Online Fault Adaptive Control for Efficient Resource Management in Advanced Life Support Systems

S. Abdelwahed J. Wu G. Biswas J. Ramirez E. Manders

Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN 37235

Abstract

In this paper we present the design and implementation of a controller scheme for efficient resource management in Advanced Life Support Systems. In the proposed approach, a switching hybrid system model is used to represent the dynamics of the system components and their interactions. The operational specifications for the controller are represented as a utility function, and the corresponding resource management problem is formulated as a safety control problem. A limited-horizon online supervisory controller is used for this purpose. The online controller explores a limited region of the state-space of the system at each time step and uses the utility function to decide on the best action. The feasibility and accuracy of the online algorithm can be assessed at design time. We demonstrate the effectiveness of the scheme by running a set of experiments on the Reverse Osmosis (RO) subsystem of the Water Recovery System (WRS).

1 Introduction

This paper discusses an online hybrid control approach for robust fault adaptive resource management and control in Advanced Life Support 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 ([2, 3, 11, 18] and the references therein).

The complex nature of hybrid systems limits the applicability of traditional optimal control techniques and supervisory control techniques that can be 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., [16]). Supervisory control design with abstracted hybrid system models has been investigated in [15, 21, 9]. Efficient control synthesis for reachability specifications through mode switching has been presented in [14].

In this paper, we develop an online control approach for efficient resource management in Advanced Life Support Systems. 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 "optimal" performance may be obtained over an extended period of time. In more detail, the control algorithm is to 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. We further demonstrate the application of this approach for fault-adaptive control by introducing faults in system components. We apply fault diagnosis methods online to isolate and identify the fault, and once this is done, the controller uses the updated system model to derive a new set of performance requirements. The online decision-theoretic control scheme is then applied at runtime to optimize performance in the faulty system.

To achieve these objectives, we propose a receding horizon online supervision algorithm. This algorithm selects the next set of control actions (i.e., the input to the system) based on information available about the current state and a utility-based evaluation of the consequences of various action sequences over a finite lookahead window. In this setting, the selection of the next step is based on two maps; a distance map that that defines how close the current state is to the desired set point trajectory, and a utility map that defines the current level of performance of the system. Both maps are generated at design time from system specifications.

The proposed control approach is conceptually similar to model predictive control schemes [17, 20], where a limited time forecast of the process behavior at each state is optimized according to a given criterion (cost function) over the set of controlled inputs. The application of model predictive techniques for the control of hybrid systems was investigated in [22, 13]. The online control approach is also conceptually related to the online limited lookahead supervision of discrete event systems [8].

A second component of our approach is an online fault diagnosis methodology for hybrid systems. This approach derives fault signatures from a temporal causal graph derived from the system model, and uses them in an innovative fault isolation scheme. Search methods are employed to track system behavior across discrete mode changes. After the qualitative fault isolation process, a parameter estimation scheme is employed to estimate the fault parameter value. This approach is discussed in more details in [4].

The fault isolation and online control scheme is applied to subsystems of the Water Recovery System (WRS) of Advanced Life Support Systems (ALS) being designed by NASA for long-duration planetary missions. We demonstrate the effectiveness of the fault isolation scheme on the Reverse Osmosis (RO) process, and demonstrate the online controller keeps system performance within reasonable bounds for a variety of degradations and faulty conditions.

This paper is organized as follows. The switching hybrid system model used to represent the RO system is introduced in Section 2. In this section, we discuss also different forms of performance specifications for this class of systems. In Section 3, the ALS water recover system is introduced and the main elements system are presented. Section 4 presents the online approach for control of switching hybrid systems. The proposed approach is demonstrated for the control of a ALS water recovery system in Section 5. The fault detection and isolation approach is briefly discussed in this section. Results for control in failure scenarios are presented in this section. Conclusion and future work are discussed in Section 6.

2 Switching hybrid systems

As discussed, we consider a special class of hybrid systems in which the controlled input to the system is characterized by a finite control set. The continuous dynamics of this class of hybrid systems is described by the following discrete-time form of the state space

$$\begin{aligned} x(k+1) &= \Phi_{q(k+1)}(x(k)), \\ q(k+1) &= \delta(q(k), u(k)) \end{aligned}$$

where $k \in 0, 1, \ldots$ is the time index, $x(k) \subset \mathbb{R}^n$ is the sampled form of the continuous state vector at time k, $u(k) \subset \mathbb{R}^m$ is the discrete valued input vector at time k, and $q(k) \in Q$ is the mode (discrete state) at time k. Note that Q is a finite set of modes (discrete states) that the system can be in. δ is the (partial) transition relation. We will use X and U to denote the state space and the finite input set for the system, respectively. For each input $q \in Q$, 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.

A switching hybrid system is a special class of hybrid automata [1], therefore, in general, it is possible to represent switching hybrid systems as a hybrid automaton. The discrete-time dynamic specification above can be extended to a hybrid automaton representation by adding the guard and invariant conditions as shown in Figure 1. In each mode, the system dynamics is described by the set of discrete-time state equations of the form described above. An invariant condition may be added describe the domain of operation in each mode. Transitions between modes are defined by guard conditions on the system states, and a transition between automaton states may cause some of the state variables to be reset.

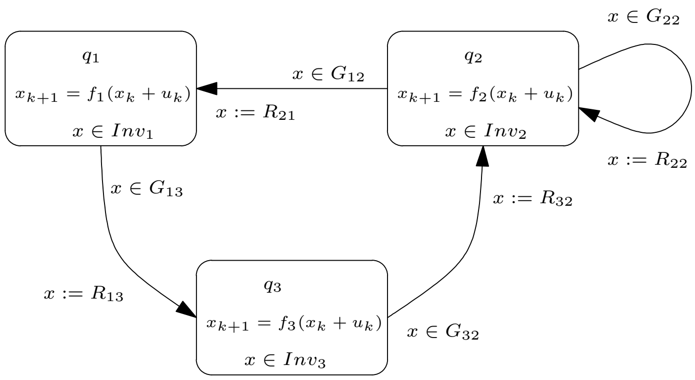

Figure 1: A Hybrid Automaton

For switching hybrid systems, discrete inputs drives the system into different modes of operation. In contrast with the general hybrid automata model there is no continuous input that can drive the system dynamics within the discrete modes. However, the switching hybrid system model is general enough to describe a wide class of practical hybrid systems. The requirement that the input set R is finite is typical in many computer-controlled systems, where the input is usually discrete and restricted to a finite set. However, the proposed online control approach is more suitable for systems with small number of control inputs as, in general, 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.

Specification patterns

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 cruse 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 a given performance measure 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 set-point objective for the system controller. This is the case for instance when the performance function can be translated into a linear or integer programming problem. In this paper we assume that optimal points for performance functions can be computed at any given time instance and therefore the requirement specification is given as a set-point, multiple set-points, or a state-space region. Such specification may change during the operation. The proposed approach can accommodate such changes as will be described in the next section.

3 The ALS Water Recovery System

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. NASA's Advanced Life Support Systems (ALS), a component of the Advanced Human Support Technology (AHST) Program, was created to explore new technologies required to support extended manned missions in space [10]. Potential applications include a Lunar base, manned missions 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.

The ALS system is typically made up of multiple loosely-coupled subsystems [6], 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, and (vi) a Waste Processing subsystem. 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 dynamic systems, which capture both the both continuous and discrete dynamics [19].

In this paper, we focus on the water recovery system (WRS). This subsystem recycles urine and waste water into potable and utility water. Critical requirements for such a system are that it consume low power, minimize the use of consumable resources, and be able to run in a fully autonomous mode for long periods of time. The WRS, as shown in Figure 2, is comprised of (i) a Biological Water Processor (BWP) to remove organic and inorganic compounds including ammonia; (ii) a Reverse Osmosis (RO) System to remove particulate matter after the BWP, (iii) an Air Evaporation subsystem (AES) to purify the remaining concentrated brine that is purged from the RO system; and (iv) a post processing system (PPS) to remove the trace organic and trace inorganic compounds by ultra-violet treatment to bring the water within potable limits. As shown in the figure, 85% of the water outcome of the reverse osmosis module is directed for post processing while the remaining 15% (most unclean water) is directed to the Air evaporation unit.

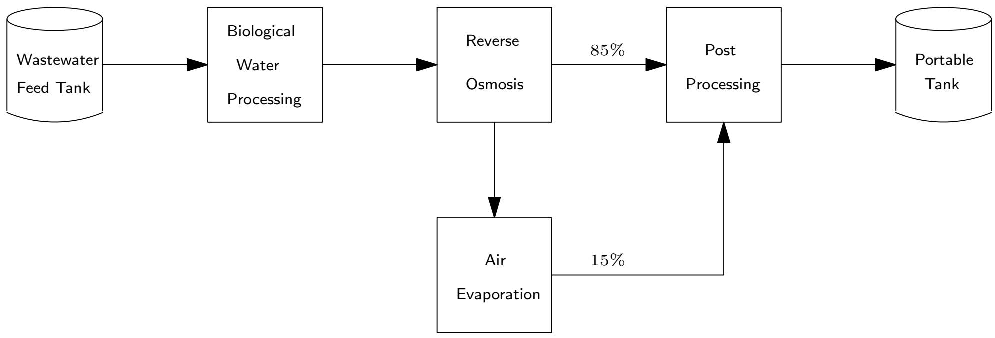

Figure 2: The Water Recovery system

The reverse osmosis (RO) system, as shown in Figure 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.

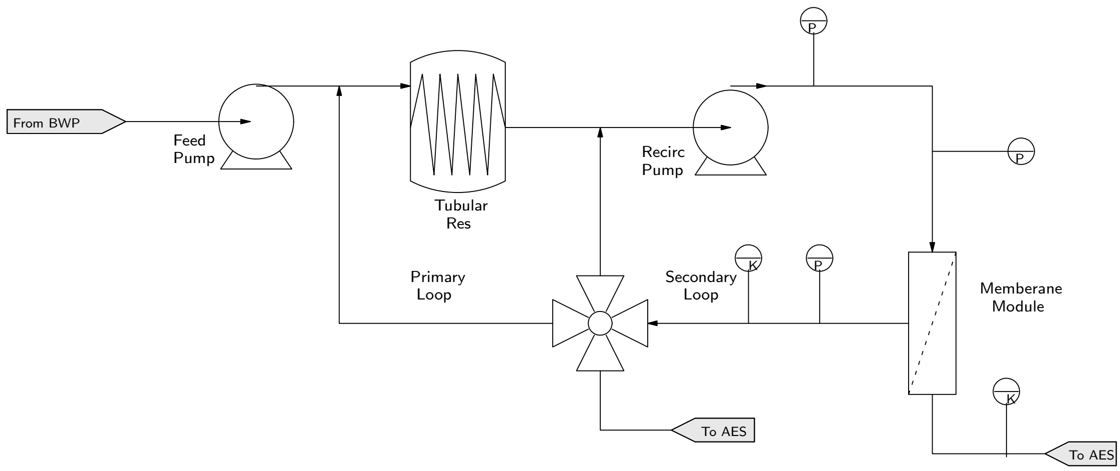

Figure 3: Diagram of the Reverse Osmosis system

For proper operation, the RO is designed to go through four distinct modes. The primary phase draws water into a coiled section of pipe that acts like a reservoir, while processing permeate in the outer loop of pipes. At some point as the brine concentration increases, the system is switched to a secondary mode, where the brine circulates in a smaller inner loop with the recirculation pump, therefore, its speed increases and it is pushed harder against the membrane. This keeps the production of clean water 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.

Control engineers working on the WRS devised a control scheme where the primary loop was run till about 70% of the water in the RO loop was processed. At this point, the four-way valve was switched to the secondary phase, where the dirtier water now circulates at a faster speed, and some water is injected from the coil into the secondary loop because of the effect of the feed pump. When 90% of the original reservoir amount has been processed, the four way valve is switched to the purge mode, and the concentrated brine drains into the reservoir of the AES system. In experimental studies run on a testbed at NASA JSC, control engineers derived average times the RO system spent in the primary and secondary loops for the above conditions, and built a time-based controller for the RO system.

3.1 System model

For our study, the input voltage to the feed pump, e_1, input current to the recirculating pump, i_1, pressure of liquid in the coil, p_3, pressure drop at the membrane, p_4, and the conductivity of the water in the loop, K are used as state variables. The output (observed) variables are the outflow of clean water to the PPS.

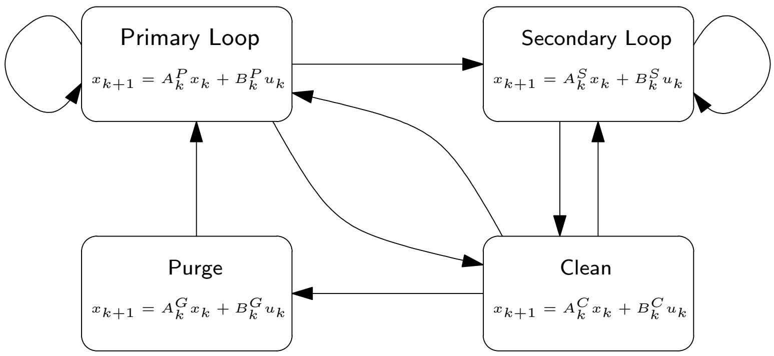

Figure 4: The switching hybrid model of the Water Recovery System

In the system model, the feed pump provides a constant flow of water from the BWP to the RO subsystem. Control actions are take by switching the 4-way valve and by changing the pump speeds, i.e., the input voltage and current, respectively. x, the state vector of the system is given by $[e_1\ i_1\ p_3\ p_4\ K]^T$, and u the controlled input variable vector is $[S_e\ S_f\ V]^T$, where V is the valve position. As discussed earlier, the system starts in the primary mode and the initial value of the state vector is given by $[0\ 0\ 0\ 0\ 12000]^T$. Figure 4 shows the hybrid system model for the RO system. The figure reiterates that the switching hybrid model of the system has the four operating modes: primary loop, secondary loop, purge, and clean.

The key component that governs the behavior and performance of the RO model is the membrane, which behaves as a combination of a capacitor and a time-varying resistance. The increase in resistance in the primary and secondary modes of operation are a function of the flow rate of water in the loop and the water conductivity. This is because the amount of particulate matter that sticks to the membrane and clogs its pores increases with time. The rate of increase in resistance is greater in the secondary loop because the loop is shorter and the conductivity of the water is higher (i.e., the water is dirtier). The membrane resistance is a quadratic function of the flow rate and a linear function of conductivity. The outflow from the RO system, f_3 is given by the equation:

$$f_3(k) = c_1/(d_1 + d_2\, r_m(k))\; i_1(k) + c_2/(d_3 + d_4\, r_m(k))\; p_3(k) + c_3/(d_5 + d_6 * r_m(k))\; p4(k)$$

where the c_i and d_i parameters in the equation depend on the system mode. This outflow is used to evaluate the system performance, which is discussed in more detail later.

4 Online control of switching systems

The problem of optimal safety control is stated as follows.

> Given a switching hybrid system H and a set of safe states X_s and a set of initial states $X_o \subseteq X$, where $X_s \subset X_o$, design a supervisor S that can drive the system from any state in X_o to X_s in a finite number of time steps using a finite set of switching events with minimum cost.

In addition, the supervisor is required to keep the system stable within the set X_s. In this setting, the supervisor is simply considered an agent that applies a given sequence of events (possibly changing the discrete input) in order to achieve a certain objective.

In the online supervision approach, the controller explores only a limited forward horizon in the system state space and selects the next event based on the available information. For the safety control problem, the selection of the next step is based on a distance map $D_s : \mathbb{R}^n \to \mathbb{R}$ that defines how close the current state is to the safe region. The distance map can be generally defined as follows; for each point $x \in \mathbb{R}^n$,

$$D_s(x) = \inf_{x' \in X_s} \|x - x'\|,$$

where $\|.\|$ is a proper norm for $\mathbb{R}^n$. In other words, $D_s(x)$ defines the minimum distance between x and the closure of the safe region X_s.

The online supervision algorithm starts by constructing the tree of all possible future states from the current state x_c 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 with the minimum distance from X_s based on the distance map D_s. 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 x_c and the event leading to x_m is used for the next step.

In general, the complexity of the online control approach is exponential in the depth of the exploration tree. However, it is possible to reduce the search tree significantly using some offline analysis of the system dynamics. For instance, write $\hat{\delta}_{X_o}$ for the maximal single step absolute change to any component in a state $x \in X_o$ under any input from R, namely $\hat{\delta}_{X_o}$ is equal to

$$\inf_{x \in X, u \in U} \|\Phi(x,u) - x\|$$

It is easy to see that $|\hat{\delta}_{X_o}|$ is an upper bound for the the distance covered by the system in one step. The computation of $\hat{\delta}_{X_o}$ is simple for discrete-time linear and piecewise-linear systems. This upper bound can be used to reduce the search tree in the online control algorithm. The algorithm can safely stop exploring if there is no prospect of further reduction in the current minimal distance along any path starting from the current node up to the limit of the search tree. The algorithm for online control with efficient termination is given below.

Algorithm 1 Online Control Algorithm

```
Tree(x_c) := {x_c}; terminal(x_c) = ∅
MinDistance := ∞
while Tree(x_c).depth < N do
  newStates := Post(Tree(x_c).states) - (Tree(x_c).states ∪ terminal(x_c))
  if newStates = ∅ then
    break
  end if
  for all x ∈ newStates do
    if D_s(x) + (N − x.depth) ∗ δ̂_{X_o} > MinDistance then
      terminal(x_c).add(x)
    else
      if D_s(x) < MinDistance then
        MinDistance := D_s(x)
        OptState := x
      end if
      Tree(x_c).addState(x)
    end if
  end for
end while
return OptState
```

In the above algorithm, the function $Post(X)$ returns the set of all states that are reachable from the set X in one time steps for all possible inputs $u \in U$. Upon termination the state OptState is traced backward to the root state x_c and the initial input leading to OptState is selected as the next input.

5 Online Control of the ALS Water Recovery System

This section briefly describe the online control method and formulates the resource management problem that includes the system performance and the utility function used to measure this performance.

The optimizing component to safety control is introduced in the form of a multi attribute utility function, $V = \sum_i V_i(p_i)$, 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. Figure 5 shows the controller structure comprising the processor model, estimators, and the optimizer.

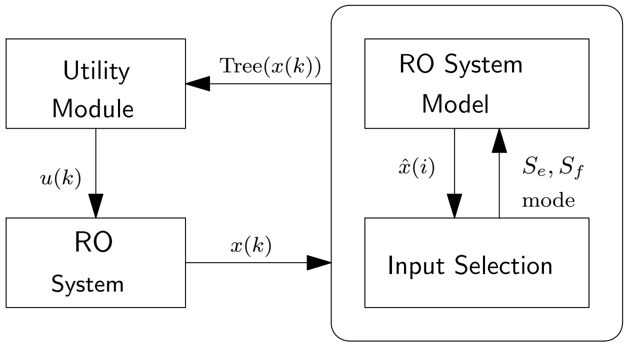

Figure 5: The basic structure of the online controller

In the ALS system, the quality (utility) of the system is measured in terms of its water outflow f_3, the conductivity of the water K, and the number of valve switches S_v. The corresponding multi-attribute utility function is expressed as

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

S_v is included as a penalty term to avoid too much switching. As discussed, the objective of the online controller is to maximize the utility function, $V(k)$. The parameters a_K, a_f, and a_{S_v} are designer-specified weights denoting the relative importance of the corresponding performance parameters. In the above equation N denotes the number of lookahead steps. The online controller uses the estimated future state vector $\hat{x}(k+1)$ for given set of inputs up to a finite number of forward steps, N, to decide the best input to maximize the utility.

Figure 6 summarizes the controller performance corresponding to one simulation run. The figure to the left shows various state variables under the online control action. The figure to the right shows the corresponding mode switching signals generated by the online controller. In the above experiment we chose $N = 3$ and the heuristics discussed earlier limits the search space to an average of 37 states to explore per time step. Also, we assumed a 4% noise in the system measurement. It clear from the figure that the controller maintained a uniform cycling in spite of the measurement noise.

In the following sections, we briefly describe the fault isolation algorithm, and the use of the utility-based controller to accommodate for fault effects.

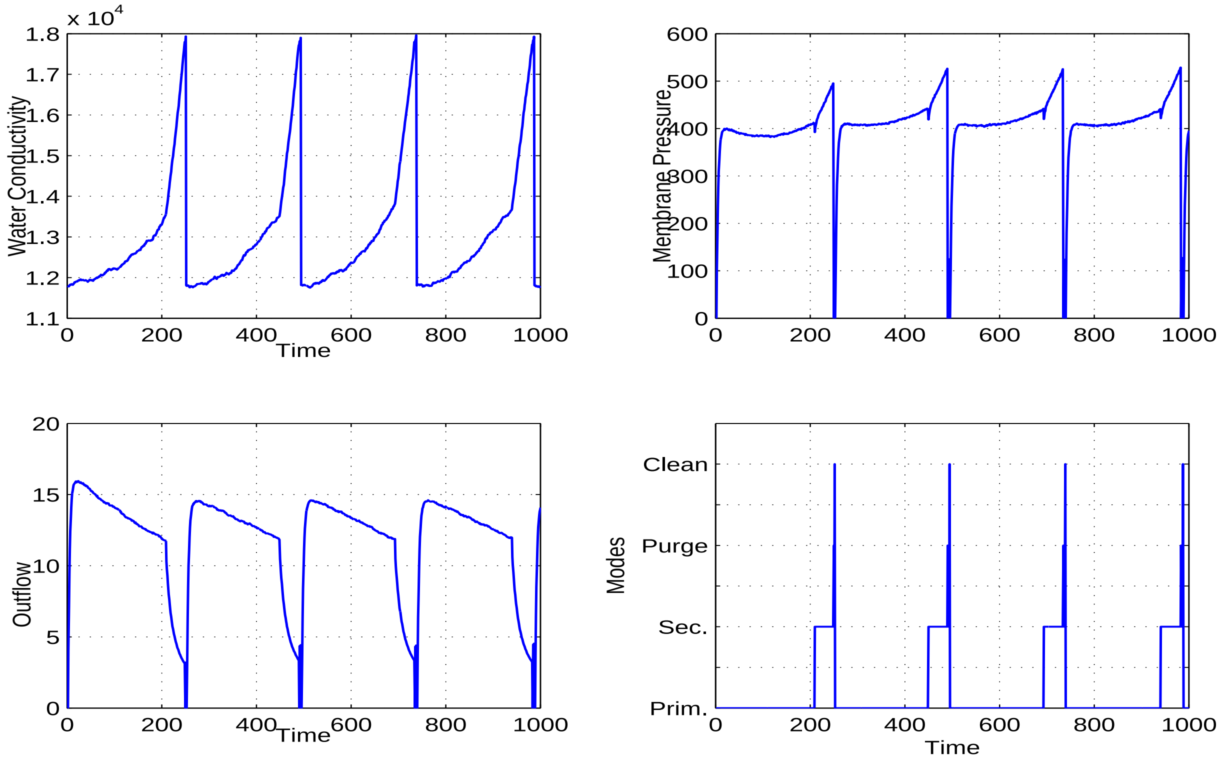

Figure 6: System performance under the online control

5.1 Fault Detection and Isolation

Our model-based approach to fault detection and isolation (FDI) combines robust tracking of nominal system behavior using extended Kalman filter techniques [7], 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 [5]. 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 [19]. An automaton model is employed to switch system models when mode changes occur[12].

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. Table 1 presents the comprehensive diagnosis results for a set of faults in the RO system. The fault magnitudes were chosen to ensure detection (after some delay). For each scenario, the qualitative fault isolation scheme reduced the initial candidate set considerably, and parameter estimation converged to the correct fault candidate. The estimated parameter values were also quite acceptable for all scenarios.

Fault	$t - t_f$	Step	Symbolic	Candidate set + parameter estimation
R_{ep}^+, 35% t_f : 20000	88	0	$e37 : (-, \cdot)$	C_c^+, C_{memb}^+ I_{fp}^+, I_{ep}^+, R_{brine}^-, TF^+, R_{pipe}^-, R_{memb}^-, C_k^+, R_{fp}^+, R_{ep}^+, GY^-
	640	1	$f25 : (-, \cdot)$	I_{fp}^+, I_{ep}^+, R_{brine}^-, TF^+, R_{fp}^+, R_{ep}^+, GY^-
	720	2	$e1 : (-, \cdot)$	I_{ep}^+, R_{brine}^-, R_{ep}^+, GY^-
	960	3	$e37 : (-, -)$	R_{brine}^-, R_{ep}^+
	4640	4	$e35 : (-, \cdot)$	R_{ep}^+
				parameter estimation: R_{ep}^+ changed by 0.374
GY^-, 5% t_f : 18000	200	0	$e37 : (-, \cdot)$	C_c^+, C_{memb}^+, I_{fp}^+, I_{ep}^+, R_{brine}^-, TF^+, R_{pipe}^-, R_{memb}^-, C_k^+, R_{fp}^+, R_{ep}^+, GY^-
	880	1	$f25 : (-, \cdot)$	I_{fp}^+, I_{ep}^+, R_{brine}^-, TF^+, R_{fp}^+, R_{ep}^+, GY^-
	1240	2	$e1 : (-, \cdot)$	I_{ep}^+, R_{brine}^-, R_{ep}^+, GY^-
	1960	3	$e35 : (-, \cdot)$	I_{ep}^+, R_{ep}^+, GY^-
				parameter estimation: GY^- changed by 0.934

Table 1: Comprehensive FDI/diagnosis results for selected faults in the RO system.

5.2 Fault Adaptation

The online control approach can accommodate possible changes in the system parameters that may occur as a result of a fault or parameter changes in time-varying systems. The online control is usually robust enough to accommodate the consequence of fault for certain limited time. Robustness of the controller is essential in managing faults in practical system as most fault detection and isolation systems requires certain time to isolate faults.

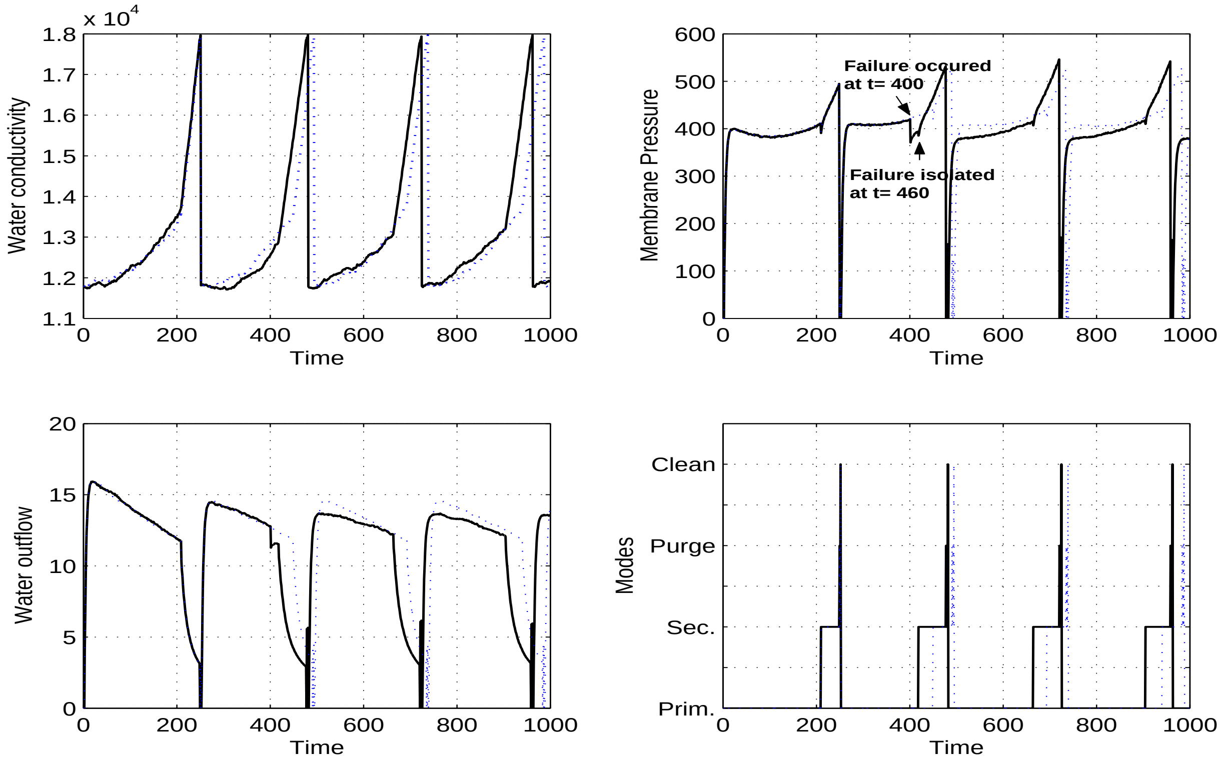

Figure 7: System performance under online control with pipe blocking failure

Figure 7 shows the behavior of the system under online control in the presence of fault. A blocking in a pipe (resulting in 35% increases its resistance) is introduced at time $t = 400$ min and was isolated at time $t = 430$ sec (see Table 1). The online controller managed to compensate for the fault by increasing the time of the primary mode. The overall average utility in this case was only 0.93% less than the utility in the non-faulty situation. In the above figure, the original system output (no failure) is shown in dotted line for comparison.

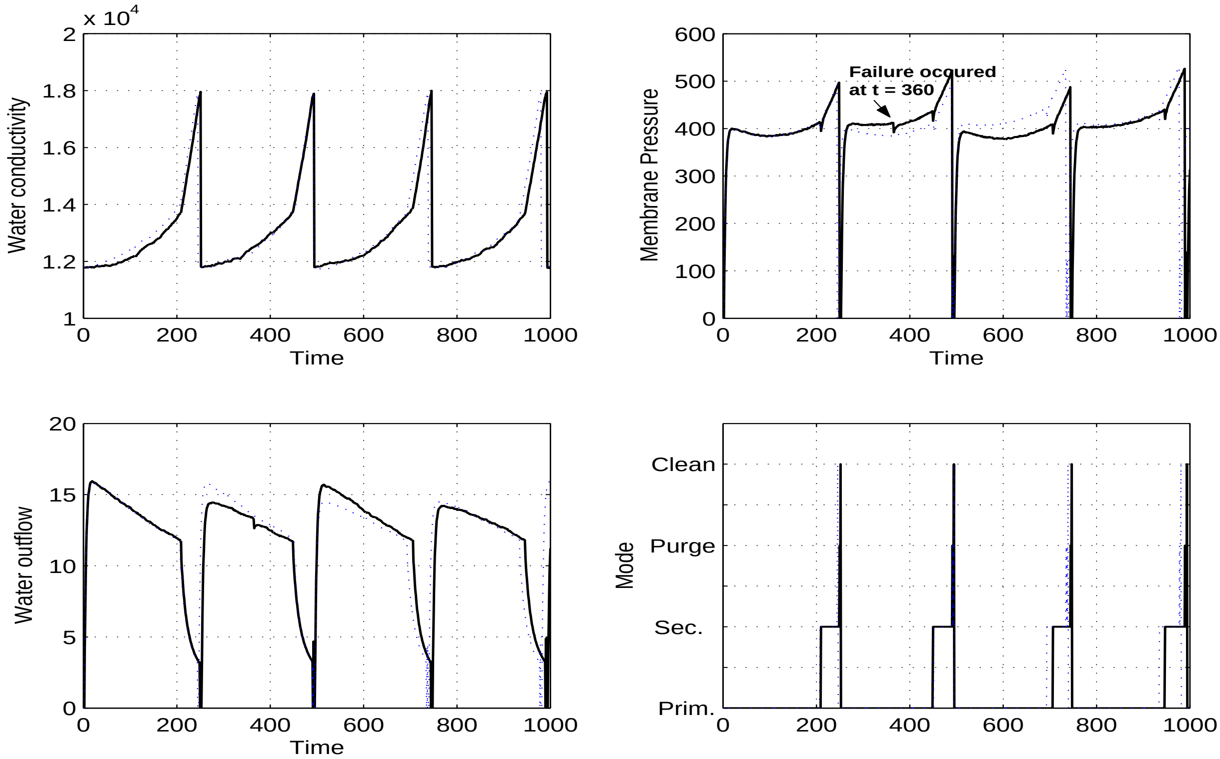

Figure 8: System performance under the online control with pump failure

Figure 8 shows the behavior of the system in the presence of under online control in the presence of fault. A fault in the feeding pump (5% reduction in the out flow) is introduced at time $t = 360$ sec and was isolated at time $t = 400$ sec.

6 Conclusions

Advanced life support systems must be operated in an resource-efficient fashion to maximize their lifetimes while satisfying certain performance requirements. We have proposed an online control approach to efficiently manage the system resources and maximize the system utility. The controller uses finite control set to adjust the system performance over a set of possible modes. We have described the system model, formulated the resource management problem, and derived the corresponding controller. Finally, the proposed controller was evaluated via detailed experiments using similar to simulated data from NASA.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[2] P. Antsaklis, editor. *Special Issue on Hybrid Systems*, volume 88 of *Proceedings of the IEEE*, July 2000.

[3] P. Antsaklis and A. Nerode, editors. *Special Issue on Hybrid Control Systems*, volume 43 of *IEEE Trans. Autom. Control*, April 1998.

[4] G. Biswas, E. Manders, J. Ramirez, N. Mahadevan, and S. Abdelwahed. Online model-based diagnosis to support autonomous operation of the advanced life support systems. *submitted to Habitation, an An International Journal for Human Support Research*, 2004.

[5] G. Biswas, Gy. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, and G. Karsai. A robust method for hybrid diagnosis of complex systems. In *5th IFAC Symp. SAFEPROCESS*, Washington, DC, 2003.

[6] P. Bonasso, D. Kortenkamp, and C. Thronesbery. Intelligent control of a water recovery system: Three years in the trenches. *AI Magazine*, pages 19–43, 2003.

[7] K. Brammer and G. Siffling. *Kalman-Bucy Filters.* Artec House, Norwood MA, 1989.

[8] S. L. Chung, S. Lafortune, and F. Lin. Limited lookahead policies in supervisory control of discrete event systems. *IEEE Trans. Autom. Control*, 37(12):1921–1935, December 1992.

[9] J. Cury, B. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Trans. Autom. Control*, 43(4):564568, 1998.

[10] Duffield, Hanford, and Lange. Advanced life support requirements document. Technical Report Tech. Rep. JSC-38571, Rev. B, NASA-Lyndon B.Johnson Space Center, Houston, TX, September 2002.

[11] R. J. Evans and A. V. Savkin, editors. *Special Issue on Hybrid Control Systems*, volume 38 of *Systems and Control Letters*, 1999.

[12] 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.

[13] E. Kerrigan, A. Bemporad, D. Mignone, M. Morari, and J. Maciejowski. Multiobjective prioritisation and reconfiguration for the control of constrained hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1694–1698, 2000.

[14] 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*, number 2034 in LNCS, pages 333–346. Springer Verlag, 2001.

[15] X. D. Koutsoukos, P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon. Supervisory control of hybrid systems. *Proc. IEEE*, 88(7):1026–1049, July 2000.

[16] J. Lunze. Qualitative modeling of linear dynamical systems with quantised state measurements. *Automatica*, 30(3):417–431, 1994.

[17] M. Morari and J. Lee. Model predictive control: Past, present and future. *Computers and Chemical Engineering*, 23:667–682, 1999.

[18] A. Morse, C. Pantelides, S. Sastry, and J. Schumacher, editors. *Special Issue on Hybrid Systems*, volume 35 of *Automatica*, 1999.

[19] P. J. Mosterman and G. Biswas. A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute*, 334B(6), 1997.

[20] S. Qin and T. Badgewell. An overview of industrial model predictive control technology. *Chemical Process Control*, 93(316):232–256, 1997.

[21] J. Raisch and S. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Trans. Autom. Control*, 43(4):568–573, 1998.

[22] O. Slupphaug and B. A. Foss. Model predictive control for a class of hybrid systems. In *Proceedings European Control Conference*, Brussels, Belgium, 1997.