2005-01-2965

Hierarchical Online Control Design for Autonomous Resource Management in Advanced Life Support Systems

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

ABSTRACT

This paper presents a distributed, hierarchical control scheme for autonomous resource management in complex embedded systems that can handle dynamic changes in resource constraints and operational requirements. The developed hierarchical control structure handles the interactions between subsystem and system-level controllers. A global coordinator at the root of the hierarchy ensures resource requirements for the duration of the mission are not violated. We have applied this approach to design a three-tier hierarchical controller for the operation of a lunar habitat that includes a number of interacting life support components.

INTRODUCTION

The increasing complexity of engineering systems and their use in safety- and mission-critical applications has imposed strict requirements on their reliability, robustness, and availability. Achieving reliable and efficient performance will depend on the system's ability for online monitoring of processes to determine their performance, and to respond to changing conditions in a manner that important functionalities are not degraded

In this paper, we present a control-based approach for resource management in engineering systems. The proposed approach uses a utility-based optimization scheme to maintain the best possible performance from the individual subsystems. The methodology developed targets a class of hybrid dynamic systems that have finite control sets [2]. The underlying model, referred to as a switching 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 challenging and computationally complex. Considerable research has been dedicated recently to the study of hybrid system dynamics [8].

The complex nonlinearities of switched systems and the underlying finite control set makes it hard to apply traditional optimal control techniques, and in most cases a closed expression for a feedback control map cannot be established. The situation is further complicated by the fact that in practical systems the model is usually imprecise and system parameters may change during runtime due to component failures and degradations.

To address the above challenges, we propose a limited-lookahead online control policy for handling set-point specifications for switched systems. The controller directs the system toward a desired state from a given set of initial states, and then ensures that the system remains within a neighborhood of the desired state. The controller achieves this using a search technique that expands a limited set of successor states, and then chooses the action that will drive the system closer to the desired state based on a given distance metric. The control strategy proposed in this paper is conceptually similar to model predictive control [3,,10], where a limited time forecast of the process behavior at each state is optimized according to a given cost function over the set of inputs. Also, related to this work is the limited lookahead supervision of discrete event systems [4].

This paper extends our earlier work on centralized supervisory control [5] to distributed systems. To avoid the complexity of distributed systems, resource management is handled using a hierarchical control structure, where the top-level controller manages the interaction between system-level units and maintains global operational requirements using extended forecasting of environmental conditions and inputs. The lowest level is composed of a set of local control units that optimize the performance of the individual subsystems of the system, taking into account the resource constraints and desired output specified by the higher-level controllers. We implement a control scheme, where predefined set-point specifications for system operation are used to derive optimizing utility functions for the subsystem controllers.

The control scheme is applied to a two subsystem testbeds designed to evaluate subsystem performance for the proposed NASA lunar habitat [6] with its corresponding Advanced Life Support (ALS) System. The ALS itself comprises of a number of systems with interacting control loops, such as the fluid flow loop, the energy management loop, the bio-regeneration and gas transfer loop, and the chemical production loop. These loops cover multiple physical (energy) domains and operating regimes, and operate at multiple time scales. One of the key tasks in controller design is to build models of all of the systems that captured their continuous and discrete dynamics.

THE LUNAR HABITAT

A part of the lunar habitat, showing the crew chamber (this can house up to four astronauts) and two primary components of the ALS system, the ARS and the WRS, are illustrated in Fig. 1. We also include a Power generation system. Critical requirements for such systems are that they consume low power, minimize the use of consumable resources, and run in a fully autonomous closed loop for long periods of time. The rest of this section briefly describes the four systems and the biological crew model.

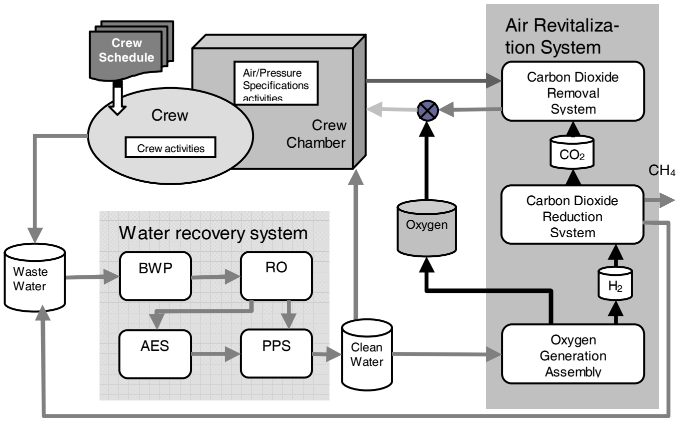

Crew Schedule
Air/Pressure Specifications activities
Crew
Crew activities
Crew Chamber
Air Revitalization System
Carbon Dioxide Removal System
CO_2
Carbon Dioxide Reduction System
CH_4
H_2
Oxygen
Oxygen Generation Assembly
Water recovery system
Waste Water
BWP
RO
AES
PPS
Clean Water

Figure 1: The Advanced Life Support (ALS) system

THE WATER RECOVERY SYSTEM - This system recycles urine and wastewater into potable water. The WRS, shown in Fig. 2, is comprised of the Biological Waste Processor (BWP) that removes organic compounds including ammonia, the Reverse Osmosis (RO) system that removes particulate matter after the BWP, the Air Evaporation System (AES) that purifies the 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 of concentrated brine that is passed to the AES from the RO subsystem.

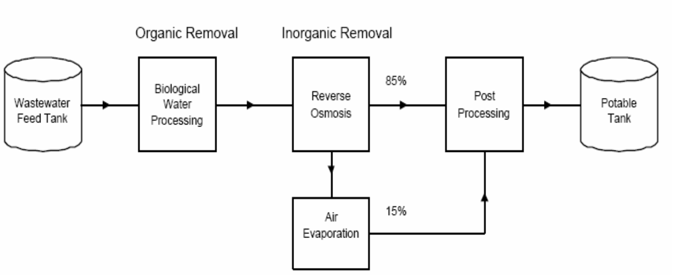


Figure 2: The Water Recovery System

The RO 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 pushes it through a cylindrical membrane that acts like a molecular sieve at high speed. The clean water permeate is passed on to the PPS.

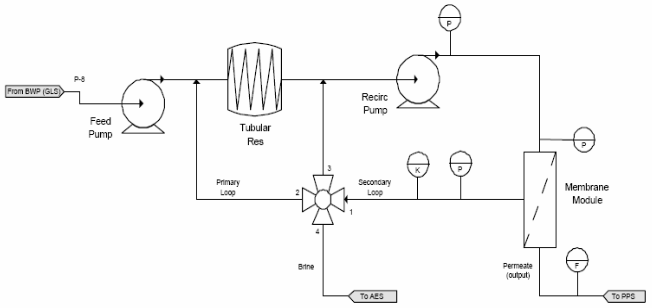


Figure 3: RO system schematic

The RO is designed to go through six modes of operation. 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 is circulated faster in the smaller inner loop with the recirculation pump, to push it 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 GLS, 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.

The AES subsystem contains a reservoir to collect the brine. The brine is absorbed onto a wick and evaporated using hot air. 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 AIR REVITALIZATION SYSTEM (ARS) - The purpose of this subsystem is to replenish the oxygen that the crew consumes, and remove excess carbon dioxide before the air is circulated back to the crew chamber. This task is performed by the Carbon Dioxide Removal

Assembly (CDRA). A second task is to recover the oxygen from the carbon dioxide exhaled by the crew. This is done in two steps. First, hydrogen and oxygen are generated by electrolysis of water using an Oxygen Generation Assembly (OGA). The oxygen goes into a storage tank, and the hydrogen is fed into a reactor (CRS) to reduce the carbon dioxide to water and methane. In the current configuration, the water is sent back to the WRS for purification, and the methane is vented. Fig. 4 shows the main components of the ARS systems and their interaction with other ALS subsystems.

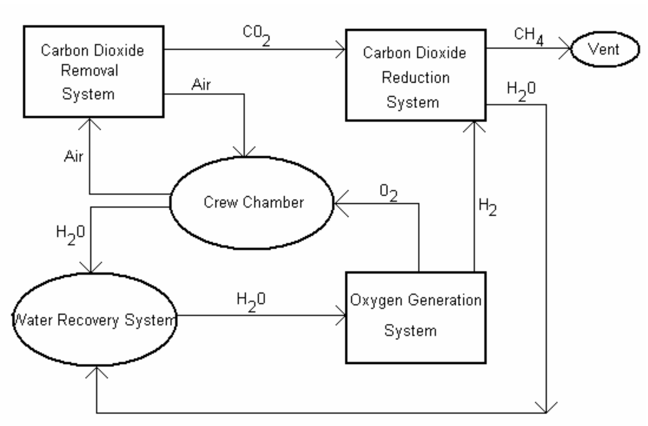


Figure 4: Components of the ARS

The CDRA subsystem uses an adsorption-based device known as a “four-bed molecular sieve” to remove CO_2 exhaled by the crew. There are two primary operations in this subsystem. At any time two of the beds are adsorbing CO_2, while the other two are releasing adsorbed CO_2 as they are heated. When the adsorbing beds become saturated, they are switched to the desorption mode, and the two desorbing beds are used for adsorption. This cycle occurs several times a day. A compressor takes the desorbed CO_2 and stores it in a tank before it is passed on to the reduction system.

The CRS uses a Sabatier reactor with a catalyst to react the CO_2 and H_2. Optimal reaction performance is maintained by controlling the temperature, pressure, and the molar ratio of the gases.

THE CREW HABITAT – This is the crew living and working quarters. The goal of the controller is to maintain the air quality (29% oxygen with nitrogen as the dilutent gas) and temperature in the habitat. We assume the crew consumes O_2, H_2O, and food, and the habitat provides these resources, while removing waste water and solid wastes. The lunar mission [11] assumes that the chamber is occupied by a team of four crew members. A biological model for a typical crew member determines the amount of resources consumed by the crew while performing different activities.

During the mission the crew engage in a pre-defined set of activities, which includes a two-shift schedule, where each schedule involves 8 hours of work, 8 hours of sleep, and the remaining time divided into eating, exercising, maintenance and leisure activity. In general, the crew can either be in the habitat or outside on an EVA (Extra Vehicular Activity) mission. The difference between the main habitat and EVA environment is that the main crew habitat is in the ALS loop, whereas resources produced/consumed in EVA are considered losses from the system.

THE POWER SYSTEM – A simplistic generation and storage model patterned after International Space Station (ISS) technology is employed. An array of solar cells generates the required energy to sustain all of the ALS systems and provide the thermal energy to keep the crew chamber at $298^{o}K$, while also generating excess energy that is stored in Ni-Cd batteries for use during the night (dark) periods when no power can be generated by the solar array. The day-night cycle on the lunar surface is assumed to be 28 earth days with 14 days of sunlight and 14 days of night.

MODELING THE ALS SYSTEM

Building models at the right level of detail is a critical first step in the success of a model-based fault-adaptive control scheme. Our approach is to build physics-based models that capture subsystem dynamics, and use parameter estimation techniques to identify the model parameters so that the model matches observed system behavior. The model of the RO system was derived by decomposing the system into three principal domains of operation. The mechanical and fluid domains are the primary energy domains that define the flow behavior in the system. However, the effect of time-varying impurities in the water on the flow process is accounted for by explicitly modeling the fluid conductivity domain and its interactions with the flow process. The model for the AES consists of three domains: hydraulic, pneumatic and thermal. The hydraulic domain models 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 the AES loop.

The CDRA system involves complex spatial-temporal dynamics. In this work, we build a simplified lumped parameter model in MatLab/Simulink, with multiple lumps to capture the spatial dynamics. The input parameters for CDRA include system pressure, cycle time, airflow rate, temperature, and the inlet CO_2 level. The key outputs for this unit are CO_2 concentration and flow rate. The CRS (Sabatier) model simulates the behavior of one primary reactor zone and two secondary reactor zones. For this subsystem, the input parameters are: system pressure, temperatures and inlet H_2/ CO_2 Molar Ratio. The key simulated steady state output parameters are at the end mass (percentages) of each component involved in the reaction which show the conversion situation of CO_2 and H_2.

The crew habitat model encapsulates activity scheduling, resource utilization based on that activity level, and the interaction between the resource consumption/production and the environment where the crew member resides. It includes the main crew habitat and the EVA habitat. In the developed model, each crew activity has an intensity level associated with it. This intensity level is mapped on a heart-rate value and in turn the heart-rate value is used to compute the oxygen consumption and CO_2 production. This approach mimics that taken in the BioSim simulation engine [11]. The habitat model has the amount of each gas as a state variable. The state variables are updated by the gas flow through the habitat that is imposed by the ARS, and the gases consumed/produced by the crew. Similarly, both the potable water consumed and waste-water produced are coupled to the WRS system.

The power generation model is implemented as a dynamic process with constant production rate during the day and no power generation at night. Excess power generated during the day cycle charges up the Ni-Cd cells to their maximum capacity. At night, the batteries discharge linearly.

ONLINE CONTROL OF SWITICHING SYSTEMS

Applying control techniques for resource management requires a suitable model of the underlying system that captures the relationship between the system variables, and the control inputs. In this section, we introduce the limited lookahead control approach, the underlying form of the mathematical models, and the extended hierarchical control structure.

SWITCHING HYBRID SYSTEMS - The control approach proposed in this paper targets a special class of hybrid systems in which the controlled input takes on a finite set of values. The state space equation describing the continuous dynamics of this class of systems is:

$$x(k+1) = f(x(k), u(k), \lambda(k)),$$

where k is the time index, $x(k) \in X \subseteq \Re^n$ is the sampled form of the continuous state vector, $u(k) \in U \subseteq \Re^m$ is the discrete valued input vector, and $\lambda(k) \in \Re^r$ is the environment input at time k. The set U is finite. For example, in the RO system, the recirculation pump may operate at one of four speeds.

In practical systems, environmental inputs are typically uncontrollable and unobservable. For example, the flow rate of dirty water into the WRS system depends on a variety of factors that cannot be (or are not) explicitly modeled in our system. However, in most situations, they may be predictable within predefined bounds. Forecasting techniques, such as the Box-Jenkins ARIMA modeling approach [13] and Kalman filters [12] may be used to predict variations in the environmental inputs and conditions. In both approaches, a prediction model is devised through analysis or simulation of relevant parameters of the underlying system environment. Formally, a prediction model for the environment input λ will have the form.

$$\underline{\lambda}(k+1) = \theta(\Lambda(k), a(k))$$

where $\underline{\lambda}(k+1)$ is the estimated value of the environment input, $\Lambda(k)$ is the set of all previous measured values of environment inputs, and $a(k)$ are the estimation parameters. The estimator, in general, updates $a(k)$ at each time step in order to minimize the estimation error $e(k) = ||\underline{\lambda}(k) - \lambda(k)||$. Given that the current value of the environment input cannot be measured until the next sampling instance, the next state cannot be computed precisely. Instead, we can estimate of the next state based on the following equation.

$$\underline{x}(k+1) = f(x(k), u(k), \underline{\lambda}(k))$$

In many situations, $a(k)$ is bounded. The bounds are obtained through simulation or analysis of the underlying environment.

REQUIREMENT SPECIFICATIONS – Any system designed for a particular purpose must achieve specific objectives, and, at the same time not violate resource constraints and interactions with the environment. For example, the WRS may be required to produce certain amount of clean, potable water per day to meet the needs of astronauts on a long mission. In other situations, the ratio of gray water (for plant use) to potable water (for human consumption) may follow a specified pattern over different parts of a mission. In general, *cost optimization* can be used to optimize a given performance measure represented as a function of system states and inputs. A weighted norm of the form,

$$J(k) = \| x(k) - r(k) \|_Q + \| u(k) \|_R$$

is typically used as a performance function in which a weighted sum of relevant variables is computed, with the weights reflecting their contribution to the system utility and operation cost. In this paper, we consider the case when $r(k)$ is a point, say x^*. This form of specification is generally called a set-point specification.

Operational requirements for distributed computation systems may involve additional strict and soft constraints on the system variables and control inputs. In general, strict constraints can be expressed as a feasible domain for the composite space of a set of system variables and control inputs, and they can be represented in general by a set of inequalities, $h(x, u') \leq 0$ and a restricted set of inputs $U' \subseteq U$.

The optimizing component to safety control is introduced ias a 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 is a simple weighted functions of the form $V_i(P_i) = w_i\, P_i$.

THE LOOKAHEAD CONTROL APPROACH – The objective of the control algorithm is to achieve the desired set-point specifications, maintain the system stable at the desired value, and optimize the given performance function while satisfying a set of constraints. In general, there is no closed-form solution for such non-linear optimization problems. We propose an approximate solution approach based on limited-lookahead. To this end, the control problem is defined as

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=k+1}^{k+N} \left\| x(i) - x^* \right\|_Q + \left\| u(i) \right\|_R \\ \text{Subject to} \quad & x(i+1) = f(x(i), u(i), \lambda(i)) \\ & h(x(i)) \le 0, u \in U' \end{aligned}$$

Given that the control set is finite, the above control problem is clearly solvable. In general, it may not produce the optimal trajectory solution for the original control problem. However, in many practical situations, the main concern is the feasibility of the online controller, namely, its ability to drive the system towards the desired operation domain ”quickly" and maintain it in this region under typical variations in the system or environmental conditions. The feasibility of the online control approach is discussed briefly in the next section.

Based on the above settings, the online controller aims to satisfy the desired set-point specifications by continuously monitoring the current system state and selecting the inputs that best satisfy them. In this setting, the controller is simply considered an agent that applies a given sequence of events in order to achieve a certain objective. The controller explores only a limited forward horizon, *N* time steps, in the system state space. The controller then selects the trajectory that minimizes the cost function while satisfying the constraints, *h*. The input at the first look-ahead step in this trajectory is chosen as the next input, and this process is repeated at each subsequent time step.

The above control policy takes into account the effect of possible variations in the environment inputs by requiring that the selected input satisfy a worst case scenario constraint with respect the estimation bounds. Figure 5 shows the components of the look-ahead controller. Relevant parameters of the operating environment are estimated and used by the system model to forecast future behavior over a limited look-ahead horizon. The controller optimizes the system behavior by selecting the best control inputs while making sure the specified constraints are not violated.

CONTROL STABILITY - Giving the limited exploration nature of the online algorithm, it is important to obtain a measure of feasibility to determine if the online control will be able to reach the desired region in a finite time. The controller is feasible for a given set-point and tolerance domain containing the set-point if it can drive the system (in finite time) from any initial state in a given operation region to a neighborhood (contained in the tolerance domain) of the set-point and maintain the system within this neighborhood. In [1], the feasibility of the proposed online control approach is formulated as a joint containability and attraction problem. A novel computational procedure based on nonlinear programming is presented to compute a containable region.

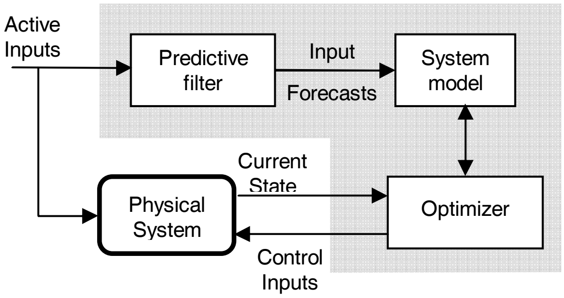


Figure 5: The online control structure

MULTILEVEL ONLINE CONTROL

This section extends the control approach to distributed systems comprising of multiple systems, each having its own specification. Typically, these systems must interact to achieve a desired global objective. This suggests a multi-level distributed control structure where systems have independent controllers, and system interaction is managed by a global controller that addresses overall requirement specifications. Figure 6 shows the multi-level control structure.

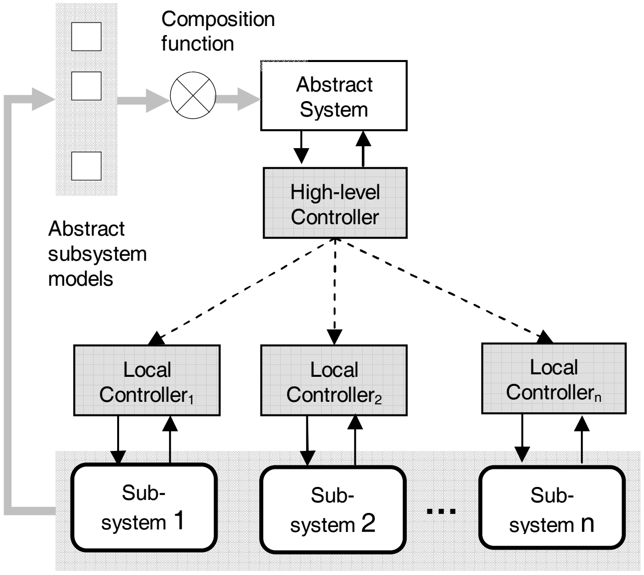


Figure 6: A Multi-level Control Structure

Since a detailed behavioral model of the underlying distributed system may be very complex, the global controller uses an abstract (simplified) model to describe the composite behavior of the system components that is relevant to the overall requirements and operational constraints. The abstract model uses a set of global variables that are related by the input-output interactions between the individual systems. Moreover, the global controller's decisions are based on aggregate behaviors, which are determined over longer time frames compared to the individual systems. We assume here that these time-frames are harmonically related, i.e., $T_g = MT_l$ where T_g and T_l are the global and local time steps, respectively. Consequently, for a set of systems, the global state vector, $y(k_g)$, at global time instance k can be represented as,

$$y(k_g) = \Omega(x^1(k_l,M), \ldots , x^L(k_l,M)),$$

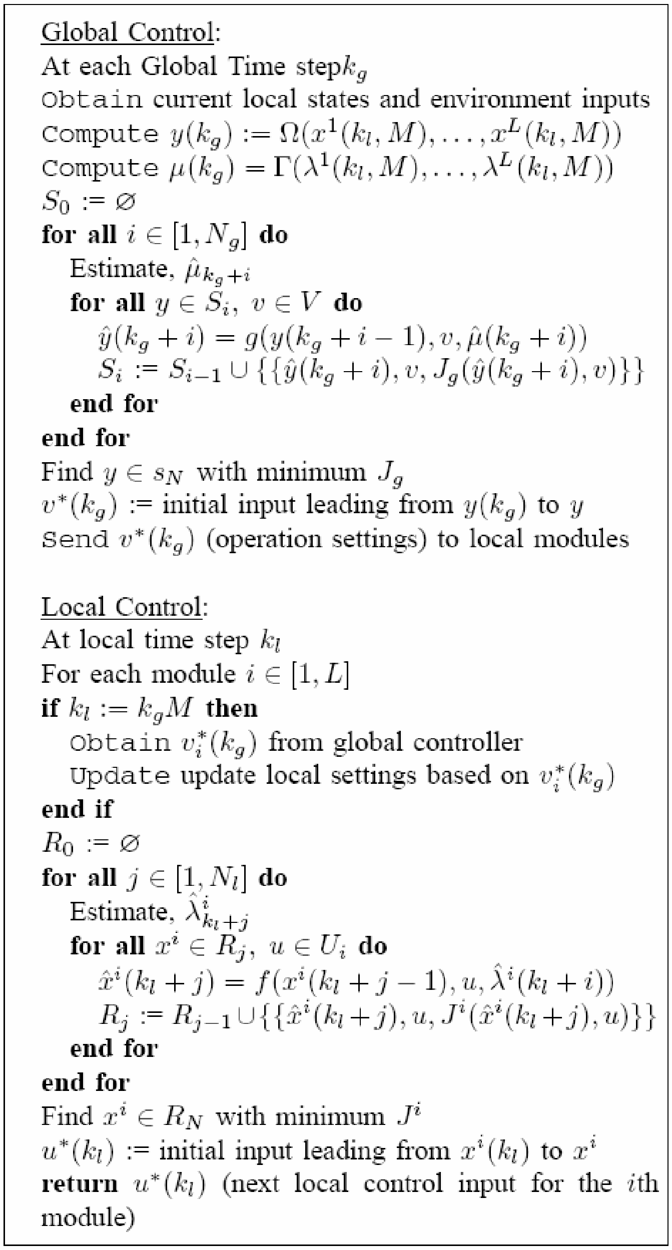
where $k_l = M\ k_g$ and M is a positive integer, and $x^j(k_l,M) = \{x^j(k_l - M + 1), \ldots, x^j(k_l)\}$ is the set of states for the *i*th system, and Ω is the abstraction map defining the relationship between the global state vector y at the global time instance k_g and the local state variables over the local time instances spanning $[k_{g-1}\ k_g]$. Similarly, we can define the global environment inputs, $\mu(k_g)$, for the global controller at time k_g as an aggregation of the local environment inputs $\lambda^j(k_l)$, over the global time frame, namely, $\mu(k_g) = \Gamma(\lambda^1(k_l;M), \ldots , \lambda^L(k_l;M))$, where $\lambda^j(k_l;M) = \{\lambda^j(k_l-M +1), \ldots , \lambda^j\ (k_l)\}$. The global model is represented by

$$y(k + 1) = g(y(k), v(k), \mu\ (k)),$$

where $v(k) \in V$ and V is the set of global control inputs, which represents a set of local control settings for the local modules. We assume that the set of such local control settings that can be manipulated by the system controller is finite. The map g defines how the global state variables respond to relevant changes in environment inputs with respect to the global control inputs. This abstract behavior can be obtained analytically (in case of simple local dynamics) or more likely through simulation where the arguments are the input set V and a quantized approximation of the domain of μ. It is typical that an initial model is built through simulation and then adjusted through continuous observation of the actual system behavior. The objective of the system controller is to minimize a given cost function $J_g(y,v)$ over the operation span of the system. We also assume that J_g takes the form of the set point specification described earlier for local controllers. Based on the assumption that global specification is of higher priority than local ones, the outcome of the system controller is communicated to local modules. The local controllers then try to optimize the performance of the local components while ensuring that conditions imposed by the system controller are not violated. To summarize, in the hierarchical control scheme, the system controller performs the following functions:

- Forecasts long-term trends of the environment and based on the abstract system model examines the effect on the overall performance of the system.
- Optimizes the system performance by changing the operational settings of local module, or the distribution of loads and resources among these modules
- Obtains performance feedback from local modules, which then used to identify the current global state.

Figure 7 shows the distributed online control algorithm. The algorithm is composed of two main procedures: global and local. The global control is invoked at each global time instant k_g, it accepts the current local states and environment inputs and returns the best operation setting for local modules $v^*(k_g)$ to apply at k_g.



```
Global Control:
At each Global Time step$k_g$
Obtain current local states and environment inputs
Compute $y(k_g) := \Omega(x^1(k_l, M), \ldots, x^L(k_l, M))$
Compute $\mu(k_g) = \Gamma(\lambda^1(k_l, M), \ldots, \lambda^L(k_l, M))$
$S_0 := \varnothing$
for all $i \in [1, N_g]$ do
   Estimate, $\hat{\mu}_{k_g+i}$
   for all $y \in S_i,\ v \in V$ do
      $\hat{y}(k_g + i) = g(y(k_g + i - 1), v, \hat{\mu}(k_g + i))$
      $S_i := S_{i-1} \cup \{\{\hat{y}(k_g + i), v, J_g(\hat{y}(k_g + i), v)\}\}$
   end for
end for
Find $y \in s_N$ with minimum $J_g$
$v^*(k_g)$ := initial input leading from $y(k_g)$ to $y$
Send $v^*(k_g)$ (operation settings) to local modules

Local Control:
At local time step $k_l$
For each module $i \in [1, L]$
if $k_l := k_g M$ then
   Obtain $v_i^*(k_g)$ from global controller
   Update update local settings based on $v_i^*(k_g)$
end if
$R_0 := \varnothing$
for all $j \in [1, N_l]$ do
   Estimate, $\hat{\lambda}^i_{k_l+j}$
   for all $x^i \in R_j,\ u \in U_i$ do
      $\hat{x}^i(k_l + j) = f(x^i(k_l + j - 1), u, \hat{\lambda}^i(k_l + i))$
      $R_j := R_{j-1} \cup \{\{\hat{x}^i(k_l + j), u, J^i(\hat{x}^i(k_l + j), u)\}\}$
   end for
end for
Find $x^i \in R_N$ with minimum $J^i$
$u^*(k_l)$ := initial input leading from $x^i(k_l)$ to $x^i$
return $u^*(k_l)$ (next local control input for the $i$th module)
```

Figure 7: The multilevel control algorithm

THE ALS CONTROL STRUCTURE

We have applied the hierarchical control approach presented in the previous section to the ALS system. The developed computational structure, shown in Figure 9, is a tree of controllers arranged in three levels each addressing different aspects of the overall system behavior. In the following we will describe the control structure at different levels and discuss the relationship between different control modules and the system components.

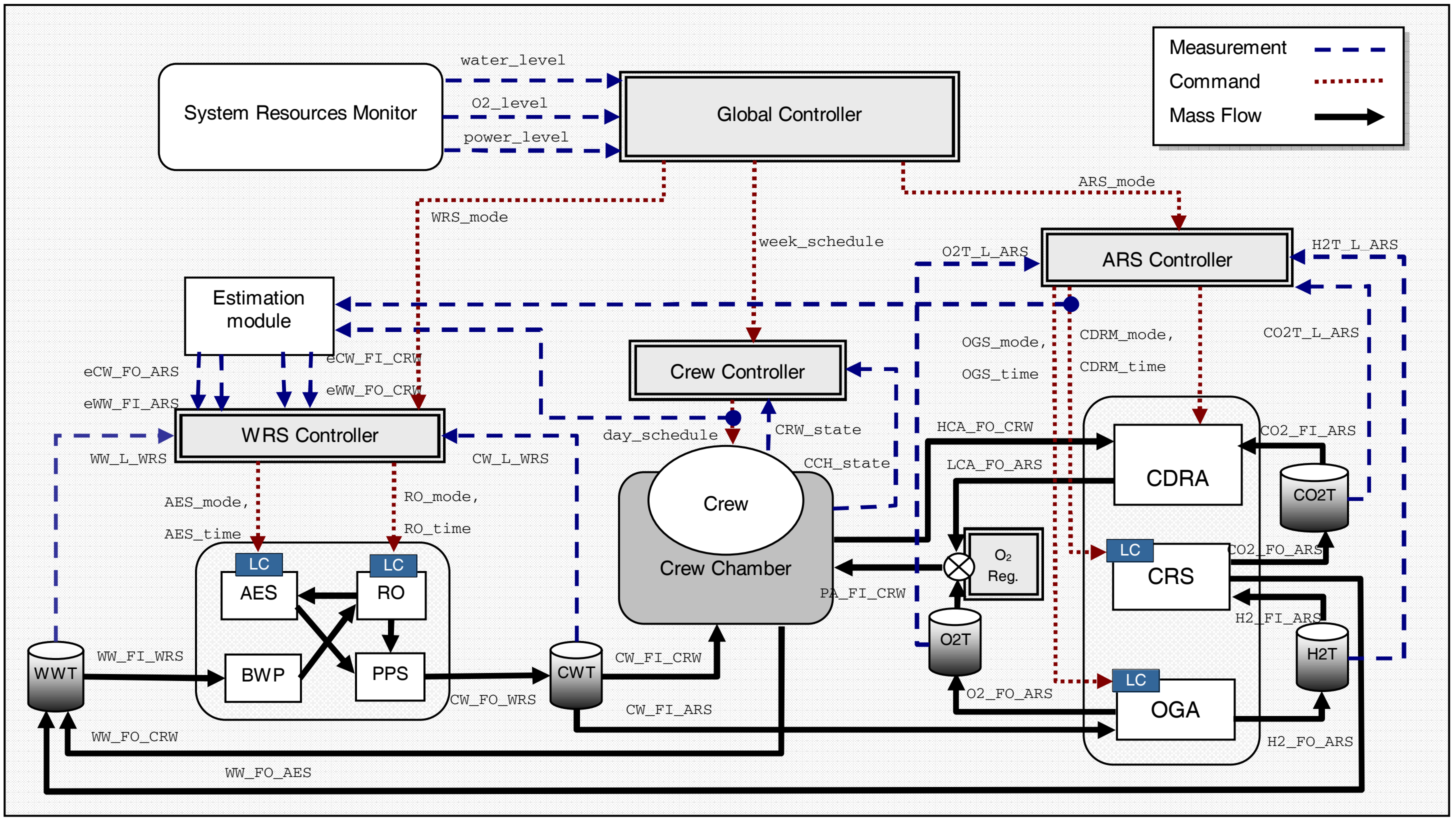


Figure 9: The ALS Control Structure

LOCAL CONTROLLERS – At the first level of the control structure a set of local controllers manage the individual subsystems of the systems. Each subsystem has an individual optimizing controller, which does not directly interact with other subsystems. Interactions are handled by the system controller and by placing physical buffers between subsystems. The local controllers receive commands in the form of input-output requirements and resource constraints from the system-level controllers at periodic intervals. These requirements and constraints are specified as modes of operation, which have accompanying control objectives and system parameters, such as control input restrictions.

SYSTEM CONTROLLERS – These controllers are responsible for managing the interactions between controlled subsystems by managing their interactions through the intervening buffers, and by distributing resources, such as power, in a manner that all of the subsystems can produce their desired output in an efficient way. The controllers at this level use an abstract model defining the average behavior of the subsystems and how they affect the level at the connected buffers. Note that system controllers target only the buffer quantities and not the dynamics of subsystem operation. The main objective is to maintain buffer levels based on mass flow predictions considering the crew schedules.

Like local controllers, the function of the system-level controllers is influenced by the global controller. The global controller chooses from among a finite number of options to adjust the behavior of these controllers. The habitat controller features the WRS, ALS, and crew chamber controllers.

GLOBAL CONTROLLER – This controller ensures the mission success by balancing resource consumption with the available level of resources. The daily crew schedule is computed based on the constraints and performance goals determined by the global controller. Specifically, the global controller allocates resources for maintenance, exercise, and EVA activities.

EXPERIMENTS

We present a set of simulation experiments based on a 90 day challenge scenario that was developed at NASA JSC to illustrate multi-level online control of the system.

The scenario assumes four crew members on the habitat. The crew members use 9 liters of water and 1 kg of O_2 on the average, while producing 1 kg of CO_2 per day.

The effect of the global control schedule is shown in Figure 10. This figure shows the evolution of levels of waste water tank and potable water tank for the 90 days. The maximum value of waste tank is never more than 25 liters. This not only demonstrates the controller's effectiveness, but also provides a good reference for the selection of water buffer size.

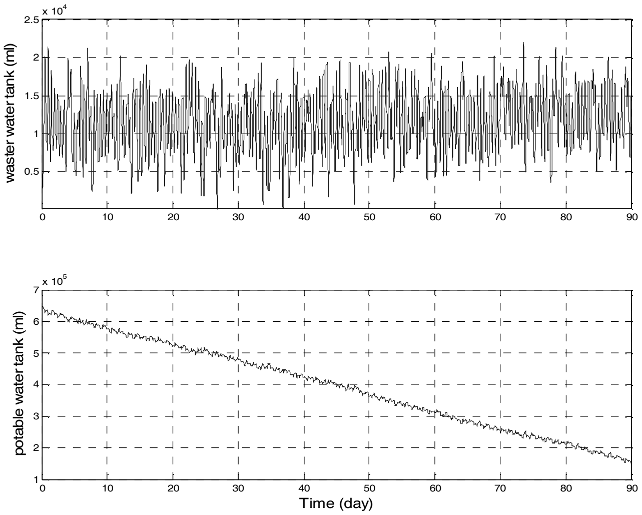


Figure 10: The level of waste and potable water tanks during the 90 mission

As discussed in the previous section, controllers at the subsystem-level use the following information to compute their control actions: (i) estimated crew activities; e.g., for the WRS, according to the daily schedule, the waste water created per hour and potable water spent per hour can be estimated, and for ARS, the CO_2 level can be estimated; (ii) average production and treatment rates and corresponding average power consumed for each preset modes; and (iii) values of relevant buffers. The derived performance index for the WRS is

$$J = \sum_{i=k}^{k+N-1} c_1 \left\| f^{c}_{m_i} - \lambda^{c}_{i} + L^{c}_{i} \right\| + c_2 \left\| - f^{w}_{m_i} + \lambda^{w}_{i} + L^{w}_{i} \right\| + c_3 \left\| P_{m_i} \right\|$$

where $f^{c}_{m_i}$, $f^{w}_{m_i}$ and $P^{ro}_{m_i}$ are average production and waste water treatment flow rates and power consumed for the preset modes of WRS, λ^{c}_{i} and λ^{w}_{i} are the potable-water-consumption and the waste-water-creation flow rates by the crews, respectively, N is the number of horizons, and c_1, c_2 and c_3 are the weights for relevant terms. Here c_1 is negative as more clean water is better than less, and the other two are positive as J, the cost function, has to be minimized. An important design issue can be addressed by the choice of the cost function, i.e., pick a cost function that minimizes buffer size of waste water and potable water while keeping performance at required levels. Figure 11 shows the modes of the WRS, estimated average waste water creation rate and potable water consummation rate over 20 cycles each of which lasts approximately 4 hours. Mode 1 represents the mode in which the WRS is off and collecting waste water. Modes 2, 3 and 4, respectively represent the low, the normal and the high production modes of the WRS when the AES is off. Mode 5 represents the mode in which the AES is on and the BWP and the RO are both off.

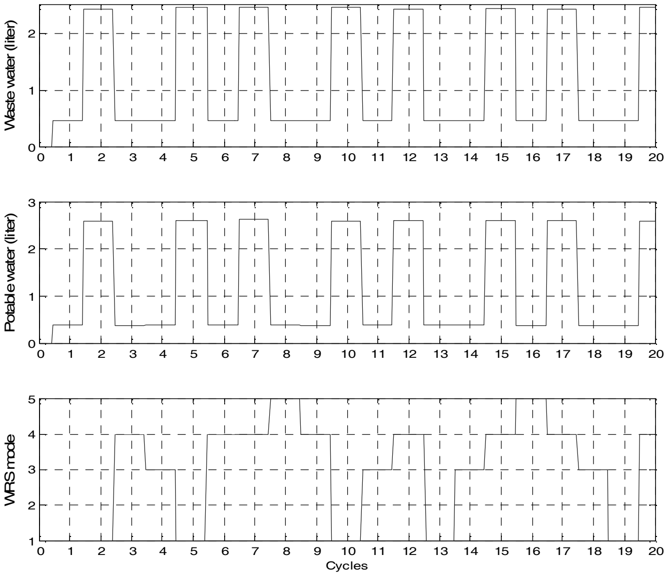


Figure 11: Modes and the associated water levels in the WRS

At the WRS subsystem level, the BWP and AES operate only in the on/off mode. The utility associated for both modes are obtained through simulation. The RO has 6 modes, and the system operates according to the utility function:

$$V=\sum_{i=k}^{k+N} a_K[K_i/K_{\max}]+a_f[f_i/f_{\max}]+a_{S_v}S_v+a_p[P_i/P_{\max}]$$

[a_K, a_f, a_S, a_P] are the parameters of the utility function, and they take on different values for the different modes of operation. Those modes are determined by the average flow rate and power consumed, which according to the potable water production level, are partitioned to {high, middle, low, off} modes. K_i is conductivity value, f_i is production outflow rate, S_v is a binary switching value and P_i is power value. Fig. 12 shows the changes to key variables of WRS subsystems. Note that the utility of the WRS is the composition of the utility associated with its local subsystems. Such composite utility clearly depends on the local mode of each subsystem. The composite utility at each composite mode of the WRS is obtained through simulation. Also, it is important to note that, the choice of different operation modes is a design choice depending on the system requirements. The proposed multi-level control approach can accommodate any form of operational modes as long as the number of modes are finite and the correspondence with the overall system objective can be established (analytically or by simulation.)

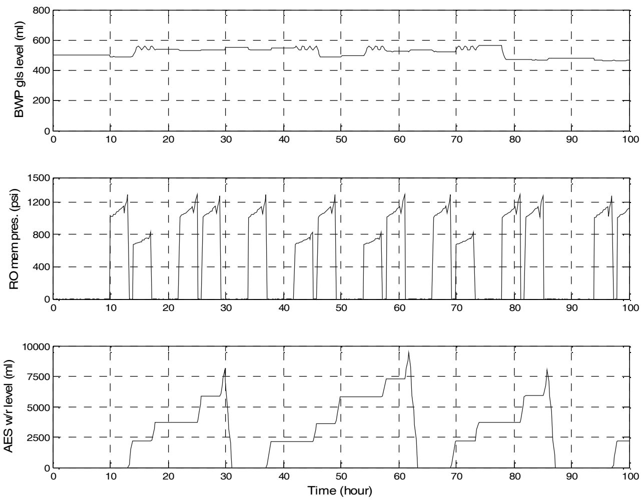


Figure 12: Controlled local variables in the WRS

A similar technique was applied to design the controllers for the ARS subsystems. Fig. 13 shows changes to three key variables in the CDRA adsorption process over four hours. Dashed lines and solid lines represent the input and output variables, respectively. The adsorption cycle time is roughly 2.4 hours.

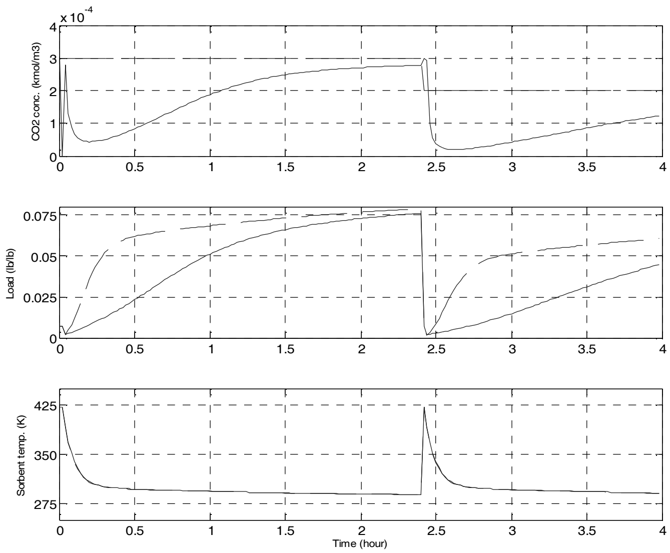


Figure 13: Controlled variables in the ARS system

Fig. 14 shows the O_2 and CO_2 buffer levels. With the controller they can be maintained within a preset range.

CONCLUSIONS

In this paper, we demonstrated a successful scheme for hierarchical control of complex embedded closed-loop systems to ensure resource constraints are not violated over long-duration missions. Resource management at the global level is successfully combined with optimizing individual subsystem behavior at the local level. This work shows a lot of promise in achieving smaller buffer sizes, as well as autonomy, robustness and reliability through model-driven fault adaptive control.

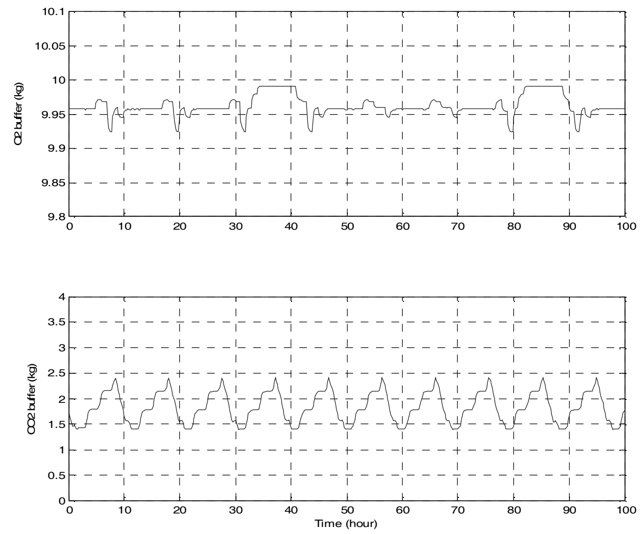


Figure 14: O_2 and CO_2 levels in the ALS buffers.

ACKNOWLEDGMENTS

This work is supported in part by grants from the NASA-ALS program (Contract: NCC 9-159). We acknowledge the help provided by the members of the ALS group at NASA Johnson Space Center, primarily, Pete Bonasso, David Kortenkamp, and Molly Anderson.

REFERENCES

1. R. Su, S. Abdelwahed, and S. Neema, “Computing finitely reachable containable region for switching system," IEE Proceedings-Control Theory and Applications, 2005, to be published.
2. S. Abdelwahed, G. Karsai, and G. Biswas, “Online safety control of a class of hybrid systems," in Proc. 41st Conference on Decision and Control, 2002.
3. S. Qin and T. Badgewell, “An overview of industrial model predictive control technology,” Chemical Process Control, vol. 93, no. 316, 1997.
4. S. L. Chung, S. Lafortune, and F. Lin, “Limited lookahead policies in supervisory control of discrete event systems,” IEEE Trans. Autom. Control, vol. 37, no. 12, pp. 1921–1935, Dec. 1992.
5. S. Abdelwahed, N. Kandasamy, and S. Neema, “Online control for self-management in computing systems,” in IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.
6. B. Duffield, A. J. Hanford, and K. Lange, “Advanced life support requirements document,” Tech. Rep. JSC-38571, Rev. B, NASA-Lyndon B. Johnson Space Center Houston, TX, Sept. 2002.
7. P. Bonasso, D. Kortenkamp, and C. Thronesbery, “Intelligent control of a water recovery system: Three years in the trenches,” AI Magazine, 2003.

8. P. Antsaklis and A. Nerode, editors. Special Issue on Hybrid Control Systems, volume 43 of IEEE Trans. Automatic Control, April 1998.
9. J. Lunze. Qualitative modeling of linear dynamical systems with quantized state measurements. Automatica, 30(3):417-431, 1994.
10. M. Morari and J. Lee. Model predictive control: Past, present and future. Computers and Chemical Engineering, 23:667-682, 1999.
11. 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.
12. K. Brammer and G. Siffling, Kalman-Bucy Filters, Artec House, Norwood MA, 1989.
13. 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.