2004-01-2441

Online Hierarchical Fault-Adaptive Control for Advanced Life Support Systems

Sherif Abdelwahed, Jian Wu, Gautam Biswas, John W. Ramirez, and Eric J.-Manders
Dept. of EECS and ISIS, Vanderbilt University, Nashville, TN 37235.

ABSTRACT

This paper discusses a hierarchical online fault-adaptive control approach for Advanced Life Support (ALS) Systems. ALS systems contain a number of complex interacting subsystems. To avoid complexity in the models and online analysis, diagnosis and fault-adaptive control is achieved by local units. To maintain overall performance, the problem of resource management for contending concurrent subsystems has to be addressed. We implement a control structure, where predefined set-point specifications for system operation are used to derive optimizing utility functions for the subsystem controllers. We apply this approach in situations where a fault occurs in a system, and once the fault is isolated and identified, the controllers use the updated system model to derive new set point specifications and utility functions for the faulty system.

INTRODUCTION

The support of human life in the hostile environment of space critically depends on a set of complex technical systems that contain or interact with biological and chemical processes. The NASA Advanced Life Support Systems (ALS) program, itself a component of the larger Advanced Human Support Technology (AHST) Program, was created to explore new technologies required to support extended manned missions in space [1]. Potential applications include a Lunar base, a manned mission to Mars, and the International Space Station (ISS). An ALS must exhibit a high level of autonomy, so as not to detract from the mission specific tasks of the crew. This requirement translates to a high level of availability of the individual components of the ALS. It also requires that the integrated system have the ability to adapt to changing mission objectives and crew configurations, mainly in response to unplanned events.

Achieving good operational performance will depend critically on the ability to monitor the operation of the physical system, analyze its performance, and respond in a manner such that important functionalities are not lost or degraded. Process diagnosis refers to the capability to explain a deviation from nominal system behavior. Diagnosis combines the fault detection, fault isolation, and fault identification tasks [2]. These elements are a prerequisite to the ability to respond to any fault, especially if the goal is to continue operations in spite of the fault occurrence. Note that fault tolerance is not the desired objective in this case. Rather, the objective is to develop techniques so that the system can *adapt* to faults by reconfiguring itself and/or its controllers. Our goal is to develop autonomous systems, therefore, achieve this adaptivity by designing appropriate control schemes.

The design of an ALS presents complex challenges, including those related to control [3]. The system is made up of multiple loosely coupled subsystems, such as (*i*) a Water Recovery System (WRS), (*ii*) an Air Revitalization System (ARS), (*iii*) a Power generation system, (*iv*) a Thermal control system, (*v*) a Biomass production system, (*vi*) a Food production subsystem, and (*vii*) a Solid waste collection and conditioning system. These subsystems comprise a number of interacting control loops, such as the fluid flow loop, the energy management loop, the thermal control loop, the bio-regeneration and gas transfer loop, and the chemical production loop. These loops also cover multiple physical (energetic) domains and operating regimes, and operate at multiple time scales. An effective way to describe the behavior of the controlled physical subsystems is to model them as hybrid dynamical systems, which capture both the both continuous and discrete dynamics [4].

This paper discusses an online control approach for efficient resource management in Advanced Life Support (ALS) Systems. The methodology developed targets a class of hybrid dynamic systems that have finite control sets. The underlying model, referred to as a *switched hybrid system* model, can describe the dynamics of a wide variety of practical real-life systems. General hybrid systems can be described by a transition structure on a state space, which is a cross product of two domains: (i) discrete-event and (ii) continuous-time dynamics. The interaction of discrete-event and time-based variables makes the behavior generation and analysis tasks quite challenging and computationally complex. Considerable amount of research work has been dedicated recently to the study of hybrid systems dynamics [5, 6].

The complex nature of hybrid systems limits the applicability of traditional optimal control techniques and supervisory control techniques that can be applied directly to hybrid systems. Several promising approaches have been proposed in the literature to deal with the complexity of hybrid systems. For example, abstraction techniques have been developed to reduce the complexity of the hybrid models while preserving features of the original model relevant to the analysis/control objectives (e.g., [7]). Supervisory control design with abstracted hybrid system models has been investigated in [8, 9]. Efficient control synthesis for reachability specifications through mode switching has been presented in [14].

Section 2 introduces the basic building blocks of our model-based FDI and fault-adaptive control schemes, and emphasizes the importance of component-based modeling and the link between fault isolation, identification, and fault-adaptation. Section 3 presents the models of two coupled components of the Water Recovery system (WRS) of the ALS that we have chosen as the testbed for our fault-adaptive control studies. Section 4 discusses our diagnosis scheme and the hierarchical decision-theoretic control scheme to achieve optimal performance in the system given resource constraints and a set point trajectory that applies for nominal operation. Section 5 presents the results of the experiments we have conducted on the WRS, and section 6 presents the conclusions of this work.

MODEL-BASED FAULT-ADAPTIVE CONTROL

Our approach to fault-adaptive control, illustrated in Fig. 1, is centered on model-based approaches for fault detection, fault isolation and estimation, and hierarchical online supervisory control for hybrid systems. The plant is assumed to be a hybrid system [4, 5]. The heart of the *Fault Adaptive Control Unit* is the *Hybrid Observer* [11] that tracks the behavior of the plant under nominal conditions. When the *Fault Detector* detects a discrepancy between the measured and the expected behavior, the diagnosis units are triggered. The *Hybrid Diagnosis* unit combines qualitative reasoning with quantitative parameter estimation. Qualitative diagnosis is based on dynamic plant models represented as Temporal Causal Graphs [12]. The set of candidates picked by the qualitative diagnoser are passed to the *Parameter Estimation Unit* that reduces the candidate set to a single fault candidate by computing the degree of degradation, and retaining that candidate that has the least prediction error.

The online hybrid control approach focuses on optimal resource management and robust fault-adaptive control using a decision-theoretic control scheme. The proposed approach is designed to ensure distribution of a finite amount of resources among contending subsystems of a larger system in a way that *near optimal* performance may be obtained over an extended period of time. In more detail, the control algorithm is designed to achieve a set of pre-specified performance requirements for the system over finite time intervals, while simultaneously optimizing a given utility/cost function for the composite system and maintaining overall system stability. To achieve fault adaptivity, the results of fault diagnosis are used to update the system model online so that the observer may again track system behavior accurately under faulty conditions. The online supervisory controller uses the updated system model to derive a new set of performance requirements. The decision-theoretic control schemes for the individual subsystems are then applied at runtime to optimize performance in the faulty system.

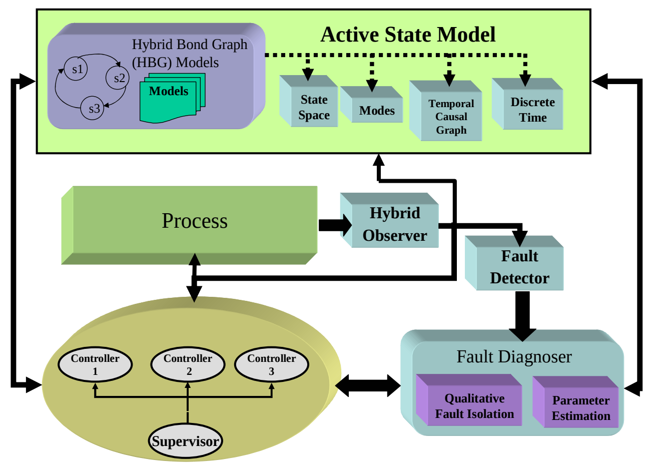

Figure 1: Fault-Adaptive Control Architecture

MODELING THE WRS SYSTEM

The ALS system is made up of multiple loosely-coupled subsystems [3], such as a Water Recovery System (WRS), an Air Revitalization System (ARS), a Biomass Production system, and a Power generation system. These subsystems comprise a number of interacting control loops, such as the fluid flow loop, the energy management loop, and the bio-regeneration and gas transfer loop. These loops also cover multiple physical domains, and operate at multiple time scales. An effective way to describe the behavior of the controlled subsystems is to model them as hybrid dynamic systems [4].

In this paper, we focus on the WRS, in particular on an experimental system that was developed and tested at the NASA Johnson Space Center (JSC) [13]. This subsystem recycles urine and wastewater into potable water. Critical requirements for such a system are that it consumes low power, minimize the use of consumable resources, and run in a fully autonomous mode for long periods of time. The WRS, as shown in Fig. 2, is comprised of a Biological Water Processor (BWP) to remove organic compounds including ammonia, a Reverse Osmosis (RO) System to remove particulate matter after the BWP, an Air Evaporation subsystem (AES) to purify the remaining concentrated brine that is purged from the

RO system, and a post processing system (PPS) to remove the trace organic and trace inorganic compounds by ultra-violet treatment to bring the water to potable limits. The combination of the BWP and RO subsystems produce about 85% of the clean water. The remaining 15% is produced by an evaporation and condensation process in the AES from the concentrated brine that is purged to it from the RO. In this work, we focus on controllers for the RO and AES systems, and the interactions between these systems to assure desired output given limited resources, which is primarily the energy available for subsystem operation.

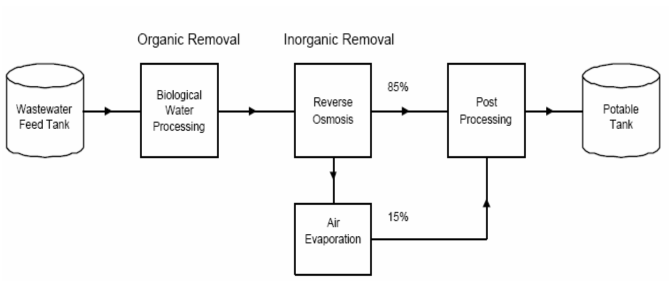

Figure 2: The Water Recovery System

THE RO SUBSYSTEM - This subsystem, shown in Fig. 3 is the linchpin subsystem in the WRS loop. It pulls water from the GLS (gas liquid separator) of the BWP, and delivers purified water (permeate) to the PPS and concentrated brine to the AES. The RO removes inorganic compounds and particulate matter by pushing the input water at high speed through a cylindrical membrane that acts like a molecular sieve. The clean water permeate is passed on to the PPS, and the dirty water (brine) continues to circulate in the RO loop.

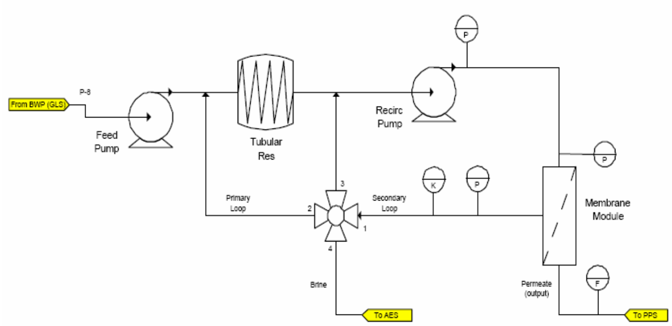

Figure 3: RO system schematic

The RO is designed to go through six modes. The primary mode draws water into a coiled section of pipe that acts like a reservoir, while processing permeate in the outer loop. When the brine concentration increases above a preset level, the system is switched to a secondary mode, where the brine circulates faster in a smaller inner loop with the recirculation pump, therefore, it is pushed harder against the membrane. This keeps the clean water production at a reasonable rate, but the concentration of brine in the inner loop continues to increase. At some point, the concentration of brine becomes high enough to reduce the output from the RO system significantly, so the brine is purged into the AES, a new batch of water is drawn in from the BWP, and the primary cycle starts again. Periodically, however, as particulate matter accumulates in the membrane, it needs to be cleaned by running the water backwards in the inner loop. This is known as the slough phase. The primary power consumers in the RO system are the two pumps, which circulate the water through the system.

THE AES SUBSYSTEM - This subsystem contains a reservoir where the brine is collected. The brine is absorbed onto a wick and evaporated using hot air blown over the wick. The evaporated water is condensed by passing it through a heat exchanger, and collected in a tank before it is sent to the PPS system. The primary power consumers in this subsystem are the blower, which moves the air through the system, and the heating unit, which heats the air to facilitate evaporation of water.

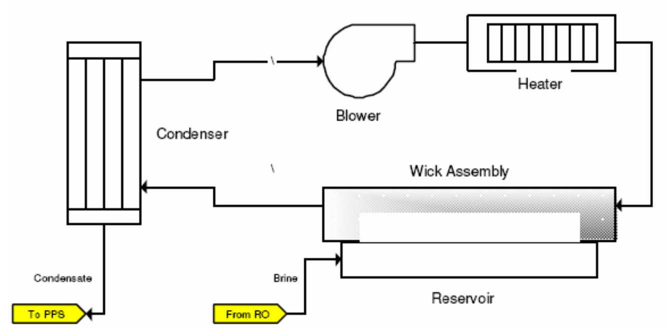

Figure 4: AES schematic

MODELING THE RO & AES SUBSYSTEMS – Building models at the right level of detail is a critical first step in the success of a model-based fault-adaptive control scheme. The choice of the model representation and the level of detail included in the model determine the set of faults that are linked to model parameters, and the set of inputs that can be controlled. In the bond-graph modeling paradigm [7] that we have adopted, faults in components that are linked to parameters in the bond-graph model can be isolated, and the controlled inputs are represented as sources of flow and effort.

Bond graphs (BG) define a domain-independent topological modeling language that captures energy-based interactions among the different physical processes that constitute a dynamic system [14]. The vertices in the graph are components or subsystems modeled as generic physical processes, such as capacities and inertias (energy storage processes), dissipators (dissipative process), transformers and gyrators (transformation between energy domains) and sources (interactions of system with environment). Component behavior can be linear or nonlinear. Additional vertices impose conservation of energy at idealized connecting points between components. Hybrid Bond graphs (HBG) are an extension of the bond graph formalism that allow some elements to have discrete states, giving the modeler the ability to create domain-independent models that can describe both continuous and discrete behaviors of a system [4]. A unique property of the HBG is the use of switching signals to turn energetic connections between HBG components *on* and *off*. Nonlinear systems are modeled by components that have time-varying parameters, i.e., their parameter values are functions of system variables.

The HBG models for the RO system and the AES are shown in Figure 5. The HBG model of the RO system was derived by decomposing the system into three principal domains of operation. Given the pump-fluid system, the mechanical and fluid domains are the primary energy domains that define the flow behavior in the system. However, to take into account the effects of impurities in the water on the flow process, and the fact that these impurities are time varying, we explicitly model the fluid conductivity domain and its interactions with the flow process using bond graph elements. The energy interaction between the mechanical and the hydraulic domains is governed by the pump characteristics, which in our simplified models of the pump are represented by the pump efficiency. A primary innovation in our model design is the ability to capture the interaction between the hydraulic and conductivity domains in the bond graph using modulating signals.

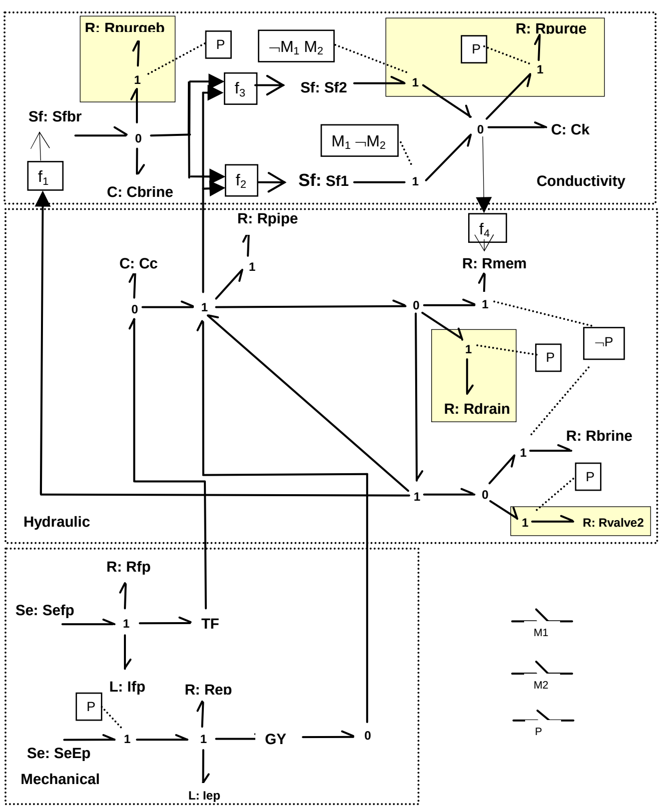

Figure 5: HBG model of RO system

The HBG for the AES consist of three domains: hydraulic, pneumatic and thermal. The hydraulic domain uses modulated source flows to model the amount of vapor being generated in the wick and the amount of vapor condensed in the heat exchanger. The pneumatic domain is modeled simply with a blower pushing air through a pipe modeled as a resistance. The thermal domain defines the primary behavior of the AES, and uses capacities to model the heat capacity in each section of the AES loop: between the heating coil and the wick, at the wick, at the heat exchanger, and after the heat exchanger. Heat transfer between sections is determined by whether the blower is on or off. When the blower is on heat is transferred by conduction, and when the blower is off heat is transferred by radiation.

We have developed tools for translating the HBG models into (i) state space equations used by the observer for tracking system behavior, (ii) Temporal Causal Graph (TCG) models for qualitative fault diagnosis, and (iii) discrete time models that are employed in the decision-theoretic controller. The translation tools are presented elsewhere [16] and not discussed in this paper.

DIAGNOSIS & FAULT-ADAPTIVE CONTROL

FAULT DIAGNOSIS SYSTEM - Our model-based approach to fault detection and isolation (FDI) combines robust tracking of nominal system behavior using extended Kalman filter techniques [15], statistical fault detection and symbol generation techniques, and a novel fault isolation method that is based on the qualitative analysis of the system dynamics immediately after the time point of fault occurrence followed by quantitative parameter estimation to uniquely isolate and identify the fault [12]. The extension of these methods to hybrid systems complicates the analysis in that discrete mode changes, and, therefore, model switches occur while tracking and analyzing system behavior. An automaton model is employed to switch system models when mode changes occur [11].

We have conducted extensive FDI experiments on a number of simulated fault scenarios on the RO system. These correspond to faults in the pump (loss of efficiency and increased friction in the bearings), membrane (clogging), and the connecting pipes (blocks). Faults were introduced as abrupt changes in parameter values, i.e., a discrete change in the parameter value that is assumed to occur at a point in time. We consider two faults in the RO system: (1) a 5% decrease in the recirculating pump efficiency (fault introduced at time step = 380), and (2) a 35% increase in bearing friction for the recirculating pump (fault introduced at time step = 400). The fault magnitudes were chosen to ensure detection (after some delay). In both cases, there was a delay of about 100 to 200 units in detecting the fault. For each scenario, the qualitative fault isolation scheme required a set of measurement deviations to reduce the initial candidate set considerably, and parameter estimation converged to the correct fault candidate. The estimated parameter values were within 5% of the actual change and quite acceptable for control purposes. Details of the diagnosis scheme are presented elsewhere [11, 12].

FAULT ADAPTIVE CONTROL - To utilize control theory for resource management, a suitable model for the underlying system has to be established. The system model captures the relationship between the observed system parameters, particularly those relevant to the requirement specifications, and the control inputs used to adjust these parameters [16]. Typically, an initial model is built for those system components with known dynamics, while parameter estimation techniques are used to identify the unknown parameters of the system.

SWITCHING HYBRID SYSTEMS – The control approach proposed in this papers targets a special class of hybrid systems in which the controlled input to the system is characterized by a finite control set. The following discrete-time form of the state space equations describes the continuous dynamics of this class of hybrid systems:

$$x(k+1) = \Phi(x(k), q(k))$$
$$q(k+1) = \delta(q(k), x(k))$$

where k is the time index, $x(k) \subseteq \Re^n$ is the sampled form of the continuous state vector at time k, $x(k) \subseteq \Re^m$ is the discrete valued input vector at time k, and $q(k) \in Q$ is the mode (discrete state) at time k. Q is a finite set of discrete states that the system can be in. δ is the (partial) transition relation. We use X and U to denote the state space and the finite input set for the system, respectively. For each mode, $q \in Q$ the function Φ_q is continuous in X and meets the conditions for existence and uniqueness of solutions for a set of initial states $X_o \subseteq X$. Note that in the above representation, at any time step k the system input defines the next mode of the system and the next state is computed from the corresponding state equation.

The above model is general enough to describe a wide class of hybrid systems, including nonlinear systems and piecewise linear systems. The requirement that the input set is finite is not uncommon in practical computer-controlled systems, where the control inputs are usually discrete and take values from a finite set. It is important to note, however, that the proposed online control approach is more suitable for systems with small number of control inputs, since the size of the search tree grows exponentially with the number of input switching signals which is proportional to the size of the input set. Many real-time computation systems have a limited finite (quantized) set of control inputs and, therefore, can be adequately captured using the above model.

REQUIRMENT SPECIFICATIONS – In many real-life systems performance specifications can be classified into two categories. The first type is set-point specifications in which the underlying parameter or variable is required to be maintained at specific level or follow a certain pattern (trajectory). Examples of this type include car speed in cruise mode and water quality in a water supply system. The other type of specification, referred to as performance specifications, is used to optimize the system performance by minimizing or maximizing given performance measures, such as power consumption and system utilization. The performance measure is a function of the state, input, and output variables, typically, a weighted norm in which these variables are added together with different weights reflecting their contribution to the overall system utility and/or cost.

The objective of the control structure is to achieve the desired level of the set-point specifications in “reasonable” time, maintain the system stable at the desired value, and optimize the given performance function. Note that, due to the nature of the system environment, it is common that the variables used to optimize the performance functions are evaluated over a quantized finite domain. For example, the quality of the result of a given subsystem varies with respect to the size of the input, which can only take a finite set of values.

In certain situations, the optimal operation point can be computed at design time, and used as a set-point objective for the system controller. In this case, the performance function can be translated into a linear or integer programming problem. We assume that optimal points for performance functions can be computed, therefore, the specification is given as one or more set-points, or a state-space region. The specifications may change during operation, and the proposed approach can accommodate the changes.

THE ONLINE CONTROL APPROACH – The online controller tries to satisfy the specification by continuously monitoring the current state of the system and selects the input that can best satisfy the given specification. In addition, the controller is required to keep the system stable within the domain that satisfies the specification. In this setting, the controller is simply considered an agent that applies a given sequence of events in order to achieve a certain objective.

In the online control approach, the controller explores only a limited forward horizon in the system state space and selects the next event based on the available information. Considering the case of set-point specification, the selection of the next step is based on a distance map that defines how close the current state is to the desired set point. The distance map can be defined for each state $x \subseteq \Re^n$ as $D(x) = ||x - x_s||$, where $||.||$ is a proper norm for $\Re^n$. In the case of performance specification, the input that minimizes (maximizes) a given utility function is selected. This function assigns to each state of the system a cost associated with reaching and maintaining that state.

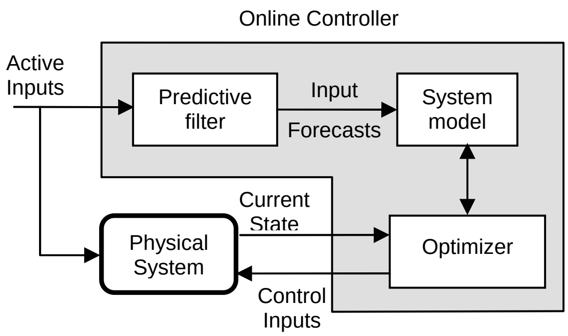

Figure 6: The basic structure of online control

The online supervision algorithm starts by constructing the tree of all possible future states from the current state up to a specified depth. To avoid Zeno effects, where the controller may try to preempt time indefinitely by switching over very small time intervals, we require at most one event switch per time unit. The exploration procedure identifies the set of states that best satisfy the given specification as discussed above. A state x_m is then chosen from this set based on certain optimality criterion (for instance minimum time from the current state), or simply picked at random. The chosen state is then traced back to the current state and the input leading to x_m is used for the next step. The basic structure for online control is shown in the Fig. 6.

In the above structure, we assume that there are two forms of system inputs; active inputs and control inputs. Active inputs are typically a set of stochastic signals received from the system environment. In order to compute possible future behavior, a *predictive filter* is implemented in the above structure to provide forecasts for future environment inputs up to the depth of the look-ahead horizon. In many practical situations, the variations in environment inputs can be predicted using an appropriate time-series model [17].

THE MULTILEVEL CONTROL STRUCTURE – In a distributed system with several components (processes), each component has its own requirement specification that defines its desired region of operation. In addition, a global performance requirement for the overall system may also be specified. The nature of such multi-component systems, suggests a decentralized control structure in which each components has a local controller. The need to share resources and possible interactions between the system components may cause conflicts among the controllers. To manage such conflicts and to address possible global specifications of the overall system, a supervisory controller is used creating a multi-level control structure shown in Fig. 7.

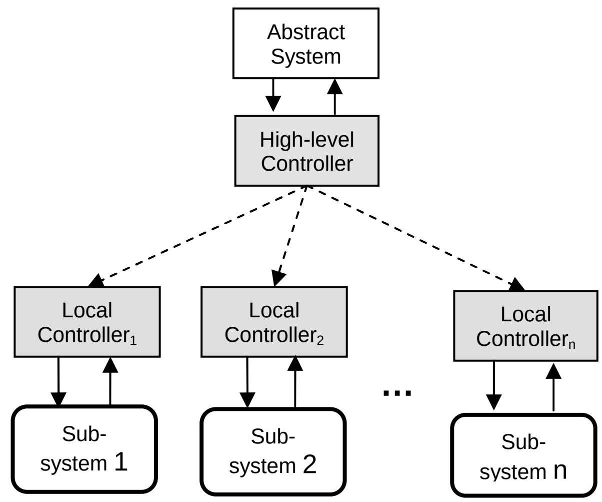

Figure 7: A Multi-level Control Structure

In the above control scheme, the high level controller manages the interactions between the local controllers using an abstract model of the system that contains information relevant to the objectives of the high-level controller. This includes, for instance, information about the interaction between the system components in terms of specific local variables that contributes to a global objective. The abstract dynamics then represents how these variables would changes in reaction to certain settings that the global controller can enforce through commands to the local controllers. The local controllers then try to optimize the performance of the local components using utility functions, but ensure conditions imposed by the global controller are not violated.

Typically, in a multi-level control structure, the high level controller takes a long-term perspective, while the local controllers act to optimize their components on a short-term basis. The high-level commands can then be viewed as a set of long-term restrictions on the local controllers directed towards satisfying a global objective. The local controller then acts to optimize the underlying component subject to the high level restriction.

UTILITY FUNCTIONS FOR CONTROL – The optimizing component to safety control is introduced in the form of a multi-attribute utility function, $\sum_i V_i(P_i)$, where each V_i corresponds to a value function associated with performance parameter, P_i. The parameters, p_i, can be continuous or discrete-valued, and they are derived from the system state variables, i.e., $P_i(t) = p_i(x(t))$. The value functions employed have been simple weighted functions of the form $V_i(P_i) = w_i P_i$, where the weights take on values in the interval [–1 1], and represent the importance of the parameter in the overall operation of the system. The supervisory controller uses the system model to predict possible behaviors corresponding to different action sequences for a finite forward time horizon, and then selects the action (i.e., control input) that maximizes the utility function. This process is then repeated for the next time step, and so on.

EXPERIMENTS

We present a set of simulation experiments to illustrate multi-level fault adaptive control of the system. As a first step, the upper level controller was designed based on the information of arrival rates to RO (predicted), and the average production rates and corresponding average power consumed for each high level preset modes of the RO and AES systems. The corresponding performance index is defined as

$$J = \sum_{i=k}^{k+N-1} \left\| f_{m_i}^{ro} - \lambda_i \right\| + \left\| P_{m_i}^{ro} \right\| + \left\| \alpha f_{m_i}^{ro} - f_{m_i}^{aes} \right\| + \left\| P_{m_i}^{aes} \right\|$$

where $f_{m_i}^{ro}$ and $P_{m_i}^{ro}$ are average production flow rates and power consumed for the preset modes of the RO. $f_{m_i}^{aes}$ and $P_{m_i}^{aes}$ are the flow rates for the AES, and N is the number of horizons. By minimizing *J* with the arrival rates for 10 cycles, the RO and AES sequences are determined shown in Fig. 8.

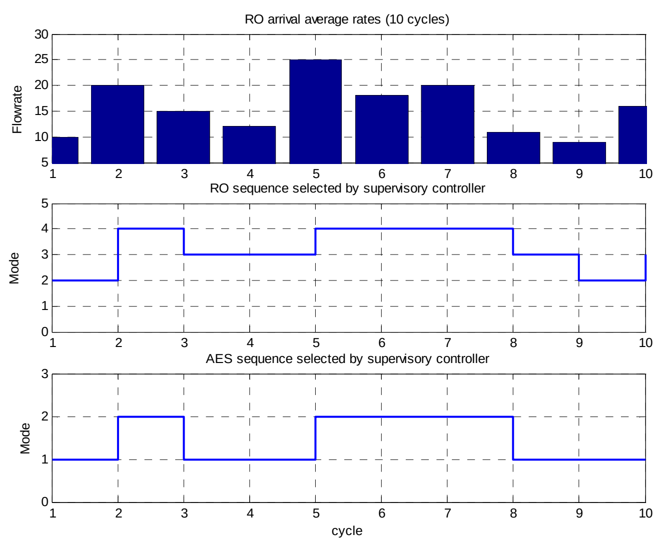

Figure 8: Arrival sequence and RO and AES modes

The local controllers operate using the control commands, i.e., the utility function coefficients provided by the high level controller. The utility function for the RO is given by:

$$V(k) = \sum_{i=k}^{k+N} a_K [K(i)/K_{\max}] + a_f [f_3(i)/f_{\max}] + a_{S_v} S_v + a_p [P(i)/P_{\max}]$$

Table 1 shows the parameters of the RO utility functions for different high level modes of operation determined by the average flow rate and power consumed, the parameters used in the high level controller. Note that Mode 1 for the RO is the off mode. Fig. 9 gives the results of RO local control in 10 cycles. Similarly, the AES runs for four normal-speed modes for 4 cycles (see Fig. 8) and is off in other modes.

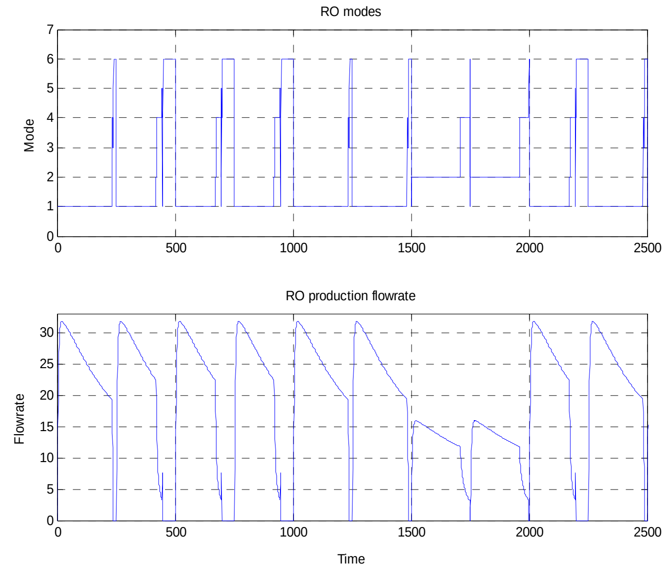

Figure 9: RO local modes changes according to the high-level sequences of Figure 8

The RO cycles through six local modes. Mode 1 represents high speed primary loop, mode 2 corresponds to the low speed primary loop, mode 3 the high speed secondary loop, mode 4 the low speed secondary loop, mode 5 represents the clean mode, and mode 6 corresponds to purge (off).

RO	Weights $[a_K, a_f, a_{S_v}, a_p]$	Average flow rate	Average power
Mode 2	[18, 1, -10, -1]	12.25	0.41
Mode 3	[26, 2, -10, -1]	18.96	0.73
Mode 4	[32, 3, -50, -1]	23.64	0.95

Table 1: Parameters of high-level RO system modes

A second experiment was conducted with faults introduced in the RO system at run time. Fig. 10 shows the behavior of the system under online control in the presence of fault.

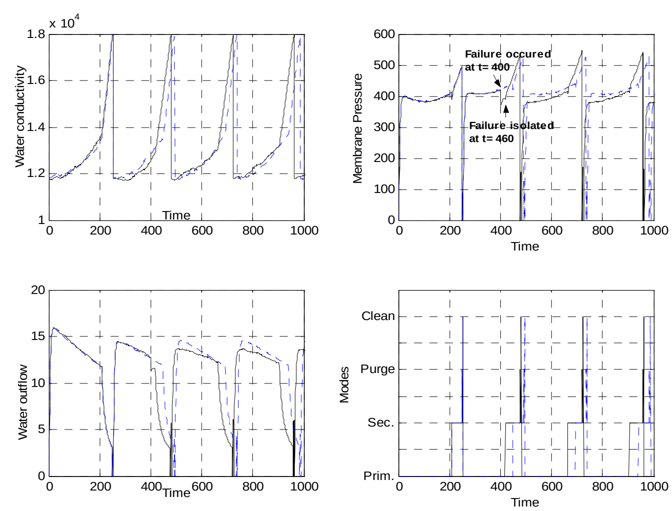

Figure 10: System performance under online control with pipe blocking failure

A block in a pipe (resulting in 35% increases its resistance) was introduced at time t = 400 sec and was isolated at time t = 430 sec. The online controller managed to compensate for the fault by increasing the time spent in the primary loop, i.e., step 1 mode of operation. The overall average utility in this case was only 0.93% less than the utility in the non-faulty situation. In Fig. 10, the original system output (no failure) is shown in dotted line for comparison.

CONCLUSIONS

In this paper, we have demonstrated a successful scheme for online model-based diagnosis and fault-adaptive control of complex hybrid systems. In addition, we have successfully developed a hierarchical control schemes that combines the management of resource constraints at a global level with optimizing individual subsystem behavior at the local level. In future work, we will extend this scheme to larger systems with more distributed components.

ACKNOWLEDGMENTS

This work was supported in part through grants from the NASA-IS program (Contract number: NAS2-37143) and the NASA-ALS program (Contract number: NCC 9-159). We acknowledge the help provided by the members of the ALS group at NASA Johnson Space Center (JSC), primarily, Pete Bonasso, David Kortenkamp, Wen-Chin Lee, and other members of the Lockheed group at NASA JSC.

REFERENCES

1. Duffield, Hanford, and Lange, "Advanced life support requirements document," Tech. Rep. JSC-38571, Rev. B, NASA-Lyndon B. Johnson Space Center Houston, TX, Sept. 2002.
2. R. Isermann, "Supervision, fault-detection and fault-diagnosis methods – an introduction," Control Engineering Practice, vol. 5, no. 5, pp. 639–652, May 1997.
3. P. Bonasso, D. Kortenkamp, and C. Thronesbery, "Intelligent control of a water recovery system: Three years in the trenches," AI Magazine, pp. 19–43, 2003.
4. P. J. Mosterman and G. Biswas, "A theory of discontinuities in dynamic physical systems," Journal of the Franklin Institute, vol. 334B, no. 6, pp. 401–439, 1997.
5. P. Antsaklis, editor. Special Issue on Hybrid Systems, volume 88 of Proceedings of the IEEE, July 2000.
6. P. Antsaklis and A. Nerode, editors. Special Issue on Hybrid Control Systems, volume 43 of IEEE Trans. Automatic Control, April 1998.
7. J. Lunze. Qualitative modeling of linear dynamical systems with quantized state measurements. Automatica, 30(3):417{431, 1994.
8. M. Morari and J. Lee. Model predictive control: Past, present and future. Computers and Chemical Engineering, 23:667-682, 1999.
9. J. Cury, B. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. IEEE Trans. Automatic Control, 43(4):564-568, 1998.
10. T. J. Koo, G. Pappas, and S.Sastry. Mode switching synthesis for reachability specifications. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, Hybrid Systems: Computation and Control, no. 2034 in LNCS, pages 333-346. Springer Verlag, 2001.
11. G. Biswas, G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, G. Karsai, "A robust method for hybrid diagnosis of complex systems," 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS), Washington, D.C., pp. 1125-1130, June 2003.
12. P. J. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," IEEE Trans. Systems, Man, and Cybernetics - A, vol. 29, no. 6, pp. 554–565, 1999.
13. D. Kortenkamp and S. Bell, "BioSim: An integrated simulation of an advanced life support system for intelligent control research," in Proc. of the 7th Intl. Symposium on Artificial Intelligence, Robotics and Automation in Space, 2003.
14. D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, Systems Dynamics: Modeling and Simulation of Mechatronic Systems, John Wiley & Sons, Inc., New York, 3rd edition, 2000.
15. K. Brammer and G. Siffling, Kalman-Bucy Filters, Artec House, Norwood MA, 1989.
16. G. Karsai, G. Biswas, S. Abdelwahed, T. Pasternak, S. Narasimhan, G. Peceli, G. Simon, and T. Kovacshazy. Towards fault-adaptive control of complex dynamical systems. In Tariq Samad and Gary Balas, editors, Software-Enabled Control Information Technology for Dynamical Systems, chapter 17, pages 347-368. Wiley Interscience, 2003.
17. S. A. DeLurgio, Forecasting Principles and Applications, McGraw-Hill Intl., Singapore, 1998.

CONTACT

Gautam Biswas, Dept. of EECS and ISIS, Box 1824 Station B, Vanderbilt University, Nashville, TN 37235.Tel: (615)-343-6204; e-mail: gautam.biswas@vanderbilt.edu