1

Introduction to Intelligent Control Systems with High Degrees of Autonomy

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

Kevin M. Passino
Dept. of Electrical Engineering
The Ohio State University
2015 Neil Ave.
Columbus, OH 43210

Abstract

Intelligent control systems with high degrees 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. Here, we provide an introduction to the area of intelligent autonomous control. The fundamental issues in autonomous control system modeling and analysis are discussed, with emphasis on mathematical modeling. Some results and directions in relevant research areas are outlined.

1. INTRODUCTION

Autonomous means having the power for self government. Control systems with *high degrees of autonomy* should have the power and ability for self governance in the performance of control functions. They are composed of a collection of hardware and software, which can perform the necessary control functions, without external intervention, over extended time periods. There are several *degrees of autonomy.* A fully autonomous controller should perhaps have the ability to even perform hardware repair, if one of its components fails. Note that conventional fixed controllers can be considered to have a *low degree* of autonomy, since they can only tolerate a restricted class of plant parameter variations and disturbances; conventional adaptive controllers have higher degree of autonomy. To achieve significantly *higher degrees* of autonomy, the controller must be able to perform a number of functions in addition to the conventional control functions such as tracking and regulation. These additional functions, which include the ability to accommodate for system failures, are discussed in this paper. This paper is based on the developments in [1,2].

Controllers with a high degree of autonomy can be used in a variety of systems from manufacturing to unmanned space, atmospheric, ground, and underwater exploratory vehicles (for a description of several applications see [3]). This introduction to autonomous control will be developed around a space vehicle application so that concrete examples for the various control functions, and fundamental characteristics of autonomous control can be given, and so that the development addresses relatively well defined control needs rather than abstract requirements. Furthermore, the autonomous control of space vehicles is highly demanding; consequently the developed architecture is general enough to encompass all related autonomy issues. It should be stressed that all the ideas presented here apply to most autonomous control systems. In other classes of applications, the architecture, or parts of it, can be used directly and the same fundamental concepts and characteristics identified here are valid.

We begin by describing a hierarchical functional controller architecture for high autonomy systems necessary for the operation of future advanced space vehicles. The concepts and methods needed to successfully design such a controller are introduced and discussed. The control system is designed to ensure highly autonomous operation of the control functions and it allows interaction with the pilot/ground station and the systems on board the autonomous vehicle. A command by the pilot or the ground station is executed by dividing it into appropriate subtasks which are then performed by the controller. The controller can deal with unexpected situations, new control tasks, and failures within limits. To achieve this, high level decision making techniques for reasoning under uncertainty and taking actions must be utilized. These techniques, if used by humans, are attributed to *intelligent* behavior. Hence, one way to achieve autonomy, for some applications, is to utilize high level decision making techniques, intelligent methods, in the autonomous controller. It should be kept in mind therefore that:

Autonomy is the objective, and intelligent controllers are one way to achieve it.

The fields of Artificial Intelligence (AI) and Operations Research offer some of the tools to add the higher level decision making abilities.

Autonomous Control Functions: High autonomy control systems must perform well under significant uncertainties in the plant and the environment for extended periods of time and they must be able to compensate for system failures without external intervention. Such autonomous behavior is a very desirable characteristic of advanced systems. A highly autonomous control system provides high level *adaptation* to changes in the plant, environment and control objectives. To achieve autonomy the methods used for control system design should utilize both *algorithmic-numeric methods*, based on the state of the art conventional control, identification, estimation, and communication theory, developed for continuous-state type systems, and *decision making-symbolic methods,* such as the ones developed in computer science (e.g., automata theory) and specifically in the field of Artificial Intelligence (AI) for discrete-state systems. In addition to supervising and tuning the control algorithms, the autonomous controller must also provide a high degree of tolerance to failures. To ensure system reliability, failures must first be detected, isolated, and identified (and if possible *contained*), and subsequently a new control law must be designed if it is deemed necessary.

The autonomous controller must be capable of planning the necessary sequence of control actions to accomplish a complicated task. It must be able to interface to other systems as well as with the operator, and it may need learning capabilities to enhance its performance while in operation. It is for these reasons that advanced planning, learning, and expert systems, among others, must work together with conventional control systems in order to achieve high degrees of autonomy.

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. Also it is important to note that autonomous controllers are *evolutionary* and not *revolutionary*. They evolve from existing controllers in a natural way fueled by actual needs, as it is now discussed.

Design Methodology - Evolution Conventional control systems are designed 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 first mathematical model to describe plant behavior for control purposes is attributed to J.C. Maxwell 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.

The control methods and the underlying mathematical theory were developed to meet the ever increasing control needs of our technology. The evolution in the control area was fueled throughout its history by three major needs:

(i) The need to deal with increasingly complex dynamical systems.
(ii) The need to accomplish increasingly demanding design requirements.
(iii) The need to attain these design requirements with less precise a priori knowledge of the plant and its environment, that is, the need to control under increased uncertainty.

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. 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. In control systems with high degrees of autonomy 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. 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.

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.

Outline: In Section 2 the functions, characteristics, and benefits of control systems with high degrees of autonomy are outlined. It is then explained that plant complexity and design requirements dictate how sophisticated a controller must be. From this it can be seen that often it is appropriate to use methods from Operations Research or Computer Science to achieve high autonomy. Such methods are studied in *intelligent control theory*. An overview of some relevant research literature in the field of intelligent and autonomous control is given together with references that outline research directions. A functional architecture for a highly autonomous intelligent control system for future space vehicles is then presented, which incorporates the concepts and characteristics described earlier. The controller is hierarchical, with three levels, the Execution Level (lowest level), the Coordination Level (middle level), and the Management and Organization Level (highest level). The general characteristics of the overall architecture, including those of the three levels are explained, and an example to illustrate their functions is given. In Section 3, fundamental issues and attributes of intelligent autonomous systems are described. Section 4 discusses several mathematical models for high autonomy systems including logical Discrete Event System models. An approach to the quantitative, systematic modeling, analysis, and design of autonomous controllers is also discussed. It is a hybrid approach since it is proposed to use both conventional analysis techniques based on difference and differential equations, together with new techniques for the analysis of systems described with a symbolic formalism such as finite automata. The more abstract, macroscopic, view of dynamical systems taken in the development of autonomous controllers, suggests the use of a model with a hybrid or nonuniform structure, which in turn requires the use of a hybrid analysis. In Section 5, several major relevant research areas are indicated. In particular, some results from the areas of Planning and Expert systems, Machine Learning, Artificial Neural Networks and the area of Restructurable Controls are briefly outlined. Finally, some concluding remarks are given in Section 6.

2. FUNCTIONAL ARCHITECTURE OF AN INTELLIGENT CONTROLLER FOR SYSTEMS WITH HIGH DEGREES OF AUTONOMY

2.1 Intelligent Control for High Autonomy Systems

Motivation: Sophistication and Complexity in Control: The complexity of a dynamical system model and the increasingly demanding closed loop system performance requirements, necessitate the use of more complex and sophisticated controllers. For example, highly nonlinear systems normally require the use of more complex controllers than low order linear ones when goals beyond stability are to be met. The increase in uncertainty, which corresponds to the decrease in how well the problem is structured or how the control problem is formulated, and the necessity to allow human intervention in control, also necessitate the use of increasingly sophisticated controllers. Controller complexity and sophistication is then directly proportional to both the complexities of the dynamical system to be controlled and to the control design requirements.

These ideas suggest a hierarchical ranking of increasing controller sophistication on the path to *intelligent* controls [4,5]. At the lowest level, deterministic feedback control based on conventional control theory is utilized for simple linear plants. As plant complexity increases, such controllers will need for instance, state estimators. When process noise is significant, Kalman or other filters may be needed. Also, if it is required to complete a control task in minimum time (or energy), optimal control techniques are utilized. When there are many quantifiable, stochastic characteristics in the plant, stochastic control theory is used. If there are significant variations of plant parameters, to the extent that linear robust control theory is inappropriate, adaptive control techniques are employed (See, e.g., the text by Astrom and Wittenmark). For still more complex plants, self-organizing or learning control may be necessary. At the highest level in their hierarchical ranking, plant complexity is so high, and performance specifications so demanding, that intelligent control techniques are used.

In the hierarchical ranking of increasingly sophisticated controllers described above, the decision to choose more sophisticated control techniques is made by studying the control problem using a controller of a certain complexity belonging to a certain class. When it is determined that the class of controllers being studied (e.g., adaptive controllers) is inadequate to meet the required objectives, a more sophisticated class of controllers (e.g. intelligent controllers) is chosen. That is, if it is found that certain higher level decision making processes are needed for the adaptive controller to meet the performance requirements, then these processes can be incorporated via the study of intelligent control theory. These intelligent autonomous controllers are the next level up in sophistication. They are *enhanced adaptive controllers,* in the sense that they can adapt to more significant global changes in the plant and its environment than conventional adaptive controllers, while meeting more stringent performance requirements.

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. Next, a number of intelligent and autonomous control research results which have appeared in the literature are outlined.

A Brief Literature Overview: In [1] the authors provided a relatively complete list of references for the field of autonomous control. Here we include some of those references together with references particularly appropriate for an introduction to the field; these are of course in addition to the excellent Chapters in this book. Note that the references included in the chapters of this book provide an excellent and comprehensive bibliography of the intelligent control area.

Hierarchical systems are treated in [6,7]. In [8] the authors explain how AI techniques will be useful in enhancing space station autonomy, capability, safety, etc. Aerospace applications are also discussed in [9]. For a book on AI and autonomous systems see [10], for one on cybernetics and intelligent systems see [11], and for one on intelligent manufacturing systems see [12].

In [1,2] the authors introduce an intelligent autonomous controller and discuss in detail the fundamental characteristics of autonomous control. In [13] the author offers a decentralized control-theoretic view on intelligent control. Functional and structural hierarchies are studied in [14] and further in Chapter 4 of this book. Fundamentals of intelligent systems such as the principle of increasing intelligence with decreasing precision, are discussed in [15], [16], and [17] (See Chapter 6). The work in [18,19,15,16,20], and [21], [22,23] probably represents the most complete mathematical approach to the analysis of intelligent machines. In [24] and the references therein the authors study distributed intelligent systems (for an introduction to this area see Chapter 5). In [25] the author introduces a theory of intelligent control that has received considerable attention since then (for a comprehensive overview of this theory see Chapter 2). There have been numerous studies on the use of Expert Systems to control various processes especially for chemical process control [26] (See also Chapter 7); expert systems have also been used extensively in failure detection and identification for processes (See Chapter 16). There are interesting relationships between the type of problems examined in intelligent autonomous control, fuzzy control [27] (See Chapter 9), and automated reasoning [28]. Simulation of high autonomy systems and modeling and architectural issues have been studied extensively in [29,30] and the references therein (See Chapter 3 for more details). Neural Networks in control is an emerging area of increasing importance in high autonomy intelligent control systems (See Chapters 9 and 10). Other key components of intelligent controllers include planning systems (See Chapter 8) and learning systems (See Chapters 9-11). In addition to the applications mentioned above there have been many applications to robotic systems (See Chapters 13 and 14) and aircraft (See Chapter

15). The collection of Chapters in this book by leaders in the field provides a comprehensive picture of the different approaches in the area.

2.2 An Intelligent High Autonomy Control System Architecture For Future Space Vehicles

Here, a functional architecture of an intelligent controller that is used to attain high degrees of autonomy in future space vehicles is introduced and discussed. This hierarchical architecture has three levels, the Execution Level, the Coordination Level, and the Management and Organization Level. The architecture exhibits certain characteristics, as discussed below, 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 Figure 2.1. 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 2.1 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.* For instance, see the architecture developed in [31]. 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 2.1 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 provides the interface between the actions of the other two levels and it uses a combination of conventional and intelligent decision making methods.

The sensors and actuators are implemented mainly with hardware. They are the connection between the physical system and the controller. 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. For example, there may be one control manager which directs a number of different adaptive control algorithms to control the flexible modes of the vehicle via appropriate sensors and actuators. Another control manager is responsible for the control functions of a robot arm for satellite repair. The control executive issues commands to the managers and coordinates their actions.

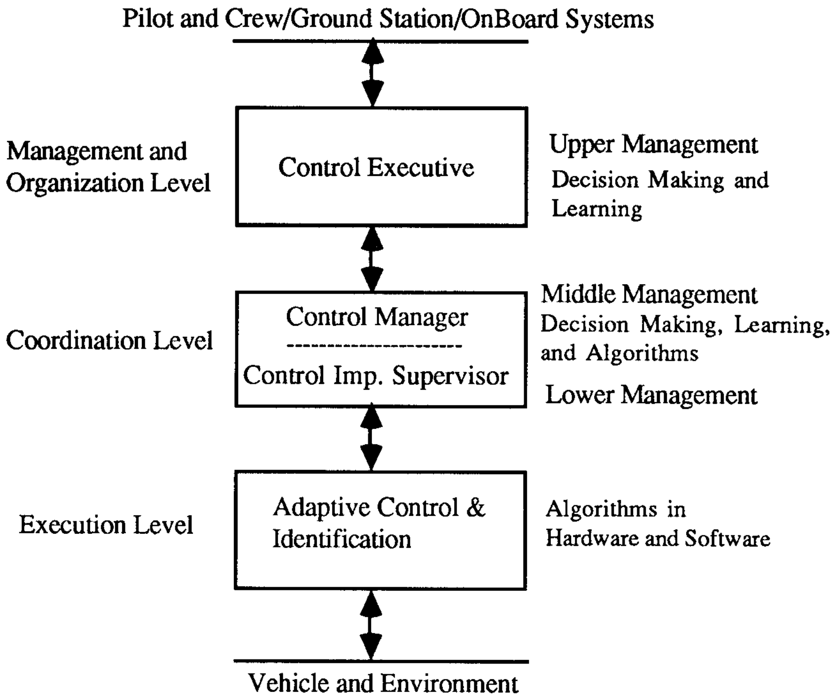

Figure 2.1. Autonomous Controller Functional Architecture

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.

Functional Operation: Figure 2.2 describes the overall architecture in more detail. Commands are issued by higher levels to lower levels and response data flows from lower levels upwards. Parameters of subsystems can be altered by systems one level above them in the hierarchy. There is a delegation and distribution of tasks from higher to lower levels and a layered distribution of decision making authority. At each level, some preprocessing occurs before information is sent to higher levels. If requested, data can be passed from the lowest subsystem to the highest, e.g., for display. All subsystems provide status and health information to higher levels. Human intervention is allowed even at the control implementation supervisor level, with the commands however passed down from the upper levels of the hierarchy.

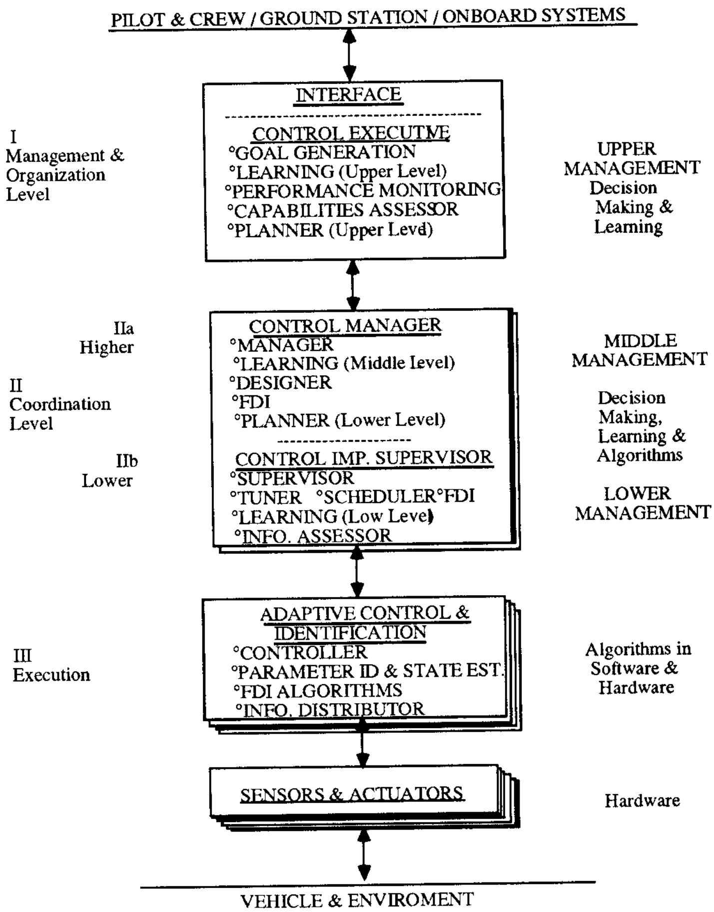

Figure 2.2. Autonomous Controller Architectural Schematic

The specific functions at each level are described in detail in later sections. Here we present a simple illustrative example to clarify the overall operation of the autonomous controller. Suppose that the pilot desires to repair a satellite. After dialogue with the control executive via the interface, the task is refined to "repair satellite using robot A". This is arrived at using the capability assessing, performance monitoring, and planning functions of the control executive. The control executive decides if the repair is possible, under the current performance level of the system, and in view of near term planned functions. The control executive, using its planning capabilities, sends a sequence of subtasks sufficient to achieve the repair to the control manager. This sequence could be to order robot A to: "go to satellite at coordinates xyz", "open repair hatch", "repair". The control manager, using its planner, divides say the first subtask, "go to satellite at coordinates xyz", into smaller subtasks: "go from start to $x_1y_1z_1$", then "maneuver around obstacle", "move to $x_2y_2z_2$",..., "arrive at the repair site and wait". The other subtasks are divided in a similar manner. This information is passed to the control implementation supervisor, which recognizes the task, and uses stored control laws to accomplish the objective. The subtask "go from start to $x_1y_1z_1$", can for example, be implemented using stored control algorithms to first, proceed forward 10 meters, to the right 15 degrees, etc. These control algorithms are executed in the controller at the Execution Level utilizing sensor information; the control actions are implemented via the actuators.

It is important at this point to discuss the *dexterity* of the controller. The Execution Level of a highly dexterous controller is very sophisticated and it can accomplish complex control tasks. The implementation supervisor can issue commands to the controller such as "move 15 centimeters to the right", and "grip standard, fixed dimension cylinder", in a dexterous controller, or it can completely dictate each mode of each joint (in a manipulator) "move joint 1, 15 degrees", then "move joint 5, 3 degrees", etc. 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 autonomous controller 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.

The Execution Level (III) The functional architecture for the Execution Level of the autonomous controller is shown in Figure 2.3 below. Its main function is to generate, via the use of numeric algorithms, low level control actions as dictated by the higher levels of the controller, and apply them to the vehicle. It senses the responses of the vehicle and environment, processes them to identify parameters, estimates states, or detects vehicle failures, and passes this information to the higher levels.

The **Sensor** and **Actuator** subsystems are depicted in Figure 2.3. These devices which physically accomplish the functions for the autonomous controller are at the lowest level of the architecture. The complexity of these devices depends on the dexterity of the controller. All sensors which provide information from the vehicle and environment to any component in the autonomous controller are included here. On the Execution Level, the controller will need feedback information about control variables. The state estimator and parameter identifier also use such outputs for their respective tasks. The Failure Detection and Identification (FDI) algorithms need these outputs and those of special failure sensors to enable them to detect failures. To perform "execution monitoring" for the planning systems at the higher levels the dynamical response of the system must be sensed and passed to the planning system so that it can determine if a plan has failed. The implementation supervisor also needs sensor information so that it can, for instance, make the smooth transition in the implementation of a newly designed control law. Sensory information is also used in performance monitoring, capabilities assessing, tuning, scheduling, and display to the pilot, crew, ground station, or other onboard systems. The actuators are the usual control actuators (transducers) which translate the outputs of the controller to actions meaningful to the vehicle. For a highly dexterous controller, a whole manipulator may be considered to be an "actuator".

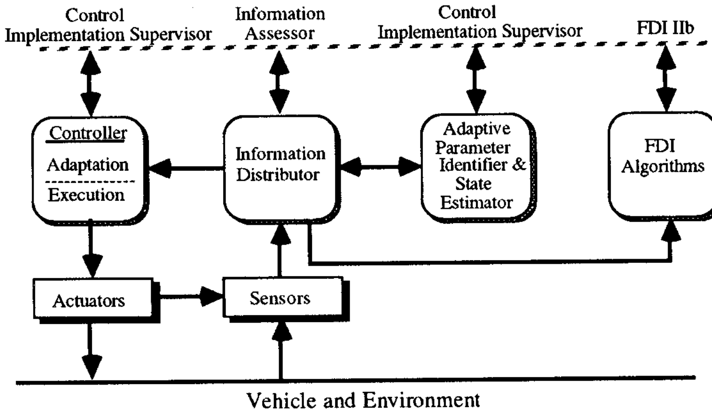

Figure 2.3. Execution Level

The main function of the **Controller** in Figure 2.3 is to execute the control algorithms and to issue commands to the actuators. It performs advanced conventional adaptive control functions. It receives, in real time, all the necessary data (from the information distributor) to execute the current control algorithm. The information consists of current output values from the sensors, model parameter estimates and state estimates, as they are generated from the identifier. The *adaptation* part of the controller algorithmically interprets the values of the measured plant variables and the estimated plant parameters and states; and it adjusts, on-line, the coefficients of the control law which runs in the *execution* part of the controller. These functions correspond to conventional adaptive control. The adaptation algorithm can contain information about the model to be followed, thus implementing "model reference" adaptive control. Since the model parameters are explicitly estimated and then used in the control law adaptation, the structure appears to suggest an "indirect" adaptive control approach. However, notice that this is not necessarily the case since the model parameter estimates from the identifier can simply be ignored and the adaptation algorithm can directly process the information from sensors to directly estimate the control law coefficients, thus implementing "direct" adaptive control. If a fixed control law is used, then the appropriate sensor data are simply fed back to the control law which is being executed. The sensor data are values of measured variables (e.g., states).

All possible conventional control functions can be performed via the proposed architecture. For fixed control laws, one could envision a loop containing the sensors providing feedback information, through the information distributor, to the controller; the control actions are performed via the actuators. For adaptive control this also involves the model parameter identifier. In addition to advanced adaptive control functions, the controller has the following capabilities: The controller allows intervention from above. It of course allows the introduction of reference signals for set points and tracking as conventional controllers do. In addition it receives commands: (i) To alter the parameters of adaptation (as determined by the tuner in the coordination level), and (ii) To switch to different control laws altogether suggested by the scheduler or the control redesigner.

If the higher levels of the architecture are ignored, the intervention to the controller can be envisioned as being that of a human operator who adjusts certain parameters depending on performance, sets the set points, switches to different algorithms from time to time or to new control laws when failures occur. It returns status information to the higher level, such as what particular control law is currently running; also information about the health of the system (errors in implementation, etc.). The controller has access to a variety of stored control laws. The particular location of the stored programs is not important in this functional architecture. They could be located in the controller, or in the level above (implementation supervisor). If they are located above, then one should allow for down loading these programs. Since control law switching is desirable, transition programs, for smooth control law switching are necessary. When the scheduler and control redesigner send new control laws to be implemented they should also attach a program to ensure the smooth transition from the current to the new control law.

The main function of the **Information Distributor** shown in Figure 2.3, is to distribute sensor, parameter , and state information where it is needed. Since the control models and therefore the control, identification, estimation, and FDI algorithms do change, it is essential to guarantee that the Execution Level subsystems receive each time the correct information. Information about the current control models and current algorithms is provided from above. Using stored information, the distributor provides the correct sensor information to the controller for control feedback purposes, to the identifier for model parameter identification and state estimation, and to the FDI for detection and isolation. After perhaps some preprocessing, it also provides this information to higher levels.

The main function of the **Adaptive Parameter Identifier and State Estimator** shown in Figure 2.3, is to execute parameter identification algorithms and state estimation algorithms, and to continuously pass this information to the controller, to the FDI algorithms, and to higher levels. It receives all appropriate sensor information from the information distributor. The parameters and the states, the estimates of which are sought, depend on the particular control model used. Since the control model and the control law do change, the parameter identifier and state estimator should be able to switch control models and identification and estimation algorithms. This information is given from above. It provides the necessary parameter and state estimates to the controller and to the FDI algorithms via the information distributor. It returns to the higher level, parameter estimates and state estimates of the current model (via the information distributor) and information as to the status and health of the system directly.

The main function of the **FDI Algorithms** shown in Figure 2.3, is to execute FDI algorithms for failures detected at the execution level of the autonomous controller. It receives all appropriate sensor information via the information distributor. This includes information from sensors specifically located to detect failures at the actuator level of the control system; it also includes model parameter and state estimates from the identifier. It has the ability to switch algorithms and plant models. The FDI algorithms return information to the higher level FDI subsystems.

Coordination Level (IIb) The functional architecture for Coordination Level IIb is shown in Figure 2.4. Coordination Level IIb receives commands to perform predetermined specific control tasks from the control manager in the level above. It provides the appropriate sequence of control and identification algorithms to the Execution Level below. Its ability to deal with extensive uncertainties is limited.

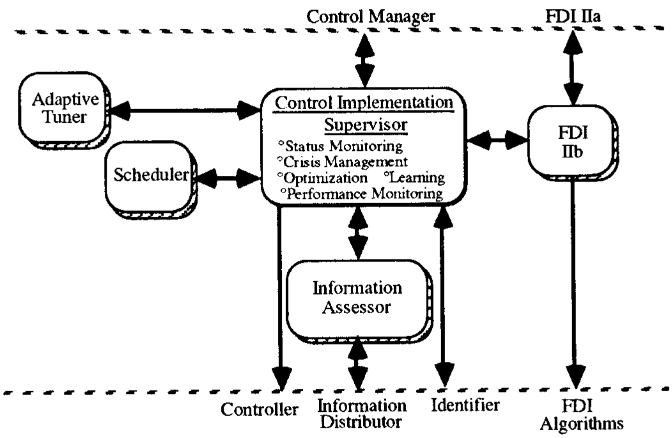

Figure 2.4. Coordination Level IIb

The main function of the **Control Implementation Supervisor** shown in Figure 2.4, is to carry out the sequence of control actions dictated by the control manager. It can accomplish predetermined control actions and cope with limited predetermined crisis situations. The supervisor receives the sequence of control tasks to be accomplished from the control manager and it has access to a variety of control models, and control, identification and estimation algorithms. It selects appropriate reference signals for the controller and it *optimizes* the subsequences of actions to accomplish the tasks dictated by the above levels in the best way possible. The supervisor uses the scheduler to decide what models and algorithms to use in the controller and identifier; it uses the tuner to decide how to adapt parameters in the algorithms, which are currently used, and it sends this information to the execution level. It monitors the *status* of the system at IIb and III, i.e., what algorithms and models are currently used, and the *health* of the systems. The supervisor does *performance monitoring* on IIb and III levels using information provided by the information assessor and FDI IIb. It contains a *crisis management* facility to deal with certain failures. This includes a number of methods to maintain performance or to maintain a certain degree of safety in operations, while degrading performance gracefully. For example, if a failure in an actuator or sensor is detected, it can switch to an alternative control method using other actuators or sensors to maintain performance. If performance cannot be maintained, it should degrade gracefully, guaranteeing safety (stability). It will take the necessary steps to maintain stability after a failure is detected and it is isolated and identified. The control implementation supervisor uses *learning* to improve the implementation of the (predetermined) control forms. It thus improves the speed and accuracy of tuning with experience, it improves its crisis management and the scheduling of algorithms, and it learns how to more efficiently optimize the overall operations, as a good human supervisor would do; it also learns completely new control methods sent from the level above. It informs the control manager about the health of the system in levels IIb and III, about its status (the progress in performing the tasks) and it notifies the manager if failures, and unexplained (at that level) performance degradation is occurring.

The main function of the **Scheduler** shown in Figure 2.4, is to determine, during the performance of a specific control function, if certain conditions are met in order to switch to alternative control laws (and plant models) and to appropriate identification, estimation and FDI algorithms. It receives information from the implementation supervisor as to control function to be performed, together with information about the plant models and their validity range, the corresponding control laws, and the rest of the algorithms. Based on information it receives from the supervisor, it decides when to switch to the proper algorithms and models. The criteria for switching are predetermined, in perhaps tabular form, and they also depend on information from environmental sensors. This information is transmitted from the higher level through the supervisor. Examples will be the schedulers used for docking control. Depending, for example, on approaching speed and attitude, an appropriate new control law is selected. Here, the scheduler also selects appropriate parameter identification, state estimation, and FDI algorithms; it also selects the corresponding plant model when necessary. The scheduler does not deal with crisis situations.

The main function of the **Adaptive Tuner** shown in Figure 2.4 is to determine, during the execution of particular algorithms, if specific conditions are met in order to adjust, tune, certain parameters in the adaptation laws. It receives information from the implementation supervisor as to the current algorithm being executed, control and identification algorithms, and also information from the information assessor (via the supervisor) necessary to decide first if timing is appropriate. Then based on predetermined criteria, it selects the new values for the parameters in the adaptation laws. The criteria for tuning will be based on excessive output, state, and parameter errors, and the selection of the new adaptive parameter values will depend on algorithms or heuristic rules using performance measures and actual past and present inputs and outputs. In this way parameter tuning of identification and control algorithms (adaptive, robust, optimal) is accomplished.

The main function of the **Information Assessor** shown in Figure 2.4 is to process and distribute sensor, state and parameter information to the information distributor (execution level) and the implementation supervisor. It receives information from the supervisor as to the current plant model, control, estimation and identification, and FDI algorithms, and it instructs the information distributor to pass the necessary sensor information to the controller, identifier, and FDI systems. It receives, from the identifier via the distributor, information about the current model parameter and state estimates. After instruction from the supervisor it supplies to the supervisor processed information such as errors in state and parameter estimation. To do this, it uses sensor data supplied by the distributor and models supplied by the supervisor. This processed information is used by the tuner, the scheduler, and the control implementation supervisor for performance monitoring.

The main function of the **FDI IIb** subsystem shown in Figure 2.4 is to supervise the FDI algorithms (execution level) and to detect and identify, using algorithms and heuristic methods, failures that occurred at the execution level. It passes the information about the current models used from the supervisor to the FDI algorithms. It sends appropriate FDI algorithms to be executed to the lower level. It receives the outputs of those algorithms. It compares them with additional information from the supervisor, and it proceeds, after detecting a failure, to isolate and identify it. It informs the FDI IIa subsystem about the status of the failure and it also informs the supervisor so that predetermined crisis measures can be taken if necessary and possible. If the crisis cannot be dealt with at that level, the information is passed to the FDI IIa, and the designer via the control manager.

Coordination Level (IIa) The functional architecture for Coordination Level IIa is shown in Figure 2.5. Coordination Level IIa receives commands from the management level which it must determine how to perform using the designer and planner and considering information from FDI IIa and the control implementation supervisor.

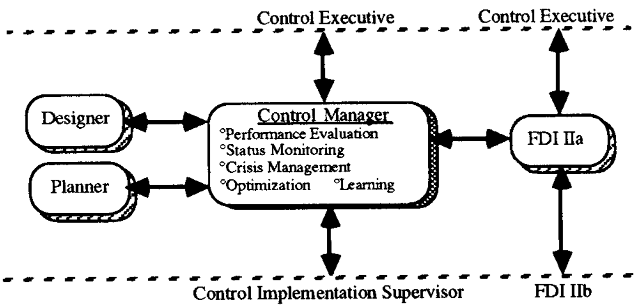

Figure 2.5. Coordination Level IIa

It generates a sequence of control actions that the control implementation supervisor can recognize and passes them to it. This coordination level has abilities to deal with significant uncertainties.

The main function of the **Control Manager** shown in Figure 2.5 is to accomplish the control tasks given by the control executive. It can accomplish predetermined control actions, using the lower levels, but also it can cope with failures to a large degree. In general it is equipped to successfully carry out the control tasks under a wide variety of unanticipated vehicle and environmental conditions. It can also be directed to prepare for future requirements, building new control laws and contingency methods using the designer. When a control task is given, it breaks it down into a sequence of control actions, using the planner, and it passes it to the implementation supervisor. It receives processed sensor information from the supervisor about the current positions and information from above about the goals so that it can plan its actions. It also passes to the supervisor newly designed algorithms and contingency plans. It receives, from the implementation supervisor, *status and health* information and it passes its status and health information to the executive. It does *performance evaluation and monitoring* on II and III levels. For example, it evaluates the performance of a sequence of tasks so that, changes in the next sequence are implemented if necessary. It also contains a *crisis management facility* to deal with failures. It is similar to the one in the implementation supervisor but it deals with higher level contingencies. It has significant *learning abilities*, to improve its performance. It does *optimization* of system below it using the planned actions, and it suggests new strategies in algorithm selection to the implementation supervisor.

The main function of the **Designer** shown in Figure 2.5 is to develop methodologies to deal with novel situations for which no prior designs have been made. These include detected failures via the FDI system, in which case new control laws must be designed on-line, using the models and specifications provided by the control manager. They also include dealing with new control tasks suggested by the manager or the higher level. When there are no requirements to develop new methods in real time, the designer, under direction of the manager, works on developing new methods to build up the crisis management algorithm inventory, and the inventory of algorithms to deal with new control tasks needed some time in the future. These algorithms are passed to the control implementation supervisor at the direction of the control manager. In this way, the system is enriched and improved, it becomes more experienced, as it can deal with a greater number of contingencies and tasks. The designer uses decision making under uncertainty (possibly symbolic based methods) to select design algorithms. When the designer must react in a very short time to deal with, say a major failure, it may decide to initially suggest a method which will preserve the system integrity without meeting all the performance specifications. In the mean time it can work to produce a full solution to the problem.

The main function of the **Planner**, shown in Figure 2.5, is to plan the sequence of control actions, to be given to the control implementation supervisor (IIb), in order to accomplish a higher level control task. If, for example, the control executive orders the robot to move to a specific location, the planner, based on the current and possible future robot locations, will devise a sequence of actions to be taken so that tasks will be accomplished. It will, perhaps at the beginning, suggest a skeleton plan which will be refined as it is being executed. For example, start moving to the right 15 degrees, report if passage blocked, etc.

The main function of the **FDI IIa** system shown in Figure 2.5 is to detect and identify failures which occurred at levels IIb and III. It also supervises the FDI IIb system. It receives failure information from the Execution Level (III) via FDI IIb and additional information from the control manager. It informs the manager about the failure location and its severity, so that measures can be taken, using perhaps the services of the designer. It directly informs the control executive about the status of the failures detected at any level since they are very important in capability assessment. It uses high level decision making involving heuristics and few algorithms.

Management and Organization Level (I) The functional architecture for the Management and Organization Level (I) is shown in Figure 2.6. It interfaces to the pilot, crew, ground station, and other onboard systems and performs the highest level control functions. It oversees and directs all the activities at both the Coordination and Execution levels. It is the most "intelligent" of the three levels.

The main function of the **Control Executive** shown in Figure 2.6 is to accomplish high level control tasks given by the pilot, crew, ground station, or other onboard systems. Such a task could be: Change orbit to ..., deploy satellite (open door, turn, etc, then deploy), repair satellite via robot A (send robot to satellite, open hatch, repair), retrieve satellite, etc. It performs high level *planning*. It *optimally* breaks down the "macro commands" into simpler commands for the control manager (IIa). It performs *capability assessing* of the control system. It receives information about faults from FDI and about *status* and *health* from the control manager and performs high level *performance monitoring*. It evaluates the current situation and it predicts what can be reasonably expected to be accomplished in a certain time. For example, "docking procedures under way, estimated docking in 30 seconds". It provides this information to the *goal generation* facility, which by exchanging information, having a dialogue, with other onboard systems, pilot, or ground station via the interface, generates attainable realistic goals to be accomplished by the autonomous controller. For example, in view of the current situation, "docking can be achieved in 30 seconds but not in 20 seconds as requested". These goals are then used by the planner in the control executive, to plan the necessary steps which lead to their accomplishment. The control executive has significant *learning abilities*. It uses past experience to increase its efficiency and to improve its capability assessment. It is informed by the control manager about new capabilities possible, by newly generated control methods. It suggests preparation for future control tasks. It uses decision making exclusively. It interprets reports from below and execution commands from above. It can request, through the interface, additional information from the pilot, crew, ground station, or other onboard systems which may be useful in the control system. This includes navigation information, future uses of the autonomous controller, etc.

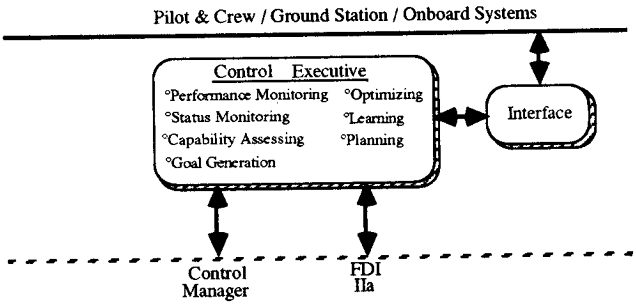

Figure 2.6. Management and Organization Level

Learning is essential to the development of a true autonomous system. High level learning will occur at the Management and Organization level. At each level of learning, beginning at Coordination level IIb, information is for instance, successively generalized via induction. The controller may need to learn the model of the plant, the problem solving strategy, the goals to obtain, and the required performance level.

The main function of the **Interface**, shown in Figure 2.6, is to provide the liaison, interface, between the autonomous control system and the Pilot and Crew / Ground Station / Other Onboard Systems. It is an *intelligent interface* as it allows user friendly dialogue. It is a *language translator*, translating language of other systems or the crew or ground station into a language familiar to the autonomous controller. It *displays data* from the control subsystems if requested. It passes the *control status* to the crew etc., and desired behavior and goals, to the control executive.

2.3 Some Design Guidelines for Autonomous Controllers

There are certain functions, characteristics, and behaviors that autonomous systems should possess [31,8]. These are outlined below. Some of the important characteristics of autonomous controllers are that they relieve humans from time consuming mundane tasks thus increasing efficiency, enhance reliability since they monitor health of the system, enhance performance, protect the system from internally induced faults, and they have consistent performance in accomplishing complex tasks.

There are autonomy guidelines and goals that should be followed and sought after in the development of an autonomous system. Autonomy should reduce the work load requirements of the operator or, in the space vehicle case discussed here, of the pilot/crew/ground station, for the performance of routine functions, since the gains due to autonomy would be superficial if the maintenance and operation of the autonomous controller taxed the operators. Autonomy should enhance the functional capability of the system. Since the autonomous controller will be performing the simpler routine tasks, persons will be able to dedicate themselves to even more complex tasks.

There are certain autonomous system architectural characteristics that should be sought after in the design process. The autonomous control architecture should be amenable to evolving future needs and updates in the state of the art. The autonomous control architecture should be functionally hierarchical; for lower level subsystems to take some actions, they have to clear it with a higher level authority. The system must, however, be able to have lower level subsystems, that are monitoring and reconfiguring for failures, act autonomously to certain extent to enhance system safety. There are also certain operational characteristics of autonomous controllers. Persons should have ultimate supervisory override control of autonomy functions. Autonomous activities should be highly visible, "transparent", to the operator the maximum extent possible.

Finally, there must be certain features inherent in the autonomous system design. Autonomous design features should prevent failures that would jeopardize the overall system mission goals or safety. These features should enhance safety, and avoid false alarms and unnecessary hardware reconfiguration. This implies that the controller should have self-test capability. Autonomous design features should also be tolerant of transient errors, they should not degrade the reliability or operational lifetime of functional elements, they should include adjustable fault detection thresholds, avoid irreversible state changes, and provide protection from erroneous or invalid external commands.

3. CHARACTERISTICS OF HIERARCHICAL INTELLIGENT CONTROLLERS FOR HIGH AUTONOMY SYSTEMS

Based on the architecture described in Section 2 we identify the important fundamental concepts and characteristics that are needed for an autonomous control theory. Note that several of these have been discussed in the literature and this book as outlined above. Here, these characteristics are brought together for completeness. Furthermore, the fundamental issues which must be addressed for a quantitative theory of intelligent autonomous control are introduced and 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" [20]. 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 [32].) 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 and this is now discussed. It was pointed out in Section 2 that complex control problems required a controller sophistication that involved the use of AI methodologies. It is interesting to observe the following [33]: 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".

4. QUANTITATIVE MODELS FOR INTELLIGENT AUTONOMOUS SYSTEMS: DISCRETE EVENT SYSTEMS AND HYBRID SYSTEMS

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 (e.g. [34]) or such as Petri nets [35] are quite useful for modeling the higher level decision making processes in the intelligent autonomous controller. It was shown in [36,37] 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 of course because of their novelty, but for the most part, it is due to the *hybrid* structure (nonuniform, nonhomogeneous nature) of the dynamical systems under consideration; they include both continuous-state and discrete-state systems. The systems are hybrid since in order to examine autonomy issues, a more global, macroscopic view of a dynamical system must be taken than in conventional control theory. Modeling techniques for intelligent autonomous systems must be able to support this macroscopic view of the dynamical system, hence it is necessary to represent both numeric and symbolic information. We need modeling methods that can gather all information necessary for analysis and design. For example, we need to model the dynamical system to be controlled (e.g., a space platform), we need models of the failures that might occur in the system, of the conventional adaptive controller, and of the high level decision making processes at the management and organization level of the intelligent autonomous controller (e.g., an AI planning system performing actions that were once the responsibility of the ground station). 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. For an extended discussion on the modeling of hybrid systems consult [38].

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 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; for some initial results along these lines see [39,40]. In any case, our modeling philosophy requires the examination of *hierarchical* models. Much work needs to be done on hierarchical DES modeling, analysis, and design, let alone the full study of hybrid hierarchical dynamical systems. Some research has begun to address hierarchical DES [34].

A practical but very important issue is the simulation of hybrid systems. This requires simulation of both conventional differential equations and symbolic decision making processes or DES. Normally, numeric-algorithmic processing is done with languages like FORTRAN and symbolic decision making can be implemented with LISP or PROLOG while DES are often simulated with SLAM. Sometimes several types of processing are done on computers with quite different architectures. There is then the problem of combining symbolic and numeric processing on one computer. If the computing is done on separate computers, the communication link normally presents a serious bottleneck. Combining AI, DES, and conventional numeric processing is currently being addressed by many researchers and some promising results have been reported. Some very promising results have been reported in [29,30] and the references therein.

5. SOME RESEARCH RESULTS AND DIRECTIONS

In this Section we will discuss results obtained on the analysis and design of several components of the intelligent controller architecture for high autonomy systems. 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. Below we provide only a sampling of the results to introduce the reader to these research areas. Many more research directions are identified in other Chapters of this book.

To determine how to utilize AI techniques it is productive to study the relationships between AI and conventional control methods. In this way one can determine what AI techniques have to offer over conventional control methods. For instance, the authors in [36] have provided a systems and control theoretic perspective on AI planning systems. In this work, the authors explain how AI planning systems are in fact control systems where the input and output variables are symbols rather than numbers. It is shown that the techniques used in the implementation of AI planning systems are actually generalized open and closed loop control, state estimation, system identification, and adaptive control.

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 [41,42] the authors introduce a fault detection and identification (FDI) system that is composed of AI decision making mechanisms and conventional FDI algorithms. The hybrid algorithmic-decision making FDI system detects and identifies failures for an intelligent restructurable controller on board an advanced aircraft.

Some control theoretic techniques offer modeling, analysis, and design techniques for the higher level decision making mechanisms in the intelligent autonomous controller. For instance, in [43,37] the authors show that AI planning problems can be studied in a discrete event system (DES) theoretic framework by utilizing the A^* algorithm. Moreover, there are many recent results developed in a DES-theoretic framework that can be used for the study of components of the intelligent autonomous controller (e.g., results from the Ramadge-Wonham formulation for the study of logical DES models).

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 [44]. Although the literature on higher level learning performed in conjunction with low level adaptation is limited, in [45] the authors show 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 ([48], [49], [50-53], [54]). 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 [42,55] studies techniques to reconfigure controllers when significant failures occur. Recently there have been advances in the theory of restructurable controls [56] where the authors develop stability bounds on the allowable parameter variations, induced by system failures; also in [57] a model following approach to the problem is introduced.

It is our viewpoint that 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 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 (Coord. Level). There is also a need for systematically extracting the necessary information from lower level symbolic models to generate higher level symbolic models to be used in the hierarchy where appropriate [39]. Tools for the implementation of this *information extraction* also need to be developed (see for instance [49]). 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.

6. CONCLUDING REMARKS

The fundamental issues in intelligent high autonomy control system modeling and analysis were identified and briefly discussed, thus providing an introduction to the research problems in the area. A hierarchical functional controller architecture for systems with high degree of autonomy was also presented. It was proposed to utilize a hybrid approach to modeling and analysis of autonomous systems. This will incorporate conventional control methods based on differential/difference equations for continuous-state systems, and new techniques for the analysis of systems described with a symbolic formalism, developed for discrete-state systems. In this way, the well developed theory of conventional control can be fully utilized. It should be stressed that autonomy is the design requirement and intelligent control methods appear, at present, to offer some of the necessary tools to achieve autonomy for some classes of applications. A conventional approach may evolve and replace some or all of the intelligent functions.

Acknowledgements: This work was partially supported by the Jet Propulsion Laboratory. In addition, the second author was also partially supported by an Engineering Foundation Research Initiation Grant.

REFERENCES

[1] Antsaklis P.J., Passino K.M. and Wang S. J., "Towards Intelligent Autonomous Control Systems: Architecture and Fundamental Issues", *Journal of Intelligent and Robotic Systems*, Vol.1, pp. 315-342, 1989.

[2] Antsaklis P.J., Passino K.M. and Wang S. J., "An Introduction to Autonomous Control Systems", *IEEE Control Systems Magazine*, Vol. 11, No. 4, pp.5-13, June 1991.

[3] Special Issue on Autonomous Intelligent Machines, *IEEE Computer*, Vol. 22, No. 6, June 1989.

[4] Saridis G.N.,"Toward the Realization of Intelligent Controls", *Proc. of the IEEE*, Vol. 67, No. 8, pp. 1115-1133, August 1979.
[5] Gevarter W.B., *Artificial Intelligence*, Noyes, NJ, 1984.
[6] Mesarovic M., Macko D. and Takahara Y., *Theory of Hierarchical, Multilevel, Systems*, Academic Press, 1970.
[7] Findeisen W.,*et al*, *Control and Coordination in Hierarchical Systems*, Wiley, New York, 1980.
[8] Firschein O.,*et al*, *Artificial Intelligence for Space Station Automation*, Noyes, New Jersey, 1986.
[9] Heer E., Lum H., *ed.*, *Machine Intelligence and Autonomy for Aerospace Systems*, AIAA, Wash. DC, 1988.
[10] Dougherty E.R. and Giardina C.R., *Mathematical Methods for Artificial Intelligence and Autonomous Systems*, Prentice Hall, 1988.
[11] Glorioso R.M. and Colon Osorio F.C., *Engineering Intelligent Systems*, Digital Press, 1980.
[12] Kusiak A., *Intelligent Manufacturing Systems*, Prentice Hall, 1990.
[13] Ozguner U.,"Decentralized and Distributed Control Approaches and Algorithms", *Proc. of the 28th IEEE Conference on Decision and Control*, pp.1289-1294, Tampa, Florida, Dec 1989.
[14] Acar L., Ozguner U., "Design of Knowledge-Rich Hierarchical Controllers for Large Functional Systems", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 4, pp. 791-803, July/Aug. 1990.
[15] Saridis G.N.,"Foundations of the Theory of Intelligent Controls", *Proc. IEEE Workshop on Intelligent Control*, pp 23-28, 1985.
[16] Saridis G.N.,"Knowledge Implementation: Structures of Intelligent Control Systems", *Proc. IEEE International Symposium on Intelligent Control*, pp. 9-17, 1987.
[17] Meystel A.,"Intelligent Control: Issues and Perspectives", *Proc. IEEE Workshop on Intelligent Control*, pp 1-15, 1985.
[18] Saridis G.N.,"Intelligent Controls for Advanced Automated Processes", *Proc. of the Automated Decision Making and Problem Solving Conf.*, NASA CP-2180, May 1980.
[19] Saridis G.N.,"Intelligent Robot Control", *IEEE Trans. on Automatic Control*, Vol. AC-28, No. 5, pp. 547-556, May 1983.
[20] Saridis G.N., "Analytic Formulation of the Principle of Increasing Precision with Decreasing Intelligence for Intelligent Machines", *Automatica*, Vol.25, No.3, pp. 461-467, 1989.
[21] Valavanis K.P., *A Mathematical Formulation For The Analytical Design of Intelligent Machines*, PhD Dissertation, Electrical and Computer Engineering Dept., Rensselaer Polytechnic Institute, Troy NY, Nov. 1986.
[22] Valavanis K.P. and Saridis G.N., "Information Theoretic Modeling of Intelligent Robotic Systems, Part I: The Organization Level", *Proc. of the 26th Conf. on Decision and Control*, Los Angeles, pp. 619-626, Dec. 1987.
[23] Valavanis K.P.,and Saridis G.N., "Information Theoretic Modeling of Intelligent Robotic Systems, Part II: The Coordination and Execution Levels", *Proc. of the 26th Conf. on Decision and Control*, Los Angeles, pp. 627-633, Dec. 1987.
[24] Jin V.Y., Levis A.H., "Compensatory Behavior in Team Decision Making", *Proc. of the IEEE Int. Symp. on Intelligent Control*, pp. 107-112, Philadelphia, Sept. 1990.

[25] Albus J., et al, "Theory and Practice of Intelligent Control", *Proc. 23rd IEEE COMPCON*, pp 19-39, 1981.

[26] Astrom K.J., *et al*, "Expert Control", *Automatica*, Vol. 22, No. 3, pp. 277-286, 1986.

[27] Zadeh L.A., "Fuzzy Logic", *Computer*, pp. 83-93, April 1988.

[28] Wos L., *Automated Reasoning: 33 Basic Research Problems*, Prentice Hall, NJ, 1988.

[29] Zeigler B.P. "DEVS Representation of Dynamical Systems: Event Based Intelligent Control", *Proc. of the IEEE*, Vol. 77, No. 1, pp. 72-80, 1989.

[30] Zeigler B.P., Chi S.D., "Model-Based Concepts for Autonomous Systems", *Proc. of the IEEE Int. Symp. on Intelligent Control*, pp. 27-32, Philadelphia, Sept. 1990.

[31] Turner P.R., *et al*, "Autonomous Systems: Architecture and Implementation", Jet Propulsion Laboratories, Report No. JPL D-1656, August 1984.

[32] 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.

[33] Mendel J. and Zapalac J.,"The Application of Techniques of Artificial Intelligence to Control System Design", in *Advances in Control Systems*, C.T. Leondes, ed., Academic Press, NY, 1968.

[34] Zhong H. and Wonham W.M., "On the Consistency of Hierarchical Supervision in Discrete-Event Systems", *IEEE Trans. on Automatic Control*, Vol. 35, No. 10, pp. 1125-1134, 1990.

[35] Peterson J.L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[36] 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.

[37] Passino K. M. and Antsaklis P. J., "On the Optimal Control of Discrete Event Systems" , *Proc. of the 28th IEEE Conf. on Decision and Control*, pp. 2713-2718, Tampa, FL, Dec. 13-15, 1989.

[38] Zeigler B.P., "Knowledge Representation from Newton to Minsky and Beyond", *Journal of Applied Artificial Intelligence*, 1:87-107, 1987.

[39] Stiver J. M. and Antsaklis P. J. , "A Novel Discrete Event System Approach to Modeling and Analysis of Hybrid Control Systems", *Proc. 29th Allerton Conference on Communication, Control and Computing*, Univ. of Illinois at Urbana-Champaign, October 2-4, 1991.

[40] Passino K.M., Ozguner U., "Modeling and Analysis of Hybrid Systems: Examples", *Proc. of the 1991 IEEE Int. Symp. on Intelligent Control*, pp. 251-256, Arlington, VA, Aug. 1991.

[41] 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.

[42] Passino K.M., "Restructurable Controls and Artificial Intelligence", *McDonnell Aircraft Internal Report*, IR-0392, April 1986.

[43] 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.

[44] Antsaklis P.J., "Learning in Control", in Panel Discussion on "Machine Learning in a Dynamic World", M.M.Kokar Organizer, *Proc. of the 3rd IEEE*

Intern. Symposium on Intelligent Control, pp. 500-507, Arlington, VA, Aug. 24-26, 1988.

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

[46] Antsaklis P.J., "Neural Networks in Control Systems", Guest Editor's Introduction, Special Issue on 'Neural Networks in Control Systems' of the *IEEE Control Systems Magazine*, Vol.10, No.3, pp.3-5, April 1990.

[47] Antsaklis P.J., Special Issue on 'Neural Networks in Control Systems' of the *IEEE Control Systems Magazine*, Vol. 12, April 1992.

[48] Antsaklis P.J.and Sartori M.A., "Neural Networks in Control Systems", *Encyclopedia of Systems and Control*, Second Supplementary Volume, M.G. Singh Editor-in-chief, 1992.

[49] 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.

[50] Sartori M. A. and Antsaklis P. J., "Neural Network Training via Quadratic Optimization", Technical Report #90-05-01, Dept. of Electrical and Computer Engineering, University of Notre Dame, May 1990.

[51] Sartori M. A. and Antsaklis P. J., "A Simple Method to Derive Bounds on the Size and to Train Multi-Layer Neural Networks", *IEEE Trans. on Neural Networks*, Vol 2, No 4, pp. 467-471, July 1991.

[52] Sartori M. A. and Antsaklis P. J., "A Gaussian Neural Network Implementation for Control Scheduling", *Proceedings of the 1991 IEEE Intern. Symposium on Intelligent Control*, pp 400-404, Arlington, VA, August 13-15, 1991.

[53] Sartori M. A. and Antsaklis P. J., "Implementation of Learning Control Systems Using Neural Networks", *IEEE Control Systems Magazine*, Vol. 12, April 1992.

[54] Sartori M.A., Passino K.M., Antsaklis P.J., "A Multi-Layer Perceptron Solution to the Match Phase Problem in Rule-Based Artificial Intelligence Systems", To appear in the *IEEE Transactions of Knowledge and Data Engineering*, 1992.

[55] Stengel R.F., "AI Theory and Reconfigurable Flight Control Systems", Princeton University Report 1664-MAE, June 1984.

[56] 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.

[57] Gao Z. and Antsaklis P.J., "Reconfigurable Control System Design via Perfect Model-Following", *Proceedings of the 1991 AIAA Guidance, Navigation and Control Conference*, pp 239-246, New Orleans, LA, August 12-14, 1991.