\dot{\xi} \\
&= -W(z) + \frac{\partial V}{\partial z}g(z)y \\
&\quad +\xi^T(\mathbf{P}(\mathbf{A} + b\mathbf{K}) + (\mathbf{A} + b\mathbf{K})^T\mathbf{P})\xi + 2\xi^T\mathbf{P}bv \\
&\leq -W(z) = \xi^T\mathbf{Q}\xi + \frac{\partial V}{\partial z}g(z)y + 2\xi^T\mathbf{P}bv
\end{aligned}$$

We are interested in the last two terms and note that if we select $v$ to cancel the term $\frac{\partial V}{\partial z}g(z)y$, then $\dot{V}_c < 0$.

To select $v$ in this way, we require first that $\mathbf{P}$ satisfy the Lyapunov equation,

$$(\mathbf{A} + b\mathbf{K})^T\mathbf{P} + \mathbf{P}(\mathbf{A} + b\mathbf{K}) = -\mathbf{Q}. \tag{73}$$

In addition to this, however, we also require

$$\mathbf{P}b = c \tag{74}$$

This last condition holds when the linear system is said to be *feedback positive real*.

Assuming that we can find $\mathbf{P}$ satisfies equations (73- 74), then we can see that

$$\begin{aligned}\dot{V}_x(z,\xi) &\leq -W(z) - \xi^T\mathbf{Q}\xi + \frac{\partial V}{\partial z}g(z)y + 2\xi^T\mathbf{P}bv \\ &= -W(z) - \xi^T\mathbf{Q}\xi + \frac{\partial V}{\partial z}g(z)y + 2\xi^T cv \\ &= -W(z) - \xi^T\mathbf{Q}\xi + \frac{\partial V}{\partial z}g(z)y + 2\xi^T yv\end{aligned}$$

Since $y$ and $v$ are scalars, we can annihilate the last two terms by letting

$$v = -\frac{1}{2}\frac{\partial V}{\partial z}g(z)$$

and so $\dot{V}_c < 0$ thereby establishing the Lyapunov stability of the cascaded system's origin. The control law $u$ that therefore ensures the stabilized linear system cannot destabilize the driven nonlinear system is

$$u = \mathbf{K}\xi - \frac{1}{2}\frac{\partial V}{\partial z}g(z)$$

## 7. Feedback Passivation

A key part of the constructive method involves supposing we had Lyapunov-like functions from which we can construct the controller. It is not always apparent where such *control Lyapunov functions* might come from. But recall that when we discussed passivity in chapter 4, we found one of its main features was that its storage function arose in a natural way from the kinetic and potential energy of the system. So this observation suggests that another avenue for stabilizing a nonlinear system might be to use passivity concepts. The following sections discuss the use of feedback to passivate nonlinear systems.

Consider the problem of stabilizing the origin of the system

$$\dot{x} = f(x) + g(x)u$$

with $u$ as the input. The important thing here is that we assume we are free to pick the output of this system

$$y = h(x)$$

to make the input-output system from $u$ to $y$ passive. Clearly, this may not be possible in all applications. We view, however, the problem of determining where to place sensors as part of the system design process. The control $u$ is then computed assuming we only have access to $y$. Based on the preceding section's discussion we know that if we select $y$ to ensure the system is passive, then we can easily use a feedback law $u = -ky$ to ensure the origin is Lyapunov stable. Moreover, if we can also verify that the system with the given output is ZSD then we will known the interconnection is globally asymptotically stable.

By theorem 28, our ability to find a $y = h(x)$ that renders the system passive with a positive definite storage function means that the system is stable when $u = 0$. This is overly restrictive and so we seek a more flexible way of passivating the plant that does not require the original system to be stable. Instead we simply assume that the uncontrolled system is stabilizable and use a feedback law in conjunction with a selected output function, $h$, to passivate the system. In particular, this means we need to find a feedback transformation

$$u = \alpha(x) + \beta(x)v$$

with $\beta(x)$ invertible and an output $y = h(x)$ such that the system

$$\begin{aligned} \dot{x} &= [f(x) + g(x)\alpha(x)] + g(x)\beta(x)v \\ y &= h(x) \end{aligned}$$

is passive. If such a transformation exists then we say the original system is *feedback passive*. The process of selecting the feedback transformation and the output is called *feedback passivation*. Feedback passivation represents a useful tool in the design of asymptotically stabilizing nonlinear controllers. In particular, if we are able to establish that the feedback passivated system is ZSD, then we can use measurement feedback $v = -ky$ to ensure the origin is asymptotically stable.

One of the key issues we encountered in our earlier study of "feedback linearization" methods was identifying the set of affine systems for which such linearizations were possible. Essentially, we asserted that feedback linearization is possible when the underlying system is controllable and if its set of vector fields is involutive. We face a similar question when we consider feedback passivation. In that case, however, the conditions that are needed to ensure the system is feedback passive are that

- the system has a relative degree no greater than one
- and the zero dynamics must be at least weakly minimum phase (i.e. not unstable)

To see how restrictive these conditions are we examine what these conditions mean for LTI systems. For LTI systems, the relative degree requirement means there is exactly one more pole than zero; a condition that is essential for high-gain stabilization. For LTI systems the minimum phase condition requires that all zeros lie on the left hand side of the complex plane. Both of these requirements are extremely restrictive and therefore suggest that the feedback passivation scheme may be of limited value unless we can find a way to sidestep these limitations.

The following example provides a simple walk through the steps used in feedback passivating a system. In particular, consider the system

$$\begin{aligned}\dot{x}_1 &= x_1^2 x_2 \\ \dot{x}_2 &= u\end{aligned}$$

If we select the output $y = x_2$, then we readily see we have a relative degree one system that is already in its normal form. Since it is in normal form, we can easily identify the zero dynamics $\dot{x}_1 = 0$. This is only stable and so the system is weakly minimum phase. We can then use the following feedback transformation

$$u = v + x_1^3$$

to passivate the system. In particular with this choice of output and feedback transformation the input-output system becomes

$$\begin{aligned} \dot{x}_1 &= x_1^2 x_2 \\ \dot{x}_2 &= -x_1^3 + v \\ y &= x_2 \end{aligned}$$

If we then consider a candidate storage function $V(x) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$ we can see compute $\dot{V}$ to get

$$\begin{aligned} \dot{V} &= x_1\dot{x}_1 + x_2\dot{x}_2 \\ &= x_1^3 x_2 + x_2(-x_1^3 + v) \\ &= x_2 v = yv \end{aligned}$$

which establishes the passive (lossless) nature of the transformed system.

Note that if $y(t) = v(t) = 0$ for all $t$, then $x_2(t) = 0$ which means that $\dot{x}_2 = 0$ and so $x_1 = 0$. In other words, this system is zero-state observable (ZSO) and so we know the output feedback law $v = -ky$ for any $k > 0$ will render the system's origin globally asymptotically stable.

While the preceding example showed how the feedback transformation and selection of $h$ conspire to make a system passive, the example provided little guidance in terms of how to choose the feedback transformation or output. For this to be a truly useful tool we need to identify a systematic method for feedback passivating a system. This may be difficult to do in general, but when we focus our attention on cascaded systems it then becomes possible to formulate a method. This is the topic covered in the next section.

## 8. Passivation of Cascades

This section examines feedback stabilization designs for the cascade of two nonlinear systems with subsystem states $z$ and $\xi$ as shown in Fig. 11. In this cascade, the control only enters the $\xi$ subsystem. The interconnected subsystems are also assumed to satisfy the following ordinary differential equations

$$\begin{aligned} \dot{z} &= f(z) + \psi(z, \xi) \\ \dot{\xi} &= a(\xi) + b(\xi)u \end{aligned}$$

As discussed earlier, the stability of the driving and driven system need not imply the stability of the cascade. This was an issue we faced in I/O feedback linearization designs in which we found that having minimum phase zero dynamics may not be sufficient to assure asymptotic stability of the entire system.

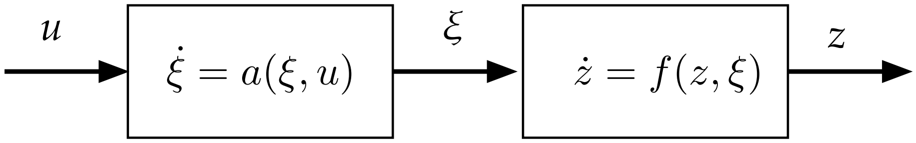


FIGURE 11. Cascaded System

This section investigates the possibility of using feedback passivation to passivate the cascade emerging from I/O feedback linearization. The main assumption we need to achieve this objective is that the unforced driven system $\dot{z} = f(z)$ is globally stable with a $C^2$ radially unbounded positive definite function $W(z)$ such that $L_f W \leq 0$. So we are only assuming that the driven system $\dot{z} = f(z)$ is globally Lyapunov stable. In the context of I/O feedback linearization, of course, this means that the zero dynamics are weakly minimum phase.

Motivated by an I/O feedback linearized system where the driving system is a chain of integrators and the driven system is a weakly minimum phase zero dynamic, we consider the problem of feedback passivating the following

$$\begin{aligned}\dot{z} &= f(z) + \psi(z,\xi) \\ \dot{\xi} &= \mathbf{A}\xi + \mathbf{B}u\end{aligned}$$

For this particular case, it is possible to identify a systematic method for determining the feedback transformation and output needed to feedback passivate the cascade.

This is done by identifying two passive input-output subsystems, $G_1$ and $G_2$. The first subsystem $G_1$ is obtained by factoring the interconnection function in the driven system as

$$\psi(z,\xi) = \tilde{\psi}(z,\xi)\mathbf{C}\xi$$

where $\mathbf{C}$ is chosen so the linear transfer function

$$G_1(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

is positive real, since this implies the linear system is passive. To make this choice for $\mathbf{C}$ clearer, we need to review a few results regarding passive linear systems.

Consider the linear system $G \overset{s}{=} \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]$ with transfer function $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$. The transfer function matrix $G(s)$ is *positive real* if

- poles of all elements of $G(s)$ are in $\mathrm{Re}(s) \leq 0$.
- For all real $\omega$ where $j\omega$ is not a pole of any element of $G(s)$, the matrix $G(j\omega) + G^T(j\omega) \geq 0$.
- Any purely imaginary pole $j\omega$ for any element of $G(s)$ is a simple pole and the residue $\lim_{s\to \jmath\omega}(s - j\omega)G(j\omega)$ is positive semidefinite Hermitian.

The passivity properties of positive real transfer functions are based on the *positive real lemma* also known sometimes as the KYP lemma which we state below without proof.

THEOREM 40. **(Positive Real Lemma)** *Let* $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ *be a* $p \times p$ *transfer function matrix where* $(\mathbf{A}, \mathbf{B})$ *is controllable and* $(\mathbf{A}, \mathbf{C})$ *is observable. Then* $G(s)$ *is positive real if and only if there exist matrices* $\mathbf{P} = \mathbf{P}^T > 0$, $\mathbf{L}$, *and* $\mathbf{W}$ *such that*

$$\begin{aligned} \mathbf{PA} + \mathbf{A}^T\mathbf{P} &= -\mathbf{L}^T\mathbf{L} \\ \mathbf{PB} &= \mathbf{C}^T - \mathbf{L}^T\mathbf{W} \\ \mathbf{W}^T\mathbf{W} &= \mathbf{D} + \mathbf{D}^T \end{aligned}$$

From the positive real lemma, one can then show that if $G(s)$ is positive real it must also be passive. In particular, consider $V(x) = \frac{1}{2}x^T\mathbf{P}x$ as a candidate storage function and compute

$$\begin{aligned} u^Ty - \frac{\partial V}{\partial x}(\mathbf{A}x + \mathbf{B}u) &= u^T(\mathbf{C}x + \mathbf{D}u) - x^T\mathbf{P}(\mathbf{A}x + \mathbf{B}u) \\ &= u^T\mathbf{C}x + \frac{1}{2}u^T(\mathbf{D} + \mathbf{D}^T)u - \frac{1}{2}x^T(\mathbf{PA} + \mathbf{A}^TP)x - x^T\mathbf{PB}u \\ &= u^T(\mathbf{B}^TP + \mathbf{W}^T\mathbf{L})x + \frac{1}{2}u^T\mathbf{W}^T\mathbf{W}u \\ &\quad + \frac{1}{2}x^T\mathbf{L}^T\mathbf{L}x - x^T\mathbf{PB}u \\ &= \frac{1}{2}(\mathbf{L}x + \mathbf{W}u)^T(\mathbf{L}x + \mathbf{W}u) \geq 0 \end{aligned}$$

which means $G(s)$ is passive. The important thing to note from the positive real lemma is that conditions for $G(s)$ to be positive real (i.e. passive) are recast in terms of three of matrix equations. The third equation

$$\mathbf{W}^T\mathbf{W} = \mathbf{D} + \mathbf{D}^T$$

determines $\mathbf{W}$ from the known system matrix $\mathbf{D}$. The first equation is a Lyapunov equation

$$\mathbf{PA} + \mathbf{A}^T\mathbf{P} = -\mathbf{L}^T\mathbf{L}$$

where we pick $\mathbf{L}$ and then have a linear equation for $\mathbf{P}$ that we can solve for. The second equation can be rewritten as

$$\mathbf{C}^T = \mathbf{PB} + \mathbf{L}^T\mathbf{W}$$

which shows that to make $G$ passive (positive real) we need to pick $\mathbf{C}$ (i.e. the output ) to satisfy the above equation. So in this case the choice of $\mathbf{C}$ to render $G_1$ passive is relatively easy.

With this choice for $\mathbf{C}$, the block $G_2$ for the driven system becomes

$$\dot{z} = f(z) + \tilde{\psi}(z, \xi) u_2$$

with input $u_2 = y_1$. We still need to define the output $y_2$ for this system to ensure the $G_2$ is passive. In particular, since we know $\dot{z} = f(z)$ is stable with Lyapunov function $W(z)$, we use this as a candidate storage function for $G_2$. Computing the derivative of $W$ yields

$$\dot{W} = \frac{\partial W}{\partial z}(f(z) + \tilde{\psi}(z, \xi) y_1) \leq \left[ \frac{\partial W}{\partial z} \tilde{\psi}(z, \xi) \right] u_2$$

where we used the fact that $L_f W \leq 0$ and $y_1 = u_2$. This will be passive if we take the output of $G_2$ to be

$$y_2 = h_2(z, \xi) = \left[ \frac{\partial W}{\partial z} \tilde{\psi}(z, \xi) \right]^T$$

With this choice for $y_2$, the second block $G_2$ will be passive and we can then use the feedback transformation

$$u = -h_2(z, \xi) + v$$

to create a feedback interconnection which, by the passivity theorem **??**, will ensure the cascasde from $v$ to $y_1$ is passive. Moreover, if the cascade is ZSD, this means that the usual output feedback law $v = -ky$ will render the origin of the passivated system globally asymptotically stable.

The preceding example was confined to cascades in which the driving system was linear. This obviously is of great use in assuring the asymptotic stability of system realizations obtained from an I/O feedback linearization. However, we also know that such linear systems are topologically equivalent to affine nonlinear systems and this suggests that we should be able to extend the above development to cascades in which the driving system is nonlinear and affine in the control. This is in fact the case. The resulting controller is simply stated below without proof.

THEOREM 41. **(Feedback Passivation of Cascade)** *Suppose that for the cascade*

$$\begin{aligned} \dot{z} &= f(z) + \psi(z,\xi) \\ \dot{\xi} &= a(\xi) + b(\xi)u \end{aligned} \tag{75}$$

*in which the equilibrium $z = 0$ of $\dot{z} = f(z)$ is globally stable with a $C^2$ radially unbounded positive definite function $W(z)$ such that $L_f W \leq 0$. Suppose there exists an output $y = h(\xi)$ such that*

- *the interconnection $\psi(z,\xi)$ can be factored as $\psi(z,\xi) = \tilde{\psi}(z,\xi)y$,*
- *the subsystem*

$$\begin{aligned} \dot{\xi} &= a(\xi) + b(\xi)u \\ y &= h(\xi) \end{aligned} \tag{76}$$

*is passive with a $C^1$ positive definite, radially unbounded storage function $U(\xi)$.*

*Then the entire cascade in (75) is rendered passive with the feedback transformation*

$$u = -(L_{\tilde{\psi}} W)^T(z,\xi) + v \tag{77}$$

*and $V(z,\xi) = W(z) + U(\xi)$ is its storage function. If, with the new input $v$ and the output $y$, the cascade is ZSD, then $v = -ky$ with $k > 0$ achieves global asymptotic stability of the equilibrium $(z,\xi) = (0,0)$.*

**Example:** Let us consider the system

$$\begin{aligned} \dot{z} &= -z + z^2\xi \\ \dot{\xi} &= u \end{aligned} \tag{78}$$

We will consider two strategies for stabilizing this system. In the first strategy, we use partial feedback of $\xi$ to force the driving $\xi$-subsystem to go to zero. The idea is that by driving $\xi$ to zero that the stability of the unforced driving system $\dot{z} = -z$ will be sufficient to ensure the stability of the cascade. The second strategy will use full state feedback to implement a feedback passivating control.

While both strategies achieve local stabilization of the origin, there is a subtle difference in global nature of the stability achieved. In particular, we will show that the first partial feedback strategy only achieves what is called *semiglobal* asymptotic stability, by which we mean that there is a control for any initial condition that assures asymptotic stability, but the gain required to achieve that convergence grows with the distance of the initial state from the equilibrium. On the other hand, the passivating control is able to achieve *global* asymptotic stability in the sense that there is a fixed control law that assures convergence to the origin for *any* initial condition.

Let us first look at the partial feedback approach. In this case we use a linear feedback law $u = -k\xi$ with $k > 0$ to achieve asymptotic stability of $(z, \xi) = (0, 0)$. We use the Lyapunov function $V(z, \xi) = z^2 + \xi^2$ to estimate the region of attraction. The derivative of $V$ is

$$(79)\dot{V} = -2(z^2 + k\xi^2 - \xi z^3) = -\begin{bmatrix} z & \xi \end{bmatrix} \begin{bmatrix} 2 & -z^2 \\ -z^2 & 2k \end{bmatrix} \begin{bmatrix} z \\ \xi \end{bmatrix}$$

negative for $z^2 < 2\sqrt{k}$. An estimate of the region of attraction is the largest set $V = c$ in which $\dot{V} < 0$. This shows that with a feedback gain $k > \frac{c^2}{4}$ that we can guarantee any prescribed $c$. In other words, we can guarantee convergence from an initial condition, $x_0$, with $V(x_0) > c$ only if the gain $k > c^2/4$. So this what we mean when we say the system is *semi-globally* asymptotically stable.

Let us now consider a passivating design that employs full-state feedback to achieve *global* stabilization. We use $y_1 = \xi$ to first create a linear passive system $G_1$. Then by selecting $W(z) = \frac{1}{2}z^2$ as a storage function, we establish that the first equation in (78) defines a passive system $G_2$ with $u_2 = \xi$ as the input and $y_2 = z^3$ as the output. Hence with the feedback transformation

$$u = -y^2 + v = -z^3 + v$$

the cascade (78) becomes a feedback connection of two passive systems. The ZSD property is also satisfied because in the set $y_1 = \xi = 0$, the system reduces to $\dot{z} = -z$. Therefore a linear feedback control $v = -ky_1$ with $k > 0$ will render the whole cascade globally asymptotically stable (GAS).

## 9. Backstepping Feedback Passivation

One of the major limitations of feedback passivation is that it requires the system to have relative degree one and be weakly minimum phase. This is a very small class of systems and to make feedback passivation practical we need to extend it beyond this class of systems. This section shows how backstepping can be used to bypass the relative degree one obstacle.

We discuss backstepping feedback passivation by first presenting an example and then formalizing what we see in the example. In particular, let us consider the following system

$$\begin{aligned}\dot{x}_1 &= x_2 + \theta x_1^2\\ \dot{x}_2 &= x_3\\ \dot{x}_3 &= u\end{aligned}$$

The relative degree of the system depends on what we choose for the system output. The output in I/O feedback linearization would be $x_2$ since it is at the top of the chain of integrators. If $y$ is chosen that way then the system has a relative degree 2. The zero dynamics are embedded in the first equation and satisfy $\dot{x}_1 = \theta x_1^2$, which is clearly unstable. So this system with $y = x_2$ fails to satisfy the conditions needed for the cascade to be feedback passive.

To use the feedback passivation methods, therefore, we need to make some different choices for the output $y$. If we are free to choose $y$, then we should choose it so the system has relative degree one. We'll denote this output as $y_3$ and define it to be

$$y_3 = x_3 - \alpha_2(x_1, x_2)$$

Note that since $y_3$ contains $x_3$ and $\dot{x}_3 = u$, the system from $u$ to $y_3$ will have a relative degree of one.

The second term, $\alpha_2(x_1, x_2)$, will be chosen to satisfy the minimum phase requirement. In particular, with the chosen output $y_3$, this means that the zero dynamics are given by the first two equations. In particular when $y_3 = 0$, then we know $x_3 = \alpha_2(x_1, x_2)$ and inserting this into the first two equations yields the zero dynamics

$$\begin{aligned} \dot{x}_1 &= x_2 + \theta x_1^2 \\ \dot{x}_2 &= \alpha_2(x_1, x_2) \end{aligned}$$

Let us assume that we select $\alpha_2(x_1, x_2)$ so the system is stable, but note that it has the same form as our original system, except that it is now of order $2$ instead of order $3$. Since it has the same structure as our original system, we can try the same trick.

In particular, we consider the system

$$\begin{aligned} \dot{x}_1 &= x_2 + \theta x_1^2 \\ \dot{x}_2 &= v \end{aligned}$$

where we treat $v$ as the control input. We select an output $y_2$ so that this system is of relative degree one, which following our earlier strategy means that

$$y_2 = x_2 - \alpha_1(x_1)$$

When $y_2 = 0$, which means that $x_2 = \alpha_1(x_1)$, we obtain the system's zero dynamics as

$$\dot{x}_1 = \alpha_1(x_1) + \theta x_1^2$$

with the output $y_1 = x_1$. This is obviously a first order system and we need to select $\alpha_1(x_1)$ to asymptotically stabilize the zero dynamics. The selection of $\alpha_1(x_1)$ is relatively easy now since we could use something like the damping theorem to dominate the nonlinear term.

The preceding construction generated a sequence of passivating outputs $y_1$, $y_2$, and $y_3$ that proceed in a bottom-up manner from the full system equations, reducing the order of the system by $1$ at each step, until we are left with a scalar system. Once we stabilize that scalar system, we can then use our usual backstepping control to generate the control $\alpha_2(x_1, x_2)$ that stabilizes the second order system, and then use that stabilized system to construct the control $u = \alpha_3(x_1, x_2, x_3)$ that stabilizes the full original system. In this regard, the construction of the control laws is done in a top down manner; starting from the scalar system and proceeding step by step until we've stabilized the full system. As one should recognize, this is nothing more than the *backstepping* procedure that we introduced previously and shows another application of backstepping as a tool that, in this case, sidesteps the relative degree one restriction we encountered in the use of feedback passivation. The basic step described in the above example can now be formalized into the following theorem

THEOREM 42. **(Backstepping Feedback Passivation)** *Assume that for the system*

$$\dot{z} = f(z) + g(z)u \tag{80}$$

*there is a $C^1$ feedback transformation $u = \alpha_0(z) + v_0$ and a $C^2$ positive definite radially unbounded storage function $W()$ such that this system is passive from input $v_0$ to output $y_0 = [L_g W]^T(z)$ (i.e. $\dot{W} \leq y_0^T v_0$). Then the* *augmented system*

$$\begin{aligned} \dot{z} &= f(z) + g(z)\xi \\ \dot{\xi} &= a(z,\xi) + b(z,\xi)u, \end{aligned} \tag{81}$$

*where $b^{-1}(z,\xi)$ exists for all $z$ and $\xi$, is feedback passive with respect to the output $y = \xi - \alpha_0(z)$ and the storage function $V(z,y) = W(z) + \frac{1}{2}y^T y$. A particular control law that renders this augmented system passive is*

$$u = b^{-1}(z,\xi)\left(-a(z,\xi) - y_0 + \frac{\partial \alpha_0}{\partial z}(f(z) + g(z)\xi) + v\right) \tag{82}$$

*Furthermore the augmented system (81) with this control law (82) is ZSD with respect to input $v$ if and only if the base system (80) is ZSD for the input $v_0$.*

**Proof:** Substituting $\xi = y + \alpha_0(z)$, we rewrite the augmented system (81) as

$$\begin{aligned} \dot{z} &= f(z) + g(z)(\alpha_0(z) + y) \\ \dot{y} &= a(z, y + \alpha_0(z)) + b(z, y + \alpha_0(z))u - \dot{\alpha}_0(z,y) \end{aligned}$$

Using the feedback transformation in equation (82) this system becomes

$$\begin{aligned} \dot{z} &= f(z) + g(z)(\alpha_0(z) + y) \\ \dot{y} &= -y_0 + v \end{aligned} \tag{83}$$

The passivity of the system from $v$ to $y$ is established with the storage function $V(z,y) = W(z) + \frac{1}{2}y^T y$. The time derivative of $V$ is

$$\dot{V} = \dot{W} + y^T(-y_0 + v) \leq y^T v$$

where we used the fact that $\dot{W} \leq y_0^T v_0$ and that $y = v_0$.

To verify the ZSD property of (83), we set $y \equiv v \equiv 0$ which implies $y_0 \equiv 0$. Hence the system (83) is ZSD if and only if $z = 0$ is attractive conditionally to the largest invariant set of $\dot{z} = f(z) + g(z)\alpha_0(z)$ in the set where $y_0 = (L_g W)^T = 0$. This is equivalent to the ZSD property of the original system for input $v_0$ and output $y_0$. $\diamondsuit$

## 10. Example: The TORA System

This section shows how feedback passivation, backstepping, and I/O feedback linearization can be used together to stabilize a well known benchmark problem in nonlinear control known as the TORA (translation oscillator with rotating actuator) system.

The TORA system in Fig. 12 consists of a platform that can oscillate with damping in the horizontal plane (no gravity effect). On the platform is a rotating eccentric mass that is actuated by a DC motor. The motion of this mass applies a force to the platform that can be used to damp the translational oscillations. The motor torque is the control variable and the problem is to find a control law that asymptotically stabilizes the system at a desired equilibrium.

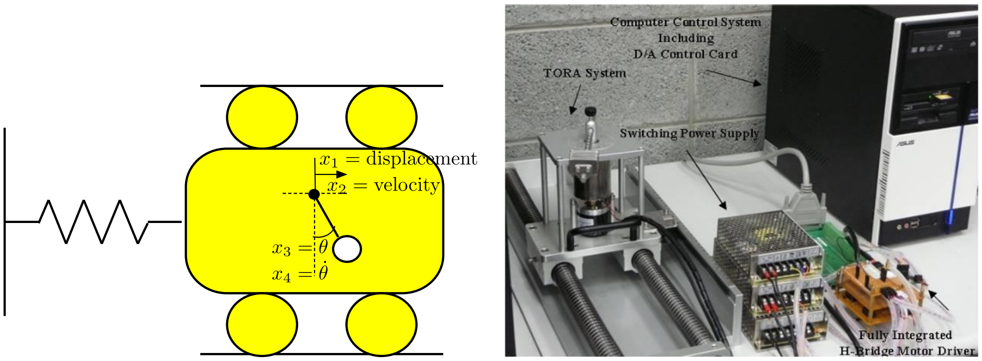


FIGURE 12. (right) TORA system at UMich (left) Diagram of TORA system

We let $x_1$ be the displacement of the platform from the equilibrium position. The state $x_2 = \dot{x}_1$ denotes the velocity of the platform's displacement. $x_3 = \theta$ is the angle of the rotor with $x_4 = \dot{x}_3$ denoting the angular velocity of the rotor. In these coordinates the system dynamics can be written as

$$\dot{x} = f(x) + g(x)u$$

where $u$ is the torque applied to the eccentric mass. The vector fields $f$ and $g$ are

$$f(x) = \begin{bmatrix} x_2 \\ \frac{-x_1+\epsilon x_4^2 \sin x_3}{1-\epsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\epsilon \cos x_3 (x_1 - \epsilon x_4^2 \sin x_3)}{1-\epsilon^2 \cos^2 x_3} \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ \frac{-\epsilon \cos x_3}{1-\epsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1-\epsilon^2 \cos^2 x_3} \end{bmatrix}$$

with $\epsilon$ being a constant parameter that depends on the rotor, platform masses, and eccentricity. Typically the value of this constant is $\epsilon = 0.1$.

We will find it convenient to use a feedback linearizing transformation to transform this system to a simpler form. Recall that the relationship between the feedback linearized system and the original system is a diffeomorphism, which means these systems are equivalent. In particular we use the following state variables

$$\begin{aligned} z_1 &= x_1 + \epsilon \sin x_3 \\ z_2 &= x_2 + \epsilon \cos x_3 \\ \xi_1 &= x_3 \\ \xi_2 &= x_4 \end{aligned}$$

with the feedback transformation

$$\begin{aligned} v &= \frac{1}{1-\epsilon^2 \cos^2 \xi_1} \left[ \epsilon \cos \xi_1 \left( z_1 - (1-\xi_2^2)\epsilon \xi_1 \right) + u \right] \\ &= \alpha(z_1, \xi) + \beta(\xi) u \end{aligned}$$

to bring the system to the following normal form

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= -z_1 + \epsilon \sin \xi_1 \\ \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= v \end{aligned} \tag{84}$$

For this system we will examine three methods for asymptotically stabilizing the origin. The first method uses feedback linearization. The second method uses integrator backstepping and the third will use feedback passivation.

**10.1. Feedback Linearization of TORA system.** The system equation (84) is already in normal form since a feedback linearizing transformation was used above to put it in that form. This is a common preliminary step in the design of nonlinear controllers. To more easily see how this relates to our prior discussion of feedback linearization, let us rename the $z$ variables as $\eta = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ so that the system equations becomes

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= v \\ \dot{\eta}_1 &= \eta_2 \\ \dot{\eta}_2 &= q(\xi, \eta) \end{aligned}$$

where $q(\xi, \eta) = -\eta_1 + \epsilon \sin \xi_1$. The first two equations represent the linearized I/O dynamics of the system in which $\xi_1$ is taken as the output and the last two equations model contain the zero-dynamics of the system.

We know that we can stabilize the I/O map from $v$ to $y$ using a control of the form,

$$v = -k_1 \xi_1 - k_2 \xi_2$$

and in particular we select the gains $k_1$ and $k_2$ so the origin of the $\xi$-subsystem is globally asymptotically stable. With this control, we simply need to check and see if the cascade of this stabilized linear I/O dynamic with the zero-dynamics is also asymptotically stable.

Let us examine the input-to-state stability of the $\eta$-subsystem

$$\begin{aligned} \dot{\eta}_1 &= \eta_2 \\ \dot{\eta}_2 &= -\eta_1 + \epsilon \sin \xi_1 \end{aligned}$$

where $\xi_1$ is treated as an external disturbance. The $\eta$-subsystem is not asymptotically stable when $\xi = 0$ because it is linear with eigenvalues of $\pm j$. This means, therefore, that the zero-dynamics are weakly minimum phase. So no matter what we choose to stabilize the upper system, we will not be able to assure the asymptotic stability of the zero dynamics. This example, therefore, is a case where input-output feedback linearization as we described it above does not synthesize an asymptotically stabilizing controller. The best we can hope for is that the cascaded system is stable.

This conclusion is supported by a MATLAB simulation for the feedback linearization controller with $k_1 = k_2 = 1$. The script and results from this simulation are shown in Fig. 13. The top plot shows the translational states, $x_1$ and $x_2 = \dot{x}_1$. The bottom plots shows the rotational states. As expected the rotational states are controlled to the origin, whereas the translational states continue oscillating.

```
clear; close all; clc;

tstop = 50; dt = 0.01;
eps = 0.1;

x=[1;1;1;1];
k1 = 1; k2 = 1;

data = [];
for time=0:dt:tstop;

   z(1,1) = x(1)+eps*sin(x(3));
   z(2,1) = x(2)+eps*x(4)*cos(x(3));
   y(1,1) = x(3);
   y(2,1) = x(4);

   v = -k1*y(1) - k2*y(2);
   u = (1-eps^2*cos(x(3))^2)*v - eps*x(1)*cos(x(3))+eps^2*x(4)^2*cos(x(3))*sin(x(3));


   f(1,1) = x(2);
   f(2,1) = (-x(1)+eps*x(4)^2*sin(x(3)))/(1-eps^2*(cos(x(3)))^2);
   f(3,1) = x(4);
   f(4,1) = (eps*cos(x(3))*(x(1))-eps*x(4)^2*sin(x(3)))/(1-eps^2*cos(x(3))^2);

   g(1,1) = 0;
   g(2,1) = (-eps*cos(x(3)))/(1-eps^2*cos(x(3))^2);
   g(3,1) = 0;
   g(4,1) = 1/(1-eps^2*cos(x(3))^2);

   xdot = f+g*u;

   x    = x + xdot*dt;

   data = [data; time x' z' y' u  v];
end;

figure(1);
subplot(2,1,1);
plot(data(:,1),data(:,2:3),'linewidth',2);
legend('x1','x2');
title('translational states');
grid on;
subplot(2,1,2);
plot(data(:,1),data(:,4:5),'linewidth',2);
title('rotational states');
```

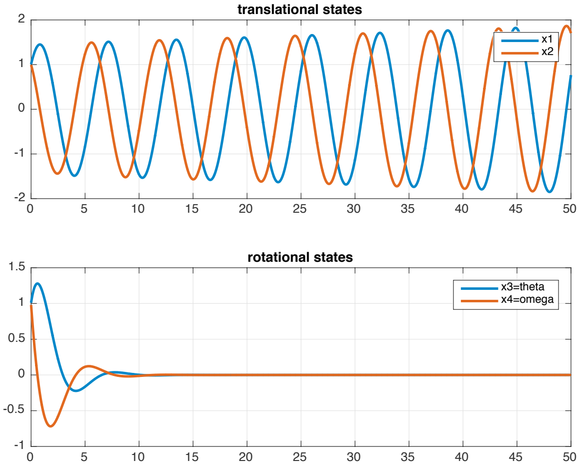


FIGURE 13. (left) script (right) simulation results for TORA system using I/O feedback linearizing control

**10.2. Integrator Backstepping Controller for TORA system.** We now introduce an integrator backstepping control for the TORA system. In this case it is more convenient to keep the system in $(z, \xi)$ coordinates we introduced above.

$$\dot{z}_1 = z_2 \tag{85}$$

$$\dot{z}_2 = -z_1 + \epsilon \sin \xi_1 \tag{86}$$

$$\dot{\xi}_1 = \xi_2 \tag{87}$$

$$\dot{\xi}_2 = v \tag{88}$$

Integrator backstepping starts by treating $\xi_1$ in equation (86) as a "virtual control" input and synthesizes a control law, $\phi_1(z)$ that stabilizes the $z$ subystem (eqs. 85-86). The particular virtual control we will use is

$$\phi_1(z) = -c_0 \tan^{-1} z_2$$

where $0 < c_0 < 2$. To verify that this control asymptotically stabilizes the $z$-subsystem, we consider the candidate Lyapunov function $V_0(z) = \frac{1}{2}(z_1^2 + z_2^2)$ and compute the directional derivative to get

$$\dot{V}_0(z) = -\epsilon z_2 \sin(c_0 \tan^{-1} z_2)$$

which can be shown to be nonpositive. If we then consider the set $E = \{z \in \mathbb{R}^2 : \dot{V}_0(z) = 0\}$, one may easily deduce that $E = \{z \in \mathbb{R}^2 : z_2 = 0\}$. So from the Invariance principle (theorem **??**) we know $z$ asymptotically converges to the largest invariant set in $E$. That invariant set is, of course, just the origin so the proposed virtual control $\phi_1(z)$ indeed globally asymptotically stabilizes the origin of the $z$-subsystem in equations (85-86).

We now rewrite equations (85-87) in a form consistent with our backstepping procedure where we treat $\xi_2$ as the control input $u = \phi_2(z, \xi_1)$ we need to find.

$$\begin{aligned} \dot{z} &= f(z) + g(z) \sin \xi_1 = F(z, \xi_1) \\ \dot{\xi}_1 &= u \end{aligned}$$

and

$$f(z) = \begin{bmatrix} z_2 \\ -z_1 \end{bmatrix}, \quad g(z) = \begin{bmatrix} 0 \\ \epsilon \end{bmatrix}$$

Note that this is not quite in the form required by theorem 38 because the input from the lower system $\xi_1$ does not enter linearly, it is modified by a sine function. This means we cannot directly use theorem 38.

We will modify the backstepping procedure so it applies to systems such as

$$\begin{aligned} \dot{z} &= F(z, \xi_1) \\ \dot{\xi}_1 &= u \end{aligned}$$

in which the upper system may not be affine in $\xi_1$. This modification is done the same steps we followed to establish theorem 38. We use our control $\phi_1(z)$ from above which is known to asymptotically stabilize the origin of $\dot{z} = F(z, \phi_1(z))$ and then rewrite the augmented system is

$$\begin{aligned} \dot{z} &= F(z, \phi_1(z)) + [F(z, \xi_1) - F(z, \phi_1(z))] \\ \dot{\xi}_1 &= u \end{aligned}$$

and we introduce the backstepping variable

$$y_1 = \xi_1 - \phi_1(z)$$

With this change of variables the system becomes

$$\begin{aligned} \dot{z} &= F(z, \phi_1(z)) + \psi(z, \xi_1) y_1 \\ \dot{y}_1 &= u - \dot{\phi}_1(z) \\ &= u - \frac{\partial \phi_1(z)}{\partial z} F(z, \xi_1) \end{aligned}$$

where

$$\psi(z, \xi_1) = \frac{1}{\xi - \phi_1(z)} \left[ F(z, \xi_1) - F(z, \phi_1(z) \right]$$

At this point, the only difference from our earlier analysis is the introduction of the function $\psi(z, \xi_1)$ rather than using $g(z)$. We needed to do this because in our system $g(z)$ does not inject $\xi_1$ into the upper system in a linear manner. Other than this change, the rest follows as expected and can be shown (homework) to give a control of the form

$$u = \phi_2(z,\xi_1) \quad = \quad -\frac{\partial V_0}{\partial z}\psi(z,\xi_1) - k(\xi - \phi_1(z)) + \frac{\partial \phi_1(z)}{\partial z} F(z,\xi_1)$$

where $V_0$ is the Lyapunov function for $\dot{z} = F(z,\phi_1(z))$.

So we now return to our example. For the proposed control, $\phi_1(z)$, we can readily see that

$$F(z,\xi_1) = \begin{bmatrix} z_2 \\ -z_1 - \epsilon \sin(c_0 \tan^{-1} z_2) \end{bmatrix}$$

with $\psi(z,\xi_1)$ being

$$\psi(z,\xi_1) = \begin{bmatrix} 0 \\ \frac{\epsilon\left[\sin\xi_1 + \sin(c_0 \tan^{-1} z_2)\right]}{(\xi_1 + c_0 \tan^{-1} z_2)} \end{bmatrix}$$

and the modified backstepping control is

$$\begin{aligned} \phi_2(z,\xi_1) \quad = \quad & -c_1(\xi_1 + c_0 \tan^{-1} z_2) + c_0 \frac{z_1 - \epsilon \sin\xi_1}{1 + z_2^2} \\ & - \frac{\epsilon z_2}{(\xi_1 + c_0 \tan^{-1} z_2)} \left[\sin\xi_1 + \sin(c_0 \tan^{-1} z_2)\right] \end{aligned}$$

with $c_1 > 0$. This therefore gives a globally asymptotically stabilizing control for the first 3 equations of the system (85-87). We still have one last backstepping operation to do before we obtain the actual control for the entire system.

We now use backstepping to obtain a stabilizing control for the entire system (85-88). Note that this system can be seen as a cascade of the system given by equations (85-87) with the last subsystem (88). To emphasize this let $\zeta = \begin{bmatrix} z \\ \xi_1 \end{bmatrix}$ then we can rewrite the entire system equation as

$$\begin{aligned} \dot{\zeta} &= F(\zeta, \xi_2) = \begin{bmatrix} z_2 \\ -z_1 + \epsilon \sin \xi_1 \\ \xi_2 \end{bmatrix} \\ \dot{\xi}_2 &= v \end{aligned}$$

where $v$ is the control input to the system in equations (85-88) that we need to find. Applying our earlier backstepping formalism will give

$$v(z, \xi) = -c_2(\xi_2 - \phi_2(z, \xi)) + K_1(z_2) z_2 + K_2(z, \xi_1)(-z_1 + \epsilon \sin \xi_1) \tag{89}$$

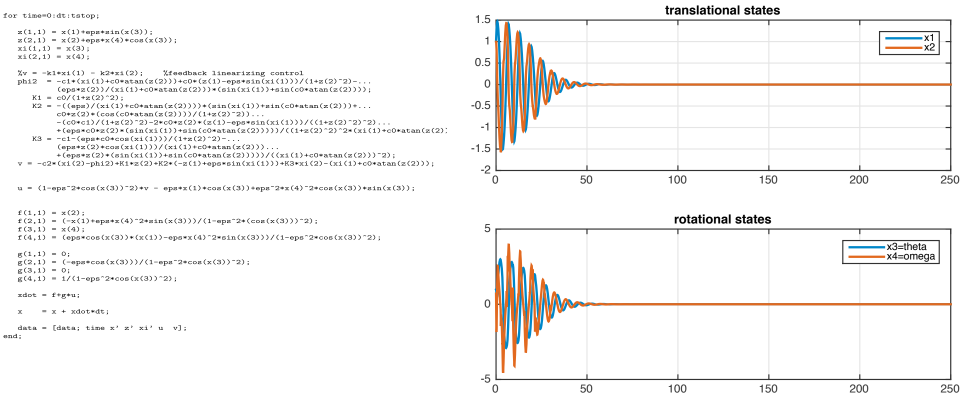
$$+K_3(z, \xi_1)\xi_2 - (\xi_1 + c_0 \tan^{-1} z_2) \tag{90}$$

with $c_2 > 0$ and where

$$\begin{aligned} K_1(z_2) &\equiv \frac{c_0}{1 + z_2^2} \\ K_2(z, \xi_1) &\equiv -\frac{\epsilon}{\xi_1 + c_0 \tan^{-1} z_2} \left[ \sin \xi_1 + \sin(c_0 \tan^{-1} z_2) + c_0 z_2 \frac{\cos(c_0 \tan^{-1} z_2)}{1 + z_2^2} \right] \\ &\quad -\frac{c_0 c_1}{1 + z_2^2} - 2 c_0 z_2 \frac{z_1 - \epsilon \sin \xi_1}{(1 + z_2^2)^2} + \frac{\epsilon c_0 z_2 [\sin \xi_1 + \sin(c_0 \tan^{-1} z_2)]}{(1 + z_2^2)(\xi_1 + c_0 \tan^{-1} z_2)^2} \\ K_3(z, \xi_1) &\equiv -c_1 - \frac{\epsilon c_0 \cos \xi_1}{1 + z_2^2} - \frac{\epsilon z_2 \cos \xi_1}{\xi_1 + c_0 \tan^{-1} z_2} \\ &\quad + \frac{\epsilon z_2 [\sin \xi_1 + \sin(c_0 \tan^{-1} z_2)]}{(\xi_1 + c_0 \tan^{-1} z_2)^2} \end{aligned}$$

We simulated the action of this controller with $c_0 = c_1 = c_2 = 3$. The script and results from this simulation are shown in Fig. 14. The top plot shows the translational states, $x_1$ and $x_2 = \dot{x}_1$. The bottom plots shows the rotational states. This controller indeed asymptotically stabilizes the origin of the entire system, but it is extremely complicated as can be seen from equation (90) as well as the MATLAB script.

**10.3. Feedback Passivating System for TORA systems.** We now investigate the use of feedback passivation in controlling the TORA system.



```
for time=0:dt:tstop;

   z(1,1) = x(1)+eps*sin(x(3));
   z(2,1) = x(2)+eps*x(4)*cos(x(3));
   xi(1,1) = x(3);
   xi(2,1) = x(4);

   %v = -k1*xi(1) - k2*xi(2);    %feedback linearizing control
   phi2  = -c1*(xi(1)+c0*atan(z(2)))+c0*(z(1)-eps*sin(xi(1)))/(1+z(2)^2)-...
           (eps*z(2))/(xi(1)+c0*atan(z(2)))*(sin(xi(1))+sin(c0*atan(z(2))));
      K1 = c0/(1+z(2)^2);
      K2 = -((eps)/(xi(1)+c0*atan(z(2))))*(sin(xi(1))+sin(c0*atan(z(2)))+...
           c0*z(2)*(cos(c0*atan(z(2))))/(1+z(2)^2))...
           -(c0*c1)/(1+z(2)^2)-2*c0*z(2)*(z(1)-eps*sin(xi(1)))/((1+z(2)^2)^2)...
           +(eps*c0*z(2)*(sin(xi(1))+sin(c0*atan(z(2)))))/((1+z(2)^2)^2*(xi(1)+c0*atan(z(2)
      K3 = -c1-(eps*c0*cos(xi(1)))/(1+z(2)^2)-...
           (eps*z(2)*cos(xi(1)))/(xi(1)+c0*atan(z(2)))...
           +(eps*z(2)*(sin(xi(1))+sin(c0*atan(z(2)))))/((xi(1)+c0*atan(z(2)))^2);
   v = -c2*(xi(2)-phi2)+K1*z(2)+K2*(-z(1)+eps*sin(xi(1)))+K3*xi(2)-(xi(1)+c0*atan(z(2)));


   u = (1-eps^2*cos(x(3))^2)*v - eps*x(1)*cos(x(3))+eps^2*x(4)^2*cos(x(3))*sin(x(3));


   f(1,1) = x(2);
   f(2,1) = (-x(1)+eps*x(4)^2*sin(x(3)))/(1-eps^2*(cos(x(3)))^2);
   f(3,1) = x(4);
   f(4,1) = (eps*cos(x(3))*(x(1))-eps*x(4)^2*sin(x(3)))/(1-eps^2*cos(x(3))^2);

   g(1,1) = 0;
   g(2,1) = (-eps*cos(x(3)))/(1-eps^2*cos(x(3))^2);
   g(3,1) = 0;
   g(4,1) = 1/(1-eps^2*cos(x(3))^2);

   xdot = f+g*u;

   x    = x + xdot*dt;

   data = [data; time x' z' xi' u  v];
end;
```

FIGURE 14. (left) script (right) simulation results for TORA system using integrator backstepping control

To start we write out the system equations, but expanding out $\dot{x}_2$ term to show more clearly the actual control input $u$.

$$
\begin{array}{rcl}
\dot{z}_1 & = & z_2 \\
\dot{z}_2 & = & -z_1 + \epsilon \sin \xi_1 \\
\dot{\xi}_1 & = & \xi_2 \\
\dot{\xi}_2 & = & a(z_1, \xi_1) + b(\xi_1) u
\end{array} \tag{91}
$$

where

$$
\begin{array}{rcl}
a(z_1, \xi_1) & = & \dfrac{\epsilon \cos \xi_1 [z_1 - (1 + \xi_2^2) \epsilon \sin \xi_1]}{1 - \epsilon^2 \cos^2(\xi_1)} \\
b(\xi_1) & = & \dfrac{1}{1 - \epsilon^2 \cos^2(\xi_1)}
\end{array}
$$

Feedback passivation, recall, requires the choice of an output function $y = h(z, \xi)$ so that the system from $u$ to $y$ has relative degree one and the zero-dynamics are weakly minimum phase. To be relative degree one, we require the control input $u$ appear after a single differentiation of of the output. From the system equation (91) we see this means that $y = \xi_2$ with the corresponding zero dynamics

$$
\begin{array}{rcl}
\dot{z}_1 & = & z_2 \\
\dot{z}_2 & = & -z_1 + \epsilon \sin \xi_1 \\
\dot{\xi}_1 & = & 0
\end{array} \tag{92}
$$

This means that $\xi_1(t)$ is a constant. Since the underlying system is linear, we can readily see that the output remains bounded for all time, thereby establishing that the zero-dynamics are stable. Since this choice of output ensures the entire system has relative degree one and a weakly minimum phase zero dynamic, we can feedback passivate the system.

Recall that to feedback passivate a system we also need a Lyapunov function, $W(z, \xi_1)$, for the zero dynamics in equation (92). In particular, we can use

$$W(z, \xi_1) = \frac{1}{2}(z_1 - \epsilon \sin \xi_1)^2 + \frac{1}{2}z_2^2 + \frac{k_1}{2}\xi_1^2$$

where $k_1$ is a design parameter. It can be shown that the time derivative of $W$ along system trajectories of (92) is nonpositive and is, in fact, $\dot{W} = 0$.

The feedback transformation required to asymptotically stabilize this system is given by

$$v = -L_{\tilde{\psi}}W + w = (z_1 - \epsilon \sin \xi_1)\epsilon \cos \xi_1 - k_1\xi_1 + w$$

which renders the system passive from the new input $w$ to the output $y = \xi_2$ with respect to the storage function

$$V(z, \xi) = W(z, \xi_1) + \frac{1}{2}\xi_2^2 \tag{93}$$

where, in fact $\dot{V} = \xi_2 v = yv$.

Next we verify if the system with output $y = \xi_2$ and input $w$ is ZSD. From $y = \xi_2 \equiv 0$, we get $\dot{\xi}_2 \equiv 0$, which with $w \equiv 0$ gives

$$0 \equiv \epsilon \cos \xi_1 (z_1 - \epsilon \sin \xi_1) - k_1 \xi_1$$

This means $\xi_2 \equiv 0$ implies that $\xi_1$ is constant and from the above equation $z_1$ is also a constant so that $\dot{z}_1 = z_2 \equiv 0$. Then $\dot{z}_2 = z_1 - \epsilon z_3 \equiv 0$ which together with the above relation shows $\xi_1 \equiv 0$. This proves that $y \equiv 0$ and $w \equiv 0$ only if all of the states are 0, thereby establishing the system is ZSD.

Because the system is passive and ZSD, with the positive definite, radially unbounded storage function, we can achieve global asymptotic stability with $w = -k_2 y = -k_2\xi_2$. The resulting control therefore can be shown to be

$$\begin{aligned} u &= b^{-1}(-a - \frac{\partial W}{\partial \xi_1} - k_2\xi_2) \\ &= \epsilon^2 x_4^2 \sin x_3 \cos x_3 - \epsilon^3 \cos^2 x_3 (z_1 - \epsilon \sin x_3) \\ &\quad -(1-\epsilon^2\cos^2 x_3)(k_1 x_3 + k_2 x_4) \end{aligned} \tag{94}$$

A script was written to simulate this feedback passivating controller with $k_1 = 1$ and $k_2 = 0.15$. The results are shown below in Fig. 15 and they indeed show that the translational dynamics have been asymptotically stabilized. There are, however, two things to notice. The first is that the resulting controller is much simpler than the backstepping controller we designed in the preceding subsection. The second thing to note, however, is that the convergence rate for the controlled system is extremely slow.

```
for time=0:dt:tstop;

   z(1,1) = x(1)+eps*sin(x(3));
   z(2,1) = x(2)+eps*x(4)*cos(x(3));
   xi(1,1) = x(3);
   xi(2,1) = x(4);

  u = eps^2*x(4)^2*sin(x(3))*cos(x(3))-
eps^3*cos(x(3))^2*(z(1)-eps*sin(x(3)))...
      -(1-eps^2*cos(x(3))^2)*(k1*x(3)+k2*x(4));



   f(1,1) = x(2);
   f(2,1) = (-x(1)+eps*x(4)^2*sin(x(3)))/(1-
eps^2*(cos(x(3)))^2);
   f(3,1) = x(4);
   f(4,1) = (eps*cos(x(3))*(x(1))-
eps*x(4)^2*sin(x(3)))/(1-eps^2*cos(x(3))^2);

   g(1,1) = 0;
```

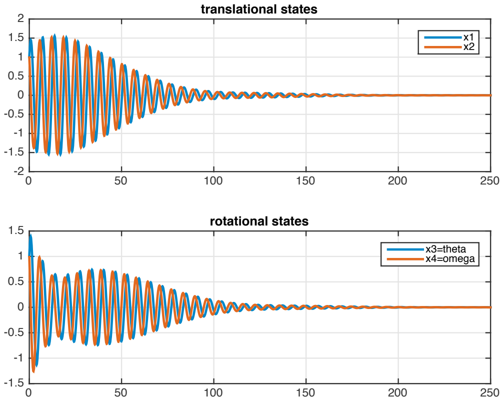


FIGURE 15. Simulation results for feedback passivating control (94) for the TORA system based on the storage function in equation (93).

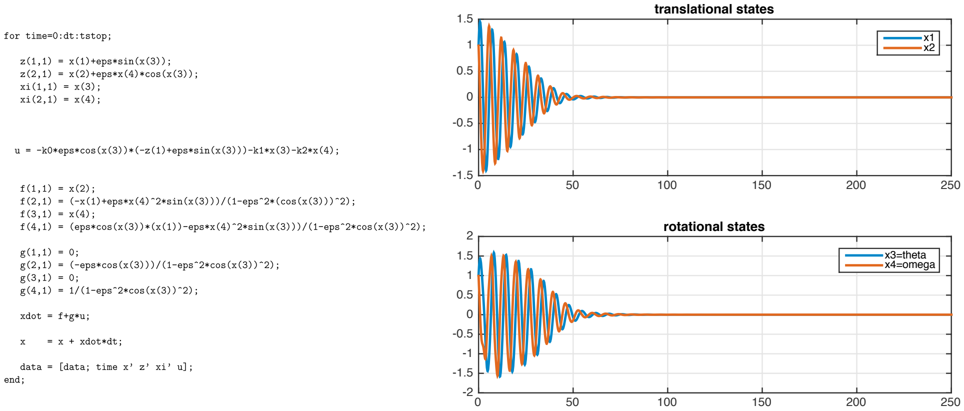
The passivity-based controller given above cannot be made faster through a different selection of the controller gains. The only way to achieve faster response is to include $z_1$ in the feedback law and this might be accomplished by modifying the storage function to increase the penalty on large deviations in the translation coordinate, $z_1$. So we use the following storage function upon which to feedback passivate the system

$$V(z,\xi) = \frac{k_0+1}{2}\left[(z_1 - \epsilon\sin x_3)^2\right] + \frac{k_1}{2}x_3^2 + \frac{1}{2}x_4^2(1-\epsilon^2\cos^2 x_3) \tag{95}$$

For this choice of storage function we can show that the control becomes

$$u = -k_0\epsilon\cos x_3(-z_1 + \epsilon\sin x_3) - k_1 x_3 - k_2 x_4 \tag{96}$$

A simulation for this feedback passivating control is shown in Fig. 16 with gains $k_0 = 10$, $k_1 = 5$, and $k_2 = 0.5$. This simulation retains much of the simplicity of the original feedback passivating control with a transient response that is consistent with what was seen with the backstepping control law.



```
for time=0:dt:tstop;

   z(1,1) = x(1)+eps*sin(x(3));
   z(2,1) = x(2)+eps*x(4)*cos(x(3));
   xi(1,1) = x(3);
   xi(2,1) = x(4);



  u = -k0*eps*cos(x(3))*(-z(1)+eps*sin(x(3)))-k1*x(3)-k2*x(4);


   f(1,1) = x(2);
   f(2,1) = (-x(1)+eps*x(4)^2*sin(x(3)))/(1-eps^2*(cos(x(3)))^2);
   f(3,1) = x(4);
   f(4,1) = (eps*cos(x(3))*(x(1))-eps*x(4)^2*sin(x(3)))/(1-eps^2*cos(x(3))^2);

   g(1,1) = 0;
   g(2,1) = (-eps*cos(x(3)))/(1-eps^2*cos(x(3))^2);
   g(3,1) = 0;
   g(4,1) = 1/(1-eps^2*cos(x(3))^2);

   xdot = f+g*u;

   x    = x + xdot*dt;

   data = [data; time x' z' xi' u];
end;
```

FIGURE 16. Simulation results for feedback passivating control (96) for the TORA system based on the storage function in equation (95).

## 11. Summary

This chapter presented a constructive approach to the stabilization of nonlinear systems based on Control Lyapunov functions. The material on feedback linearization is drawn from Khalil (2002) and Isidori (1995). Much of the discussion on the stabilization of the cascades generated by feedback linearization was drawn from Sepulchre et al. (2012). The application of these methods to the TORA system Wan et al. (1996) were drawn from Jankovic et al. (1996).

CHAPTER 6

# Data Driven Control Systems

The last five chapters presented methods for control system design that assume the prior knowledge of an accurate mathematical model for the plant. Even in the uncertain plants of chapter 3, we assumed there were bounds on the model uncertainty. This need for a prior *accurate* model is an important part of model-based engineering design. In the past, engineers went to great lengths to ensure that the *physical* system fit within the assumed modeling framework. In recent years it has become increasingly difficult to ensure that the ”model” matches ”real-life. Our systems are increasingly complex and they are open to the environment in a way that does not allow us to “engineer” away complexity. Controlling such systems, therefore, means that we must ”learn” a model for the plant or directly learn a controller that is stabilizing.

*Data-driven* control seeks to ”learn” stabilizing controllers for a plant for which we have no prior model using data obtained from watching how the system responds to observed inputs. While there are a number of approaches to data-driven control, this chapter provides a high level review of only three methods. The first methods is an indirect approach that uses input/output data to identify a model for the plant and then design the controller. Another approach known as adaptive control seeks uses online data to adjust an existing controller and thereby improve its performance. Finally, we review the use of Machine learning in data-driven control. In particular we build on chapter 2's results regarding the control of Markov Decision Processes and outline a current popular approach known as Reinforcement Learning.

## 1. Dynamic Mode Decomposition with Control (DMDc)

Dynamic mode decomposition with control (DMDc) is a method that uses regression to identify the $\mathbf{A}$ and $\mathbf{B}$ matrices for an LTI discrete time system of the form

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}w(k) \tag{97}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times n}$ This identification is done using a sequence of $M$ observed states $\mathcal{X} = \{x(i)\}_{i=0}^{M}$ and inputs $\mathcal{W} = \{w(i)\}_{i=0}^{M}$. These two sets form the *data* used in determining the system matrices, $(\mathbf{A}, \mathbf{B})$. This data is used to form the following data matrices

$$\mathbf{X} = \left[\begin{array}{cccc} x(0) & x(1) & \cdots & x(M-1) \end{array}\right], \quad \mathbf{W} = \left[\begin{array}{cccc} w(0) & w(1) & \cdots & w(M-1) \end{array}\right]$$

We also form a time-shifted version of the $X$ matrix

$$\mathbf{X}^{+} = \left[\begin{array}{cccc} x(1) & x_2 & \cdots & x(M) \end{array}\right]$$

By the linear dynamics in equation (97) we have

$$\mathbf{X}^{+} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{W} = \left[\begin{array}{cc} \mathbf{A} & \mathbf{B} \end{array}\right] \left[\begin{array}{c} \mathbf{X} \\ \mathbf{W} \end{array}\right] \overset{\text{def}}{=} \mathbf{F}\mathbf{\Omega}$$

We can then see that an least squares approximation for $\mathbf{F}$ can be obtained using the standard formula

$$\left[\begin{array}{c} \widehat{\mathbf{A}} \\ \widehat{\mathbf{B}} \end{array}\right] = \widehat{\mathbf{F}} = \mathbf{X}^{+}(\mathbf{\Omega}\mathbf{\Omega}^{T})^{-1}$$

In general the inverse will exist since the number of data samples $M$ is chosen to be much greater than $m+n$. In practice, computing this inverse is numerically unstable due to the size of the matrices and so we usually use singular value decompositions of $\mathbf{\Omega}$ to do this computation in a more numerically stable manner.

The preceding discussion used data on the system state and inputs to identify the $(\mathbf{A}, \mathbf{B})$ pair for a linear time-invariant system. But, as noted before, this ”linear” model is an idealization of the physical plant generating the data. Moreover, we may only have access to the system’s inputs and *outputs*, $y$, rather than direct access to the system's states. Even in this situation we can use DMDc to identify a suitable "linearized" model of the nonlinear plant

$$\begin{aligned}\dot{x}(k+1) &= f(x(k)) \\ y(k) &= h(x(k))\end{aligned}$$

where $x(k) \in \mathbb{R}^n$ and $y(k) \in \mathbb{R}$ This is done using the notion of *time-delay embedding* where we take a vector formed from the delayed system outputs as a *state* for the system.

Taken's embedding theorem provides the basis for using time-delayed outputs of a scalar systems,

$$\mathbf{y}_n = \begin{bmatrix} y(n-d+1) \\ y(n-d+2) \\ \vdots \\ y(n) \end{bmatrix}$$

in place of the state $x(n)$ at time $n$. The vector $\mathbf{y}_n$ represents a time-delayed set of the past $d$ system outputs before time $n$. Taken's theorem says that under certain regularizing assumptions, there is a one-to-one map from the original state $x(n)$ onto the $\mathbf{y}_n$ vector provided $M \geq 2n+1$. On the basis of this theorem we would then form the DMDc data matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{M-1} \end{bmatrix}$$

and then proceed to estimate the system pair $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$ as before. The justification for using this linear approximation is based on the fact that $\mathbf{y}_n$ is a time-shifted version of $\mathbf{y}_{n+1}$. But that linear approximation clearly only holds if the dimensionality, $d$, of the embedding vector is infinite. We would then expect that we can always linearize a nonlinear system's dynamics by projecting it onto an infinite dimensional linear systems and then using a lower finite dimensional approximation to design the control system.

One important observation about an input-output system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

is that if the system is *passive* then there are a number of extremely robust stabilization schemes that we can use. In particular, if we can use learn a linearized model of the system, then it is relatively easy to modify the output from that model to transform our identified linear system into a passive system.

## 2. Data-driven Discovery of Koopman Eigenfunctions

Let us consider the following state-based system

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_0$$

We define the *flow*, $\mathbf{\Phi}_t : \mathbb{R}^n \to \mathbb{R}^n$ as an indexed automorphism on the state space such that

$$\mathbf{\Phi}_t(x_0) = x(t; x_0)$$

We define an *observable*, $h : \mathbb{R}^n \to \mathbb{R}$ as any real-valued function taking the state $x$ onto a real scalar (observation). We denote the linear space of all $C^1$ observables as $\mathcal{H}$.

The *Koopman operator*, $\mathcal{K}_t : \mathcal{H} \to \mathcal{H}$ is an operator mapping observables onto othe observables. The operators are indexed with respect to time, $t \in \mathbb{R}$ so they form a one parameter semigroup of operators that take values

$$\mathcal{K}_t[h](x_0) = h(\mathbf{\Phi}_t(x_0)) = h(x(t; x_0))$$

for any time $t \in \mathbb{R}$ and any initial state $x_0 \in \mathbb{R}^n$. So $\mathcal{K}_t$ is a linear transformation defined on the linear function space, $\mathcal{H}$, and may be seen as mapping an observable for the "current" output onto an observable for the output at time $t$ in the future.

Since the Koopman operator is a linear transformation on $\mathcal{H}$, it has an eigendecomposition. So there exists $\lambda \in \mathbb{C}$ and an associated function, $\phi : \mathbb{R}^n \to \mathbb{C}$ such that

$$\mathcal{K}_t[\phi](x_0) = \phi(\mathbf{\Phi}_t(x_0)) = \lambda\phi(x_0)$$

Note that the eigensubspace generated by each eigenfunction is an invariant of the Koopman operator so that the subspace formed by a finite number of these Koopman eigenfunctions is finite dimensional and the dynamics of $h(x)$ generated by $\mathcal{K}_t$ will be ”linear”. In other words, if we choose observables that are Koopman eigenfunctions then the dynamics of that transformation will be linear.

For a given eigenvalue, $\lambda$, we can obtain an approximation of the Koopman eigenfunctions can be obtained in a data-driven manner. The simplest approach would frame the problem as a regression over a library of basis functions. In particular, we consider a set of $p$ candidate basis functions $\theta_i : \mathbb{R}^n \to \mathbb{R}$ and arrange them as a matrix

$$\mathbf{\Theta}(x) = \begin{bmatrix} \theta_1(x) & \theta_2(x) & \cdots & \theta_p(x) \end{bmatrix}$$

For the chosen $\lambda$, the associated eigenfunction would be

$$\phi(x) = \sum_{k=1}^{p} \theta_k(x)\xi_k = \mathbf{\Theta}(x)\xi$$

The solution vector $\xi$ would be obtained by finding the vectors in the null space of the following linear transformation

$$(\lambda\mathbf{\Theta}(x) - \mathbf{\Theta}(\mathbf{X}^+))$$

Note that this approach assumes we have specified the eigenvalue $\lambda$. In general, we would have to also determine what $\lambda$ should be, which would require a more sophisticated approach that simultaneously searches for $\lambda$ and $\phi$.

## 3. Nonlinear Adaptive Control

Control synthesis is based on prior knowledge of the plant's dynamics. This prior knowledge might be obtained in the data-driven methods described above. But in some cases, we may have an existing control system and the plant or controller's parameters are changing over time. Robust controllers ensure a minimum performance level over the entire range of parameter variation, but that minimum level may be overly conservative in practice. *Adaptive control* seeks to tune an existing control law in a manner that improves its performance even though the system parameters are initially unknown. One approach to adaptive control uses the nonlinear damping theorem and backstepping from chapter 5 to adapt nonlinear control systems in a manner that stabilizes the unknown system.

As is usually done in constructive nonlinear control, we start by considering how to stabilize an unknown scalar system and then use backstepping to extend that strategy to a systems consisting of a chain of integrators driving that scalar system. The scalar system of interest to us is

$$\dot{x} = u + \theta\phi(x) \tag{98}$$

This is a special case of the uncertain system considered in the nonlinear damping theorem of chapter 5, except that now the uncertainty is denoted as $\theta$. In our case, we now think of $\theta$ as an *unknown parameter*, rather than an external uncertain disturbance. Even if we do not know a bound for $\theta$, the nonlinear damping theorem provides a way to design a static nonlinear controller that guarantees the global boundedness of $x$. Recall from the damping theorem that this robust control is

$$u = -cx - kx\phi^2(x)$$

with the resulting closed loop system equation

$$\dot{x} = -cx - kx\phi^2(x) + \theta\phi(x)$$

We can use $V = \frac{1}{2}x^2$ as an ISS control Lyapunov function for this system and verify that its directional derivative satisfies

$$\dot{V} \leq -cx^2 + \frac{\theta^2}{4k}$$

which implies that $x(t)$ converges to the interval

$$|x| \leq \frac{|\theta|}{2\sqrt{kc}}$$

The size of this interval can be reduced by increasing $k$ or $c$, but we cannot force $x(t)$ to asymptotically converge to zero if $\theta$ is a nonzero constant. Increasing $k$ or $c$ to reduce the size of the residual region is undesirable because it increases the ”jumpiness” of the system to noise, which may accentuate the impact that neglected uncertainties have on the system. So we are interested in finding a way to reduce the size of this residual set. In particular, we want it to go to zero so we can ensure asymptotic convergence to the equilibrium.

To ensure asymptotic convergence of $x(t)$ to zero, we employ adaptation. In particular, if $\theta$ were known, then we could use the control

$$u = -\theta\phi(x) - c_1 x$$

to render the derivative of $V_0(x) = \frac{1}{2}x^2$ negative definite and thereby ensure convergence to zero. This control, however, cannot be used since $\theta$ is unknown and so one way to address this issue is to use the *certainty-equivalence principle* which assumes that we use a control of the form

$$u = -\widehat{\theta}\phi(x) - c_1 x$$

where $\widehat{\theta}$ is an *estimate* of the actual parameter $\theta$ such that some averaged measure of the estimation error

$$\widetilde{\theta} = \theta - \widehat{\theta}$$

is zero. In this case the system would on average be able to cancel the dynamics associated with $\theta$ and replace them with the linear control $-c_1 x$. We call this the certainty-equivalence approach since we are asserting that the use of the parameter estimate is equivalent to using the true or ”certain” parameter in our controller.

The question we need to answer is whether certainty-equivalence indeed assures that the system will asymptotically converge to zero. So let us consider the candidate Lyapunov function $V_0(x) = \frac{1}{2}x^2$ and compute its directinal derivative

$$\begin{aligned} \dot{V}_0 &= \frac{\partial V_0(x)}{\partial x}\dot{x} = x(-cx - \widehat{\theta}\phi(x) + \theta\phi(x)) \\ &= -cx^2 + \widetilde{\theta}x\phi(x) \end{aligned}$$

For a linear system, we an take $\phi(x) = x$, so that $\theta$ represents the unknown dynamic and we get

$$\dot{V}_0 = -(c - \widetilde{\theta})x^2$$

Provided we can ensure $\widetilde{\theta} < c$, then we have asymptotic convergence to zero. In particular, for linear systems, if we know that $\widetilde{\theta} \to 0$ as $t \to \infty$, then eventually this stabilization condition holds and we should be able to assure the convergence to zero.

For general nonlinear systems, however, we cannot rely on certainty equivalence. In fact we see that in

$$\dot{V}_0 = -cx^2 + \widetilde{\theta}x\phi(x)$$

the second term is indefinite and depending on the form of $\phi(x)$, it may be that this second term dominates the system’s behavior. Moreover, we have already seen examples of nonlinear systems in which ”peaking” in the estimation error $\widetilde{\theta}$ may destabilize the upper system. This suggests we need to abandon certainty-equivalence as a design principle in nonlinear adaptive control.

So let us augment $V_0$ with a quadratic term in the parameter error $\widetilde{\theta}$.

$$V_1(x, \widetilde{\theta}) = \frac{1}{2}x^2 + \frac{1}{2\gamma}\widetilde{\theta}^2$$

where $\gamma > 0$ is a design constant we call the *adaptation gain*. The derivative of this candidate Lyapunov function is

$$\begin{aligned}\dot{V}_1 &= x\dot{x} + \frac{1}{\gamma}\widetilde{\theta}\dot{\widetilde{\theta}} \\ &= -c_1x^2 + \widetilde{\theta}x\phi(x) + \frac{1}{\gamma}\widetilde{\theta}\dot{\widetilde{\theta}} \\ &= -c_1x^2 + \widetilde{\theta}\left[x\phi(x) + \frac{1}{\gamma}\dot{\widetilde{\theta}}\right]\end{aligned}$$

The second term is still indefinite, but it contains the dynamics of the estimation error in it,

$$\dot{\widetilde{\theta}} = -\dot{\widehat{\theta}}$$

So we make an appropriate selection for the parameter estimator, $\dot{\widehat{\theta}}$, that cancels the indefinite term. In particular we choose the update law

$$\dot{\widehat{\theta}} = \gamma x\phi(x)$$

then the directional derivative of $V_1$ becomes

$$\dot{V}_1 = -c_1x^2 \leq 0$$

The resulting adaptive system now consists of the original system with the control and the update law,

$$\begin{aligned}\dot{x} &= -c_1x + \widetilde{\theta}\phi(x) \\ \dot{\widetilde{\theta}} &= -\gamma x\phi(x)\end{aligned}$$

This then gives us a nonlinear estimator for $\theta$. The preceding discussion assumed there was no other disturbance driving the system, but we could clearly have introduced disturbances as well in a manner that would render the adaptively controlled system ISS to this external disturbance.

The preceding discussion assumed the uncertainty generated by the estimation error could be cancelled by the control. This cannot be done if the parameter uncertainty is not *matched* to the control. This is the case in the following system

$$\dot{x}_1 = x_2 + \theta\phi(x_1) \tag{99}$$

$$\dot{x}_2 = u \tag{100}$$

This particular form, suggests that we can use *backstepping* to get the desired "adaptive" control law.

## 4. Extremum Seeking Stabilization

Extremum seeking stabilization (ESS) uses a dithering sinusoidal perturbation to the control input that allows one to determine the gradient of a control Lyapunov function without having a prior model for the system's dynamics. Following this gradient allows the system state to seek the minimum of the CLF, thereby stabilizing the system state about the origin.

We can see the basic idea in this approach on the following open-loop unstable scalar system

$$\dot{x} = x + b(t)u$$

where the control direction $b(t)$ is unknown. For such a system, the standard controllers would fail if the sign of $b(t)$ was incorrect or if $b(t)$ had a change in sign. If, however, we consider the action of a control

$$u = \sqrt{\alpha\omega}\cos(\omega t) + kV(x))$$

where $V(x) = x^2$ is a candidate Lyapunov function, then for $\omega$ sufficient large, the "averaged" state of the system is

$$\overline{x}(t) = x(0) + \int_0^t (x(\tau)b(\tau)u(\tau)d\tau)$$

converges to the origin. This assertion can be justified by noting that $\overline{x}$ satisfies the differential equation

$$\dot{\overline{x}} = (1 - k\alpha b^2(t))\overline{x}$$

Note that the ”unknown” $b(t)$ is now replaced by $b^2(t) \geq 0$. So if $b(t)$ is nonzero often enough, then for large enough $k\alpha > 0$, we can ”practically” stabilize the system in the sense that the averaged state $\overline{x}$ goes to zero and the true state remains in a bounded neighborhood of the origin.

We can illustrate this in a simple example where the unknown system is

$$\dot{x} = x + (0.5x + \sin(10t))u$$

In this case $b(t) = 0.5x(t) + \sin(10t)$. We use the candidate Lyapunov function $V(x) = x^2$ and the control $u$ is then

$$u = \sqrt{\alpha\omega}\cos(\omega t + kV(x))$$

where we select $\omega = 1000$, $\alpha = 10$. Fig. 1 shows the resulting state trajectory that on ”average” converges to the origin. The true state trajectory oscillates about this minimum in a bounded way, thereby exhibiting uniform ultimately bounded behavior (or what is sometimes called *practical stability*).

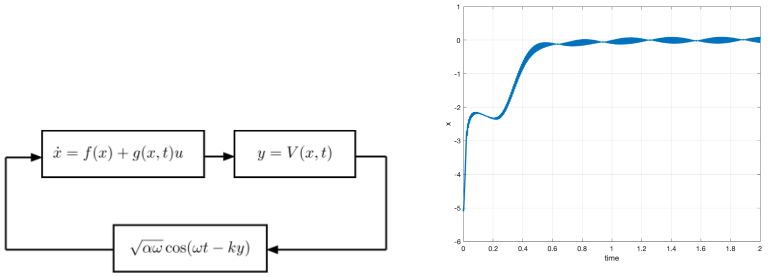


FIGURE 1. Extremem Seeking Control

Note that the control in this example assumes very little about the actual system dynamics. All we really required was that the output from the plant was a Lypunov-like function $V$. The control itself simply dithers back and forth around the the current value of $V$. The dithering control causes variations in $V$ whose phase with respect to the cosine can be detected and used to guide the control in a way that seeks the minimum point of the control

Lyapunov function $V$. This strategy works for more general MIMO systems. It does not achieve ”asymptotic stability”, rather it achieves practical stability.

## 5. Reinforcement Learning

Reinforcement Learning teaches an agent how to act in an unknown environment. Consider an agent that interacts with an external environment. The environment is a dynamical system that provides to the agent, at each time instant, the environment’s current *state* and a *reward* in response to the agent’s current *action*. The agent does not know the environment’s reward function or dynamics. It must learn these things by seeing how the environment responds to the agent’s actions. The agent then uses what it learns about the environment to identify an action (control) policy that selects actions in response to the current environmental state. That action is selected to maximize the aggregate discounted reward the agent receives over a finite time horizon. Reinforcement learning can therefore be seen as trial-and-error learning since it learns how to act in response to the positive/negative consequences of its actions. This learning process is ”data-driven” in the sense that the ”data” are the states/rewards seen in response to each action.

RL can be understood as an optimal control problem for Markov Decision Processes (MDP), which we discussed in chapter 2. That chapter defined the agent/environment interaction as a tuple, $(S, A, p, r, S_0, S_K)$ where $S$ is a finite set of environmental states, $A$ is a finite set of agent actions. The sets $S_0, S_k \subset S$ are initial and terminal state sets. The map $p : X \times A \to \mathcal{P}(S)$ maps the current state action pair at time $k$, $(s_k, a_k)$ onto the next state through the probability distribution

$$p(y \,|\, x, a) = \Pr \left\{ \mathbf{s}_{k+1} = y \,|\, \mathbf{s}_k = x, \mathbf{a}_k = a \right\}$$

The other map, $r \,:\, S \times A \times S \to \mathbb{R}$ maps the current state-action-next state triple $(s_k, a_k, s_{k+1})$ onto the numerical reward $r_{k+1} \in \mathbb{R}$.

The agent and environment interact over a sequence of time steps, $k = 0, 1, 2, 3, \ldots, K$. The initial state $s_0$ lies in $S_0$. At each time instant $k$, the agent selects and action $a_k \in A$ using a policy $\pi : S \rightarrow \mathcal{P}(A)$. The policy uses the current state $s_k$, to randomly select the action $a_k$ with respect to distribution $\pi(\mathbf{a}_k \,|\, \mathbf{s}_k = s)$. The environment then takes this action and returns its next state $s_{k+1} \sim p(\cdot \,|\, s_k, a_k)$ and the next reward $r_{k+1} = r(s_k, a_k, s_{k=1}$. This interaction therefore generates a sequence of state-action-reward triples

$$(s_0, a_0, r_1) \rightarrow (s_1, a_1, r_2) \rightarrow \cdots \rightarrow (s_{K-1}, a_{K-1}, r_K)$$

where the stopping time $K$ occurs when the system state $s_K \sim p( : \, |s_{K-1}, a_{K-1})$ enters the terminal state set $S_K$ for the first time. Each run is called an *episode*

What we saw, before, was that we wanted to find a policy, $\pi$ that maximized the total discounted reward the agent receives for an episode, averaged over all episodes the agent might see. This optimal value is called the value function and for a given policy $\pi$, the value received by an agent starting in state $s$ is

$$V^\pi(s) = \mathbb{E}^\pi \left\{ \sum_{k=0}^{K-1} \gamma^k r(s_k, \pi(s_k), s_{k+1}) \,|\, s_0 = s \right\}$$

where $\gamma \in (0,1)$ is a discount factor. We seek a policy $\pi^*$ such that

$$V^{\pi^*}(s) \geq V^\pi(s)$$

for all $s \in S$ and over all feasible action policies, $\pi$. We learned from chapter 2 that $V^\pi$ satisfies the Bellman equation

$$V^\pi(s) = \sum_{a \in A} \pi(a \,|\, s) \sum_{s' \in S} p(s' \,|\, s, a)(r(s, a, s') + \gamma V^\pi(s'))$$

We also say that it was sometimes valuable to write this ”value” as a function of the state action pair $(s, a)$, This $Q$-function also satisfied a Bellman like equation of the form

$$Q^\pi(s,a) = \sum_{s' \in S} p(s' \mid s,a) \left( r(s,a,s') + \gamma \max_{a \in A} Q^\pi(s',a) \right)$$

The optimal policy could then be obtain

$$\pi^*(s) = \arg\max_a Q^*(s,a) = \arg\max_a \left\{ \max_{\pi \in \mathbf{\Pi}} Q^\pi(s,a) \right\}$$

Finally, we saw that if the state transition distribution $p$ and reward function were known, then we could use the value iteration to generate a sequence value functions $\widehat{V}_\ell$ that asymptotically converge to the optimal value function $V^*$, from which we could then obtain the optimal policy $\pi^*$.

Reinforcement learning (RL) is built on the framework described above, the only difference being that we do not have prior information about the state transition kernel, $p$, or the reward function, $r$. Instead, we have the agent who uses an action policy to select an action, $a_k$, and then observe the environment's reward, $r_{k+1}$ and state $s_{k+1}$ generated in immediate response to that selected action. RL, therefore, uses a trial and error scheme to "learn" the optimal action policy for the MDP. This is done by recursively estimating the value function using a *value gradient method* or by directly learning the policy using *policy gradient methods*. The following subsections review some of the basic RL algorithms used for both methods.

**5.1. RL Value Gradient Algorithms:** Value gradient methods estimate the gradient of the value function using the Bellman equation. They use these gradients to estimate the state-action value function from which the optimal policy can be readily determined. These value gradient algorithms are known a temporal-difference (TD) learning and this section examines two specific versions of TD learning, the SARSA algorithm and $Q$-learning.

TD learning may be seen as an efficient way of using Monte Carlo methods to estimate the value function of a given policy. Monte Carlo (MC) methods use simulations to estimate the likelihood of events through a simulation model of the environment. That simulation is used to generate a large number of process trajectories, each trajectory being called an *episode*. That set of trajectories is then used to estimate fundamental statistics of the process, such as the expectation of the value function at each state in the state space.

Let $V^\pi(s)$ denote the value of environmental state $s \in S$ under a fixed policy $\pi : S \to A$. The simulation model is used to generate a large collection of episodes and we examine those episodes that pass through a give state $s \in S$. We can then estimate the value function from $s$ by simply taking the empirical mean of the total reward generated by all of these episodes. Consider a single episode that generates the state and reward sequence $\{s_\ell\}_{\ell=0}^\infty$, then the value from a state $s = s_k$ for a specific $k$ can be estimated as

$$\widehat{V}^\pi(s_k) \leftarrow \widehat{V}^\pi(s_k) + \alpha \left[ \sum_{\ell=0}^{\infty} \gamma^\ell r(s_{k+\ell}, \pi(s_{k+\ell}), s_{k+\ell+1}) - \widehat{V}^\pi(s_k) \right]$$

where $\alpha \in (0, 1)$ is a hyperparameter called the *learning rate*. If we do this for all episodes passing through the same $s$ then we are essentially averaging the value received from state $s$. Note that this update can only be computed at the end of each episode. This means we have to wait until the entire episode has been completed before updating our estimate of $\widehat{V}^\pi(s)$ at that state. This means, of course, that MC methods represent a very inefficient way of estimating the value function and so are not used in practice.

We would like a method that updates the estimate of $V^\pi$ every time we get a new reward and state from the environment in response to the agent's action. By the Bellman equation, we know that the value function at state $s_k$ is equal to the value function at $s_{k+1}$ plus the immediate reward, $r(s_k, a_k, s_{k+1})$ that the agent receives at time $k + 1$ in response to using action $a_k$. So, rather than waiting to the end of the episode to compute an estimate of the value function based on the total reward obtained from that state, i.e., $\sum_{\ell=0}^\infty \gamma^\ell r(s_{k+\ell}, a_{k+\ell}, s_{k+\ell+1})$, we substitute with the Bellman approximation to get

$$\begin{aligned}\widehat{V}^{\pi}(s_k) &\leftarrow \widehat{V}^{\pi}(s_k) \\ &\quad +\alpha\left[r(s_k,a_k,s_{k+1})+\gamma\widehat{V}^{\pi}(s_{k+1})-\widehat{V}^{\pi}(s_k)\right] \end{aligned} \tag{101}$$

where $a_k = \pi(s_k)$. This update can be computed immediately after the agent using action $a_k = \pi(s_k)$ at time $k$ has received the environment's updated state, $s_{k+1}$, and reward $r_{k+1}$. So rather than updating our estimate of the value function after the episode has finished we can compute it as we are generating each episode. Equation 101 is also called the temporal-difference or TD prediction equation.

The TD prediction in equation (101) can then be used to find the optimal policy. This is done following a similar strategy that was portrayed in chapter 2 in equation (17) for the Value or Policy iteration. We would use the TD-prediction equation (101) to compute estimates of the value function and then use that value function to improve the policy by selecting actions that maximize the value. In general, the original Policy iteration improved its policy through $Q^{\pi}$ (state-action value function) rather than the state value function, $V^{\pi}$. But estimating $Q^{\pi}$ through the Bellman recursion is nearly identical to what we do in estimating $V^{\pi}$. To see this recall that an episode consists on an alternating sequence of (state,action) pairs and rewards,

$$(s_k,a_k)\to r_{k+1}\to(s_{k+1},a_{k+1})\to r_{k+2}\to(s_{k+2},a_{k+2})\cdots$$

Now consider the subsequence from state-action pair to state-action pair and use that to update the state-action value function. Formally, this is identical to the earlier TD-prediction equation (101) except that it is in terms of our estimate $\widehat{Q}^{\pi}$ of the state-action value function,

$$\begin{aligned}\widehat{Q}^{\pi}(s_k,a_k) &\leftarrow \widehat{Q}^{\pi}(s_k,a_k) \\ &\quad +\alpha\left[r_{k+1}+\gamma\widehat{Q}^{\pi}(s_{k+1},a_{k+1})-\widehat{Q}^{\pi}(s_k,a_k)\right] \end{aligned} \tag{102}$$

where $a_k = \pi(s_k)$ and $r_{k+1} = r(s_k,a_k,s_{k+1})$. Note that the variables used in update equation (102) are

$$(s_k,a_k,r_{k+1},s_{k+1},a_{k+1})$$

which spell out the word s-a-r-s-a. As a result the update in equation (102) is called the SARSA TD learning algorithm.

There are several variations of the SARSA TD updating equation. One particular important variation is the $Q$-learning update equation

$$\begin{aligned} \widehat{Q}(s_k, a_k) &\leftarrow \widehat{Q}(s_k, a_k) \\ &+\alpha \left[ r_{k+1} + \gamma \max_a \widehat{Q}(s_{k+1}, a) - \widehat{Q}(s_k, a_k) \right] \end{aligned} \tag{103}$$

where $r_{k+1} = r(s_k, \pi(s_k), s_{k+1})$. In this recursion, we are trying to estimate the optimal state-action value function, $Q^*$, directly rather than computing it for a specific policy $\pi$. The $Q$-learning algorithm in equation (103) is particularly important because it was the first TD algorithm for which one could formally prove convergence Watkins and Dayan (1992).

The policy $\pi$ that is used in both the SARSA and Q-learning algorithms can simply be

$$\pi(s) = \arg\max_a Q(s, a)$$

But the convergence of both algorithm requires that the policy ensures all states are eventually visited infinitely often. It is quite possible that if we always use the ”optimal” action in a *greedy* manner that we fail to fully explore the state space. As a result, the policy that we actually use switches between the ”optimal” action and a ”random” action. This is called an $\epsilon$-greedy policy. In particular, it means that given some probability distribution $p(A)$ (usually chosen to be uniform)

$$\pi(s) = \begin{cases} a \sim p(A) & \text{with probability } \epsilon \\ \arg\max_a Q(s, a) & \text{with probability } 1 - \epsilon \end{cases}$$

In other words, we choose a random action with probability $\epsilon$ and the optimal ”greedy” action with probability $1 - \epsilon$. The $\epsilon$-greedy policy provides agents with the capacity to switch between *exploration* of the state space (i.e. selecting the random action) versus *exploitation* of the prior experience emboded in the $\widehat{Q}^\pi$ function (i.e. picking the greedy optimal action). The $\epsilon$-greedy policy is, perhaps, the simplest way of ensuring the learning algorithm explores the entire state space. Other methods employ the notion of *regret*. For infinite horizon MDPs with finite diameter, the expected average reward is not a function of state and so we can use the difference between this expected reward and the reward obtained using the current policy to trigger actions that ”explore” the state space. The value of these methods is that they can be used to provide probabilistic bounds on the rate of convergence of RL algorithms Auer et al. (2008).

I am now going to use the Frozen Lake Environment from Fig. 5 in chapter 2 to illustrate how a $Q$-learning algorithm performs. We will then compare the outcome against the value function obtained using the Value Iteration in chapter 2. The basic script is shown below

```
Q = np.zeros((state_size,action_size))
num_episodes = 500000      #max number of episodes
num_steps       = 5000          #max length of each episode
lr = .5                                     # learning rate
gamma = .9                           #discount factor
epsilon = .5                          #initial epsilon

for episode in range(episodes):
     s = env.reset()[0]
     done = False
     for step in range(num_steps):
          #epsilon-greedy
          if random.uniform(0,1) < epsilon:
               a = env.action_space.sample()
          else
               a = np.argmax(qtable[s])

           s_new, reward, done, trunc, info = env.step(a)

           qtable[s,a] = Q[s,a] + lr*(reward+gamma*np.max(Q[s_new])-Q[s,a])
           s = s_new

           if done:
                break
```

Note that this is a Q-learning algorithm since it is using update equation (103). To evaluate how well this algorithm is learning the policy, we can estimate the total discounted reward using a moving average of the actual reward obtained in each episode. Let us first evaluate the success rate of the learned policy in the same way we evaluated the success rate of the value iteration policy in chapter 2. The resulting value function and optimal policy are

$$V^* = \begin{bmatrix} 0.069 & 0.057 & 0.032 & 0.015 \\ 0.071 & & 0.102 & \\ 0.189 & 0.27 & 0.207 & \\ & 0.455 & 0.858 & \mathbf{G} \end{bmatrix}, \quad \pi^* = \begin{bmatrix} \text{W(0)} & \text{N(3)} & \text{N(3)} & \text{N(3)} \\ \text{W(0)} & & \text{W(0)} & \\ \text{N(3)} & \text{S(1)} & \text{W(0)} & \\ & \text{E(2)} & \text{S(1)} & \end{bmatrix}$$

Comparing this to the earlier value function and policy obtained using the Value iteration, we see they are very similar, though not exactly the same. If we used this policy to evaluate the success rate of the $Q$-learning policy we see it is about $82\%$, comparable to that obtained using the optimal policy.

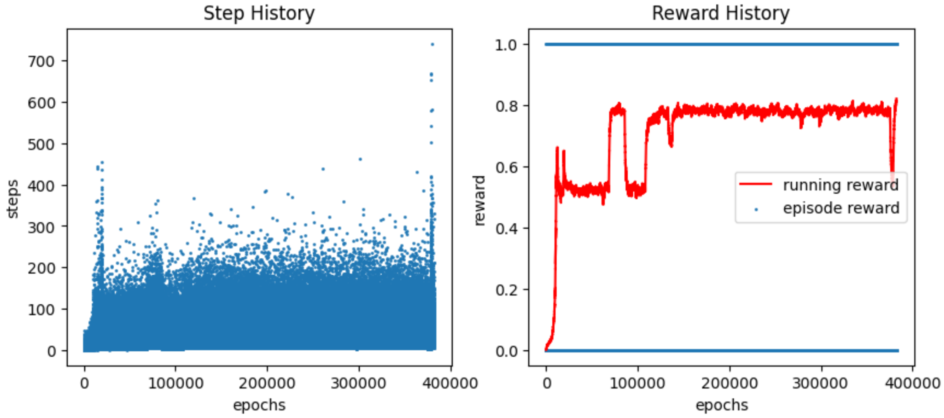


FIGURE 2. (left) step history for Q-learning in frozen lake (right) running reward for Q-learning in frozen lake

Fig. 2 shows plots of the step count for each episode and the running reward as a function of training episode. This shows that the length of successful episodes that finish at the goal can b extremely long. This is due to the slippery nature of the transitions. The other thing we note is that the improvement in the running reward is not monotonic in nature. As the running reward increases, we see that it will ”stall” out for a period of time, increase, and then fall back down again before increasing. This is an example of a *bad episode*. Due to the random nature of the transitions, there is always a probability that there will be a sequence of transitions that cause the agent to fall in the hole several times in a row, thereby lowering the estimated value function for those states. If we had simply adopted a greedy strategy for selecting actions, then this would settle down into a low performing policy. The use of the $\epsilon$-greedy strategy ensures that there is always a chance to ”explore” the space in a way that can overcome the accumulated regret in being too exploitative in our policy.

**5.2. Policy Gradient Methods:** Another approach to RL is to directly ”learn” the policy, rather than first trying to estimate the value function. These methods are called *policy gradient* method for the policy is first written as $\pi(a \,|\, s, \theta)$ where $\theta$ is a parameter vector we need to ”learn”. In this case we search for policy parameters, $\theta$, that maximize the performance of the policy. So we first define a performance function $J(\theta)$ for the policy model and then use gradient *ascent*

$$\theta_{k+1} = \theta_k + \alpha \frac{\widehat{\partial J(\theta_k}}{\partial \theta}$$

where $\widehat{\frac{\partial J}{\partial \theta}}$ is a stochastic estimate whose expectation is close to the gradient of $J(\theta)$. In the following we will introduce one such policy gradient algorithms known as the REINFORCE algorithm.

Let us consider an episodic version of a policy graident method where the policy is updated after an episode has been completed. We define the performance function $J$ to be the value function from the initial state $s_0$,

$$J(\theta) = V^{\pi_\theta}(s_0)$$

where $\pi_\theta(a \,|\, s)$ is the policy with parameter $\theta$ and $V^{\pi_\theta}$ is the value function under policy $\pi_\theta$. It can be shown Sutton and Barto (2018) that the gradient of $J(\theta)$ can be approximated as

$$\begin{align}
\nabla_\theta J(\theta) &= \propto \sum_s \sum \mu(s) \sum_a Q^{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(a \,|\, s) \nonumber \\
&\approx \mathbb{E}_{\pi_\theta}\left[ \sum_a Q^{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(a \,|\, s) \right] \tag{104} \\
&= \mathbb{E}_{\pi_\theta}\left[ G \frac{\nabla_\theta \pi_\theta(a \,|\, s)}{\pi_\theta(a \,|\, s)} \right] \tag{105}
\end{align}$$

where $\mu(s)$ is the fraction of time spent in state $s$ over the given episode, $Q^{\pi_\theta}(s,a)$ is the state-action value function under $\pi_\theta$ and $G$ is the total return for the episode. If we let $\theta_n$ denote our policy parameters during the $n$th episode, and equation (105) to compute the gradient of $J(\theta_n)$, then we can get a new set of parameters for the $n+1$st episode through the gradient ascent,

$$\begin{align}
\theta_{n+1} &= \theta_n + \alpha G_n \frac{\nabla \pi(a \,|\, s, \theta_n)}{\pi(a \,|\, s, \theta_n} \nonumber \\
&= \theta_n + \alpha G_n \ln \pi(a \,|\, s\theta_n) \tag{106}
\end{align}$$

where $G_n$ is the total reward obtained in the $n$th episode.

This algorithm is easily implemented on the Frozen Lake environment. In this case, our policy model $\pi_\theta$ will be taken to be a deep neural network that we instantiate using the Tensorflow/Keras deep learning library. The model I'll use is

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, optimizers

inputs = keras.Input(shape = (state_size,))
x = layers.Dense(16, activation="relu")(inputs)
x = layers.Dense(8, activation="relu")(x)
outputs = layers.Dense(action_size,activation="softmax")(x)
model = keras.Model(inputs=inputs, outputs = outputs)
```

The actual learning algorithm is now shown below in the following script. In this script we use Tensorflow/Keras `GradientTape` object to build a computation graph used in computing the gradient of the loss function. The model uses one-hot encoded versions of the state

```
for episode in range(num_episodes):
    s = env.reset()[0]
    with tf.GradientTape() as tape:
        for step in range(max_steps):
            s_hot =np.zeros(shape=(1,state_size)).astype("float32")
            s_hot[0,s] = 1.
            action_probls = model(s_hot)
            action = np.random.choice(action_size, p = np.squeeze(action_probs))
            action_probs_history.append(tf.math.log(action_probs[0, action]))
            s, reward, done, trunc, info = env.step(action)
            rewards_history.append(reward)
            if done:
                break

        returns = []
        discounted_sum = 0
        for r in rewards_hstiory[::-1]:
            discounted_sum = r + gamma * discounted_sum
            returns.insert(0, discounted_sum)
        history = zip(action_probs_history, returns)
        actor_losses = []
        for log_prob, ret in history:
            actor_losses.append(-log_prob * ret)
        loss_value = sum(actor_losses)

    grads = tape.gradient(loss_value, model.trainable_variables)
    optimizer.apply_gradients(zip(grads, model.trainable_variables))

    if (termination condition satisfied)
        break
```

Fig. 3 illustrates the step and running reward history for our REINFORCE algorithm. This implementation completes its training in about 35000 episodes. Recall that the $Q$-learning algorithm required about 400,000 episodes before we got an acceptable policy. Also note that the abrupt drops in running reward due to episodes appears to be much smaller than what appeared in $Q$-learning. Overall the learning performance in REINFORCE appear to be more monontonic in nature, with the running reward being small for a long period of time and then abruptly increasing to the desired termination level. Again, if we evaluate the resulting policy we get a success rate comparable to that predicted by the optimal Value Iteration policy.

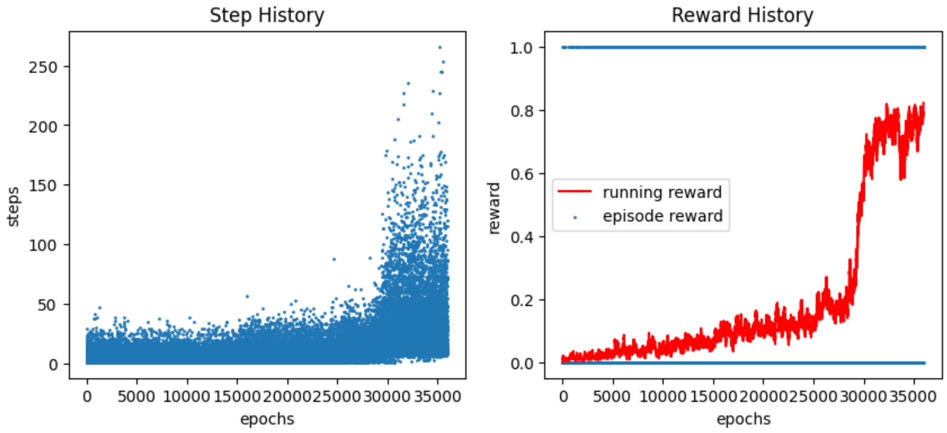


FIGURE 3. (left) step history for REINFORCE in frozen lake (right) running reward for REINFORCE in frozen lake

## 6. Summary

This chapter is by no means a complete introduction to data-driven control methodologies. The topics regarding Koopman operators, DmD, and extreme seeking control were drawn from Brunton and Kutz (2022). The material on nonlinear adaptive control comes from Freeman and Kokotovic (2008). The material on Reinforcement learning is based on Sutton and Barto (2018) and the more recent work on data driven safety comes from Wabersich et al. (2023).

# Bibliography

Adler, I., Resende, M. G., Veiga, G., and Karmarkar, N. (1989). An implementation of karmarkar's algorithm for linear programming. *Mathematical programming*, 44(1):297–335.

Alessio, A. and Bemporad, A. (2009). A survey on explicit model predictive control. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, pages 345–369.

Antsaklis, P. and Michel, A. N. (2006). *Linear systems*. Springer Science & Business Media.

Astrom, K. and Murray, R. (2010). *Feedback systems: an introduction for scientists and engineers*. Princeton University Press.

Auer, P., Jaksch, T., and Ortner, R. (2008). Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21.

Balas, G., Chiang, R., Packard, A., and Safonov, M. (2008). Robust control toolbox user's guide. In *The Math Works, Inc*. Citeseer.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (2006). *Nonlinear programming: theory and algorithms*. John wiley & sons.

Bertsekas, D. (1995). *Dynamic programming and optimal control*. Athena scientific Belmont, MA.

Brunton, S. L. and Kutz, J. N. (2022). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.

Dorato, P., Cerone, V., and Abdallah, C. (1994). *Linear-quadratic control: an introduction*. Simon & Schuster.

Doyle, J. (1978). Guaranteed margins for lqg regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757.

Doyle, J. C., Francis, B. A., and Tannenbaum, A. R. (2013). *Feedback control theory*. Courier Corporation.

Fleming, W. H. and Rishel, R. W. (1972). *Deterministic and stochastic optimal control*. Springer-Verlag.

Freeman, R. and Kokotovic, P. V. (2008). *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science & Business Media.

Gahinet, P. and Apkarian, P. (1994). A linear matrix inequality approach to $\mathcal{H}_\infty$ control. *International journal of robust and nonlinear control*, 4(4):421–448.

Gahinet, P., Nemirovskii, A., Laub, A. J., and Chilali, M. (1994). The lmi control toolbox. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 3, pages 2038–2041. IEEE.

Gikhman, I. and Skorokhod, A. (1972). *Stochastic Differential Equations*. Springer-Verlag, Berlin, New York.

Goldstein, H. (1959). *Classical Mechanics*. Addison-Wesley, Reading, Mass.

Green, M. and Limebeer, D. J. (2012). *Linear robust control*. Courier Corporation.

Isidori, A. (1995). *Nonlinear control systems*. Springer Science & Business Media.

Isidori, A. (1999). *Nonlinear Control Systems II*. Springer Verlag.

Jankovic, M., Fontaine, D., and KokotoviC, P. V. (1996). Tora example: Cascade-and passivity-based control designs. *IEEE Transactions on Control Systems Technology*, 4(3):292–297.

Jiang, Z.-P., Teel, A. R., and Praly, L. (1994). Small-gain theorem for iss systems and applications. *Mathematics of Control, Signals, and Systems (MCSS)*, 7(2):95–120.

Kailath, T. (1976). *Lectures in Linear Least Squares Estimation*. Springer-Verlag.

Khalil, H. (2002). *Nonlinear Systems*. Prentice-Hall.

Krstic, M., Kanellakopoulos, I., and Kokotovic, P. V. (1995). *Nonlinear and adaptive control design*. Wiley.

Krstić, M. and Kokotović, P. V. (1996). Modular approach to adaptive nonlinear stabilization. *Automatica*, 32(4):625–629.

Liberzon, D. (2012). *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press.

Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE.

Mayne, D. Q. and Michalska, H. (1988). Receding horizon control of nonlinear systems. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 464–465. IEEE.

Moylan, P. and Hill, D. (1978). Stability criteria for large-scale systems. *IEEE Transactions on Automatic Control*, 23(2):143–149.

Ogata, K. (2009). *Modern Control Engineering*. Pearson, 5th edition.

Parrilo, P. A. (2003). Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320.

Prajna, S., Papachristodoulou, A., and Parrilo, P. (2002). Introducing sostools: A general purpose sum of squares programming solver. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 1, pages 741–746. IEEE.

Puterman, M. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience.

Rao, A. V. (2009). A survey of numerical methods for optimal control. *Advances in the astronautical Sciences*, 135(1):497–528.

Reznick, B. (2000). Some concrete aspects of hilbert's 17th problem. *Contemporary Mathematics*, 253:251–272.

Rohrs, C. E., Schultz, D., and Melsa, J. (1992). *Linear control systems*. McGraw-Hill Higher Education.

Rudin, W. (1964). *Principles of mathematical analysis*. McGraw-Hill New York.

Sanchez-Pena, R. S. and Sznaier, M. (1998). *Robust systems theory and applications*. John Wiley & Sons, Inc.

Sepulchre, R., Jankovic, M., and Kokotovic, P. V. (2012). *Constructive nonlinear control*. Springer Science & Business Media.

Sontag, E. D. (1989). A 'universal'construction of artstein's theorem on nonlinear stabilization. *Systems & control letters*, 13(2):117–123.

Sussmann, H. and Kokotovic, P. (1991). The peaking phenomenon and the global stabilization of nonlinear systems. *IEEE Transactions on automatic control*, 36(4):424–440.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Toh, K.-C., Todd, M. J., and Tütüncü, R. H. (1999). Sdpt3—a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1):49–95.

Wabersich, K. P., Taylor, A. J., Choi, J. J., Sreenath, K., Tomlin, C. J., Ames, A. D., and Zeilinger, M. N. (2023). Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177.

Wan, C.-J., Bernstein, D. S., and Coppola, V. T. (1996). Global stabilization of the oscillating eccentric rotor. *Nonlinear Dynamics*, 10(1):49–62.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.

Wie, B. and Bernstein, D. (1992). Benchmark problems for robust control design. *Journal of Guidance, Control, and Dynamics*, 15(5):1057–1059.

Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and optimal control*, volume 40. Prentice hall New Jersey.