A Hybrid Control System Design and Implementation for a Three Tank Testbed

Jian Wu, *Student Member, IEEE*, Gautam Biswas, *Senior Member, IEEE*, Sherif Abdelwahed, *Senior Member, IEEE* and Eric Manders, *Member, IEEE*

***Abstract* — This paper discusses methodologies for designing limited lookahead supervisory controllers for a class of embedded systems that can be modeled as switching hybrid system (SHS). In the limited lookahead control approach, a limited forward horizon of possible behaviors is explored at each time step. The controller uses a cost function to determine the immediate action that will move the system toward the "best" state on the horizon. We discuss the controller design and implementation for a three-tank system test-bed with distributed sensor and actuation units. A set of real-time fault adaptive control experiments demonstrate the effectiveness of the approach.**

I. Introduction

The mixed continuous and discrete behavior of hybrid systems limits the applicability of traditional optimal control and supervisory control techniques in designing controllers for these systems. Several approaches have been proposed to mitigate the effects of combined continuous and discrete behavior in controller design. For example, abstraction techniques that involve discretizing the state space, have been developed to reduce the complexity of the hybrid models represented as automata while preserving features of the original model relevant to the analysis/control objectives [1]. Supervisory control design with abstracted hybrid system models has been investigated in [2, 3]. Efficient control synthesis for reachability specifications through mode switching has been presented in [4].

In this paper, we develop a limited lookahead control approach [5, 6] for hybrid systems, and apply it to design and implement a controller for a three tank system testbed in the Embedded and Hybrid Systems Laboratory at Vanderbilt University, USA. The limited lookahead controller operates by continuously monitoring the current state of the system and selecting the inputs that best satisfy the given specifications while minimizing a cost function or maximizing a utility function. To achieve this, the control input to the system is discretized, and the plant model is represented as a switching hybrid system (SHS). To avoid computational intractability and to maintain real time operation for complex systems, the controller explores only a limited forward horizon in the system state space at each time step and selects the next event based on optimizing cost and utility function. For example, given a set-point specification, action selection for the next step is based on a distance map that defines how close the generated state is to the desired set point.

Hybrid Bond Graphs [7], a general-purpose hybrid modeling language, are used to derive the three-tank plant model. The control input to the three-tank system is defined by a finite control set that is composed of different settings for valve positions and a discrete set of pump speeds. This allows the system to be modeled as a switching hybrid system.

We perform real-time experiments with the implemented supervisory controller on the three-tank test-bed to study its robustness when faults occur in the system. The faults analyzed include temporary leaks in the tank and permanent change in the tank capacity. The online experimental results demonstrate the effectiveness and robustness of the hybrid control approach.

The paper is organized as follows. Section II addresses the limited lookahead supervisory control approach. It introduces the switching hybrid system (SHS) framework, and the procedures for selecting the best control action at any step. Section III gives an overview of the three-tank system set up. It includes a description of the distributed measurement and control (DMC) system, and a brief introduction to the IEEE 1451.2 standard. Section IV details the methodology used to derive the mathematical model of the system. Section V describes the real time experiments and the results obtained. Section VI concludes the paper.

II. Limited Lookahead Supervisory Control Method

The limited lookahead supervisory control approach uses a model predictive control architecture that is applied to a class of hybrid systems, called switching hybrid system (SHS). The controller design has two primary components: (i) a performance measure that captures the system behavior specification using either a cost function or a utility function, and (ii) a methodology to select the best control action at any step based on the performance measure.

A. *Switching Hybrid System*

The limited lookahead control approach targets a special class of hybrid systems in which the controlled input to the

Manuscript received Feb. 18, 2005.

Jian Wu is with Dept. of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37203, USA. (phone: 615-343-4641; fax: 615-343-7440; e-mail: jian.wu@vanderbilt.edu.)

Drs. Gautam Biswas, Sherif Abdelwahed and Eric Manders are with Dept. of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37203, USA. (emails: gautam.biswas, sheriff.abdelwahed, eric.manders @vanderbilt.edu.)

system is characterized by a finite control set. The following discrete-time form of the state space equations is used to describe the continuous dynamics of this class of hybrid systems:

$$x(k+1) = \Phi(x(k), u(k)) \qquad (1)$$

where k is the time index, $x(k) \subseteq \Re^n$ is the sampled form of the continuous state vector at time k, and $u(k) \in U \subseteq \Re^m$ is the discrete valued input vector at time k. We use X and U to denote the state space and the finite input set for the system, respectively. For each mode of system operation, $u \in U$, the function Φ 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 of SHS 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 discretized and take values from a finite 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.

B. *Requirement Specification*

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 pre-specified levels, or to follow a certain pattern (trajectory). The controller for the three-tank system we describe in this paper is required to maintain pre-defined fluid level in the three tanks.

The other type of specification, referred to as *performance specifications*, is used to optimize the system performance by minimizing or maximizing a given set of performance measures that can include power consumption and system utilization. The performance measure is a function of the state, input, and output variables, and typically defined as a weighted norm of these variables. The control goals are expressed as optimizing utility functions. Examples can be found in [6]. The objective of the designed controller is to achieve the desired specs in "reasonable" time, keep the system stable at the desired value, and optimize the given performance functions.

C. *Limited Lookahead Control Algorithm*

The limited lookahead controller tries to meet the defined specifications by continuously monitoring the current state of the system, and selecting the input that drives the system to the specifications. Action selection is performed at every time step. In addition, the controller is required to keep the system stable within the domain that satisfies the specs.

The limited lookahead control approach is illustrated in Fig. 1. Consider 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 in $\Re^n$. An alternative form can be expressed as:

$$\min J = \sum_{i=k}^{k+N} c^T \| x\ (i) - x_s \|_Q \qquad (2)$$

where $x(i)$ at time step i is defined by Equation 1.

N is the prediction horizon, Q is a proper norm, and c is a vector representing the relative importance (preference) of each variable in the cost function.

The control algorithm starts by constructing the tree of all possible future modes from the current mode up to a pre-specified depth (i.e., a finite forward horizon). For realizing this, first all the modes are numbered. A predefined adjacency matrix with rows and columns labeled by numbered modes captures all possible mode transitions. A 1 in position (i, j) implies the system can transition from mode i to mode j. A 0 implies this transition is not possible. Finally, based on the adjacency matrix, a trajectory matrix with N+1 state vectors all starting from the current mode are derived. This set of trajectories represents the tree in Fig. 1. This matrix can be updated at every time step if necessary.

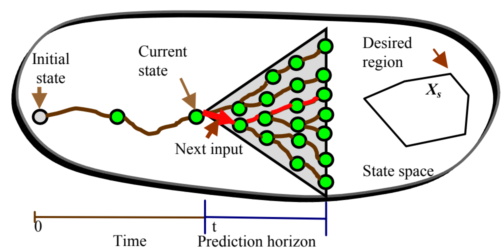

Figure 1: The limited lookahead control

The exploration procedure identifies the set of states that best satisfy the given specification as discussed above. A state x_{min} is then chosen from this set based on the certain optimality criterion. The chosen state is then traced back to the current state and the input leading to x_{min} is used for the next step. Table 1 summarizes the limited look- ahead control algorithm.

Table 1: Limited lookahead control procedure

Step 0: Generation of two part model.

i) Discrete-time SHS model (Equation 1), and

ii)Adjacency matrix A.

Steps 1-4: Limited lookahead control.

Step 1: Trajectories (Tree) generation.

i) Derive Trajectory matrix T using adjacency matrix A, and pre-specified depth of prediction horizon N.

ii)*Refine* T. All trajectories that violate system constraints are removed. A T now contains only legal trajectories.

Step 2: Compute cost for each vector in T using cost function J (Equation 2) and current state x_c.

Step 3: Select next control action. The optimal trajectory vector (give best cost value) in T is identified. The first action in this trajectory is selected.

Step 4: Update x_c with new measurement. Go to Step 2 if system mode does not change, otherwise go to Step 1.

III. THE THREE-TANK FLUID SYSTEM SETUP

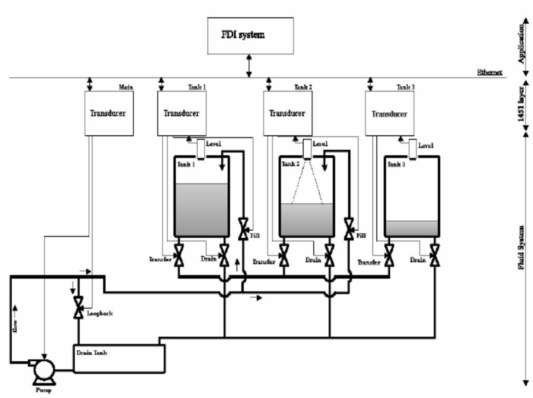

Figure 2: Schematic of the three tank fluid

The fluid system, shown in Fig. 2, consists of three identical tanks, connected in a series configuration. A closed loop fluid system is created with an additional drain tank, a variable speed electric pump, and connecting pipes. Transfer of fluid through different paths is made possible by turning a set of solenoid operated valves on and off. The fluid level in each tank is measured with an ultrasonic level sensor.

We adopt the *IEEE 1451.2* [8] specification to define a standard measurement and control interface for distributed embedded systems that can be used in a variety of applications. This standard defines a smart transducer interface for the sensors and actuators in a controlled physical system. A transducer channel is deemed "smart" in this context because (i) it is described by a machine-readable data format, (ii) the control and data associated with the channel are digital, and (iii) triggering, status, and control are provided to support the proper channel functions.

This specification provides a standard platform for developing distributed measurement and control systems (DMC), and defining programming and hardware protocols for a set of networked smart transducers. The transducer includes an onboard processor, which can run user-specified code, and this typically includes routines for analyzing measurements and generating actuator signals.

IV. DERIVING THE THREE-TANK SYSTEM MODEL

We employ a model-based approach to realize and fault adaptive control of embedded systems, like the three-tank system. To achieve this, the first step is to derive the mathematical model of the hybrid system that includes continuous behavior interspersed with discrete mode changes.

In our work, we operate the pump in three discrete modes: (i) off, (ii) low-speed, and (iii) high-speed. This determines the rate at which tanks 1 and 2 can be filled. Overall the set of control commands that govern system behavior can be defined as a finite set that includes the valve settings (on and off) and the three pump speeds. This makes it easy for us to build the three-tank system model as a switching hybrid system (SHS) as described in Section II. We use a hybrid bond graph approach to derive the mathematical model of the system in different modes of operation. The three tank system exhibits nonlinear behavior, therefore, deriving the mathematical model is a nontrivial process.

A. Hybrid Bond Graph of Three Tank Testbed

Bond Graphs (BGs) [10] are a powerful energy-based formalism to model continuous dynamics of physical systems. Hybrid Bond Graphs (HBGs) [7] extend the BG modeling paradigm to hybrid systems. The central idea in HBGs is to introduce the notion of an idealized switching junction (i.e., switching occurs instantaneously), whose discrete switching behavior is governed by a two state automata that generates on/off signals. Hybrid behavior is described by reconfiguration of the bond graph structure to model different modes of system operation. Each mode is defined by the on/off state of all the controlled junctions. State equations for can be derived from the bond graph model in each mode.

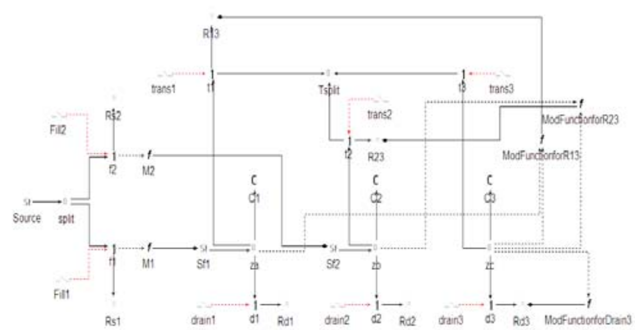

Figure 3: Hybrid Bond Graph of three-tank system

The authors' preferences for HBGs stem not only from its elegance and utility as a modeling tool, but also the fact that this representation facilitates causal diagnosis. For example, the HBG model is employed in *Transcend*, a general qualitative fault diagnosis framework [9]. The hybrid bond graph in Fig. 3 represents the general hybrid model for the three tank testbed setup in the EHS laboratory (Fig. 2). Five types of components are shown in this figure: (i) Several switching signals (switch plus dotted line represent the control signals to the individual valves). (ii) Tanks 1, 2, and 3 are modeled as capacities C_1, C_2 and C_3, respectively. (iii) Pipes are modeled as resistance components. Some pipes are modeled by linear resistances, e.g. 'Rd1,' the drain resistance of the Drain pipe in Tank 1. Others, such as the transfer pipes between pairs of tanks are modeled as nonlinear resistances. The value of the resistance in these cases, are functions of the pressures in the adjacent tanks. (iv) The pump is modeled as an idealized source of flow, and (v) 0 and 1 junctions are idealized components that govern energy transfer among the different parts of the system. The next subsections describe in some detail about the mathematical model of the three tank testbed.

B. The Discrete Modes

The mode of the system, from a modeling perspective, is the union of the modes of each distributed transducer. Creating a model of the system requires capturing every possible combination of transducer modes. This approach results in a single model of the system with many modes, which has several advantages like that it eliminates the need to interface multiple controllers and avoids concurrency and communication issues with multiple models.

There are a total of 8 two-way valves in the system, and since each of these valves can be in an *on* or *off* mode, there are $2^{8} = 256$ different configurations that can be generated for the system.

In order to make the implementation work less complex and more efficient, we discard some of the configurations. For example, we never drain fluid from Tanks 1 and 2, therefore, their drain valves are always off, and the drain valve for Tank 3 is always on. Therefore, we have 5 valves that switch during system operation, and pump can be in one of three states. Furthermore, we disallow some actions, e.g., transferring fluid between Tank 1 and Tank 2, and simultaneously transferring from Tank 1 and Tank 2 to Tank 3. Taking into account these reductions, our hybrid three-tank model has 19 discrete modes of operation, and this includes a 'Rest' mode in which all valves and the pump are off. As a next step, the adjacency matrix defined in Section II-C is constructed to define all valid mode transition for the experiment.

C. The Continuous Dynamics Modeling

As discussed, staring with the HBG model in Fig. 3 the BG model in each continuous mode of operation can be determined. We demonstrate the methodology employed to derive the parameters of the continuous model in one mode of operation. The parameters for the other modes are derived in a similar manner. More details can be found in [11].

Example: <u>Generating the model for the mode of transferring fluid between tank 1 and tank 3,</u>

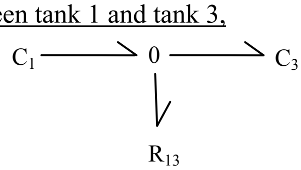

Figure 4: BG for fluid transfer between Tanks 1 and 3

To isolate the transfer resistance between Tank1 and Tank 3, i.e., R_{13}, the pump and all other valves in the system were turned off, and the system is in the mode which fluid transfers from Tank 1 to Tank 3. Fig. 4 is the simplified bond graph, extracted from the HBG model (Fig. 3). Equations 3 and 4 define the fluid transfer dynamics between Tank 1 and Tank 3. p_1 is the pressure (height) at the bottom of Tank 1, and p_3 is the pressure (height) at the bottom of Tank 3.

$$\frac{dp_1}{dt} = \frac{p_3 - p_1}{C_1 R_{13}} \tag{3}$$

$$\frac{dp_3}{dt} = \frac{p_1 - p_3}{C_3 R_{13}} \tag{4}$$

The tank capacities are assumed to be known; they are computed from the tank dimensions and tank geometry. The tank heights are measured variables, and they can be converted to pressure values. In this case, R_{13} is the only unknown parameter, and the equation can be rearranged and solved for resistance, shown in Equation 5 below:

$$R_{13} = \frac{p_3 - p_1}{C_1\, dp_1 / dt} \tag{5}$$

Using the data of Tank 1 and Tank 3 heights and the least squares fit module available in Matlab, a best-fit polynomial function for the resistance value as a function of the pressures is derived. The derived least squares function becomes the modulating function, *ModFunctionR13,* (see Fig. 3), that governs the value of resistance R_{13}. This function takes two inputs, one is the effort value (pressure) from the zero junction connected to Tank 1, i.e., C_1, and the second one is the zero junction connected to Tank 3, i.e., C_3. Equation 6 shows the modulating function of R_{13} from Tank 1 to Tank3.

$$R_{13} = 4.265E8(\Delta h)^3 - 4.091E8(\Delta h)^2 + 1.732E8(\Delta h) + 3.498E6$$
$$where\ \Delta h = (p_1 - p_3)/\rho g, \tag{6}$$

ρ is fluid density and g is gravitational constant.

D. Summary

The heights of Tanks 1, 2 and 3 are chosen as state variables $\{x\}$. For deriving the dynamic equation of system behavior, the input is represented by a finite control set $\{u\}$ which determines operation mode and is composed of the valves positions (0 or 1) and pump speeds (0, low speed, or high speed). These form the basis for deriving the state space switching hybrid system equations for the three tank testbed.

The example below shows the derived SHS model for a particular mode of system operation – Tank 2 is filling at low speed and Tank 1 transferring fluid to Tank 3. For this mode, $u = \{V_{Fill1} = 0,\ V_{Fill2} = 1,\ V_{trans1} = 1,\ V_{trans2} = 0,\ V_{trans3} = 1,\ pump = low\ (4)\}^T$, and $Flow_{low}$ is the corresponding constant flow provided by the pump with low speed. The discrete time state equations for this mode are

$$h_1(k+1) = h_1(k) + \Delta T((h_3(k) - h_1(k))/C_1/R_{13}(k)))$$
$$h_2(k+1) = h_2(k) + \Delta T(Flow_{low}/A_2) \tag{7}$$
$$h_3(k+1) = h_3(k) + \Delta T((h_1(k) - h_3(k))/C_3/R_{13}(k) - h_3(k)/C_3/R_{d3}(k))$$

where sampling time $\Delta T = 1s$, $Flow_{low} = 0.0001833 m^3/s$, cross sectional areas $A_1 = A_2 = A_3 = A = 0.01549 m^2$, and $C_1 = C_2 = C_3 = \dfrac{A}{\rho g} = 1.570E - 6 m^5/N$, and modulating function for $R_{d3} = -1.028E8 h_3^2 + 4.185E8 h_3 + 2.465E6\ Ns/m^5$, at time step k.

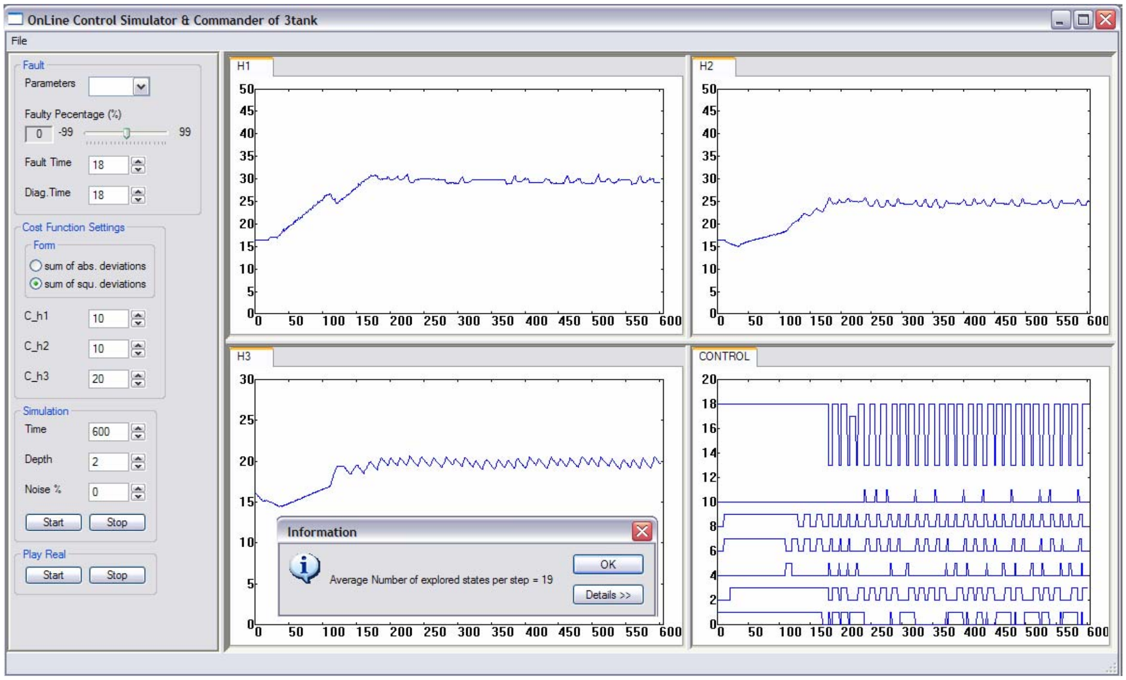

Figure 5: GUI and online observations of heights and control signals for the nominal case

V. Real time experiments

We use the SHS model of the plant to implement the online controller for the three-tank testbed. In this section, we first introduce the online controller software design and then describe three sets of real time experiments that we have conducted on the testbed.

A. Software Design

The software design comprised of two tasks: Setting up the communication between the sensor/actuator modules and controller that run on a desktop PC connected to network, and implementing the controller.

The real-time communication between the three tank testbed and the controller routines running on the desktop are implemented as two C++ functions that realize the underlying network calls and communications with the nodes. They implement the Actuate (Write) module and the Sense (Read) module. The Actuate function uses three parameters: socket address (i.e., an internet address of a transducer node), channel number to transmit the actuate signal, and a value. Likewise, the Sense function reads the sensor data (tank heights).

We implemented the controller software with Python. Python is a scripting language which is basically as powerful as C/C++ language but its execution speed is often slower than corresponding C++ programs. We use SWIG (Simplified Wrapper and Interface Generator) to automatically generate a Python extension module from these two read/write C++ functions. Then, the interface class and 1451.2 library are dynamically loaded by the Python interpreter upon importing the extension module.

The controller software includes three main parts: (i) the limited lookahead control algorithm, (ii) the complete SHS mathematical model, and (iii) the Graphical User Interface for setting experiment parameters and observing the results.

As shown in the left part of Fig. 5, the GUI provides several features to make it convenient for the users to run experiments. The experiment can be chosen to run in two modes: (i) as a simulation run, or (ii) in the real-time experimental mode, where the controller interacts with the three tank testbed. The various functionalities provided allow the user to specify features of experiments, e.g., choose the form of the cost function (1-norm and 2-norm). In the right part of the Fig. 5, there are four partitions each of which will display the three heights and control signals. Once ‘play real’ button is pressed, the system operates as a real-time experiment.

B. Real-time Experiment Results

As mentioned in Section IV, Tank 3 provides the system outflow. One of the primary control objectives is to maintain this outflow at a predetermined set value. Since this outflow is directly a function of the Tank 3 height, the control goal can be interpreted as a requirement to maintain the fluid level of Tank 3 at a preset value. The control task is further extended to ensure that the levels of fluid in the other two feed tanks are maintained at preset values.

Experiment I: Nominal case

The goal is to maintain the levels of Tank 1, Tank 2 and Tank 3 respectively at 30cm, 25cm and 20cm. We want to assign greater importance to maintaining the height of fluid in Tank 3. Therefore, we give it a higher penalty weight of 20, versus the weights of other two heights of 10. 2-norm form of cost function is used and prediction horizon is set to 2. Then, we have $x = \{h1, h2, h3\}^T$, $N = 2$; $x_s = \{30, 25,$

$20\}^T$ *and* $c = \{10, 10, 20\}^T$. We can find the above lookahead depth and coefficients setting in the left part of Fig. 5.

The right part of Fig. 5 shows the results of height evolutions for each tank, and all of the control signals displayed from bottom to top - the valve settings for filling tanks 1 and 2, the transfer valves of tanks 1, 2 and 3, the Rest Mode, and the pump. Those results show that the controller designed is able to maintain the specified fluid heights. The bottom part of Tank 3 display gives the information that, on the average, 19 states are explored at every lookahead time step.

Experiment II: Faulty case – temporary leak of Tank 2

The goal is kept the same as Experiment I. A leak fault is introduced into Tank 2 for the time interval [175, 200], i.e., the intermittent leak starts at time = 175 seconds, and goes away at time = 200 seconds. The system behavior is shown in Fig. 6. Right after the fault happens, the pump is kept on for a longer time in the high-speed mode to compensate the loss of fluid in Tank 2. The Rest Mode of the system is also skipped. Fig. 6 also shows that when a fault occurs, there is a big drift of Tank 2 as well as small drifts in Tank 1 and Tank 3. The controller brings all three tank heights to their set-point values in a very short period of time.

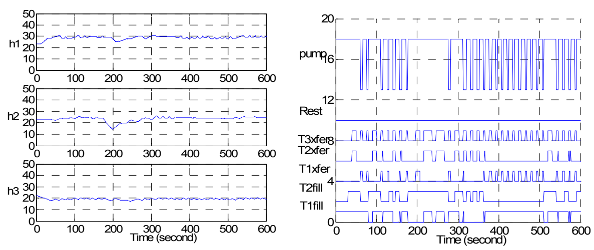

Figure 6: Tank heights and control signals in Experiment II

Experiment III: Faulty case – permanent capacity change of Tank 3

In this experiment, a permanent fault, in the form of a decrease in capacity of Tank 3 is introduced at time = 252 seconds. We do this by dropping an object into Tank 3. At the point of failure, we observe a jump in the Tank 3 height, see Fig. 7. In fact the initial jump magnitude is incorrect, because the height sensor detected the object that was introduced into the tank. After the fault occurrence, the pump quickly decelerates from high speed to low-speed and stays longer in the low-speed state than in the nominal case as a reaction to the accidental increase of the water in Tank 3.

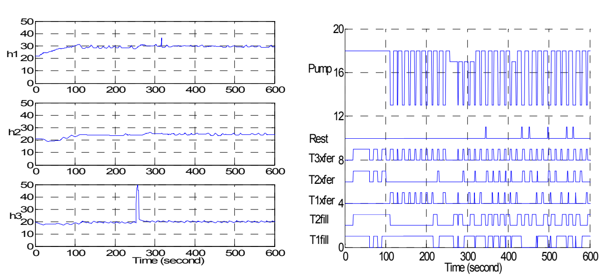

Figure 7: Tank heights and control signals in Exp. III

VI. Conclusions

In this work, we demonstrated the applicability of the limited lookahead supervisory control method for realistic systems. We exploit the distributed measurement and control system available on the three tank testbed in the EHS laboratory at Vanderbilt University to design and implement a fault-adaptive supervisory controller for this system. The first step was to derive the SHS model for the three tank testbed. The interface routines and the online controller code were then implemented. The real-time experimental results demonstrate the practicability and effectiveness of the designed system under both the nominal and faulty conditions. The study of relevant stability analysis and the preliminary results can be found in [12].

The work described in this paper represents the first phase in the development of computationally efficient model-predictive hybrid control schemes that can be applied for both nominal and faulty operations of a physical plant. The next step in this research will focus on combining the FDI technology, specifically speaking, a fault diagnoser with the current control system. We can then establish a rather complete platform for Fault Adaptive Control Technology (FACT).

References

[1] J. Lunze, "Qualitative modeling of linear dynamical systems with quantised state measurements," *Automatica*, vol. 30, pp. 417–431, 1994.

[2] J. Cury, B. Krogh, and T. Niinomi. "Synthesis of supervisory controllers for hybrid systems based on approximating automata," *IEEE Trans. Automatic Control*, vol. 43, pp. 564-- 568, 1998.

[3] X. D. Koutsoukos, P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon. Supervisory control of hybrid systems. *Proc. IEEE*, vol. 88, pp. 1026–1049, July 2000.

[4] 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, pp. 333–346. Springer -Verlag, 2001.

[5] S. Abdelwahed, N. Kandasamy, and Sandeep Neema, "Online control for resource management in computing systems." in *Proc. 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, Toronto, 2004, pp. 368- 375.

[6] S. Abdelwahed, J. Wu and G. Biswas, "Online hierarchical fault adaptive control for advanced life support system," in *Proc. 35th Intern. Conf. Environmental Sys*., Colorado Springs, CO, 2004.

[7] P. J. Mosterman, "A hybrid bond graph modeling paradigm and its application in diagnosis," Ph. D. thesis, Dept. of Electrical Engineering and Computer Science, Vanderbilt University, 1997.

[8] *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators – Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, IEEE Standard 1451.2-1997.

[9] Available at http://macs.vuse.vanderbilt.edu/macsweb.

[10] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *Systems Dynamics: Modeling and Simulation of Mechatronic Systems*, New York: John Wiley & Sons, Inc., 3rd edition, 2000.

[11] J. Lyons, "Distributed monitoring and control and physical system modeling for a laboratory three-tank system," M. S. thesis, Dept. Electrical Engineering, Vanderbilt University, Nashville, TN, 2004.

[12] S. Abdelwahed, R. Su and S. Neema, "A feasible lookahead control for systems with finite control set," accepted in Proc. of IEEE Conf. Contr. Applications, Toronto, Canada, Aug., 2005.