Estimation of Distributed Hybrid Systems Using Particle Filtering Methods

Xenofon Koutsoukos[1], James Kurien[2], and Feng Zhao[2]

[1] Department of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37235, USA
Xenofon.Koutsoukos@vanderbilt.edu
[2] Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304, USA
{jkurien,zhao}@parc.com

Abstract. Networked embedded systems are composed of a large number of components that interact with the physical world via a set of sensors and actuators, have their own computational capabilities, and communicate with each other via a wired or wireless network. Such systems are best modeled by distributed hybrid systems that capture the interaction between the physical and computational components. Monitoring and diagnosis of any dynamical system depend crucially on the ability to estimate the system state given the observations. Estimation for distributed hybrid systems is particularly challenging because it requires keeping track of multiple models and the transitions between them. This paper presents a particle filtering based estimation algorithm for a class of distributed hybrid systems. The hybrid estimation methodology is demonstrated on a cryogenic propulsion system.

1 Introduction

The work in this paper is motivated by existing and emerging applications of networked, embedded systems. Such systems contain a large number of distributed nodes, each of which performs a moderate amount of computation, collaborates with other nodes via a wired or wireless network, and is embedded in the physical world via a set of sensors and actuators. Examples include complex electro-mechanical systems with embedded controllers [13] and smart matter systems [6]. State estimation from the available measurements in such systems presents a number of interesting new challenges. The system dynamics are best described by hybrid models of computation and hybrid estimation techniques are required. The complexity that arises from the distribution of both the physical and computing components must be also addressed. A large amount of computation is potentially available, but it may be partitioned into relatively small, embedded chunks. Communication between nodes is available, but may involve unreliable delivery, power-constrained wireless networks, or large, complex topologies requiring multiple hops to connect two arbitrary nodes.

Estimation of hybrid systems is particularly challenging because keeping track of multiple models and the autonomous transitions between them is computationally very expensive. Simple extension of conventional estimation techniques, like the Kalman filter, leads to algorithms that require tracking of all possible trajectories, and therefore, are exponential in the number of time steps. Approximation by Gaussians is often used to collapse the distributions for each trajectory resulting in poor performance. A related approach to our work based on banks of extended Kalman filters has been presented in [4] where only trajectories with high confidence probability are traced. A methodology using both discrete and continuous observers based on finite state machines and linear systems has been proposed in [1]. Sequential Monte Carlo (or particle filtering) methods can support process densities that contain both continuous and discrete dynamics and have been explored for hybrid diagnosis in [11]. However, autonomous transitions between modes triggered by the continuous dynamics have not been considered. Particle filtering has been applied also for a class of hybrid systems modeled by dynamic Bayesian networks in [7] where the autonomous transitions between discrete states are defined using the so-called softmax conditional probability distributions. A fault modeling and diagnosis approach for hybrid systems based on qualitative representation of the fault hypotheses has been presented in [8]. A Bayesian approach for mode estimation of hybrid systems has been presented in [13] and has been demonstrated for monitoring and diagnosis of electro-mechanical systems. This approach uses continuous measurements and prior from a temporal discrete event model to compute the likelihood functions for the mode transitions.

In this paper, we present a particle filter based estimation algorithm that addresses the challenge of the double-sided interaction between continuous and discrete dynamics in hybrid systems. The algorithm is applicable to a large class of hybrid systems, where the continuous dynamics and the guard conditions can be nonlinear, and the noise can be represented by arbitrary multi-modal distributions. We show how we can estimate autonomous transitions based on complex guard conditions. We also describe how we can improve the performance and robustness of the algorithm by using guard conditions that cover the state space of the system. In particle filters, complex integrals are computed efficiently by approximating the belief state by finitely many samples. General process densities that can represent the interaction between discrete and continuous dynamics in hybrid systems can be used in an efficient manner. Detailed descriptions of particle filtering methods for estimation of dynamical systems can be found in [3]. Our approach is similar to algorithms with mixed-state and automatic model switching that have been successfully applied for tracking of motion boundaries in video images [5, 2]. The centralized estimation algorithm and its application to a two-tank system can be found in [9]. In this paper, we demonstrate the algorithm for the estimation and fault detection of the rocket propulsion example of an experimental NASA vehicle (X34) using simulation results. The hybrid estimation approach presented in this paper is part of a distributed, hybrid diagnostic system that has been developed for the cryogenic propulsion system; details can be found in [10].

The remainder of the paper is organized as follows. The class of distributed models and hybrid estimation problems considered in this paper are presented in Section 2. Section 3 presents the cryogenic propulsion system used to illustrate the approach. Our approach for distributed hybrid system estimation is described in Section 4. The particle filtering algorithm is presented in Section 5. Simulation results for the cryogenic propulsion system are presented in Section 6. Finally, conclusions and future work are discussed in Section 7.

2 Problem Statement

Hybrid systems contain interacting discrete and continuous dynamics. The discrete dynamics are usually described by discrete event models with a finite state space Q. Every discrete state (or mode) q corresponds to a unique differential (difference) equation $\dot{x} = f(q, x)$ that governs the continuous dynamics. The state of the hybrid system is described by $s = (q, x)$. The state can change either by time delay as described by the differential/difference equation or by a transition. Mode transitions $e = (q_1, q_2)$ may occur either upon receiving an external control command or when the continuous state satisfies a guard $x \in G(e)$ that labels the transition. The state may be reset after the occurrence of such a transition according to the reset map $x' = R(e, x)$. Mode transitions that depend on the continuous behavior of the system are called autonomous.

In the hybrid system literature, it is often assumed that the state is directly observable. However, in real-world applications, the state has to be reconstructed from the observations. In this paper, we follow a Bayesian state estimation approach using a discrete-time representation of the system dynamics. The continuous dynamics of the system can be described, using zero-order hold sampling for example, by the discrete-time model

$$\begin{aligned} x_{t+1} &= f_q(x_t, u_t) + \nu_t \\ y_t &= g_q(x_t) + \xi_t \end{aligned}$$

and ν_t and ξ_t denote process and measurement noise respectively. The evolution of the discrete state can be described by the transition function

$$q_{t+1} = \delta(q_t, \sigma_t, x_t)$$

where σ_t denotes events corresponding to the control commands. A discrete transition occurs when either the controller issues an appropriate command or when the continuous state satisfies the guard of the transition. The hybrid estimation problem is to compute the most likely hybrid state $s_t = (q_t, x_t)$ given the observation sequence $Y_t = (y_0, y_1, \ldots, y_t)$, the sequence of continuous control inputs $U_t = (u_0, u_1, \ldots, u_t)$, and the history of control events $(\sigma_1, \sigma_2, \ldots)$ up to time t.

A distributed embedded system consists of multiple components that can be described by interacting hybrid systems. In this paper, we consider a class of distributed hybrid systems where the coupling occurs only through the guards that govern the mode transitions. We model such systems by a collection of subsystems $\{H_n\}, n = 1, \ldots, N$. The state of the n^{th} subsystem is $s^{(n)} = (q^{(n)}, x^{(n)})$. We assume that if there exists coupling between H_n and H_m, it can be described by a guard condition of the form $G(e^{(n)}, x^{(m)})$. Thus, a mode transition $e^{(n)}$ in H_n can be triggered by a condition on the state $x^{(m)}$ of H_m. A cryogenic propulsion system that is modeled by such a distributed hybrid system is presented in Section 3.

In distributed hybrid systems, the state s_t is the aggregate state of all the subsystems $\{s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(N)}\}$. Centralized estimation algorithms are computationally very expensive because they are based on high-dimensional models. They also require high-bandwidth networks since all the remote measurements must be communicated to a centralized location at every time step. Distributed estimation algorithms offer significant computational advantages, especially, because they can exploit the computation that is embedded in several components of the system. The local hybrid estimation problem at subsystem H_n is to compute the most likely state $s_t^{(n)} = (q_t^{(n)}, x_t^{(n)})$ using local or remote observations and control inputs. For the class of systems considered in this paper, estimating the local state $s_t^{(n)} = (q_t^{(n)}, x_t^{(n)})$ only requires knowledge of remote guard conditions $G(e^{(n)}, x^{(m)})$ that affect the local behavior.

3 The Propulsion System Domain

Space launch vehicles that reach Earth orbit do so by carrying large quantities of oxygen which is combined with a fuel and burned to produce thrust. The oxygen is stored in the form of liquid oxygen (LOX) at a temperature several hundred degrees below that of the launch environment. Figure 1 illustrates the LOX venting system for the X-34, an experimental, rocket-powered vehicle designed for NASA. During flight, the tank absorbs heat and the LOX temperature and pressure are increasing. A digital controller is responsible for keeping the LOX pressure in a safe region. When the pneumatic valve is open, the LOX tank can vent to the atmosphere. The vehicle's control system does not directly actuate the pneumatic valve. Instead, the pneumatic valve opens when it is pressurized by the pneumatic system to its left. The pneumatic tank and regulators provide high pressure gas to the solenoid valve. When the control system opens the solenoid valve, the pneumatic valve is pressurized and opens.

The cryogenic propulsion system can be best modeled as a distributed hybrid system consisting of the LOX tank subsystem and the pneumatic subsystem. The dynamics of the LOX tank follow a multi-modal behavior that represents if LOX is boiling to gas oxygen (GOX) or not and if GOX is venting in the atmosphere or not. The continuous dynamics are governed by mass and energy conservation laws. The discrete dynamics include the behavior of the solenoid valve which is controlled by a digital controller that monitors the output pressure of the LOX tank. In the following, we present a dynamical model of the system. The notation including units and constants can be found in Table 1.

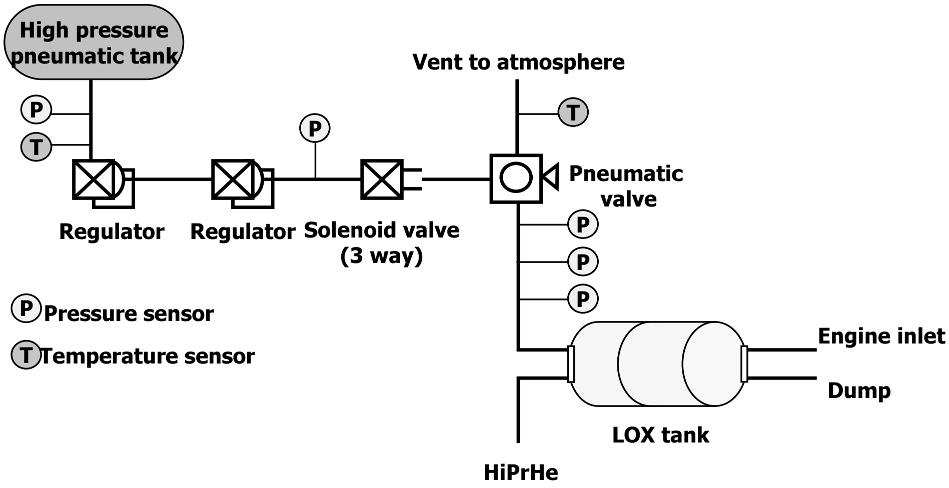

Fig. 1. LOX Tank System

The dynamics of the LOX tank are described by the following set of nonlinear differential equations:

$$
\begin{aligned}
\dot{m}_{lox} &= -B \\
\dot{m}_{gox} &= B - r_{vent} \\
\dot{T}_{lox} &= \frac{q_{lox} - h_v B}{c_p m_{lox}} \\
\dot{u}_{gox} &= \frac{1}{m_{gox}}(B h_{gox_{in}} - r_{vent} h_{gox_{out}} + q_{gox} + B u_{gox})
\end{aligned}
$$

Liquid oxygen is boiling if its saturation pressure is larger than the pressure of the GOX inside the tank according to

$$
B = \begin{cases} 0, & \text{if } P_{gox} > P_{sat} \\ r_{vent} + \frac{r_{vent}}{2}(P_{sat} - P_{gox}) - 0.001, & \text{if } P_{sat} \geq P_{gox} \end{cases}
$$

where the saturation pressure is a function of the LOX temperature approximated by

$$
\begin{aligned}
P_{sat}(T_{lox}) = 272.968 &- 9.83445 T_{lox} + 0.139169 T_{lox}^2 - 9.4813 \times 10^{-4} T_{lox}^3 \\
&+ 2.9745 \times 10^{-6} T_{lox}^4 - 3.00628 \times 10^{-9} T_{lox}^5
\end{aligned}
$$

and the GOX pressure is given using the ideal gas law

$$
P_{gox} = \frac{m_{gox}}{w_{gox}} R \frac{T_{gox}}{V_{gox}}.
$$

The GOX temperature is computed using the internal energy as $T_{gox} = (u_{gox} - 54.503)/0.1606$. The GOX volume is computed by $V_{gox} = V_{tank} - V_{lox}$ where

PARAMETER	DESCRIPTION	VALUE (for constants)	UNITS
m_{lox}	mass of LOX		lbm
m_{gox}	mass of GOX		lbm
T_{lox}	temperature of LOX		deg R
T_{lox}	temperature of LOX		deg R
u_{gox}	internal energy of GOX		BTU/lbm
B	boil rate		lbm/s
r_{vent}	venting mass rate		lbm/s
q_{gox}	heat dissipation of GOX		BTU/s
q_{lox}	heat dissipation of LOX		BTU/s
q_{total}	total heat dissipation		BTU/s
h_v	heat of vaporization of LOX	91.5	BTU/lbm
C_p	specific heat of LOX	0.4	BTU/deg R
h_{goxin}	input enthalpy of GOX		BTU/lbm
h_{goxout}	output enthalpy of GOX		BTU/lbm
P_{lox}	pressure of LOX		psi
P_{gox}	pressure of GOX		psi
P_{sat}	saturation pressure of LOX		psi
w_{gox}	molecular weight of GOX	0.0705	lbm/mol
V_{lox}	volume of LOX		ft^3
V_{gox}	volume of GOX		ft^3
V_{tank}	volume of LOX tank	16800	ft^3
d_{lox}	density of LOX	71.5	lbm/ft^3
d_{gox}	density of GOX		lbm/ft^3
m_{PVT}	mass of He in PVT		lbm
T_{PVT}	temperature of He in PVT		deg R
P_{PVT}	pressure of PVT		psi
r_{in}	PVT input mass rate		lbm/s
h_{PVT}	input enthalpy of He in PVT		BTU/lbm
V_{PVT}	volume of PVT	0.04	ft^3
d	diameter of the pneumatic valve	0.065	ft
α	loss coefficient	2	
w_{he}	molecular weight of He	0.00882	lbm/mol
R	gas constant	2.365×10^{-2}	$\frac{\text{psi ft}^3}{\text{mol mol}}$

Table 1. Nomenclature

$V_{lox} = m_{lox}/d_{lox}$. The input and output enthalpy are given by

$$h_{gox_{in}} = 0.2184 T_{lox} + 56.906$$
$$h_{gox_{out}} = 0.2184 T_{gox} + 56.906$$

and the GOX and LOX head dissipation are defined by

$$q_{gox} = 1.75 \times 10^{-5} (550 - T_{gox}) q_{total}$$
$$q_{lox} = q_{total} - q_{gox}.$$

The pneumatic subsystem consists of a pneumatic tank with high pressure helium (He), two pressure regulators, a three-way solenoid valve, and a pneumatic valve as shown in Figure 1. When the solenoid valve is open, high pressure He flows from the pneumatic tank and pressurizes the pneumatic valve. We omit the dynamical model of the pneumatic subsystem due to space limitations and we focus on the pneumatic valve that directly affects the behavior of the LOX tank subsystem.

The pneumatic valve includes a tank called pneumatic valve tank (PVT) whose dynamics are

$$\dot{m}_{PVT} = r_{in}$$

$$\dot{T}_{PVT} = \frac{r_{in} h_{PVT}}{13.2 + 3.125 m_{PVT}}$$

where $h_{PVT} = 5.2 T_{PVT} + 30.2$ is the input enthalpy. The pressure in the PVT is

$$P_{PVT} = \frac{m_{PVT}}{w_{he}} R \frac{T_{PVT}}{V_{PVT}}.$$

The behavior of the pneumatic valve is described using the pressure P_{PVT} as follows:

$$\text{pneumatic valve } = \begin{cases} \text{open,} & \text{if } P_{PVT} < 410 \\ \text{closed,} & \text{if } P_{PVT} \geq 410 \end{cases}$$

The above guard conditions on the pressure P_{PVT} define the coupling between the pneumatic system and the LOX tank. If $P_{PVT} < 410$ then the valve is open and LOX is venting in the atmosphere. The venting mass rate is given by

$$r_{vent} = \frac{\pi d^2 \sqrt{\frac{2(P_{gox} - P_{atm})4636.8}{\alpha d_{gox}}}}{4} d_{gox}.$$

If the valve is closed then there is no venting. We also assume that the flow goes only in one direction through the valve.

The venting of GOX is monitored by pressure and temperature sensors. A digital controller is used to actuate the solenoid valve based on the GOX pressure measurements using the following rule: if the pressure P_{gox} falls below 12.2, the controller closes the solenoid valve and if the pressure exceeds 18, then the controller opens the solenoid valve. The overall system can be modeled as a distributed hybrid system composed of the LOX tank and the pneumatic subsystem coupled by the guards that define the operation of the pneumatic valve and the controller. The hybrid system model for the LOX tank is shown in Figure 2. The hybrid estimation problem is to compute the most likely state including the LOX mass and temperature from the pressure and temperature measurements and the commands of the controller.

4 Hybrid System Estimation

The most challenging aspect of every hybrid estimation algorithm is monitoring the autonomous mode transitions and using the appropriate mode q for updating the estimate of the continuous state x. The probability of mode transitions triggered by control commands can be usually computed by discrete estimation techniques based, for example, on hidden Markov models. Our estimation algorithm is based on the following decomposition of the process density

$$p(q_t, x_t | q_{t-1}, x_{t-1}) = p(x_t | q_t, q_{t-1}, x_{t-1}) P(q_t | q_{t-1}, x_{t-1})$$

where the density $p(x_t | q_t, q_{t-1}, x_{t-1})$ describes the evolution of the continuous state conditioned on the mode and the distribution $P(q_t | q_{t-1}, x_{t-1})$ describes the mode transition probability conditioned on the continuous state.

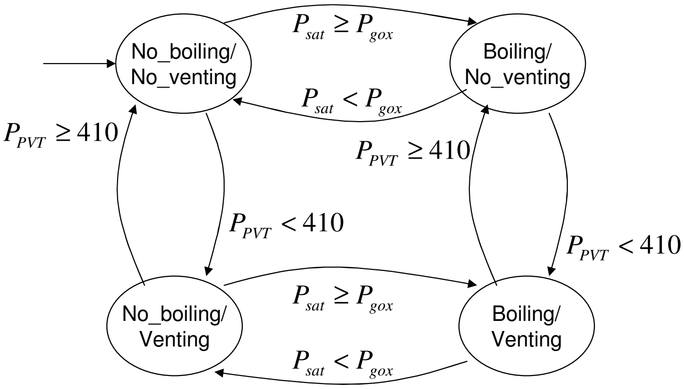

Fig. 2. Hybrid system model for the LOX tank

We define the mode transition probability matrix with elements

$$T_{ij}(t) = p(q_t = j|x_{t-1}, q_{t-1} = i), \ i, j = 1, \dots, |Q|.$$

Let G_{ij} be the guard corresponding to the transition from mode i to mode j. Assuming that the system is at mode q_i and that the probability of the transition $q_i \rightarrow q_j$ is equal to the probability the guard G_{ij} is satisfied, the mode transition probability matrix can be computed as

$$T_{ij}(t) = \int_{G_{ij}} p(x_{t-1}|Y_{t-1}, U_{t-1}, q_{t-1} = i) dx_{t-1} \qquad (1)$$

where $p(x_{t-1}|Y_{t-1}, U_{t-1}, q_{t-1} = i)$ is the conditional density of the continuous state at time $t-1$.

The above integral represents the probability of switching from mode q_i to mode q_j. The general idea of our estimation algorithm is that at every time step we evaluate the transition probability matrix based on the estimate of the continuous state. Then, we focus on the most likely modes and we update the continuous estimate by conditioning our belief on the new measurements. Our current implementation is based on a particle filtering approach described in Section 5. This approach allows the efficient computation of the transition probabilities using Monte Carlo methods. The transition probabilities are then used to dynamically assign particles to the discrete modes, thus focusing on the most likely transitions.

In the distributed algorithm proposed in this paper, local estimators communicate with each other by messages that contain the probability values of the guard conditions that define the coupling between the subsystems. Consider a local mode transition $e^{(n)}$ of the n^{th} subsystem H_n. Based on our assumptions regarding the coupling between subsystems, the mode transition probability may depend on the remote continuous state $x^{(m)}$ according to the guard condition $G(e^{(n)}, x^{(m)})$. In this case, the probability that this guard condition is satisfied can be computed at the remote subsystem H_m. This value is the only information needed at the local node in order to compute the probability of the mode transition $e^{(n)}$. Note that it's possible that the guard $G(e^{(n)})$ is a logical combination of multiple conditions of the form $G(e^{(n)}, x^{(m)})$ representing coupling between multiple components.

Next, we discuss how we can improve the performance of the algorithm by transforming the guard conditions so that they form a cover of the state space. The probability of occurrence of the autonomous transitions is represented by the transition probability matrix that can be computed at every time step. The estimation algorithm will be robust if small changes in the continuous state do not result in large changes in the probabilities T_{ij}. Practically, it is desirable to (1) avoid chattering phenomena, where the probability mass oscillates between modes at every time step, and (2) allow enough time after a mode change for the transient to converge to the steady state behavior for that particular mode. These aspects of the algorithm can be considerably improved by transforming the guard conditions so that they form a cover of the continuous state space as explained in the following.

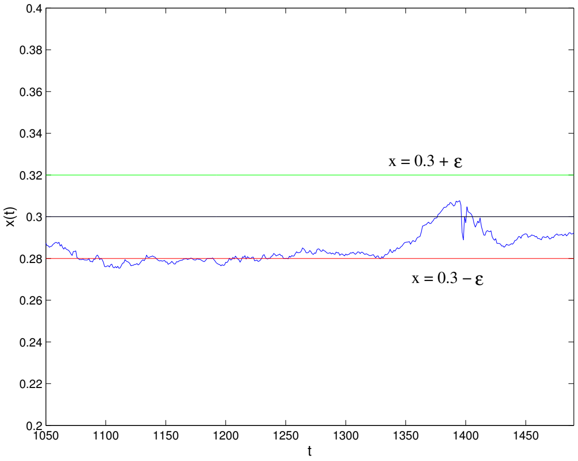

Fig. 3. Guard conditions that cover the state space

Figure 3 shows a simple example of a continuous state trajectory and a guard condition. The system switches from q_1 to q_2 if $x(t) > .3$ and from q_2 to q_1 if $x(t) < .3$. Our estimation algorithm returns a probability distribution over possible continuous states that approximates the actual state $x(t)$ at every time step. If the transition probability matrix T is computed using the original guard conditions, the performance of the algorithm is degraded by the fast switching around $t = 1380$ and leads to chattering between modes q_1 and q_2. While the most likely discrete state oscillates between q_1 and q_2, the estimation of the continuous state is unreliable.

Hybrid estimation can be considerably improved by transforming the guard conditions to form a cover of the state space as illustrated in Figure 3. The transition $q_1 \rightarrow q_2$ occurs if $x(t) > .3 + \epsilon$. Similarly, the transition $q_2 \rightarrow q_1$ occurs if $x(t) < .3 - \epsilon$. The small variations of the state around $x(t) = .3 - \epsilon$, for example, will not trigger any transitions since the system is not in mode q_2. The design parameter ϵ depends on the process and measurement noise. The transition probability matrix can be represented by the transformed guard conditions by equation (1). This method has been used also for removing Zeno behavior from hybrid system models. The guard transformation is desirable, however, even for non-Zeno hybrid systems in order to improve the robustness of the estimation algorithm in the presence of process and measurement noise. It should be noted that the continuity of analog-to-digital maps based on covers of the state space has been studied using small topologies in [12].

5 Particle Filtering Methods

In the following, we describe the particle filtering algorithm for distributed hybrid estimation. To simplify the notation, first we consider only one subsystem and we assume that there are not any control inputs. Then, we describe the step where communication between subsystems is required and how the coordination between the local estimators is accomplished. Note that detailed descriptions of particle filtering methods for estimation of dynamical systems can be found in [3].

Let $\{s_{t-1}^{(k)}, w_{t-1}^{(k)}, k = 1, \ldots, N\}$ denote the sample set at time $t-1$ where $s_{t-1}^{(k)} = (q_{t-1}^{(k)}, x_{t-1}^{(k)})$ is the k^{th} sample of the local hybrid state and $w_{t-1}^{(k)}$ its probability weight. The k^{th} sample of the predicted local state at time t is denoted by $\tilde{s}_t^{(k)} = (\tilde{q}_t^{(k)}, \tilde{x}_t^{(k)})$. The estimation algorithm consists of the following steps:

```
1. Initialization $t = 0$.
   a. sample $s_0^{(k)} = (q_0^{(k)}, x_0^{(k)}), k = 1, 2, \ldots, N$ from $p(q_0)$ and $p(x_0)$ and set $t = 1$.
2. Prediction
   a. apply $p(s_t | s_{t-1}^{(k)})$ to compute each $\tilde{s}_t^{(k)}$.
      i. compute $T_{ij}(t) = p(q_t = j | x_{t-1}, q_{t-1} = i)$ from $s_{t-1}^{(k)} = (q_{t-1}^{(k)}, x_{t-1}^{(k)})$ and $w_{t-1}^{(k)}$.
      ii. sample $\tilde{q}_t^{(k)}$ from $T_{ij}(t)$.
      iii. apply $p(x_t | x_{t-1}^{(k)}, q_{t-1}^{(k)}, \tilde{q}_t^{(k)})$ to compute $\tilde{x}_t^{(k)}$.
   b. evaluate the importance weights $w_t^{(k)} = p(y_t | \tilde{s}_t^{(k)})$.
   c. normalize the weights.
3. Re-sampling
   a. re-sample $N$ particles $s_t^{(k)}$ from $\tilde{s}_t^{(k)}$.
   b. set $t \leftarrow t + 1$ and go to step 2.
```

The interaction between the discrete and continuous dynamics is addressed at the prediction step of the algorithm in order to compute the distribution of the predicted state $\tilde{s}_t^{(k)}$. Consider that at time t the prediction $p(q_{t-1}, x_{t-1}|Y_{t-1})$ is represented by the sample set $\{q_{t-1}^{(k)}, x_{t-1}^{(k)}, w_{t-1}^{(k)}, k = 1, \ldots, N\}$. The mode transition probability matrix can be computed by

$$T_{ij}(t) = \begin{cases} \dfrac{\sum_{k \in \hat{G}_{ij}} w_{t-1}^{(k)}}{\sum_{k \in \hat{I}} w_{t-1}^{(k)}}, & i \neq j \\ \\ 1 - \sum_{\ell \neq i} T_{i\ell}(t), \; i = j \end{cases} \tag{2}$$

where $k \in \hat{G}_{ij} \Leftrightarrow q_{t-1}^{(k)} = i \ \wedge \ x_{t-1}^{(k)} \in G_{ij}$ and $k \in \hat{I} \Leftrightarrow q_{t-1}^{(k)} = i$.

The computation of the mode transition probability matrix is the only step of the distributed algorithm that requires communication between subsystems. For the class of distributed hybrid systems studied in this paper, a mode transition $e = (q_i, q_j)$ in subsystem H_n can be triggered by either a guard condition on the local continuous state or a guard condition on the continuous state of another subsystem H_m. In both cases, the probability for each mode transition can be computed locally at each subsystem. In order to proceed with the prediction of its local state, however, each subsystem must assemble the probabilities for all the local mode transitions. Therefore, at every time step each subsystem must receive messages with the probabilities of the local mode transitions that depend on the continuous state of remote subsystems.

Let $(q_{t-1}^{(k)}, x_{t-1}^{(k)}, w_{t-1}^{(k)})$ be the k^{th} particle and assume $q_{t-1}^{(k)} = i$, then we sample from the i^{th} row of the mode transition probability matrix to select the k^{th} sample $\tilde{q}_t^{(k)}$ for the discrete mode. Suppose that $\tilde{q}_t^{(k)} = j$, then we sample from the density $p_{ij}(x_t|x_{t-1}^{(k)}) = p(x_t|x_{t-1}^{(k)}, q_{t-1} = i, q_t = j)$ to compute the k^{th} sample $\tilde{x}_t^{(k)}$ for the continuous state. Next, we compute that importance weights, normalize, reinforce the predicted state using the observations, and re-sample the particles as described in the above algorithm.

At each subsystem, the estimated mode is computed as the most likely mode at every time step and the continuous state is computed using only particles from the most likely mode, that is

$$\hat{q}_t = \arg\max_i \sum_{k \in \hat{Q}_i} w_t^{(k)} \tag{3}$$

and

$$\hat{x}_t = \frac{\sum_{k \in \hat{Q}} w_t^{(k)} x_t^{(k)}}{\sum_{k \in \hat{Q}} w_t^{(k)}} \tag{4}$$

where $\hat{Q}_i = \{k|q_t^{(k)} = i\}$ and $\hat{Q} = \{k|q_t^{(k)} = \hat{q}_t\}$.

Estimation of the hybrid state based on the most likely mode is selected for computational reasons. Our objective is the development of estimation algorithms suitable for the cryogenic propulsion system and applications with real-time requirements. Particle filter methods can also support multiple hypotheses where the continuous state is estimated for every mode. However, keeping track of multiple hypotheses requires a sufficient number of particles to be assigned to each mode at every time step, thus increasing the computational requirements. A possible improvement of the algorithm is to keep multiple hypotheses using the most likely modes if they have sufficient number of particles.

6 Simulation Results

The particle filtering algorithm presented in Section 5 is used for fault detection using an observer-like scheme. The particle filter algorithm plays the role of a hybrid observer which is computing the most likely discrete mode $\hat{q}$ and continuous state $\hat{x}$ and is generating the expected output $\hat{y}$ based on the plant model. The residual signal $r_t = y_t - \hat{y}_t$ is low-pass filtered and thresholded to detect possible failures. Fault detection and isolation is performed by considering both the residual r_t and the mode $\hat{q}_t$. For example, the observer may not be able to perfectly track fast transients after each mode transition and therefore, the residual exceeding the threshold immediately after a mode transition does not necessarily correspond to a fault. Also information about the modes for which the discrepancy is present can be used for fault isolation. A leakage in a pneumatic valve, for example, will cause a discrepancy only if the valve is closed.

The particle filtering algorithm was implemented in C++ and integrated into a distributed fault detection and diagnosis system for the X34. For the purposes of distribution, we considered the LOX tank, its pneumatic valve, and the associated sensors to be one subsystem, and the pneumatic tank and remaining components to be another. The mode of the LOX tank depends upon the mode of the solenoid valve of the pneumatic subsystem. In practice, the task of the particle filter for the LOX subsystem is thus to estimate the state of the LOX tank given the observations from the pressure and temperature sensors attached to the LOX subsystem and a distribution on the current mode of the venting valve provided by the pneumatic subsystem. In the current implementation, the LOX particle filter is provided with the most likely mode of the venting valve at each sampling point, rather than a probability distribution over its mode. In addition, in our experiments the role of the vehicle and its sensors was played by a Matlab simulation developed from a model of the X34 provided by NASA.

Using these observations, the particle filter estimates the most likely state of its subsystem from a model that does not include failures. We next compute the residual between the expected output of that non-failure state and the current observations from the vehicle's sensors. A diagnostic system then detects and diagnoses faults. It uses a feature extraction algorithm and a neuro-fuzzy classifier to compute the probability of its fault hypotheses based on the residual signals. Details are beyond the scope of this paper. These fault hypotheses would then be reported to the vehicle operator or on-board control system.

The simulator, particle filter and diagnosis systems were integrated as a multi-process distributed system using the Open Control Platform, a distributed computing platform developed by Boeing for the DARPA Software Enabled Control program. The simulator and particle filter were separate OCP processes while the diagnosis system was distributed between two OCP processes. The simulation and particle filter processes each ran on a 1.7GHz Pentium 4 processor within a dual-processor PC while the diagnostic processes ran on a second 600MHz PC connected by a a 100Mbit/second Ethernet LAN. The use of OCP made the location of each process reconfigurable and transparent. The observations were sampled in the simulator and fed into the particle filter via OCP, where particle filtering was performed using $N = 100$ particles. The simulated sampling rate was 10Hz (i.e., ten samples per simulated second) but for our experiments the LOX particle filter could be reliably run 10 times faster than real time. In the following, we present simulation results for the LOX hybrid estimation portion of the distributed diagnosis system for two scenarios (1) normal behavior, and (2) leakage in the pneumatic valve.

Normal behavior. We have demonstrated that the algorithm can track the state in the case when there are no faults in the system. The continuous states corresponding to the LOX and GOX masses are shown in figure 4. The system is discretized using a sampling period $T = 100ms$. The outputs are the GOX pressure and temperature and are contaminated with Gaussian noise. Figure 5 shows when the LOX is boiling or not and the expected venting pressure, as computed using the estimated state, plotted versus the actual venting pressure. The expected pressure is tracking closely the actual pressure. The GOX pressure is increasing because the tank absorbs heat from the atmosphere. The LOX starts boiling increasing the GOX pressure above 18 *psi* which triggers the controller to open the venting valve. GOX is venting in the atmosphere and the pressure is decreasing although LOX is still boiling in the tank. When the pressure falls below 12 *psi* the controller closes the solenoid valve.

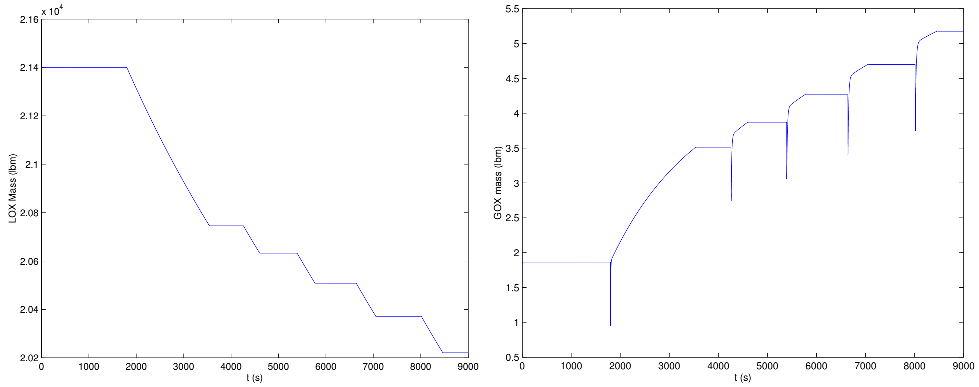

Fig. 4. LOX and GOX mass

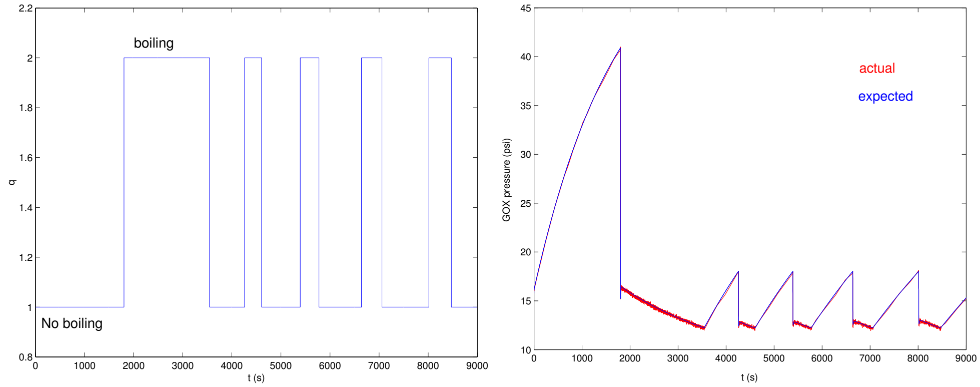

Fig. 5. Discrete mode and venting pressure

Pneumatic valve leakage. The estimation algorithm can be used also to detect continuous faults such as leakage in the pneumatic valve. The valve leakage was simulated by including an additive term in the equation that represents the flow balance when the pneumatic valve is closed. Figure 6 shows the expected and the actual venting pressure. The estimated discrete mode and the residual signal computed as the difference between the actual GOX pressure and the expected are also shown. Whenever there is no boiling then the actual pressure is less than the expected one and a fault is detected.

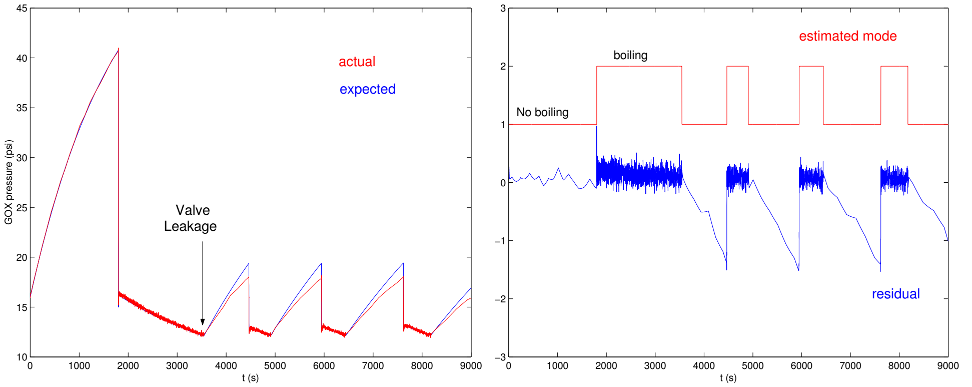

Fig. 6. Expected vs. actual venting pressure, residual, and estimated discrete mode in the case of leakage

7 Discussion and Future Work

Monitoring and diagnosis of embedded systems depends crucially on the ability to estimate the hidden hybrid state from the available measurements. In this paper, we have presented a particle filtering based method and demonstrate the algorithm using a rocket propulsion system. The algorithm can be applied to a large class of distributed hybrid systems with autonomous transitions, nonlinear system dynamics, and non-Gaussian noise. Performance characterization of the algorithm is an important and open problem. Convergence of the algorithm depends crucially on the number of particles that, in turn, depends on the dimension of the continuous state space and number of discrete modes. We have observed that the time interval between discrete transitions also affects the performance of the algorithm. Currently, we address some of these problems by increasing the number of particles and assigning a small number of particles at every mode even if the measurements indicate that some of the modes are not probable. Theoretical aspects regarding the performance characterization of the approach are subjects of current and future research. Currently, we use the estimation algorithm for diagnosis of embedded systems. However, the computational performance allows real-time estimation of the state, and therefore, the algorithm can be used for feedback control as well.

Acknowledgments This work is supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract F33615-99-C3611. Girish Baliga performed much of the implementation on OCP as a PARC summer intern. NASA Ames Research Center provided L2 and collaborated with PARC on HCC, simulation software used for the LOX system. Anupa Bajwa and Adam Sweet of NASA Ames provided expertise on the propulsion system model.

References

1. A. Balluchi, L. Benvenuti, M. D. Benedetto, and A. Sangiovanni-Vincentelli. Design of observers for hybrid systems. In C. Tomlin and M. Greenstreet, editors, *Hybrid Systems: Computation and Control (HSCC'02)*, Vol. 2289, *LNCS*, 76–89. Springer-Verlag, 2002.
2. M. Black and D. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3):231–245, 2000.
3. A. Doucet, N. D. Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001.
4. M. Hofbaur and B. Williams. Mode estimation of probabilistic hybrid systems. In C. Tomlin and M. Greenstreet, editors, *Hybrid Systems: Computation and Control (HSCC'02)*, Vol. 2289, *LNCS*, 253–266. Springer-Verlag, 2002.
5. M. Isard and A. Blake. A mixed-state condensation tracker with automatic model switching. In *Proc. of the 6th International Conference on Computer Vision*, pages 107–112, 1998.
6. W. Jackson, M. Fromherz, D. Biegelsen, J. Reich, and D. Goldberg. Constrained optimization based control of real time large scale systems: Airjet movement object system. In *Proc. of the 40th IEEE Conference on Decision and Control*, pages 4717–4720, Orlando, FL, December 2001.

7. D. Koller and U. Lerner. Sampling in factored dynamic systems. In Doucet et al. [3], pages 445–464.
8. X. Koutsoukos, F. Zhao, H. Haussecker, J. Reich, and P. Cheung. Fault modeling for monitoring and diagnosis of sensor-rich hybrid systems. In *Proc. of the 40th IEEE Conference on Decision and Control*, pages 793–801, Orlando, FL, December 2001.
9. X. Koutsoukos, J. Kurien, and F. Zhao. Estimation of Hybrid Systems Using Particle Filtering Methods. In *Proc. of MTNS 2002*, Notre Dame, IN, August 2002.
10. J. Kurien, X. Koutsoukos, and F. Zhao. Distributed diagnosis of networked hybrid systems. In *AAAI Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction*, pages 37–44, Stanford, CA, March 2002.
11. S. McIlraith, G. Biswas, D. Clancy, and V. Gupta. Hybrid systems diagnosis. In N. Lynch and B. Krogh, editors, *Hybrid Systems: Computation and Control*, Vol. 1790, *LNCS*, 282–295. Springer, 2000.
12. A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, controllability, observability. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, Vol. 736, *LNCS*, 317–356. Springer-Verlag, 1993.
13. F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, P. Cheung, and C. Picardi. Distributed monitoring of hybrid systems: A model-directed approach. In *Proc. IJCAI'2001*, pages 557–564, Seattle, WA, 2001.