P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

ON

MODELING, ANALYSIS AND DESIGN

OF

HIGH AUTONOMY CONTROL SYSTEMS

Panos J. Antsaklis
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556 USA

Ph : 219 / 239-5792
Fax : 219 / 239-8007
Email : flxfsn@irishmvs.cc.nd.edu
antsakli@saturn.ece.nd.edu

ABSTRACT

Intelligent control systems with high degree of autonomy should perform well under significant uncertainties in the system and environment for extended periods of time, and they must be able to compensate for certain system failures without external intervention. Such control systems evolve from conventional control systems by adding intelligent components, and their development requires interdisciplinary research.

After an introduction to the main ideas in Intelligent Autonomous Control Systems, a brief overview of this research area is given and certain important issues in modeling, analysis and design are discussed. Concepts and methods from Computer Science, Artificial Intelligence and Operations Research can and are being currently used quite successfully, together with methods from Control Systems, to address some of the problems. There are however additional problems in Intelligent Autonomous Control Systems of rather unique character, mainly pertaining to the real-time requirements, the dynamic characteristics of the systems involved and the need for a concrete theoretical framework to study important control properties of such systems; these problems require special consideration and need rather new concepts and methods and the researches into these problems are mostly at early stages. An outline of some of our recent work in Intelligent Autonomous Control Systems is presented, with emphasis on Reconfigurable Control and FDI, on the use of Neural Networks in Control and on Hybrid Systems modeling, analysis and design.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

Introduction

In the design of controllers for complex dynamical systems there are needs today that cannot be successfully addressed with the existing conventional control theory. They mainly pertain to the area of uncertainty. Heuristic methods may be needed to tune the parameters of an adaptive control law. New control laws to perform novel control functions to meet new objectives should be designed while the system is in operation. Learning from past experience and planning control actions may be necessary. Failure detection and identification is needed. Such functions have been performed in the past by human operators. To increase the speed of response, to relieve the operators from mundane tasks, to protect them from hazards, high degree of autonomy is desired. To achieve this, high level decision making techniques for reasoning under uncertainty and taking actions must be utilized. These techniques, if used by humans, may be attributed to *intelligent* behavior. Hence, one way to achieve high degree of autonomy is to utilize high level decision making techniques, intelligent methods, in the autonomous controller. *Autonomy is the objective, and intelligent controllers are one way to achieve it.*

The need for quantitative methods to model and analyze the dynamical behavior of such autonomous systems presents significant challenges well beyond current capabilities. It is clear that the development of autonomous controllers requires significant interdisciplinary research effort as it integrates concepts and methods from areas such as Control, Identification, Estimation, and Communication Theory, Computer Science, Artificial Intelligence, and Operations Research. In [1] topics important to realization of Intelligent Autonomous Control systems are discussed at length. The chapters of this book together with the references therein provide a good introduction to the area. In [2] many of the concepts and ideas discussed in the first, general part of this paper are elaborated upon.

Conventional Control - Evolution

The first feedback device on record was the water clock invented by the Greek Ktesibios in Alexandria Egypt around the 3rd century B.C. This was certainly a successful device as water clocks of similar design were still being made in Baghdad when the Mongols captured the city in 1258 A.D.! The first mathematical model to describe plant behavior for control purposes is attributed to J.C. Maxwell, of the Maxwell equations' fame, who in 1868 used differential equations to explain instability problems encountered with James Watt's flyball governor; the governor was introduced in 1769 to regulate the speed of steam engine vehicles. Control theory made significant strides in the past 120 years, with the use of frequency domain methods and Laplace transforms in the 1930s and 1940s and the development of optimal control methods and state space analysis in the 1950s and 1960s. Optimal control in the 1950s and 1960s, followed by progress in stochastic, robust and adaptive control methods in the 1960s to today, have made it possible to control more accurately significantly more complex dynamical systems than the original flyball governor.

Lessons from the Flyball Governor : When J.C Maxwell used mathematical modeling and methods to explain instability problems encountered with James Watt's flyball governor, it demonstrated the importance and usefulness of mathematical models and methods in understanding complex phenomena and signaled the beginning of mathematical system and control theory. It also signalled the end of the era of intuitive invention. The flyball governor worked fine for a long time meeting the needs. As time progressed and more demands were put on the device there came a point when better and deeper understanding of the device was necessary as it started exhibiting some undesirable and unexplained behavior, in particular oscillations. This is quite typical of the situation in man made systems even today. Similarly to the flyball governor, one can rely on systems developed based mainly on intuitive inventions so much. To be able to control highly

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

complex and uncertain systems we need deeper understanding of the processes involved and systematic design methods, we need quantitative models and design techniques. This is the lesson from the flyball governor.

Conventional control design : Conventional control systems are designed today using mathematical models of physical systems. A mathematical model, which captures the dynamical behavior of interest, is chosen and then control design techniques are applied, aided by CAD packages, to design the mathematical model of an appropriate controller. The controller is then realized via hardware or software and it is used to control the physical system. The procedure may take several iterations. The mathematical model of the system must be "simple enough" so that it can be analyzed with available mathematical techniques, and "accurate enough" to describe the important aspects of the relevant dynamical behavior. It approximates the behavior of a plant in the neighborhood of an operating point.

The control methods and the underlying mathematical theory were developed to meet the ever increasing control needs of our technology. The need to achieve the demanding control specifications for increasingly complex dynamical systems has been addressed by using more complex mathematical models such as nonlinear and stochastic ones, and by developing more sophisticated design algorithms for, say, optimal control. The use of highly complex mathematical models however, can seriously inhibit our ability to develop control algorithms. Fortunately, simpler plant models, for example linear models, can be used in the control design; this is possible because of the *feedback* used in control which can tolerate significant model uncertainties. When the fixed feedback controllers are not adequate, then adaptive controllers are used. Controllers can then be designed to meet the specifications around an operating point, where the linear model is valid and then via a scheduler a controller emerges which can accomplish the control objectives over the whole operating range. This is, for example, the method typically used for aircraft flight control and it is a method to design fixed controllers for certain classes of nonlinear systems. Adaptive control in conventional control theory has a specific and rather narrow meaning. In particular it typically refers to adapting to variations in the constant coefficients in the equations describing the linear plant:; these new coefficient values are identified and then used, directly or indirectly, to reassign the values of the constant coefficients in the equations describing the linear controller. Adaptive controllers provide for wider operating ranges than fixed controllers and so conventional adaptive control systems can be considered to have *higher degrees of autonomy* than control systems employing fixed feedback controllers.

Intelligent Control for High Autonomy Systems

There are cases where we *need to significantly increase the operating range*. We must be able to deal effectively with significant uncertainties in models of increasingly complex dynamical systems in addition to increasing the validity range of our control methods. We need to cope with significant unmodelled and unanticipated changes in the plant, in the environment and in the control objectives. This will involve the use of intelligent decision making processes to generate control actions so that certain performance level is maintained even though there are drastic changes in the operating conditions. It is useful to keep in mind an example which I call the *Houston control example* . It is an example that sets goals for the future and it also teaches humility as it indicates how difficult demanding and complex autonomous systems can be. Currently, if there is a problem on the space shuttle, the problem is addressed by the large number of engineers working in Houston Control, the ground station. When the problem is solved the specific detailed instructions about how to deal with the problem are sent to the shuttle. Imagine the time when we will need the tools and expertize particular problem ares of all Houston Control engineers aboard the space shuttle, space vehicle, for extended space travel.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

In view of the above it is quite clear that in the control of systems there are requirements today that cannot be successfully addressed with the existing conventional control theory. They mainly pertain to the area of uncertainty, present because of poor models due to lack of knowledge, or due to high level models used to avoid excessive computational complexity. Heuristic methods may be needed to tune the parameters of an adaptive control law. New control laws to perform novel control functions should be designed while the system is in operation. Learning from past experience and planning control actions may be necessary. Failure detection and identification is needed. These functions have been performed in the past by human operators. To increase the speed of response, to relieve the pilot from mundane tasks, to protect operators from hazards, autonomy is desired. It should be pointed out that several functions proposed in later sections, to be part of the high autonomy control system, have been performed in the past by separate systems; examples include fault trees in chemical process control for failure diagnosis and hazard analysis, and control system design via expert systems.

The design process emphasized here is a *bottom-up approach*. One turns to more sophisticated controllers only if simpler ones cannot meet the required objectives. The need to use intelligent autonomous control stems from the need for an increased level of autonomous decision making abilities in achieving complex control tasks.

An Intelligent High Autonomy Control System Architecture For Future Space Vehicles

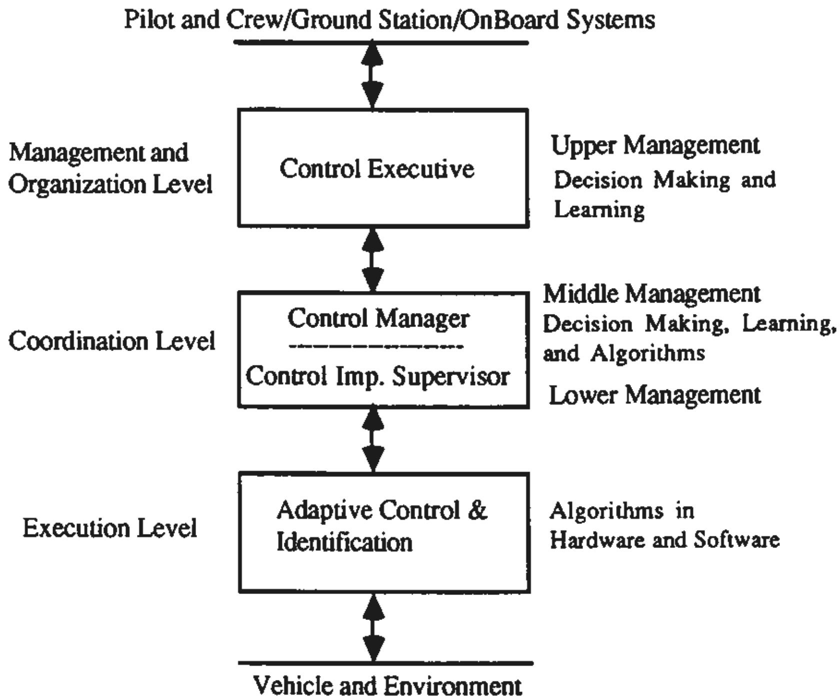
To illustrate the concepts and ideas involved and to provide a more concrete framework to discuss the issues, a hierarchical functional architecture of an intelligent controller that is used to attain high degrees of autonomy in future space vehicles is briefly outlined; full details can be found in [2]. This hierarchical architecture has three levels, the Execution Level, the Coordination Level, and the Management and Organization Level. The architecture exhibits certain characteristics, which have been shown in the literature to be necessary and desirable in autonomous systems. Based on this architecture we identify the important fundamental issues and concepts that are needed for an autonomous control theory.

Architecture Overview: Structure and Characteristics: The overall functional architecture for an autonomous controller is given by the architectural schematic of the figure below. This is a functional architecture rather than a hardware processing one; therefore, it does not specify the arrangement and duties of the hardware used to implement the functions described. Note that the processing architecture also depends on the characteristics of the current processing technology; centralized or distributed processing may be chosen for function implementation depending on available computer technology.

The architecture in Figure has three levels. At the lowest level, the Execution Level, there is the interface to the vehicle and its environment via the sensors and actuators. At the highest level, the Management and Organization Level, there is the interface to the pilot and crew, ground station, or onboard systems. The middle level, called the Coordination Level, provides the link between the Execution Level and the Management Level. Note that we follow the somewhat standard viewpoint that there are three major levels in the hierarchy. *It must be stressed that the system may have more or fewer than three levels*. Some characteristics of the system which dictate the number of levels are the extent to which the operator can intervene in the system's operations, the degree of autonomy or level of intelligence in the various subsystems, the dexterity of the subsystems, and the hierarchical characteristics of the plant. Note however that the three levels shown here in figure are applicable to most architectures of autonomous controllers, by grouping together sublevels of the architecture if necessary. As it is indicated in the Figure, the lowest, Execution Level involves conventional control algorithms, while the highest, Management and Organization Level involves only higher level, intelligent, decision making methods. The Coordination Level is the level which

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

provides the interface between the actions of the other two levels and it uses a combination of conventional and intelligent decision making methods.

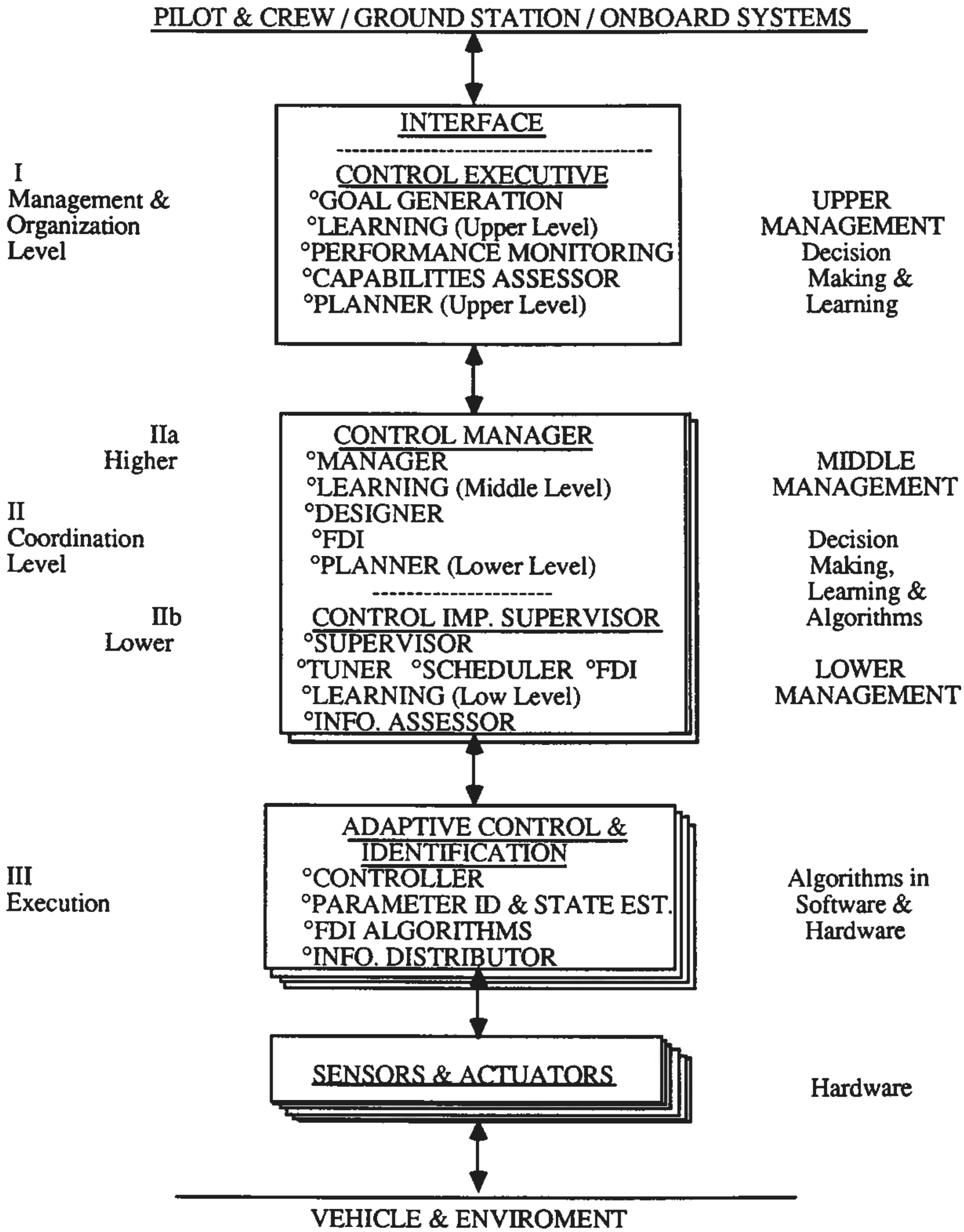


Intelligent Autonomous Controller Functional Architecture

The sensors and actuators are implemented mainly with hardware. Software and perhaps hardware are used to implement the Execution Level. Mainly software is used for both the Coordination and Management Levels. There are *multiple copies* of the control functions at each level, more at the lower and fewer at the higher levels.

Note that the autonomous controller is only one of the autonomous systems on the vehicle. It is responsible for all the functions related to the control of the physical system and allows for continuous online development of the autonomous controller and to provide for various phases of mission operations. The tier structure of the architecture allows us to build on existing advanced control theory. Development progresses, creating each time, higher level adaptation and a new system which can be operated and tested independently. The autonomous controller performs many of the functions currently performed by the pilot, crew, or ground station. The pilot and crew are thus relieved from mundane tasks and some of the ground station functions are brought aboard the vehicle. In this way the degree of autonomy of the vehicle is increased. The Figure in the next page shows more information of the hierarchical functional architecture. More details can be found in [2].

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he
T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel,
November 2-3, 1992.



Intelligent Autonomous Controller Functional Architecture Details

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

Characteristics of Hierarchical Intelligent Controllers for High Autonomy Systems

Based on the architecture described above, important fundamental concepts and characteristics that are needed for an autonomous control theory are now identified. The fundamental issues which must be addressed for a quantitative theory of intelligent autonomous control are discussed.

There is a *successive delegation of duties* from the higher to lower levels; consequently the *number of distinct tasks* increases as we go down the hierarchy. Higher levels are concerned with slower aspects of the system's behavior and with its larger portions, or broader aspects. There is then a *smaller contextual horizon at lower levels,* i.e. the control decisions are made by considering less information Also notice that higher levels are concerned with *longer time horizons* than lower levels. Due to the fact that there is the need for high level decision making abilities at the higher levels in the hierarchy, the proposition has been put forth that there is *increasing intelligence* as one moves from the lower to the higher levels. This is reflected in the use of fewer conventional numeric-algorithmic methods at higher levels as well as the use of more symbolic-decision making methods. This is the "principle of increasing intelligence with decreasing precision" of Saridis. The decreasing precision is reflected by a decrease in *time scale density,* decrease in *bandwidth* or *system rate,* and a decrease in the *decision (control action) rate.* (These properties have been studied for a class of hierarchical systems in [3].) All these characteristics lead to a decrease in *granularity of models* used, or equivalently, to an *increase in model abstractness.* Model granularity also depends on the *dexterity* of the autonomous controller. The Execution Level of a highly dexterous controller is very sophisticated and it can accomplish complex control tasks. The control implementation supervisor can issue high level commands to a dexterous controller, or it can completely dictate each command in a less dexterous one. The simplicity, and level of abstractness of macro commands in an autonomous controller depends on its dexterity. The more sophisticated the Execution Level is, the simpler are the commands that the control implementation supervisor needs to issue. Notice that a very dexterous robot arm may itself have a number of autonomous functions. If two such dexterous arms were used to complete a task which required the coordination of their actions then the arms would be considered to be two dexterous actuators and a new supervisory controller for high autonomy would be placed on top for the supervision and coordination task. In general, this can happen recursively, adding more intelligent autonomous controllers as the lower level tasks, accomplished by autonomous systems, need to be supervised.

There is an ongoing *evolution* of the intelligent functions of an autonomous controller. It is interesting to observe the following: Although there are characteristics which separate intelligent from non-intelligent systems, as intelligent systems evolve, the distinction becomes less clear. Systems which were originally considered intelligent evolve to gain more character of what are considered to be non-intelligent, numeric-algorithmic systems. An example is a route planner. Although there are AI route planning systems, as problems like route planning become better understood, more conventional numeric-algorithmic solutions are developed. The AI methods which are used in intelligent systems, help us to understand complex problems so we can organize and synthesize new approaches to problem solving, in addition to being problem solving techniques themselves. AI techniques can be viewed as research vehicles for solving very complex problems. As the problem solution develops, purely algorithmic approaches, which have desirable implementation characteristics, substitute AI techniques and play a greater role in the solution of the problem. It is for this reason that we concentrate on achieving autonomy and not on whether the underlying system can be considered "intelligent".

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

Research Areas

A number of research areas important to intelligent autonomous systems may be identified. They include the areas of:

Intelligent Systems, Hierarchical Systems, Planning and Expert Systems, Machine Learning, Fuzzy Control, Discrete Event Systems Theory and Simulation, Hybrid Systems, Restructurable Control, Failure Detection and Identification (FDI), and Neural Networks.

In the following we shall discuss some research results in the areas of Hybrid Systems, in Restructurable Control and FDI and in Neural Networks in Control Systems. To further stress the importance of mathematical modeling and methods and provide an introduction for the discussion of hybrid systems that follows some concepts and approaches of quantitative modeling are now presented:

Quantitative Models: For highly autonomous control systems, normally the plant is so complex that it is either impossible or inappropriate to describe it with conventional mathematical system models such as differential or difference equations. Even though it might be possible to accurately describe some system with highly complex nonlinear differential equations, it may be inappropriate if this description makes subsequent analysis too difficult or too computationally complex to be useful. The complexity of the plant model needed in design depends on both the complexity of the physical system and on how demanding the design specifications are. There is a tradeoff between model complexity and our ability to perform analysis on the system via the model. However, if the control performance specifications are not too demanding, a more abstract, higher level, model can be utilized, which will make subsequent analysis simpler. This model intentionally ignores some of the system characteristics, specifically those that need not be considered in attempting to meet the particular performance specifications. For example, a simple temperature controller could ignore almost all dynamics of the house or the office and consider only a temperature threshold model of the system to switch the furnace off or on.

Logical Discrete Event System (DES) models such as those used in the Ramadge-Wonham framework or such as Petri nets are quite useful for modeling the higher level decision making processes in the intelligent autonomous controller. It was shown in [4, 5] that DES-theoretic models can be used to represent AI planning systems which are an important component of the intelligent autonomous controller. The "timed" or "performance" models from DES-theoretic research will also prove useful in modeling components of the higher levels in the intelligent autonomous controller. For instance, queueing network models, Markov chains, etc. will be useful. The choice of whether to use such models will, of course, depend on what properties of the autonomous system need to be studied.

The quantitative, systematic techniques for modeling, analysis, and design of control systems are of central and utmost practical importance in conventional control theory. Similar techniques for intelligent autonomous controllers do not exist. This is mainly due to the *hybrid* structure (nonuniform, nonhomogeneous nature) of the dynamical systems under consideration; they include *both continuous-state and discrete-state systems*. Modeling techniques for intelligent autonomous systems must be able to support a macroscopic view of the dynamical system, hence it is necessary to represent both numeric and symbolic information. The nonuniform components of the intelligent controller all take part in the generation of the low level control inputs to the dynamical system, therefore they all must be considered in a complete analysis.

It is our viewpoint that research should begin by using different models for different components of the intelligent autonomous controller. Full hybrid models that can represent large portions or even the whole autonomous system should be examined but much can be attained by

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

using the best available models for the various components of the architecture and joining them via some appropriate interconnecting structure. For instance, research in the area of systems that are modelled with a logical DES model at the higher levels and a difference equation at the lower level should be examined; see discussion below on hybrid systems. In any case, our modeling philosophy requires the examination of *hierarchical* models.

Summary of some research directions: One can roughly categorize research in the area of intelligent autonomous control into two areas: conventional control theoretic research, addressing the control functions at the Execution and Coordination Levels, and the modeling, analysis, and design of higher level decision making systems found in the Management and Organization Level, and the Coordination Level.

Some control theoretic techniques offer modeling, analysis, and design techniques for the higher level decision making mechanisms in the intelligent autonomous controller. For instance, it can be shown [5] that AI planning problems can be studied in a discrete event system (DES) theoretic framework by utilizing the A^* algorithm. To determine how to utilize AI techniques it is productive to study the relationships between AI and conventional control methods. [4] provides a systems and control theoretic perspective on AI planning systems. It is also important to study how to use conventional control techniques in conjunction with AI techniques to perform autonomous control functions. For instance, in [6] the authors introduce a fault detection and identification (FDI) system that is composed of AI decision making mechanisms and conventional FDI algorithms.

It is important to note that in order to obtain a high degree of autonomy it is absolutely necessary to, in some way, *adapt or learn*; in [7] it is shown how an expert learning system can be used to tune the parameters of an adaptive controller for a large flexible space antenna so to optimize its performance and then also enhance the operating range of the system by storing this information for future use. Neural networks offer methodologies to perform learning functions in the intelligent autonomous controller. In general, there are potential applications of neural networks at all levels of hierarchical intelligent controllers that provide higher degrees of autonomy to systems. Neural networks are useful at the lowest Execution level - where the conventional control algorithms are implemented via hardware and software - through the Coordination level, to the highest Organizational level, where decisions are being made based on possibly uncertain and/or incomplete information. One may point out that at the Execution level - conventional control level - neural network properties such the ability for function approximation and the potential for parallel implementation appear to be very important. In contrast, at higher levels abilities such as pattern classification and the ability to store information in a, say, associative memory appear to be of significant interest ; see discussion below and [8]. Learning is of course important at all levels.

We stress that in control systems with high degrees of autonomy we seek to significantly widen the operating range of the system so that significant failures and environmental changes can occur and performance will still be maintained. All of the conventional control techniques are useful in the development of autonomous controllers and they are relevant to the study of autonomous control. It is the case however, that certain techniques are more suitable for interfacing to the autonomous controller and for compensating for significant system failures. For instance the area of "restructurable" or "reconfigurable" control systems studies techniques to reconfigure controllers when significant failures occur (see discussion below and [9]).

Conventional modeling, analysis, and design methods should be used, whenever they are applicable, for the components of the intelligent autonomous control system. For instance, they should be used at the Execution Level of many autonomous controllers. We propose to augment and enhance existing theories rather than develop a completely new theory for the hybrid systems described above; we wish to build upon existing, well understood and proven conventional

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

methods. The symbolic/numeric interface is a very important issue; consequently it should be included in any analysis. There is a need for systematically generating less detailed, more abstract models from differential/difference equation models to be used in higher levels of the autonomous controller; see discussion below on hybrid systems. Tools for the implementation of this *information extraction* also need to be developed (see for instance [10). In this way conventional analysis can be used in conjunction with the developed analysis methods to obtain an overall quantitative, systematic analysis paradigm for intelligent autonomous control systems. In short, we propose to use hybrid modeling, analysis, and design techniques for nonuniform systems. This approach is not unlike the approaches used in the study of any complex phenomena by the scientific and engineering communities.

Intelligent Autonomous Control as a Distinct Research Areas.

There may be the temptation to classify the area of intelligent autonomous systems as simply a collection of methods and ideas already addressed elsewhere, the need only being some kind of intelligent assembly and integration of known techniques. This is of course not true and it reflects wishful thinking rather than the harsh realities. The theory of control systems is not covered by the area of applied mathematics because control has different needs and therefore asks different questions. For example while in applied mathematics the different solutions of differential equations under different initial conditions and forcing functions are of interest, in control one typically is interested in finding the forcing functions that generate solutions, that is system trajectories, that satisfy certain conditions. This is a different problem, related to the first, but its solution requires the development of quite different methods. In a rather analogous fashion the problems of interest in intelligent systems require development of novel concepts, approaches and methods. In particular while computer science typically deals with static systems and no real-time requirements, control systems typically are dynamic and all control laws, intelligent or not, must be able to control the system in real time. So in most cases one cannot really just directly apply computer science methods to these problems. Modifications and extensions are typically necessary for example in the quantitative models used to study such systems. And although say Petri nets may be adequate to model and study the autonomous behavior at certain levels of the hierarchy, these models are not appropriate to address certain questions of importance to control systems such as stability. It is not that quantitative methods developed in other fields are inferior, it is the fact that they were developed to answer different questions. There are in addition problems in intelligent autonomous control systems that are novel and so they have not studied before at any depth. Such is the case of Hybrid systems that combine systems of continuous and discrete state; they are discussed below. The marriage of all these fields can only be beneficial to all. Computer Science and Operation Research methods are increasingly used in control problems, while the control system ideas, such as feedback, and methods that are based on rigorous mathematical framework can provide the base for new theories and methods in those areas.

Hybrid System Modeling and Design

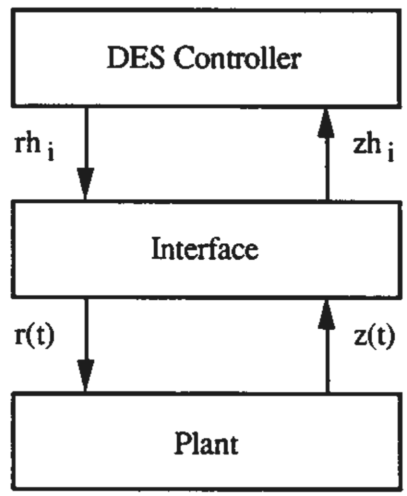
Being able to control a continuous-state system using a discrete-state supervisory controller is a central problem of autonomous control systems. The theory of hybrid system modeling and control addresses some of the important issues of extracting higher level abstract models from more detailed ones. What follows is a synopsis of the approach used in [11] and earlier papers to study hybrid systems:

A hybrid control system contains both continuous-time and discrete event components. Specifically, the plant is a continuous-time system modeled by differential or difference equations, and the controller is a discrete event system modeled by an automaton. In [11] a framework for modeling hybrid control systems including the necessary interface between the plant and controller

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he
T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel,
November 2-3, 1992.

is presented. A method to represent the entire system as a discrete event system is shown, and the concept of determinism is used to analyze hybrid control system behavior and guide in hybrid control system design. In particular, in this work, a flexible and tractable way of modeling hybrid control systems is first presented. The aim is to develop a model which can adequately represent a wide variety of hybrid control systems, while remaining simple enough to permit analysis. Second, a few methods are shown which can be used to analyze and aid in the design of hybrid control systems. These methods relate to the design of the interface which is a necessary component of a hybrid system and reflects both the dynamics of the plant and the aims of the controller.

A hybrid control system, can be divided into three parts as shown in Figure below.



The plant is modeled as a time-invariant, continuous-time system.

$$\dot{x}(t) = f(x(t), r(t))$$
$$z(t) = g(x(t))$$

where x(t), r(t), and z(t) are the state, input, and output respectively. For the purposes of this work we assume that z(t) = x(t). Note that the plant input and output are continuous-time signals.

The controller is a discrete event system which we model as a deterministic automaton. This automaton can be specified by a quintuple, $\{S, E, C, \delta, \phi\}$, where S is the (possibly infinite) set of states, E is the set of *plant events*, C is the set of *controller events*, $\delta : S \times E \rightarrow S$ is the state transition function, and $\phi : S \rightarrow C$ is the output function. The events in set C are called controller events because they are generated by the controller. Likewise, the events in set E are generated by conditions in the plant. The action of the controller can be described by the equations

$$t_i = \delta(t_{i-1}, zh_i)$$
$$rh_i = \phi(t_i)$$

where $t_i \in S$, $zh_i \in E$, and $rh_i \in C$. The index i is analogous to a time index in that it specifies the order of the states or events in a sequence. The input and output signals associated with the controller are asynchronous sequences of events, rather than continuous-time signals. Notice that there is no delay in the controller. The state transition, from t_{i-1} to t_i, and the controller event, rh_i, are generated immediately after the plant event zh_i occurs.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

Interface

The controller and plant cannot communicate directly in a hybrid control system because they each utilize a different type of signal. Thus an interface is required which can convert continuous-time signals to sequences of events and vice versa. The interface consists of two memoryless maps, γ and α. The first map, $\gamma : C \rightarrow \mathbf{R}^m$, converts each controller event to a constant plant input as follows

$$r(t) = \gamma(rh_i)$$

where rh_i is the most recent controller event, previous to time t. The plant input, r(t), can only have certain values, where each value is associated with a particular controller event. Thus the plant input is a piecewise constant signal which may change only when a controller event occurs.

The second map, $\alpha : \mathbf{R}^n \rightarrow E$, is a function which maps the state space of the plant to the set of plant events.

$$zh_i = \alpha(x(t))$$

It would appear from this equation that, as x changes, zh also continuously changes. That is, there is a continuous generation of plant events by the interface because each state is mapped to an event. This is not the case due to two reasons. First, α must be a function which induces equivalence classes on $\mathbf{R}^n$, which form contiguous regions. Second, a plant event is generated only when the state first enters one of these regions. The overall effect is that the state space of the plant is partitioned into a number of regions and each is associated with a unique plant event which is generated whenever the state enters that region. For example, if $x(t) \in \mathbf{R}^1$, α could map all positive x to one event and all negative x to a second event. As this system operated a plant event would be generated whenever x changed sign.

DES Plant

If the plant and interface of a hybrid control system are viewed as a single component, this component behaves like a discrete event system. It is advantageous to view a hybrid control system this way because it allows it to be modeled as two interacting discrete event systems which are more easily analyzed than the system in its original form. The discrete event system which models the plant and interface is called the *DES Plant* and is modeled as an automaton similar to the controller. The automaton is specified by a quintuple, $\{P, E, C, \psi, \lambda\}$, where P is the set of states, E and C are the sets of plant events and controller events, $\psi : P \times C \rightarrow P$ is the state transition function, and $\lambda : P \rightarrow E$ is the output function. The behavior of the DES plant is as follows

$$q_{i+1} = \psi(q_i, rh_i)$$
$$zh_i = \lambda(q_i)$$

where $q_i \in P$, $rh_i \in C$, and $zh_i \in E$. There are two differences between the DES plant and the controller. First, as can be seen from the equations, the state transitions in the DES plant do not occur immediately when a controller event occurs. This is in contrast to the controller where state transitions occur immediately with the occurrence of a plant event. The second difference is that the automaton which models the DES plant may be non-deterministic, meaning q_{i+1} is not determined exactly but rather is limited to a subset of P. The reason for these differences is that the DES plant is a simplification of a continuous-time plant and an interface. This simplification results in a loss of information about the internal dynamics, leading to non-deterministic behavior.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

The set of states, P, of the DES plant is based on the set of equivalence classes of α. Specifically, each state in P corresponds to a region, in the state space of the continuous-time plant, which is equivalent under α. Thus there is a one-to-one correspondence between the set of states, P, and the set of plant events, E. This can be used to develop an expression for the state transition function, ψ. Based on this, conditions for determinism in the DES plant are derived.

Partitioning

A particular problem in the design of a hybrid control system is the selection of the function α, which partitions the state-space of the plant into various regions. Since this partition is used to generate the plant events, it must be chosen to provide sufficient information to the controller to allow control without being so fine that it leads to an unmanageably complex system or simply degenerates the system into an essentially conventional control system.

The partition must accomplish two goals. First it must give the controller sufficient information to determine whether or not the current state is in an acceptable region. For example, in an aircraft these regions may correspond to climbing, diving, turning right, etc. Second, the partition must provide enough additional information about the state, to enable the controller to drive the plant to an acceptable region. In an aircraft, for instance, the input required to cause the plane to climb may vary depending on the current state of the plane. So to summarize, the partition must be detailed enough to answer: 1) is the current state acceptable; and 2) which input can be applied to drive the state to an acceptable region.

To design a partition, we can start by designing a primary partition to meet the first goal mentioned above. This primary partition will identify all the desired operating regions of the plant state space, so its design will be dictated by the control goals. The final partition will represent a refinement of the primary partition which enables the controller to regulate the plant to any of the desired operating regions, thus meeting the second goal.

An obvious choice for the final partition is one which makes the DES plant deterministic and therefore guarantees that the controller will have full information about the behavior of the plant. In addition to being very hard to meet, this requirement is overly strict because the controller only needs to regulate the plant to the regions in the primary partition, not the final partition. For this reason we define quasideterminism, a weaker form of determinism. In the DES plant, the states which are in the same region of the primary partition can be grouped together, and if the DES plant is deterministic with respect to these groups, then we say it is quasideterministic. So if the DES plant is quasideterministic, then we may not be able to predict the next state exactly, but we will be able to predict its region of the primary partition and thus whether or not it is acceptable.

This approach is illustrated in [11] by a number of examples.

Reconfigurable Control and FDI

Reconfigurable control

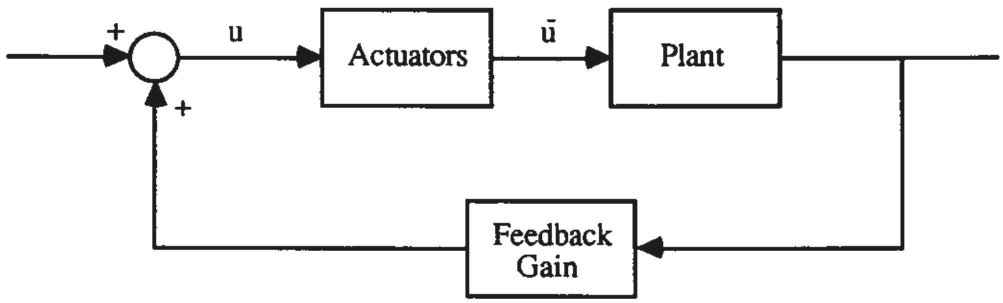
The ability to perform control reconfiguration on line is an important aspect of autonomous control systems. What follows is a synopsis of the approach used in [12] to study reconfigurable control:

In control reconfiguration it is desirable to automatically adjust feedback compensators to guarantee stability and to recover performance. Here, control reconfiguration when only incomplete failure information is available is addressed.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he
T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel,
November 2-3, 1992.

Consider the nominal plant described by $\dot{x} = Ax + Bu \quad y = Cx$ and $u = Kx$ or by $\dot{x} = (A+BK)x \quad y = Cx$. The impaired plant is described by $\dot{x} = A_f x + B_f u_f + Dw \quad y = C_f x$.

Typically it is assumed that $A_f = A$ and that B_f is known exactly. In the pseudoinverse method $A+BK$ is set equal to $A+B_f K_f$ and a solution K_f is found using the pseudoinverse of B_f; note that references on the pseudoinverse method can be found in [12]. Here the control law must be adjusted before the full report of failure is available. To illustrate, consider the problem of actuator degrading



where $\bar{u}_i = u_i(1+\delta)$ with $\delta \in [\alpha, \beta]$. Now the impaired system is $\dot{x} = Ax + B_f u \quad y = Cx$ where

$$B_f = [b_1 \ldots b_{i-1} \quad b_i(1+\delta) \quad b_{i+1} \ldots b_m]$$

with $u = K_f x$ to be determined

If

$$K_f = \begin{bmatrix} k_{f1} \\ \ldots \\ k_{fi} \\ \ldots \\ k_{fm} \end{bmatrix}$$

then

$$A + B_f K_f = (A+BK_f) + \delta(b_i k_{fi}) = A_c + \delta E$$

The problem therefore can be addressed via a two-step approach: First a stability analysis is performed off line to find the range of δ such that $A+B_f K$ is stable. If there is indication that the actuator is at fault, then the gain is adjusted on line as follows: $k_{fj} = k_j$ for $j \neq i$ and $k_{fi} = k_i/(1+\bar{\delta})$ for $j=i$ with the fault being with the actuator of the ith input; $\bar{\delta}$ represents a value, such as the middle of the range of variations for which the reconfigured system is stable; see [12] for details.

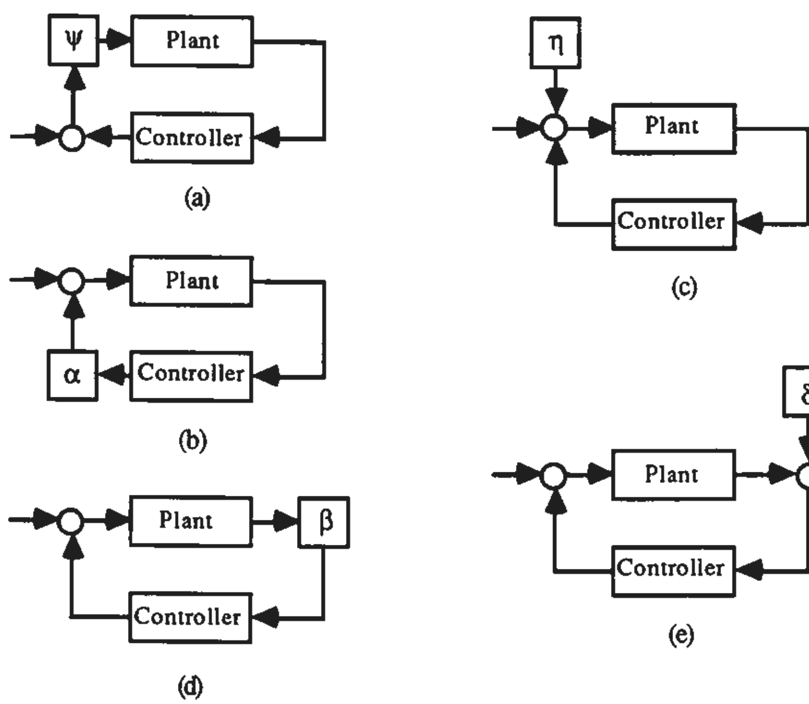
FDI

An essential ingredient of the ability to perform control reconfiguration is the ability to perform Failure Diagnosis and Identification. What follows is a synopsis of the approach used in [13] and

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

it involves neural networks. For illustration the approach is used to detect faults in certain sensors and actuators of the JPL's experimental space antenna.

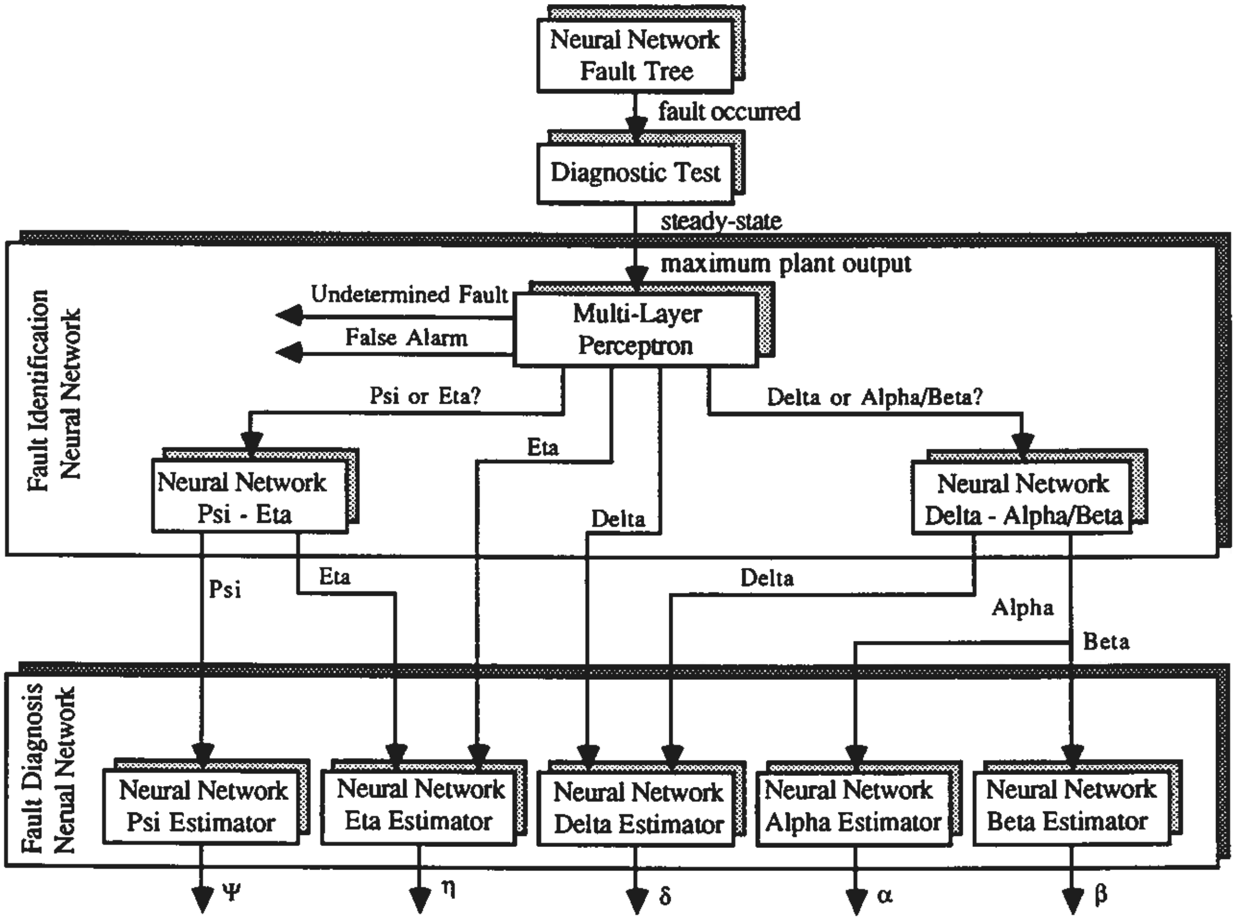
The placement of the five faults considered is illustrated in figure below. For an external input that is implemented by either software or an electrical signal, the multiplicative fault ψ is an actuator degradation fault and is in the range $0 \leq \psi \leq 1$. For an external input that is implemented by a force, the multiplicative fault α is an actuator degradation fault and is in the range $0 \leq \alpha \leq 1$. The actuator fault η is an added bias fault and is limited to $-10 \leq \eta \leq 8$, since beyond these values the system becomes unstable. The multiplicative fault β is a sensor degradation fault and is in the range $0 \leq \beta \leq 1$. The sensor fault δ is an added bias fault and is limited to $-0.01 \leq \delta \leq 0.01$, which produces values for the diagnostic test that are in the range of those produced by the other faults.



(a) Actuator degradation fault. (b) Actuator degradation fault.
(c) Actuator added bias fault. (d) Sensor degradation fault. (e) Sensor added bias fault

The overall system consists of three levels. The top detects if any fault has occured and if it has it instructs the system to apply a test signal, here a step to obtain more information. The middle level identifies the actuator or sensor which is at fault, while the bottom level estimates the exact value of the fault.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.



For the JPL space antenna, the diagnostic test consists of recording the steady-state and the maximum plant output of the plant's response to a reference step input (with the controller on), where the maximum plant output is defined here as the largest plant output, in magnitude, recorded with its phase. The values forming the five fault trajectories were plotted. Examining the various fault trajectories, all diverge from the nominal behavior of the antenna with the faults α and β following the same trajectory. The various trajectories were separated from one another with straight lines.

With the data from the α, β, and δ fault region, the quadratic optimization algorithm (developed by the authors) is used to train a two-layer neural network with 15 hidden layer neurons to separate the α and β faults from the δ fault. Testing the results, the trained neural network is given random inputs inside the fault region, and its outputs are plotted. The neural network classifies approximately half of the region as an α or β fault and classifies the other half as a δ fault.

Using the data from the ψ and η fault region, a second two-layer neural network with 18 hidden layer neurons is trained with the quadratic optimization algorithm to distinguish between the two faults. Testing the results, the trained neural network is given random inputs inside the fault region, and its outputs are plotted. The neural network classifies approximately half of the region as a ψ fault and classifies the other half as an η fault.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

Neural networks trained using the back-propagation algorithm are used as function approximators to estimate the values of the faults. The results from the diagnostic tests for the various faults are used to train the neural networks.

Due to the results used from the reference step input diagnostic test, the distinction between the two faults α and β is not possible in one step via the developed procedure since both trajectories are the same. To distinguish between the two, the following procedure can be used. When the neural network identification system determines that either α or β occurred, this information is sent to the controller. The controller first compensates for the α fault and performs another diagnostic test. If the test is within tolerance, it is assumed that an actuator degradation fault occurred. If the test is not within tolerance, it is assumed that a sensor degradation fault occurred, and the controller compensates for this fault instead of an actuator degradation fault.

The diagnosis of the identification system, which includes the fault and estimate of the fault, is sent to the controller, or another higher level system that over sees the operation of the FDI system. The controller uses this information to correct the fault. For instance, if the FDI system reported that the actuator fault ψ occurred and was 0.47, the controller would multiply its output by 1/0.47 to compensate for the fault; see [13] for further details.

Neural Networks in Control

Neural networks can be used in an intelligent autonomous controller to provide a variety of necessary functions. What follows is a synopsis of [14] which discusses a number of uses of neural networks in control systems:

In general there are potential applications of neural networks at all levels of hierarchical intelligent controllers that provide higher degree of autonomy to systems. Neural networks are useful at the lowest Execution level where the conventional control algorithms are implemented via hardware and software, through the Coordination level, to the highest Organization level, where decisions are being made based on possibly uncertain and/or incomplete information. One may point out that at the Execution level, the conventional control level, neural network properties such as the ability for function approximation and the potential for parallel implementation appear to be most relevant. In contrast, at higher levels, abilities such as pattern classification and the ability to store information in a say associative memory appear to be of most interest.

In [14], the different approaches used in the modeling of dynamical systems are discussed. The function approximation properties of multilayer neural networks are discussed at length, radial basis networks and the Cerebellar Model Articulation Controller (CMAC) are introduced and the modeling of the inverse dynamics of the plant, used in certain control methods is also discussed. The use of neural networks as controllers in problems which can be solved by supervised learning are discussed; such control problems for example would be following a given trajectory while minimizing some output error. Control problems which involve minimization over time are of interest; an example would be minimizing the control energy to reach a goal state-there is not known desired trajectory in this case. Methods such as back propagation through time and adaptive critic with reinforcement learning are briefly discussed. Other uses of neural networks in the failure detection and identification area (FDI), and in higher level control are also discussed.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

Concluding Remarks

Conventional control methods need to be enhanced to adapt to significant changes in the plant, environment and objectives. The design goal is control systems with higher degree of autonomy. It is stressed that autonomy is the design requirement and intelligent methods appear to offer some of the necessary tools to achieve higher degrees of autonomy. The research area of intelligent autonomous systems is a research area in its own right. It uses methods from a variety of areas but it modifies and extends them to address the particular problems of interest. There is need to answer questions and resolve novel problems in Planning and Expert Systems, in Learning and Neural Control, in Discrete Event Dynamical and Hybrid Systems, in Reconfigurable Control Systems and FDI Systems to mention a few. There is great need for quantitative methods and mathematical rigor in the area. There is need for systematically generating less detailed, more abstract models.

Finally two important topics are briefly discussed. They are: the issue of how to deal with anticipated and non-anticipated changes in the plant, environment and in the control goals, and the issue of lessons learned or good rules to follow in the quest for autonomy.

Reconfiguration in response to anticipated and unanticipated changes

In a space platform, for example, the actual plant may behave differently than predicted due to inadequacies in the model or changes in the plant. Model inadequacies may have occured because of inability to obtain a good model of a large flexible structure by conducting tests on the ground. Changes occur because of component and structural failures, because of evolution of the plant, and of degradation due to aging. To deal with these changes it is useful to classify them as anticipated or non anticipated.

Anticipated changes can perhaps be dealt with by having a number of appropriate models and controllers the system switches to or interpolates among when it is instructed to do so. That is, when certain conditions exist, and this has to be sensed, appropriate action is taken chosen from a list stored on board. This approach can be very useful when quick response to drastic sudden changes, failures, is needed. Conventional control methods, neural networks or other techniques may be used to derive the appropriate controllers that may be stored for example in expert systems or associate memories.

Certain changes may be considered as unanticipated because of finite memory capacity which makes it difficult to have a preprogrammed response to all changes we can think of. If a change is destabilizing and a quick reaction is of essence then it should be classified as anticipated and be dealt with via prestored response. This was the case for example in our approach discussed above as to how to deal with reconfiguration when only incomplete information.is available. For most of the unanticipated changes, the failure detection (FDI) system, or some change detection system (CDI),must provide the information needed for appropriate reconfiguration to be possible. Neural networks may be used to identify the new model (perhaps being trained as an add on, in parallel with, to an existing model) or some other inferential techniques; see above.

In summary, a combination of strategies to deal with anticipated and unanticipated changes seems to be appropriate. Fast reaction to potentially catastrophic changes is essential. As more information concerning the location and exact nature of the change comes in then the initial control action is refined to not only ensure stability but also reestablish certain level of performance to the system. A reliable way to detect and identify changes is essential to the well being of the system and its being able to perform as expected. Actuators therefore specifically designed to assist in the identification of the new plant model and conditions may be appropriate in autonomous systems.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel, November 2-3, 1992.

Lessons Learned - Rules to follow

Good rules to follow in the analysis and design of high autonomy control systems come from engineering intuition, common sense and past experience. Although they mainly represent the opinion of the author, it is felt that they are shared by many.

It is important to systematically incorporate as much existing prior knowledge as possible in the design or redesign of control systems. This has been repeatedly stressed in the above discussions. There are cases where scientists and engineers have spent considerable time and effort to derive models of complex systems from first principles and improve upon them incorporating operating experience. This knowledge captured in the existing model should be used as much as possible; this is the reason for example in neural modeling it was suggested to use the neural nets to capture only the differences between the behavior of the known model and the actual system. This provides significant savings in training speed as it reduces the computational complexity.

Computational complexity is a major issue as the systems studied are typically extremely complex. Reduced computational complexity may mean that the design can be implement. Without attempting to address the computational complexity issue it is impossible to achieve the levels of autonomy envisioned. Systematically deriving more abstract models depending on their use so that only the necessary information is dealt with is essential; it is as essential or more than designing faster computers. Some of these ideas were addressed when discussing Hybrid systems.

Improving existing control systems by adding on new features is a plausible approach having high chances for success. This bottom-up approach builds upon experience and uses existing knowledge. It is also easier to justify in applications where system failure is costly in human and material sense.

Feedback in AI planners is an essential ingredient. Using system and control theory ideas in the study of planners may lead to high performance planing systems needed to attain high autonomy. Some initial work along these lines was mentioned above.

Incorporating sensors and actuators specifically to identify changes and reconfigure the control laws may be necessary in high autonomy systems.

Technological breakthroughs are making large numbers of distributed sensors and actuators possible. This will certainly make reconfiguration and higher autonomy more common place. Areas such as sensor data fusion are becoming more important to deal with the mass of data. And methods to extract only the necessary information from the data, which is related to the above mentioned problem of extracting more abstract models, are becoming essential in the quest for higher autonomy.

References

[1] An Introduction to Intelligent and Autonomous Control, P.J.Antsaklis and K.M.Passino, Editors, Kluwer Academic Publishers, October 1992.
[2] P.J.Antsaklis and K.M.Passino, "Introduction to Intelligent Control Systems with High Degree of Autonomy", An Introduction to Intelligent and Autonomous Control, P.J.Antsaklis and K.M.Passino, Eds., Chapter 1, Kluwer,1992.
[3] Passino K.M., Antsaklis P.J., "Event Rates and Aggregation in Hierarchical Discrete Event Systems", *Journal of Discrete Event Dynamic Systems: Theory and Applications*, Vol. 1, No. 3, pp. 271-287, 1992.

P. J. Antsaklis, "On Modeling, Analysis and Design of High Autonomy Control Systems," P roc o f t he
T he 1 992 W orkshop o n I ntelligent A utonomous C ontrol S ystem s , pp. 128-147, Leshem, Israel,
November 2-3, 1992.

[4] Passino K. M. and Antsaklis P. J., "A System and Control Theoretic Perspective on Artificial Intelligence Planning Systems", *Journal of Applied Artificial Intelligence*, Vol. 3, pp. 1-32, 1989.

[5] Passino K.M., *Analysis and Synthesis of Discrete Event Regulator Systems*, Ph.D. Dissertation, Dept. of Electrical and Computer Eng., Univ. of Notre Dame, April 1989.

[6] Passino K. M. and Antsaklis P. J., "Fault Detection and Identification in an Intelligent Restructurable Controller", *Journal of Intelligent and Robotic Systems*, Vol. 1, pp.145-161, June 1988.

[7] Peek M.D. and Antsaklis P.J., "Parameter Learning for Performance Adaptation", *IEEE Control Systems Magazine*, Vol. 10, pp 3-11, December 1990.

[8] Antsaklis P.J., Special Issues on 'Neural Networks in Control Systems' of the *IEEE Control Systems Magazine*, April 1990 and April 1992.

[9] Gao Z. and Antsaklis P. J., "On the Stability of the Pseudo-Inverse Method for Reconfigurable Control Systems", *Intern. Journal of Control*, Vol 53, No 3, pp 717-729, 1991.

[10] Passino K.M., Sartori M.A. and Antsaklis P.J, "Neural Computing for Numeric to Symbolic Conversion in Control Systems", *IEEE Control Systems Magazine*, pp. 44-52, April 1989.

[11] Stiver J. M. and Antsaklis P. J. , "Modeling and Analysis of Hybrid Control Systems", *Proc. of the 31st Conference on Decision and Control* , Tucson, AZ, December 1992.

[12] Gao Z. and Antsaklis P. J., "Actuator and Sensor Failure Accommodation with Incomplete Information", Proceedings of the 1st IEEE Conference on Control Applications, pp 890-892, Dayton, OH, September 13-16, 1992.

[13] Sartori M. A. and Antsaklis P.J., "Failure Behavior Identification for a Space Antenna via Neural Networks", Proc of American Control Conference, pp.567-568, Chicago, IL, June 24-26, 1992.

[14] Antsaklis P. J. "Neural Networks for the Intelligent Control of High Autonomy Systems", Intelligent Systems Technical Report # 92-09-1, Dept of Elec Engr, Univ of Notre Dame, September 1992. Also in Mathematical Studies of Neural Networks, J.G. Taylor, Ed., Elsevier, 1992.