Nonlinear Control Systems
1. - Introduction to Nonlinear Systems

Dept. of Electrical Engineering

Department of Electrical Engineering
University of Notre Dame, USA

EE60580-01

Course Overview

- The course has three modules
  - Advanced Tools for ODEs
  - Stability Concepts
  - Nonlinear Controller Synthesis
- Text - lecture notes that are transcribed from a variety of texts and monographs
- Homework (once a week) 1-3 problems - We'll arrange recitation sessions for HW groups (2-3 students). HW due the day after recitation. Recitations are optional after the first week.
- Two exams - Midterm/Final - closed book.

Course Overview

Novel features of this course

1. Emphasis of relationships between tools and concepts, rather than detailed discussion of proofs (proofs are in notes).
2. Topics on bifurcation and structural stability
3. Use more modern approach of treating Lyapunov functions as stability certificates. Examines computational methods for certifying system stability.
4. Emphasis on three stabilization methods; constructive methods based on CLF, older geometric methods use in feedback linearization, and passivity-based methods.
5. Introduces relatively recent ideas regarding Model-free control schemes that provide a "hybrid systems" approach to the regulation of nonlinear processes.

Dynamical Systems

- Rather than starting with an ODE as the system model, let us consider a more general formulation in which the process is seen as a *mathematical system*.
- $\Sigma = (X, G, U, \phi)$ where X is a topological space (state space), G is a strongly ordered group of time instants, U is a linear space of input signals defined on G, and $\phi : G \times X \times U \to X$ is a transition map.
- We assume the transition map is continuous and

$$\begin{aligned} \phi(0, x, u) &= x \quad \text{(identity)} \\ \phi(s+t, x) &= \phi(t, \phi(s, x, u_{[0,s]}), u_{[s,s+t]}) \end{aligned}$$

- We call $x : G \to X$ a state trajectory or *orbit*. The value $x(t)$ denotes the system "state" at time instant $t \in G$.

Dynamical Systems

- If there is no input (i.e. $u \equiv 0$), then this is an unforced system (X, G, ϕ) and with the given assumptions we know that the orbits do not intersect and partition the phase space into equivalence classes. This justifies the notion of "state".
- Principle of Superposition

$$\phi(t; x, \alpha u_1 + \beta u_2) = \alpha\phi(t; x, u_1) + \beta\phi(t; x, u_2)$$

  means the system is linear. You studied this in linear systems theory (EE550) when G with either $\mathbb{Z}$ (discrete-time) or $\mathbb{R}$ (continuous-time).
- We focus on systems that do not satisfy the principle of superposition. These systems exhibit a wide range of interesting behaviors that do not appear in linear systems.
- The objective of the first couple lectures is to present examples introducing these interesting behaviors. These behaviors will provide the motivation for our subsequent study of nonlinear control systems.

Discrete-time System based on Logistic Map

- Let us consider a discrete-time system $\Sigma = (X, G, \phi)$ where $X = [0,1]$, $G = \mathbb{Z}$, and $\phi : [0,1] \to [0,1]$ is given by

$$\phi(x) = ax(1-x) \tag{1}$$

  with a being a positive real constant. This is called the *logistic map*.
- Orbits are generated by the recursive equation

$$x_{k+1} = ax_k(1-x_k), \quad x_0$$

  where $a > 0$ and $\{x_k\}_{k=0}^{\infty}$ is the orbit.
- Models population of organism in a box of fixed size. When population is low, then there is abundant food/space and x_k is increasing. When population is high, there is a shortage of food/space and x_k is decreasing.

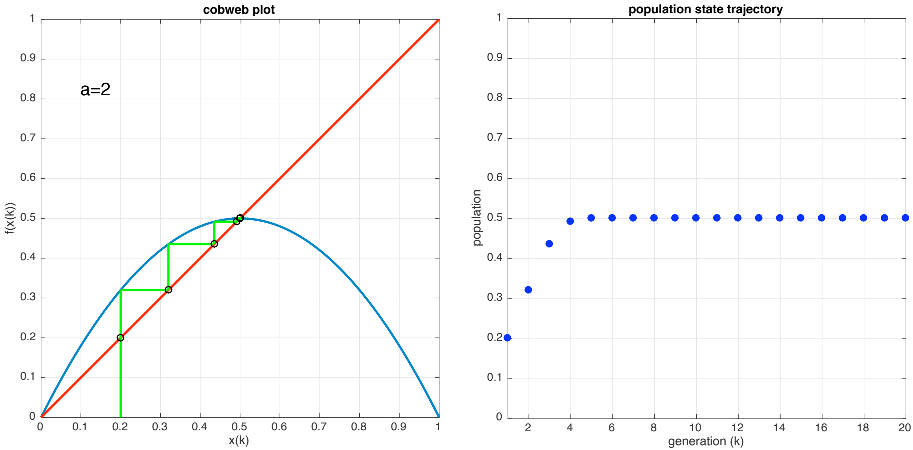


Figure: Cobweb Graph and State Trajectory for Logistic System with $a = 2$ and $x_0 = 0.2$

Logistic Map

- One can computationally examine what happens if the food supply, a, is increased without expanding the logistic map's positive support (i.e. the living space). This gives rise to a number of scenarios
-
- *Extinction:* $(0 \leq a < 1)$ - In this case the graph of $\Phi(x)$ never crosses the 45 degree line formed by $y = x$. $x[k]$ decreases with each generation and asymptotically approaches 0. In other words, for this range of a the population goes extinct due to a lack of food.
- *Stable Population:* $(1 \leq a < 3)$ - In this case the 45 degree line for $y = x$ intersects the graph $y = \Phi(x)$ at two points $x_u^* \equiv 0$ and $x_s^* \equiv 1 - \frac{1}{a}$. One can readily verify that the point $x_s^* = 1 - \frac{1}{a}$ is an *asymptotically stable equilibrium* in the sense that $x[k] \rightarrow x_s^*$ as $k \rightarrow \infty$. This case, therefore, corresponds to a scenario in which the population achieves a "stable" fixed level.

Logistic Map

- *Limit Cycling Population:* $(3 < a \leq 1 + \sqrt{6} = 3.449)$ - In this case the population goes through a "boom and bust" cycle or what will be later referred to as a *limit cycle*. Fig. 2 shows the cobweb plot and state trajectory for the case when $a = 3.4$. In this figure the population bounces between two different sizes on each generation.

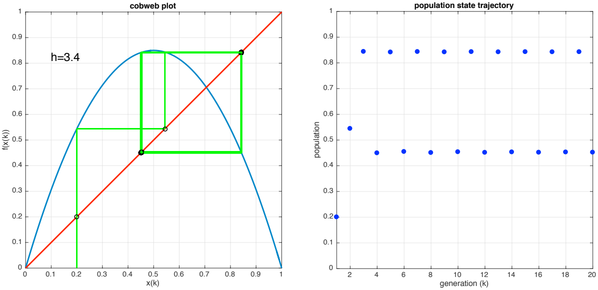


Figure: Cobweb Graph and State Trajectory for Logistic System with $h = 3.4$ and $x_0 = 0.2$

- *Road to Chaos:* $(3.449 < a \leq 3.570)$ - For a just greater than $3.449 = 1 + \sqrt{6}$, the period 2 limit cycle begins doubling until $a > 3.570$ at which point the population's state trajectory becomes *chaotic* as shown in Fig. 3. The term "chaos" formally means that the future states vary in a discontinuous manner with the initial condition x_0.

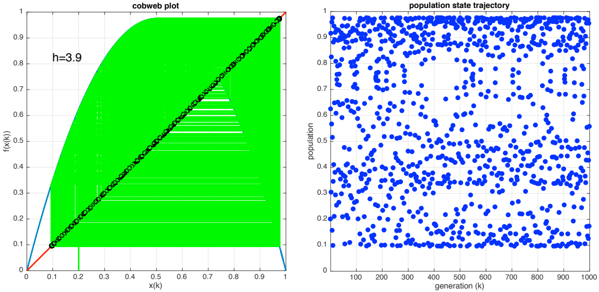


Figure: Cobweb Graph and State Trajectory for Logistic System with $h = 3.9$ and $x_0 = 0.2$

Nonlinear Continuous-time Oscillator

- A nonlinear dynamical system, $\mathbb{S} = (X, G, U, \phi)$, is said to be *continuous-time* if the index set G is the set of reals, $\mathbb{R}$. Continuous-time systems can also exhibit a range of behaviors that include convergence to fixed points, limit cycles, and chaos.
- The *Duffing oscillator* satisfies a second order differential equation of the form

$$\ddot{x} + \gamma \dot{x} - \omega^2 x + \epsilon x^3 = \Gamma \cos(\Omega t) \tag{2}$$

where ω, γ, ϵ, and Γ are all positive real constants.

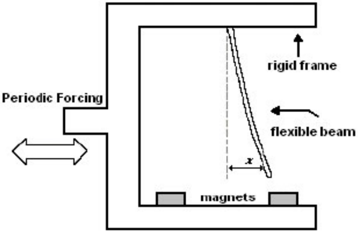


Figure 1. Mechanical interpretation of Duffing oscillator.

The Duffing equation (2) may be viewed as a forced oscillator whose spring provides a restoring force that is a cubic function of the system state. When $\epsilon < 0$ then the spring is a "hardening" spring and when $\epsilon > 0$, then this spring is "soft".

Duffing Oscillator

Numerically integrating equation (2) forward in time shows a periodic limit cycle when $\gamma = 0.1$, $\epsilon = 0.25$, $\omega = 1$, $\Gamma = 0.5$ and $\Omega = 2$.

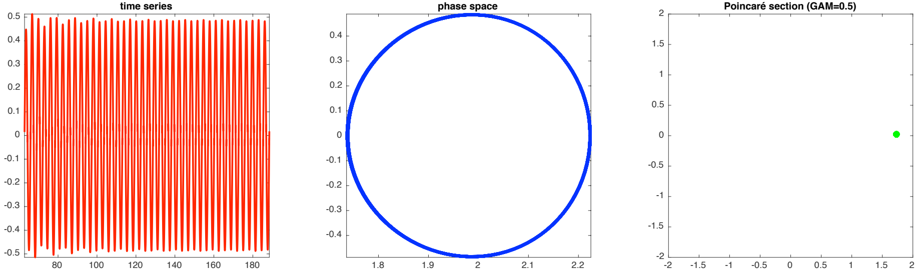
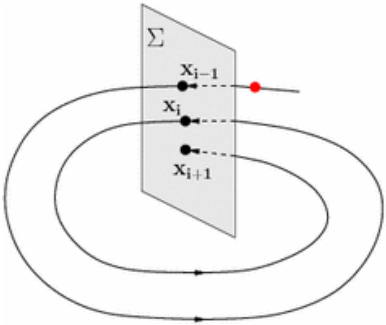


Figure: State trajectory (left), phase plane trajectory (middle), and Poincaré section (right) for forced Duffing oscillator (2) with $\gamma = 0.1$, $\epsilon = 0.25$, $\omega = 1$, $\Gamma = 0.5$, and $\Omega = 2$.

Poincaré section provides a convenient way of viewing the behavior of periodic state trajectories. The method fixes a cross section, Σ, of the phase space and determines a map F that maps a state $x \in \Sigma$ onto the state trajectory's first return, $F(x)$, to the cross section.



Duffing Oscillator Chaos

When we change $\Gamma = 1.5$ in the Duffing oscillator, the Poincaré section shows a much more complex structure. This structure is sometimes called a chaotic or *strange* attractor for the system.

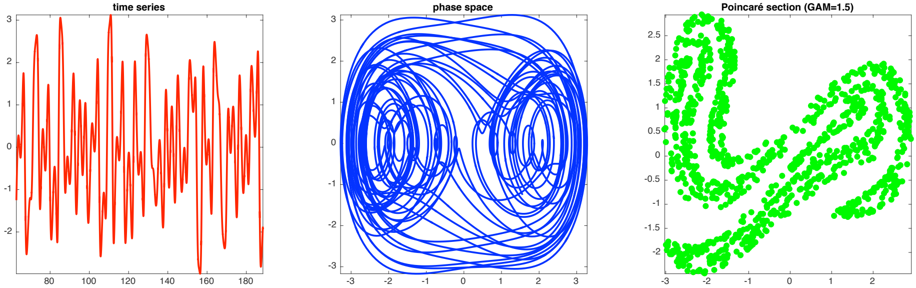


Figure: State trajectory (left), phase plane trajectory (middle), and Poincaré section (right) for forced Duffing oscillator (2) with $\gamma = 0.1$, $\epsilon = 0.25$, $\omega = 1$, $\Gamma = 1.5$, and $\Omega = 2$.

Nonlinear Regulation Problems

- Consider an input-output system whose state trajectory, $x : \mathbb{R} \to \mathbb{R}^n$, system output $y : \mathbb{R} \to \mathbb{R}^m$, and control input $u : \mathbb{R} \to \mathbb{R}^p$ satisfy the following set of equations,

$$
\begin{aligned}
\dot{x}(t) &= f(x, u) \\
y &= h(x, u) \\
u &= k(x)
\end{aligned}
$$

  We assume that the functions $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \to \mathbb{R}^p \to \mathbb{R}^m$ are known.

- Given a desired output signal, $y_d : \mathbb{R} \to \mathbb{R}^m$, the objective is to find the state feedback controller $k : \mathbb{R}^n \to \mathbb{R}^p$ such that the output signal $y(t) \to y_d(t)$ as $t \to \infty$.

Regulation through Local Linearization

- Consider a system $\dot{x} = f(x, u)$ where u is fixed and let ϕ denote the system's flow.
- A point $x^* \in \mathbb{R}^n$ is called an equilibrium point if $f(x^*, u) = 0$ for the given $u \in \mathbb{R}^m$. This equilibrium point for the unforced system (i.e. $u = 0$) is said to be *hyperbolic* if the eigenvalues of the Jacobian matrix , $\left[\frac{\partial f}{\partial x}(x^*, 0)\right]$, evaluated at x^* has no eigenvalues with zero real parts.
- The Hartman-Großman theorem establishes that the flow ϕ in a suitably small neighborhood of a hyperbolic equilibrium is *topologically equivalent* to the flows of the linearized system that satisfies

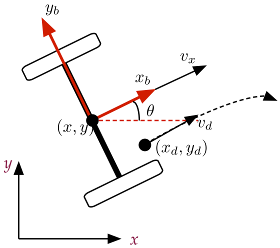
$$
\begin{aligned}
\dot{x} &= \left[\frac{\partial f}{\partial x}(x^*, 0)\right](x - x^*) + \left[\frac{\partial f}{\partial u}(x^*, 0)\right] u \qquad (3)\\
&= A(x - x^*) + Bu
\end{aligned}
$$

Regulation through Local Linearization

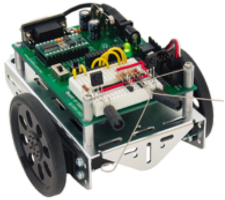
- Topological equivalence means there is a smooth invertible state-space transformation between the flows of the nonlinear system and its linearization that preserves the direction of time.
- This suggests that if one were to design a state feedback controller matrix, K, such that the control signal $u = K(x - x^*)$ asymptotically stabilizes the linearized system about this equilibrium point, then we should achieve adequate regulation of the nonlinear system.

Nonholonomic Control of Two-wheeled Robot

- Let us see how well this "linearized" control works for a two-wheeled robotic vehicle
- The "plant" is a two-wheeled robotic vehicle. Let F denote the force applied along the body's x-axis and T be the torque about the vehicle's center of mass.
- The control vector is $u(t) = [F(t), T(t)]^T$. The state variables are the plant's center of mass, x and y, the body angle, θ, the velocity, v_x, and the angular rate, ω.



$$
\begin{aligned}
\dot{x} &= v_x \cos\theta \\
\dot{y} &= v_x \sin\theta \\
\dot{\theta} &= \omega \\
\dot{v}_x &= F \\
\dot{\omega} &= T
\end{aligned}
$$



Regulation through Local Linearization

- The first step in developing such a state feedback control is to find the linearization in equation (3) 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. ?? become

$$
\begin{aligned}
\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \end{bmatrix} &= \begin{bmatrix} (z_4 + v_d)\cos(z_3 + \theta_d) - \dot{x}_d \\ (z_4 + v_d)\sin(z_3 + \theta_d) - \dot{y}_d \\ z_5 - \dot{\theta}_d \\ -\dot{v}_d \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\
&= F(x) + G(x)u
\end{aligned}
$$

Regulation through Local Linearization

- Computing the Jacobian matrix for F and Gu yields the following linearized system equation

$$
\begin{aligned}
\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \end{bmatrix}
&= \begin{bmatrix} 0 & 0 & -v_d \sin(\theta_d) & \cos(\theta_d) & 0 \\ 0 & 0 & v_d \cos(\theta_d) & \sin(\theta_d) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\
&= Az + Bu
\end{aligned}
$$

Regulation through Local Linearization

- We can then use a number of methods that design stabilizing controllers for this system. In particular, we'll compute the linear quadratic regulator (LQR) that find the state gains, K, such that the controller $u = Kz$ minimizes the cost functional

$$J[u] = \int_0^\infty (z^T z + u^T u) d\tau$$

- This controller was simulated in the following MATLAB script with the desired trajectory

$$\begin{aligned} \dot{x}_d(t) &= 50 \sin\left(\frac{2\pi t}{50}\right), \quad x_d(0) = 0 \\ \dot{y}_d(t) &= 50 \cos\left(\frac{4\pi t}{50}\right), \quad y_d(0) = 0 \end{aligned}$$

- The LQR control was recomputed at each time instant using the desired reference trajectory states. 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.

Regulation through Local Linearization

```
%initialize variables

for time=0:dt:tstop
  vd     = sqrt(xddot^2+yddot^2);
  thetad = mod(atan2(yddot,xddot)+pi, 2*pi)-pi;

  A = [0  0   -vd*sin(thetad)  cos(thetad)  0;
       0  0    vd*cos(thetad)  sin(thêta)   0;
       0  0      0               0          1;
       0  0      0               0          0;
       0  0      0               0          0];
  B = [0 0; 0 0; 0 0 ; 1 0; 0 1];

  k = lqr(A,B,eye(5,5),eye(2,2));
  u = -k*err;

  xddot  = 50*sin(2*pi*time/50);
  yddot  = 50*cos(4*pi*time/50);
  xd     = xd   +  xddot*dt;
  yd     = yd   +  yddot*dt;

  xdot      = vx*cos(theta);
  ydot      = vx*sin(theta);
  thetadot  = omega;
  vxdot     = u(1);
  omegadot  = u(2);

  x         = x + xdot*dt;
  y         = y + ydot*dt;
  theta     = mod(theta+thetadot*dt+pi,2*pi)-pi;
  omega     = omega + omegadot*dt;
  vx        = vx + vxdot*dt;
end;
```

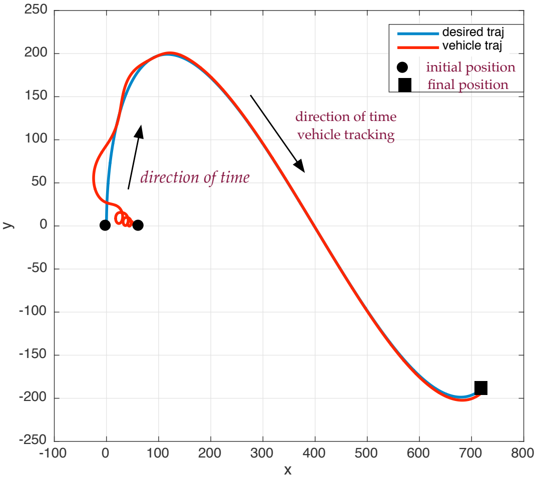
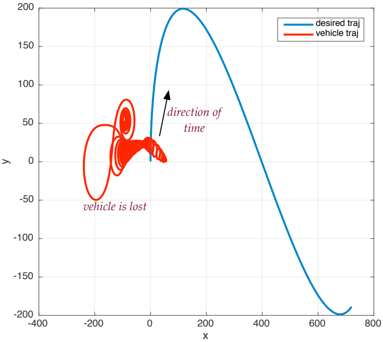


Figure: (left) MATLAB script - (right) trajectories for linearized control with $(x_0, y_0) = (50, 0)$

Regulation through Local Linearization

A 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.



Regulation through Global Linearization

- One way of overcoming this limitation is to adopt a more sophisticated *feedback linearization* method that uses feedback to force the system to appear as a linear dynamical system.
- This feedback linearization is a state transformation that transforms the original system states onto a state vector consisting of the desired tracking outputs and their derivatives.
- The advantage of this approach is that if that state transformation is "global", then the controls we develop for this "feedback" linearized system are also global and can thereby ensure asymptotic tracking of the reference trajectory for any initial vehicle condition.

Regulation through Global Linearization

- 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,

$$
\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \\ v_x \\ \omega \\ F \end{bmatrix} = \begin{bmatrix} v_x \cos\theta \\ v_x \sin\theta \\ \omega \\ F \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}
$$

Regulation through Global Linearization

- We now introduce a state transformation which is obtained by taking the derivatives of the tracking error.

$$
\begin{aligned}
z_1 &= x - x_d & z_2 &= \dot{x} - \dot{x}_d, & z_3 &= \ddot{x} - \ddot{x}_d \\
z_4 &= y - y_d, & z_5 &= \dot{y} - \dot{y}_d, & z_6 &= \ddot{y} - \ddot{y}_d
\end{aligned}
$$

- The differential equations for these components are then readily computed as

$$
\begin{aligned}
\dot{z}_1 &= v_x \cos\theta - \dot{x}_d = \boxed{z_2} \\
\dot{z}_2 &= F \cos\theta - v_x \omega \sin\theta - \ddot{x}_d = \boxed{z_3} \\
\dot{z}_3 &= \boxed{v_1 \cos\theta - v_x v_2 \sin\theta - (2F\omega \sin\theta + v_x \omega^2 \cos\theta) - \dddot{x}_d} \\
\dot{z}_4 &= v_x \sin\theta - \dot{y}_d = \boxed{z_5} \\
\dot{z}_5 &= F \sin\theta + v_x \omega \cos\theta - \ddot{y}_d = \boxed{z_6} \\
\dot{z}_6 &= \boxed{v_1 \sin\theta + v_x v_2 \cos\theta + (2F\omega \cos\theta - v_x \omega^2 \sin\theta) - \dddot{y}_d}
\end{aligned}
$$

Regulation through Global Linearization

- 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,

$$
\begin{aligned}
\left[\begin{array}{c} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \hline \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \end{array}\right] &= \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} z_1 \\ z_2 \\ z_3 \\ \hline z_4 \\ z_5 \\ z_6 \end{array}\right] + \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array}\right] \left[\begin{array}{c} -(2F\omega \sin\theta + v_x\omega^2 \cos\theta) - \dddot{x}_d \\ 2F\omega\cos\theta - v_x\omega^2 \sin\theta - \dddot{y}_d \end{array}\right] \\
&\quad + \left[\begin{array}{cc} \cos\theta & -v_x \sin\theta \\ \sin\theta & v_x\cos\theta \end{array}\right] \left[\begin{array}{c} v_1 \\ v_2 \end{array}\right] \\
\dot{z} &= Az + E(\alpha + \rho v)
\end{aligned}
$$

where A is the linear matrix representing the chain of integrators, α and ρ are matrices whose components are functions of the original system states, F, ω, θ, and v_x.

Regulation Through Global Linearization

- Note that if we select the control v to have the form

$$v = \rho^{-1}\left(-\alpha + \begin{bmatrix} \dddot{x}_d \\ \dddot{y}_d \end{bmatrix} + Kz\right) \tag{4}$$

  Then the resulting state equation is given by

$$\dot{z} = (A + EK)z \tag{5}$$

  where

$$\begin{aligned} E^T &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ K &= \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \end{bmatrix} \end{aligned}$$

- The important thing to note here is that equation (5) is a *linear* differential equation and so if we can select K so that $A + EK$ is a Hurwitz matrix, then we would have globally stabilized our vehicular system using the control in equation (4).

Regulation through Global Linearization

```
%initialize variables

for time=0:dt:tstop


  xddot   =  50*sin(2*pi*time/50);
  xdddot  =  50*(2*pi/50)*cos(2*pi*time/50);
  xddddot = -50*(2*pi/50)*(2*pi/50)*sin(2*pi(time/50);
  yddot   =  50*cos(4*pi*time/50);
  ydddot  = -50*(4*pi/50)*sin(4*pi*time/50);
  yddddot = -50*(4*pi/50)*(4*pi/50)*cos(4*pi*time/50);
  xd     = xd  +  xddot*dt;
  yd     = yd  +  yddot*dt;

  xdot     = vx*cos(theta);
  ydot     = vx*sin(theta);
  xddot    = F*cos(theta)-vx*omega*sin(theta);
  yddot    = F*sin(theta)+vx*omega*cos(theta);

  alf = [-(2*F*omega*sin(theta)+vx*omega^2*cos(theta));
         2*F*omega*cos(theta)-vx*omega^2*sin(theta)];
  rho = [cos(theta) -vx*sin(theta);
        sin(theta) vx*cos(theta)];

  err = [x-xd; xdot-xddot; xddot-xdddot;
        y-yd; ydot-yddot; yddot-ydddot];
  K   = -[1 3 3 0 0 0; 0 0 1 3 3];
  v   = inv(rho)*(-alf+K*err+[xddddot;yddddot]);

  omegadot    = v(2);
  omega       = omega + omegadot*dt;
  thetadot    = mod(theta+thetadot*dt+pi,2*pi)-pi;
  Fdot        = v(1);
  F           = F + Fdot*dt;
  vxdot       = F;
  vx          = vx+vxdot*dt;
  xdot        = vx*cos(theta);
  ydot        = vx*sin(theta);
  x           = x + xdot*dt;
  y           = y + ydot*dt;
end;
```

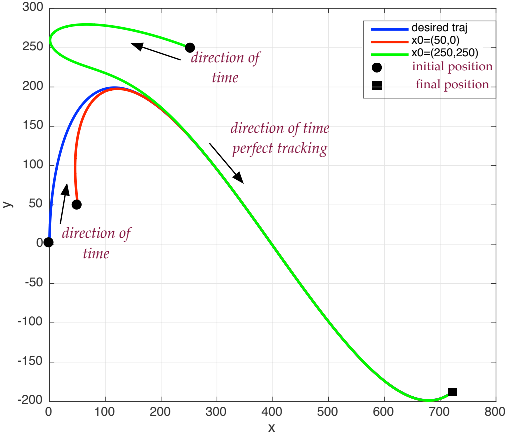


Figure: Feedback Linearized Controller (left) script - (right) trajectories

Bursting in Adaptive Systems

- Adaptive control systems are systems that adapt the controller's gains in response to how well the controlled system is actually behaving.
- Even when this approach is used to adapt a linear dynamical system, the overall system will be highly nonlinear since the control signal is a product of the gain and the system state; both of which in turn are time-varying.
- This nonlinearity makes can make it difficult to ensure the adaption rule is well behaved (i.e. stable) and it sometimes leads to unexpected behaviors such as *bursting*; a phenomenon that occurs when an adaptive system suddenly exhibits a burst of oscillatory behavior before returning back to normal.
- Bursting is a qualitative behavior found in biological systems and turbulent fluid flow. The objective of this section is to illustrate bursting in a simple adaptive control problem and show how the nonlinearities in that system give rise to the bursting phenomena.

Bursting in adaptive linear systems

- As a simple example, let us consider a linear plant whose state $x : \mathbb{R} \to \mathbb{R}$ satisfies

$$\dot{x} = ax + bu \tag{6}$$

in which a and b are unknown system constants but in which we assume we can measure the system state $x(t)$.

- The objective is to select a control signal $u : \mathbb{R} \to \mathbb{R}$ so that x tracks a reference model whose state $x_m : \mathbb{R} \to \mathbb{R}$ satisfies

$$\dot{x}_m = -a_m x_m + b_m r \tag{7}$$

where $a_m > 0$, $b_m > 0$, $r(t)$, and $x_m(t)$ are all known.

- The control input, u, is generated by the following equation

$$u(t) = k_1 x(t) + k_2 r(t) \tag{8}$$

where k_1 and k_2 are "control gains".

- The objective is to select these gains so that the model tracking error, $e = x - x_m$, asymptotically goes to zero as time goes to infinity.

Bursting in adaptive linear systems

- Let us first note that the tracking error satisfies the differential equation

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{x}_m \\ &= ax + b(k_1 x + k_2 r) + a_m x - b_m r \end{aligned}$$

- Note that if we select $k_1^* = -\frac{a+a_m}{b}$ and $k_2^* = \frac{b_m}{b}$, then

$$\begin{aligned} \dot{e} &= ax + b(k_1^* x + k_2^* r) + a_m x - b_m r \\ &= ax + b\left(-\frac{a + a_m}{b} x + \frac{b_m}{b} r\right) + a_m x - b_m r \\ &= -a_m(x - x_m) = -a_m e \end{aligned} \tag{9}$$

- Since $a_m > 0$, this would imply that for $k_1 = k_1^*$ and $k_2 = k_2^*$ that $e(t)$ asymptotically goes to 0 as $t \to \infty$.

Bursting in adaptive linear systems

- This particular choice for the control gains, however, requires that one know what a and b are ahead of time. Since these system parameters are unknown, our actual control law uses a k_1 and k_2 that are not equal to the "optimal" values.
- For this non-optimal choice, the modeling error, e, satisfies

$$\begin{aligned}\dot{e} &= ax + b(k_1 x + k_2 r) + a_m x - b_m r \\ &= ax + b((k_1 - k_1^*)x + (k_2 - k_2^*)r) + bk_1^* x + bk_2^* r + a_m - b_m r\end{aligned}$$

- We can then use equation (9) to reduce the above equation to

$$\dot{e} = -a_m e + b(k_1 - k_1^*)x + b(k_2 - k_2^*)r \qquad (10)$$

which more clearly shows how deviations in k_1 and k_2 from their optimal values impact the model tracking error.

Bursting in Adaptive Linear Systems

- In model reference adaptive control (MRAC), one seeks an adaptation rule that adjusts the controller parameters k_1 and k_2 to ensure model tracking.
- This adaptation rule may be expressed as a differential equation

$$\dot{k}_1 = \phi_1(k_1, x, r, e), \quad \dot{k}_2 = \phi_2(k_2, x, r, e) \tag{11}$$

  where ϕ_1 and ϕ_2 are the functions we need to determine.
- The adaptation rule in equation (11) takes as input the current plant output, x, the reference input r and tracking error e to adjust the gains k_1 and k_2.

- To see what a plausible form for these adaptation functions should be, let us consider a cost function

$$V(e, k_1, k_2) = \frac{1}{2}e^2 + \frac{b}{2\gamma}(k_1 - k_1^*)^2 + \frac{b}{2\gamma}(k_2 - k_2^*)^2$$

  where $\gamma > 0$ is a constant.
- Note that $V \geq 0$ and let us consider the time derivative of this cost, $\dot{V}$, as e, k_1, and k_2 follow the error dynamics in equation (10) under the adaptation rule (11).
- In particular, if one can show that $\dot{V} \leq 0$, then since V is bounded below by zero, then $V(t)$ must be a monotone decreasing function of time that converges to a limit point.
- So let us compute $\dot{V}$

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial e}\dot{e} + \frac{\partial V}{\partial k_1}\dot{k}_1 + \frac{\partial V}{\partial k_2}\dot{k}_2 \\ &= e\dot{e} + \frac{b}{\gamma}(k_1 - k_1^*)\dot{k}_1 + \frac{b}{\gamma}(k_2 - k_2^*)\dot{k}_2 \qquad (12) \end{aligned}$$

- Inserting equation (10) into (12) yields,

$$\dot{V} = -a_m e^2 + b(k_1 - k_1^*)(xe + \frac{1}{\gamma}\dot{k}_1) + b(k_2 - k_2^*)(xr + \frac{1}{\gamma}\dot{k}_2)$$

- Note that if we select

$$\begin{aligned} \dot{k}_1 &= \phi_1(k_1, x, e, r) = -\gamma xe \\ \dot{k}_2 &= \phi_2(k_2, x, e, r) = -\gamma xr \end{aligned} \quad (13)$$

  then we can ensure $\dot{V} < 0$ for all $e \neq 0$ and this would suggest $V(t)$ is a decreasing function of time.

- The adaptation rule in equation (13) is sometimes called the MIT rule and it represents one of the earliest adaptive control laws proposed for aerodynamic systems in the 1960's.

Bursting in Adaptive Linear Systems

- Various forms of the MIT rule have been used, but one well known variation takes the form

$$\begin{aligned} \dot{k}_1 &= -\gamma \tfrac{x}{\theta + x^2} e \\ \dot{k}_2 &= -\gamma x r \end{aligned} \tag{14}$$

  This is sometimes called the *normalized* MIT rule since the size of the k_1 update is normalized by the size of the current system state.

- Let us now look at simulations of the normalized MIT rule and see what happens. In this simulation, we let the plant and reference model be

$$\begin{aligned} \dot{x} &= 1.8x + 2u \\ \dot{x}_m &= -3x + 3r \end{aligned}$$

Bursting in Adaptive Linear Systems

Let us first simulate this with $\gamma = 1$ and $\theta = 0.1$ using the normalized MIT rule from equation (14). The top plot shows $e(t) \to 0$ as time goes to infinity. The lower plot shows k_1 and k_2 and the optimal gains (k_1^* and k_2^*) for this example.

```
for time=0:dt:tstop

    if (mod(time,10)==0)&&(time< 100);
        r=-r;
    end;
    xhat   = x+.0*randn(1,1);
    e      = xhat-xm;

    k1dot = -g*e*r;
    k2dot = -g*e*xhat/(.1+xhat^2);
    %d0dot = -g*e0*ypm;

    k1    = k1 + dt*k1dot;
    k2    = k2 + dt*k2dot;

    k1star = km/kp;
    k2star = -(-ap+am)/kp;

    u     = k1*r+k2*xhat;
    xdot  = -a*x+b*(u);
    xmdot = -am*xm+bm*r;
    x     = x + dt*xdot;
    xm    = xm+ dt*xmdot;

    data = [data; time x xm k1 k2 k1star k2star];
end;
```

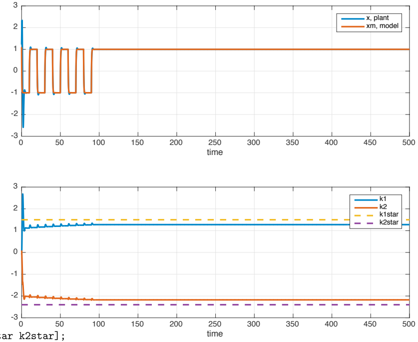


Figure: Simulation of Normalized MIT rule for $\dot{x} = 1.8x + 2u$ and $\dot{x}_m = -3x + 3r$ with no noise in the measured plant state.

Bursting in Adaptive Linear System

- An important thing to note about the simulation is that there is no modeling error in the linear plant and there is no measurement noise.
- To see how our proposed adaptive control works under these conditions, let us assume that the plant state, x, satisfies the ODE,

$$\begin{aligned} \dot{x} &= 1.8x + 0.1x^2 + 2u \\ y &= x + 0.1\nu \end{aligned}$$

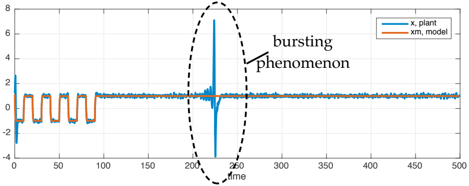
  where ν is a zero mean white noise process with unit variance.
- We now use the measured output y, rather than the true state in our feedback controller and adaptation rules,

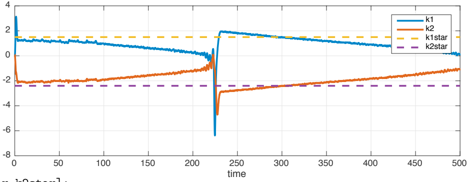
$$\begin{aligned} \hat{e} &= y - x_m \\ u &= k_1 y + k_2 r \\ \dot{k}_1 &= -\gamma \hat{e} \frac{y}{0.1 + y^2} \\ \dot{k}_2 &= -\gamma \hat{e} r \end{aligned}$$

Bursting in Adaptive Linear Systems

The simulation results show t there is a *burst* in which the system becomes highly oscillatory and then recovers tracking again.

```
for time=0:dt:tstop

    if (mod(time,10)==0)&&(time< 100);
        r=-r;
    end;
    xhat   = x+.1*randn(1,1);
    e      = xhat-xm;

    k1dot = -g*e*r;
    k2dot = -g*e*xhat/(.1+xhat^2);
    %d0dot = -g*e0*ypm;

    k1    = k1 + dt*k1dot;
    k2    = k2 + dt*k2dot;

    k1star = km/kp;
    k2star = -(-ap+am)/kp;

    u     = k1*r+k2*xhat;
    xdot  = -a*x+b*(u)+.1*x^2;
    xmdot = -am*xm+bm*r;
    x     = x + dt*xdot;
    xm    = xm+ dt*xmdot;

    data = [data; time x xm k1 k2 k1star k2star];
end;
```





The lower plot shows that k_1 and k_2 have a linear drift and bursting occurs when these gains change sign.

Bursting in Nonlinear Systems

- There are numerous other systems where the bursting phenomenon plays an important functional role.
- This occurs in the dynamics of cell membranes where the "burst" corresponds to an impulse representing the transmission of information down a nerve cell's axon.
- A simplified version of the Fitzhugh-Nagumo model for nerve conduction.

$$\begin{aligned} \dot{v} &= v - \frac{v^3}{3} - w + (y + I) \\ \dot{w} &= \phi(v + a - bw) \end{aligned}$$

  The variable v represents the cell membrane's potential, w represents a "gating" variable, and I represents the stimulating current (assumed to be constant).
- For a suitable constant I, the membrane potential exhibits a sustained oscillation as shown in the top plot of Fig. 9.

Bursting in Nonlinear Systems

Bursting occurs when a small modulation in the driving currrent $I + y$ is introduced. The burst is triggered when the current perturbation, y, exceeds a given threshold.

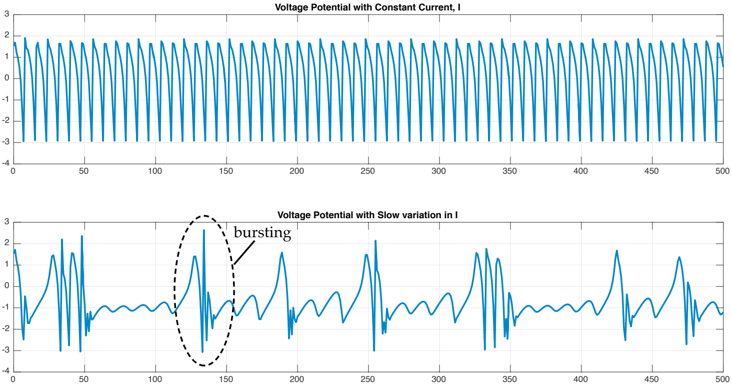


Figure: Membrane Potential v, (top) constant applied current I - (bottom) modulated current $I + y$

Reachability in Nonlinear Control Systems

- Consider a linear time-invariant system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = 0 \tag{15}$$

- We're interested in finding a control input, u that transitions the system state from $x(0) = 0$ to the state $x(T) = x_1$ where $T > 0$.
- The state x_1 is said to be *reachable* if there exists a finite time $T > 0$ and a control input $u : [0, T] \to \mathbb{R}^m$ such that

$$x_1 = \int_0^T e^{A(T-\tau)} Bu(\tau) d\tau$$

- The system is said to be *reachable* if every state in $\mathbb{R}^n$ is reachable from the origin.
- The necessary and sufficient condition for the reachability of the linear system in equation (15) is that

$$\mathrm{rank} \begin{bmatrix} A & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} = n \tag{16}$$

Reachability in Nonlinear Control Systems

- In particular, one usually examines reachability with respect to nonlinear systems that satisfy the following state equations

$$\dot{x}(t) = A(x) + B(x)u$$

where $A(x)$ is an $n \times n$ matrix of functions, $a_{ij} : \mathbb{R}^n \to \mathbb{R}^n$, in which $A(0) = 0$ and $B(x)$ is an $n \times m$ matrix of functions, $b_{ij} : \mathbb{R}^n \to \mathbb{R}^m$.

- Since we know the flows about a hyperbolic equilibrium of the nonlinear system are topologically equivalent to flows about the linearized system's equilibirum, this suggests that one might be able to use the reachability condition in equation (16) to determine whether or not the nonlinear system is reachable.
- This is, however, not as simple as it seems on the surface. A simple example can be used to illustrate how nonlinearity complicates this question.

Reachability in Nonlinear Control Systems

- Let us consider the rigid body dynamics for a satellite controlled by two gas jets.
- It will be convenient to define an inertially fixed reference frame defined a dextral set of 3 vectors, $\{e_1, e_2, e_3\}$, and a reference frame attached to the satellite's body defined by another dextral set of vectors $\{r_1, r_2, r_3\}$.
- We define the direction cosine matrix of the vehicle as a 3 by 3 matrix R such that $Re_i = r_i$ for $i = 1, 2, 3$. This matrix may be obtained through the sequence of rotations $\{\theta_1, \theta_2, \theta_3\}$ about the body frame $\{r_1, r_2, r_3\}$ assuming that the body was originally aligned with the inertial frame $\{e_1, e_2, e_3\}$. These angles are called Euler angles.
- For $i = 1, 2, 3$ let c_i denote $\cos(\theta_i)$ and s_i denote $\sin(\theta_i)$ then we obtain the following explicit expression for the direction cosine matrix,

$$R = \begin{bmatrix} c_2 c_3 & -c_2 s_3 & s_2 \\ s_1 s_2 c_3 + s_3 c_1 & -s_1 s_2 s_3 + c_3 c_1 & -s_1 c_2 \\ -c_1 s_2 c_3 + s_3 s_1 & c_1 s_2 s_3 + c_3 s_1 & c_1 c_2 \end{bmatrix}$$

Reachability in Nonlinear Control Systems

- The body's velocity vector, ω is defined about the 3 body axes, $\{r_1, r_2, r_3\}$.
- The direction cosines and the angular velocities satisfy the following set of ordinary differential equations

$$\begin{aligned} J\dot{\omega} &= S(\omega)J\omega + \sum_{i=1}^{m} b_i u_i \qquad (17) \\ \dot{R} &= S(\omega)R \end{aligned}$$

where J is the inertia matrix, $\{b_i\}_{i=1}^{m}$, are vectors characterizing how the gas jets, u_i, for $i = 1, 2, 3$ impact the body's angular acceleration, $\dot{\omega}$, and

$$S(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}$$

We could also have defined these angles through a different sequence of rotations. Fig. 10 shows how these rotations would have occurred if we had used the sequence $\{\theta_3, \theta_1, \theta_2\}$ rather than the $1-2-3$ sequence.

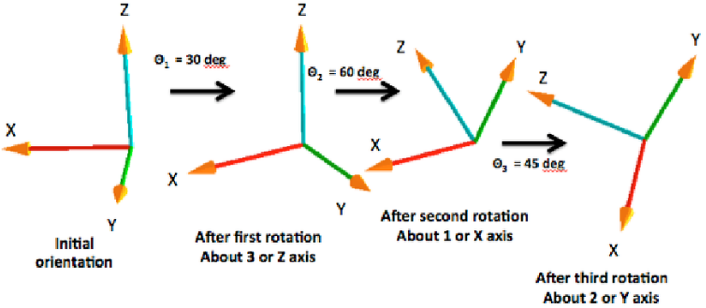


Figure: 3-1-2 Rotation of Euler Angles

- We are more interested in how the three Euler angles change over time since these are tied to the inertial frames. So rather than characterizing the attitude's time rate of change through its direction cosine matrix (i.e., $\dot{R} = S(\omega)R$), we determine the time derivatives for the Euler angles θ_i, for $i = 1, 2, 3$.
- For the $1-2-3$ sequence of body rotations it can be shown that (?, ?)

$$\dot{\theta} = \begin{bmatrix} c_2 & 0 & s_2 \\ s_2 s_1/c_1 & 1 & -c_2 s_1/c_1 \\ -s_2/c_1 & 0 & c_2/c_1 \end{bmatrix} \omega \tag{18}$$

Equations (17,18) will therefore be used in our example.

Reachability in Nonlinear Control Systems

- For this example, we will see whether or not the origin is reachable from an initial state where $\theta_1 = -0.3$, $\theta_2 = \theta_3 = 0$, and $\omega_1 = \omega_2 = \omega_3 = 0$ using only two gas jets aligned with the vehicle's pitch and yaw axes. We'll assume that the inertia matrix is $J = \mathrm{diag}(0.1, 1.0, 1.0)$.
- So the vehicle is starting out at rest aligned with the reference frame but with a roll angle of set of -0.3 radians. Since the two gas jets are aligned to the pitch and yaw axis then in equation (17) we know $m = 2$, $b_1 = [0, 1, 0]^T$ and $b_2 = [0, 0, 1]^T$.

Reachability in Nonlinear Control Systems

- Let us first look at the linearization of this system about the set point $\omega = 0$ and $\theta = 0$. We let the linearized system's state vector be $z_1 = \omega_1$, $z_2 = \omega_2$, $z_3 = \omega_3$, $z_4 = \theta_1$, $z_5 = \theta_2$, and $z_6 = \theta_3$.
- It can be readily shown that for this fixed point the linearized system's state equations are

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = Az + Bu$$

Reachability in Nonlinear Control Systems

- If we use the reachability condition in equation (16) then we see that

$$\operatorname{rank}\left[\begin{array}{ccccc} A & AB & A^2 & \cdots & A^5 B \end{array}\right] = 4 < 6$$

  which means the linearized system is not reachable.
- In particular, this means that we cannot find a control that moves the roll angle, θ_1, to zero. For the linearization this should be obvious since we see there is no control action that impacts $\dot{z}_1$.
- The same, however, may not be said of the original nonlinear system.

Reachability of Nonlinear Control Systems

Let us consider the following control sequence

$$u = \begin{bmatrix} p\left(\frac{t-0.1}{0.4}\right) - p\left(\frac{t-1.8}{0.4}\right) - p\left(\frac{t-3.1}{0.4}\right) + p\left(\frac{t-5.25}{0.4}\right) \\ p\left(\frac{t-0.1}{0.4}\right) - p\left(\frac{t-1.8}{0.4}\right) - p\left(\frac{t-6.1}{0.4}\right) + p\left(\frac{t-7.725}{0.4}\right) \end{bmatrix}$$

where $p(t) = 1$ for $0 < t < 1$ and is zero elsewhere.

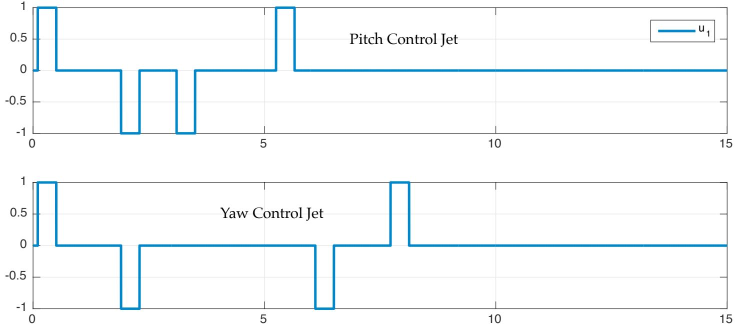


Figure: Gas jet commands used to null the roll angle with only pitch and yaw control

Reachability in Nonlinear Control Systems

When both the pitch and yaw rates accerate at the same time, there is an induced roll rate which we will null out the initial roll angle. We can then use the pitch and yaw jets to then return the other angles to zero. So even though the origin is not reachable from the initial attitude using the linearized system, it is reachable using the original nonlinear dynamics.

```
J = [.1 0 0; 0 1 0; 0 0 1];

phi=-.3;theta=0;psi=0;
p = [phi;theta;psi];pdot = zeros(3,1);
w  = zeros(3,1);wdot = zeros(3,1);
u  = zeros(3,1);

tstop = 15;
dt    = .001;
data = [];

tp1    = .1;tp2    = .5;tp3    = 1; tp4    = 1.4;


for time = 0:dt:tstop;

    data = [data ; time (180/pi)*[phi theta psi w(1) w(2) w(3) ]];

    u=zeros(3,1);
    psgn = 1;tpulse = .78;

    tpulse = 1.8;
    u = jetpulse(time,tp1,tpulse,2,psgn,u);
    u = jetpulse(time,tp2,tpulse,2,-psgn,u);
    u = jetpulse(time,tp1,tpulse,3,psgn,u);
    u = jetpulse(time,tp2,tpulse,3,-psgn,u);

    tpulse1=2.15;
    u = jetpulse(time,3+tp1,tpulse1,2,-psgn,u);
    u = jetpulse(time,3+tp2,tpulse1,2,psgn,u);


    tpulse1=1.625;
    u = jetpulse(time,6+tp1,tpulse1,3,-psgn,u);
    u = jetpulse(time,6+tp2,tpulse1,3,psgn,u);

    S = [0 w(3) -w(2);-w(3) 0 w(1); w(2) -w(1) 0];
    wdot = inv(J)*(S*J*w+u);
    s1 = sin(p(1));c1=cos(p(1));
    s3 = sin(p(3));c3=cos(p(3));
    s2 = sin(p(2));c2=cos(p(2));
    AD = [c2 0 s2; s2*s1/c1  1 -c2*s1/c1;  -s2/c1 0 c2/c1];
    pdot = AD*w;
    w        = w + wdot*dt;
    p        = p + pdot*dt;
```