Factored Sampling For Efficient Tracking of Large Hybrid Systems

Brenda Ng
Avi Pfeffer
and
Richard Dearden

TR-03-05

Computer Science Group
Harvard University
Cambridge, Massachusetts

Factored Sampling for Efficient Tracking of Large Hybrid Systems

Brenda Ng
Div. of Eng. and Applied Sci.
Harvard University
Cambridge, MA 02138
bmng@eecs.harvard.edu

Avi Pfeffer
Div. of Eng. and Applied Sci.
Harvard University
Cambridge, MA 02138
avi@eecs.harvard.edu

Richard Dearden
School of Computer Science
The University of Birmingham
Birmingham B15 2TT, UK
R.W.Dearden@cs.bham.ac.uk

Abstract

This work presents a new approach to monitoring large dynamic systems. The approach is based on factored particles, which adapts particle filtering by factoring the system into weakly interacting subsystems and maintaining particles over the factors, thus allowing much larger systems to be tracked. Our approach, hybrid factored sampling, works with systems that involve both discrete and continuous variables, including systems where discrete variables depend on continuous parents. The framework lends itself to asynchronous inference—each factor can be reasoned about independently, and the factors joined only when there exists sufficient correlation between them. This allows us to reason about each factor at its appropriate time granularity. In addition, hybrid factored sampling exploits the factorization to provide tractable look-ahead prediction, allowing sampling from the posterior probability given new observations, and considerably improving performance. Empirical results show that hybrid factored sampling is an efficient and versatile method for inference in large hybrid systems.

1 Introduction

Dealing with uncertainty in complex dynamic environments is a basic challenge to the operation of real-world autonomous systems. These systems are highly complex, involving large numbers of stochastic variables and many interacting nonlinear subsystems. An autonomous system must be able to monitor the state of its components and environment in order to form informed plans of intelligent action. The goal of probabilistic monitoring is to maintain beliefs, in the form of probability distributions over state variables, about the state of a dynamically evolving system. To monitor the state of the system, we perform inference in real time to update the new belief state given the previous belief state and the current observations. The task of monitoring is quite a challenge in large-scale systems, because the cost of updating the belief state is often exponential in the number of state variables, thus ruling out any feasibility for exact inference methods.

Hybrid systems pose a special challenge. These systems consist of discrete variables, that represent the operational modes of the system, and continuous variables that model the continuous quantities that affect the system behavior. Even in the simplest case of conditional linear Gaussian (CLG) models, the inference task is NP-hard [Lerner and Parr, 2001]. Moreover, so-called *autonomous transitions* – transitions of discrete variables induced by continuous parents – further complicate inference since they disallow any decoupling of the dynamics between the discrete and the continuous variables. As a consequence of autonomous transitions, the discrete transition probabilities change over time and must be estimated at each time step before discrete dynamics can be propagated. For a model with a high-dimensional discrete state space, this impediment further contributes to intractability.

[Lerner *et al.*, 2000] examined the problem of tracking and fault diagnosis within the framework of conditional linear Gaussian (CLG) models, a restrictive subclass of hybrid dynamic Bayesian networks that does not handle autonomous transitions. [Lerner, 2002] referred to hybrid models with autonomous transitions as *augmented* CLG*s* and have shown that exact inference can be implemented within the clique tree framework. However, exact inference comes at a very high price because even simple models lead to intractably large clique trees. Furthermore, many systems exhibit nonlinear behavior that cannot be captured by a CLG model. Others such as [Funiak and Williams, 2003; Koutsoukos *et al.*, 2002] have presented variants of particle filters (PFs) [Doucet *et al.*, 2000b] as alternatives. So far, the empirical results were based on small models where the number of particles required by PF is non-prohibitive. The issue of scalability remains an open research problem.

This work described here is directly motivated by the issue of multiple time granularities encountered in creating a state estimation system for the K-9 experimental Mars rover at NASA Ames Research Center [Willeke and Dearden, 2004]. As with many complex robotic systems, the rover contains subsystems that operate at different rates and numerous sensors that report asynchronously on the performance of its components. A common approach is to use a fixed time-step state estimator and perform inference at the finest time granularity. This naive approach is quite wasteful because much computational resources will be spent inferring about subsystems that are in stasis. Since the rover decomposes into subsystems of different time granularities, our approach is to exploit this structure to perform inference on each factor at its appropriate time granularity.

In our previous work [Ng *et al.*, 2002], we presented the idea of factored sampling using *factored particles* (FP) for approximate monitoring of discrete systems. FP shares with ordinary particle filtering (PF) the any-real-time property that the number of particles can be adjusted based on the available resource for inference at each time step, thus making FP a nice framework for resource-bounded reasoning. Unlike PF, FP exploits structure – in particular, that complex systems typically consist of weakly-interacting factors that are nearly independent of each other. By maintaining sets of factored particles over the factored state spaces, FP is able to reduce the variance of the sampling process, and thus monitor with greater accuracy using fewer particles than required by PF.

In this work, we extend the framework of factored particles to hybrid systems. This new method, known as *hybrid factored sampling* (HYBRID-FP), uses the representation of factored particles to approximate the belief state and is able to handle hybrid systems, including those that exhibit autonomous transitions. Its underlying FP representation lends readily to asynchronous inference, whereby each factor is inferred about independently. These factors are joined only when there exists sufficient correlation between the factors. This allows us to reason about each factor at its appropriate time granularity.

In addition to autonomous transitions and multiple time granularities, HYBRID-FP has the added feature of tractable look-ahead prediction. The idea of look-ahead within the framework of particle filtering was introduced in [de Freitas *et al.*, 2004]. Look-ahead prediction improves monitoring accuracy by ensuring that only the particles that best reflect the most recent observations are chosen to represent the belief state and are propagated into the next time step. However, look-ahead is an expensive procedure that is often intractable for systems with a large discrete state space. Typically, one must enumerate *all* the discrete states before one can propagate the dynamics forward to perform the look-ahead. So unless the discrete state space is small, it is virtually impossible to enumerate the full state space. In HYBRID-FP, the FP framework lends readily to *factored look-ahead*, whereby look-ahead prediction is performed in factored state space, and then pieced together to approximate the effect of the full look-ahead. We show empirically that these key features, along with the benefits of FP, make HYBRID-FP an efficient and versatile method for scalable inference of complex hybrid systems.

2 THE MARS ROVER MODEL

As a testbed, we model the K-9 Mars rover as a discrete-time model. The Mars rover employs a rocker-bogey suspension system for its six wheels. On each side, the front two wheels are attached to the bogey, which pivots around its center. The center of the bogey is attached to the rocker, which is attached to the main rover chassis and the rear wheels. Motion from the bogey can cause the rocker to pivot around the differential axle, a mechanism that centers the rover chassis between the left and right sets of wheels. From the rocker angles and the bogey angles, we can infer the rover orientation and thus the wheel heights relative to the rover chassis.

We are interested in reasoning about the rover wheel dynamics under the effects of weather, terrain, ground conditions and faulty wheel behavior. We model the rover as a hybrid dynamic Bayesian network with multiple time granularities, shown in Figure 2. Note that the DBN shows dependencies across different time steps, not just from time t to time $t + 1$.

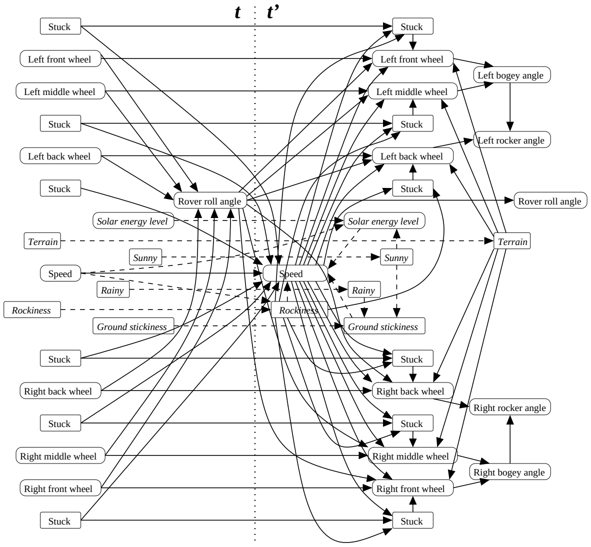

Figure 1: Rover DBN

The model consists of 20 state variables and 4 transient variables. Of the state variables, there are 11 binary and ternary discrete variables and 9 continuous variables. The variables are grouped according to time granularity. The first group of 15 variables (shown in Figure 2 with solid arrows and nodes with non-italicized labels) model rover quantities such as the solar energy level available to the rover, the rover speed, the wheel heights, the wheel stuck conditions, and the rover roll angle. The second group of 5 variables (shown in Figure 2 with dashed arrows and nodes with italicized labels) model background factors, such as weather, terrain, ground rockiness and ground stickiness. The rover variables represent quantities that change at a much quicker rate than those represented by the background variables. In our experiments, we assume that the background variables evolve at 1/25 the rate of the rover variables.

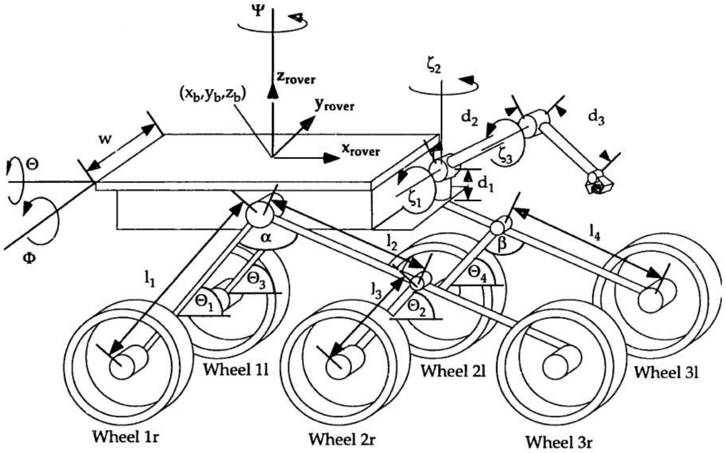

Figure 2: Rover schematic, taken from [Hacot, 1998]

The rover dynamics are adapted from the kinematic analysis [Hacot, 1998] on the model shown in Figure 2. The front wheels receive an input proportional to some height perturbation induced by the terrain. The change in the middle wheel height is proportional to the difference between the front and middle wheel height, taken relative to the rover body which experiences roll (torsional rotation) on uneven terrain. The dynamics for the back wheel is defined similarly. The change in wheel height is scaled by a proportionality constant dependent on the rover speed and the condition of whether the wheel is stuck. As the wheels get stuck, the speed decreases. The speed is also affected by the the availability of solar energy and ground surface characteristics such as stickiness and rockiness. The continuous variables that model solar energy, rover speed and rocker/bogey angles are observed as noisy measurements. These measurements arrive asynchronously, at the rate of its transition. None of the discrete variables are observed.

3 Preliminaries

3.1 Description of hybrid systems with autonomous transitions

A hybrid system is a system that contains interacting discrete and continuous dynamics. The discrete dynamics are described by probabilistic transitions to a finite set of discrete states. Discrete transitions can occur either in a controlled manner upon receiving an exogenous command or *autonomously* when the continuous state satisfies certain *guard* conditions. Every discrete instantiation corresponds to a set of unique differential or difference equations that governs the evolution of the continuous variables.

We assume knowledge of DBNs. Let Z_t denote the discrete state variables and X_t denote the continuous state variables. Let Y_t denote the observed continuous variables of a DBN. To simplify presentation, we assume here that the discrete variables are unobserved, but our framework can be easily extended to handle discrete observations. We use lowercase to denote particular instantiations of the variables. Let $y_{1:t}$ denote the history of observations from time 1 to t. To facilitate discussion about hybrid systems with autonomous transitions, we present a skeleton DBN in Figure 3.

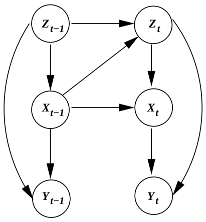

Figure 3: Hybrid model with autonomous transitions.

The model given in Figure 3 is described probabilistically by:

$$z_t \sim p(Z_t|Z_{t-1}, X_{t-1}) \tag{1}$$
$$X_t = F_{z_t}(X_{t-1}) + v_t \tag{2}$$
$$Y_t = H_{z_t}(X_t) + w_t \tag{3}$$

where $v_t \sim \mathcal{N}(0, Q)$ is the process noise and $w_t \sim \mathcal{N}(0, R)$ is the measurement noise. F and H are mappings that evolve the continuous variables from from one time point to another. Each instantiation of Z_t affects the evolution of the continuous state in a unique way. We denote $\Phi(z_t) \equiv \{F_{z_t}, H_{z_t}\}$ as the model parameters associated with the discrete state z_t.

As shown, Z_t depends on Z_{t-1} and X_{t-1}. For each instantiation of Z_{t-1}, there is a guard condition on X_{t-1} that determines a probability distribution over the discrete states for Z_t. A guard condition is a conjunction of guard events, such as $X_t > S$ for some S. The representation of autonomous transitions is important to hybrid system modelling because autonomous transitions abound in the real world. In our rover model, the wheels have higher probabilities of getting stuck when the rover is running at high speeds. Let Z_t model the discrete state of a wheel getting stuck and X_t model the rover speed. Let the guard condition g be whether the speed exceeds a safety threshold S. Given that $Z_{t-1} = False$, Z_t transitions to $True$ with probability 0.004 when $X_{t-1} > S$, and with probability 0.002 otherwise. Let γ be the probability of $X_{t-1} \leq S$. Then the probability for the wheel to be stuck at time t is:

$$\begin{aligned} & p(Z_t = True|Z_{0:t-1}, y_{1:t-1}) \\ = & \sum_{X_{t-1}} p(Z_t = True|Z_{t-1}, g(X_{t-1}))p(X_{t-1}|Z_{0:t-1}, y_{1:t-1}) \\ = & \; 0.004(1-\gamma) + 0.002\gamma \end{aligned} \tag{4}$$

A great challenge posed by autonomous transitions is that the discrete transition probability is time-variant. Thus, the transition probabilities must be estimated at each time step before discrete dynamics propagation can occur. For a model with high-dimensional state space, this impediment leads to intractability.

3.2 Hybrid monitoring

Let $S_t = (Z_t, X_t)$ denote the hybrid system state. We adopt the notation given in [de Freitas *et al.*, 2004]. The aim is to track the probability distribution $p(S_{0:t}|y_{1:t})$ of the history of states over time, conditioned upon the sequence of past observations $y_{1:t}$. The distribution can be recursively estimated given $p(S_{0:t-1}|y_{1:t-1})$ and the current observation y_t:

$$\begin{aligned} p(S_{0:t}|y_{1:t}) &= \\ p(S_{0:t-1}|y_{1:t-1}) &\times \frac{p(y_t|S_t)p(X_t|X_{t-1}, Z_t)p(Z_t|S_{t-1})}{p(y_t|y_{1:t-1})} \end{aligned} \tag{5}$$

However, exact computation leads to intractability in all but the smallest conditional linear Gaussian models. As a result, one must resort to approximate inference methods.

One approach is to exploit structure in order to parametrize the distribution as a product of marginal distributions. This method, known as Boyen-Koller algorithm (BK) [Boyen and Koller, 1998], exploits the idea of weak interaction between different system components to artificially impose independencies between weakly-interacting subsystems. BK partitions the state space into *clusters* corresponding to such near-independent subsystems and represents the belief state as a product of marginal distributions over the clusters. Although this practice introduces error into the belief state, it has been shown theoretically that this error remains bounded over time and is minimized when the clustering matches the natural structure in the system. BK performs exact belief update via junction tree propagation and is limited to domains where junction tree propagation is feasible.

A popular and widely used approach is the particle filter (PF) [Doucet *et al.*, 2000b], which approximates (5) by a discrete sum of *particles* or samples of possible states drawn from that distribution:

$$p(Z_{0:t}, X_{0:t}|y_{1:t}) \approx \frac{1}{N}\sum_{i=1}^{N} \delta(z_{0:t}^{(i)}, x_{0:t}^{(i)})$$

where $\delta(\cdot)$ denotes the Dirac delta function. Since it is difficult to sample from (5) directly, importance sampling is used, in which particles are drawn from a more tractable *proposal* distribution and each particle is weighted to account for this bias. Since variance increases as the state space increases, the number of particles required to achieve decent accuracy increases as well, so optimizations are required to make PF feasible for tracking complex, high-dimensional systems.

Rao-Blackwellization is a popular technique that is often used in conjunction with particle filtering to infer about hybrid systems. [Doucet *et al.*, 2000a] presented the Rao-Blackwellized particle filter (RBPF) that improves PF by analytically marginalizing out some of the variables. Consider the factorization

$$p(Z_{0:t}, X_{0:t}|y_{1:t}) = p(X_{0:t}|Z_{0:t}, y_{1:t})p(Z_{0:t}|y_{1:t})$$

the Gaussian distribution $p(X_{0:t}|Z_{0:t}, y_{1:t})$ can be computed analytically given the discrete distribution $p(Z_{0:t}|y_{1:t})$. Hence, RBPF can be used to reduce the sampling space from $(Z_{0:t}, X_{0:t})$ to only $Z_{0:t}$.

The sampling space can be further reduced if $p(Z_t|Z_{0:t-1}, y_{1:t})$ is used as the proposal distribution. Doing so allows one to factorize $q(Z_{0:t}|y_{1:t}) = p(Z_t|Z_{0:t-1}, y_{1:t})q(Z_{0:t-1}|y_{1:t-1})$, which approximates the posterior distribution $p(Z_{0:t}|y_{1:t}) = p(Z_t|Z_{0:t-1}, y_{1:t})p(Z_{0:t-1}|y_{1:t})$. This allows one to sample Z_t from $p(Z_t|Z_{0:t-1}, y_{1:t})$ to simulate the effect of sampling from $p(Z_{0:t}|y_{1:t})$. Thus, the sampling process is greatly simplified and the importance weights are given by:

$$\begin{aligned} w_t &= \frac{p(Z_{0:t}|y_{1:t})}{q(Z_{0:t}|y_{1:t})} \propto p(y_t|Z_{0:t-1}, y_{1:t-1}) \\ &= \sum_{Z_t} p(y_t|Z_{0:t}, y_{1:t-1}) p(Z_t|Z_{0:t-1}, y_{1:t-1}) \end{aligned} \qquad (6)$$

A further improvement on PF is *look-ahead* RBPF [de Freitas *et al.*, 2004]. The idea is based on the observation that the importance weight, given in (6), does not depend on the actual particles at time t. So one can use evidence from time t to "look ahead" and select the fittest particles from time $t-1$ to propagate into time t. This technique allows for a richer set of particles that better reflects $p(Z_t|Z_{0:t-1}, y_{1:t})$. However, look-ahead becomes quickly intractable because it requires enumerating all possible discrete states. Therefore, one must find an efficient approximation to make look-ahead feasible.

Even with RBPF, a large complex system may still require many particles to obtain a satisfactory approximation of its belief state. Thus, it may be necessary to use a factored representation of the posterior distribution, as in BK, to make inference tractable. The idea of factored particles [Ng *et al.*, 2002] stemmed from this motivation. It is a hybridization of BK and PF and was introduced as a scalable monitoring method for large discrete DBNs. Instead of maintaining particles over the entire state of the system, particles over clusters of state variables are maintained. Because the clusters have far fewer variables than the entire state space, the variance resulting from maintaining cluster distributions is smaller than from maintaining the belief state as a whole. As a result, better approximations can be obtained with small numbers of particles. But the factoring decorrelates the inter-cluster dependencies and introduces bias into the belief state. Experimental results have shown that FP reduces the variance enough that the small tradeoff of bias is worth the overall improvement in monitoring performance.

4 Hybrid factored sampling

To the best of our knowledge, there has been very little work on tracking hybrid systems with multiple time granularities. [Friedman *et al.*, 1998] raised multiple time granularities as an important issue but did not provide a solution. [Lerner *et al.*, 2002] made a first step in dealing with the issue of multiple time granularities by using the steady-state behavior of instantaneous quantities at each time step to infer about the system at a fixed time granularity. Unfortunately, their method was an ad hoc solution that applied specifically to their model. [Kwok *et al.*, 2004] introduced the real-time particle filter, that utilizes a weighted mixture of belief states to encapsulate information gained from asynchronous observations over multiple time points that occur between inference updates. But the method does not exploit multiple time granularities in the evolution of subprocesses. [Verma *et al.*, 2003] presented the variable-resolution particle filter, where the issue of state abstraction in multiple state granularities was explored. However, they did not consider multiple time granularities. Unlike previous approaches, HYBRID-FP exploits simultaneously both time granularity and state factorization to improve approximate inference of complex hybrid systems.

This work presents an improved particle filtering method for hybrid DBNs based on the framework of factored particles. Our algorithm, known as *hybrid factored sampling* (HYBRID-FP), fully leverages the factored particle representation to make asynchronous inference and look-ahead inference possible within the domain of complex hybrid systems with multiple time granularities. On a conceptual level, HYBRID-FP is as follows: Given a hybrid system, the state variables are partitioned into possibly overlapping factors that consist of variables that are strongly correlated with one another. These factors usually correspond to weakly-interacting subsystems. In our application, we partition the rover model into three factors–the first corresponds to background variables, while the other two correspond to the left and the right parts of the rover. The background variables model coarse-granularity quantities such as environment condition and power consumption, while the rover variables model fine-granularity quantities such as the wheel heights. Samples are maintained over the variables in each factor as sets of factored particles. Each set of factored particles corresponds to the belief state over the variables in a factor. At each time point, we propagate the beliefs by using RBPF. If the factor has no strong correlations with variables outside its domain, then the factor is propagated independently. Otherwise, we temporarily augment the factor to include the outside variables and proceed with inference on this augmented factor. After the inference update, the factor is "trimmed" back to its original variable domain and the samples are projected to include only the samples that represent the variables in the factor. The joining and projecting that occurs between inference updates has the effect of preserving the belief state representation as the product of marginals over the original distribution.

By maintaining the belief state as a product of factors (or sets of factored particles in our framework), this allows for two novel benefits: the freedom to reason about each factor independently according to a non-constrained time granularity and the ability to perform look-ahead inference that would otherwise be a very expensive or even intractable operation. From the perspective of resource-bounded computation, HYBRID-FP has the added benefit that one can assign more resources to the factor requiring the most attention. This can be achieved by increasing the update frequency of the critical factor and by diverting the particles to the factor in question. In addition to these benefits, our method inherits the advantages of FP and is able to reduce the number of particles needed for accurate monitoring.

We present in detail the HYBRID-FP algorithm with look-ahead prediction. Assume that a hybrid model is partitioned into K factors. Let Z_t^c and X_t^c denote the respective discrete and continuous variables in a given factor c. Each factor is represented by N factored particles, $\mathbf{s}_t^{(c)} \equiv \{s_t^{(c,i)}\}_{i=1}^N$. Each factored particle $s_t^{(c,i)}$ has three components $(z_t^{(c,i)}, \mu_t^{(c,i)}, \Sigma_t^{(c,i)})$ where $z_t^{(c,i)}$ is a sample of Z_t^c and $(\mu_t^{(c,i)}, \Sigma_t^{(c,i)})$ are the mean and covariance of $X_t^{(c,i)}$. There are five phases to the algorithm: factored look-ahead, join, resampling, dynamics propagation and projection. We present the procedural schematic in Figure 4 and the pseudocode in Algorithm 1.

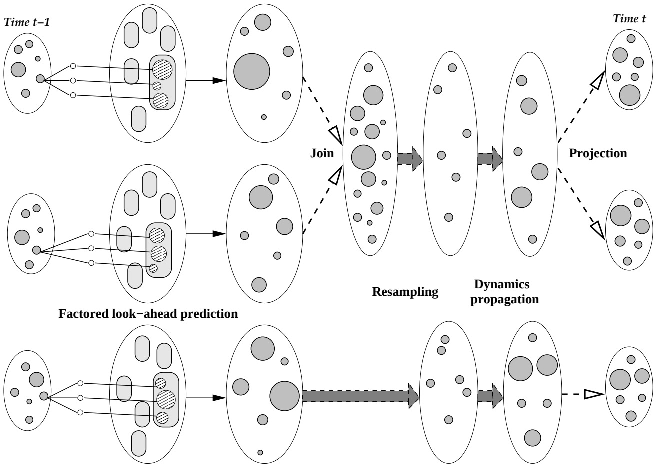

Figure 4: Schematic for HYBRID-FP.

Algorithm 1. HYBRID FACTORED SAMPLING

```
Step 1: Factored look-ahead prediction
for $c = 1$ to $K$ do
  for $i = 1$ to $N$ do
    for all $Z_t^c = j$ reachable from $z_{t-1}^{(c,i)}$ do
      let $g^c$ the local guard condition over $X_t^c$
      Compute factored transition probability:
        $\tau(i,j) = \sum_{X_{t-1}^c} p(Z_t^c = j | z_{t-1}^{(c,i)}, X_{t-1}^c) p(g^c(X_{t-1}^c) | z_{0:t-1}^{(c,i)}, y_{1:t-1})$[1]
        where $X_{t-1}^c \sim \mathcal{N}(\mu_{t-1}^{(c,i)}, \Sigma_{t-1}^{(c,i)})$
      let $\Phi^c(j)$ be the continuous dynamics models corresponding to the factored state $j$
      $(\hat{y}_t^{(c,j)}, S_t^{(c,j)}) \leftarrow$ UKF-PREDICTION$(\mu_{t-1}^{(c,i)}, \Sigma_{t-1}^{(c,i)}, \Phi^c(j))$
      Compute factored likelihood: $L_t^{(c,j)} = \mathcal{N}(y_t^c; \hat{y}_t^{(c,j)}, S_t^{(c,j)})$
      Compute factored successor weight: $w_t^{(c,j)} = L_t^{(c,i)} \tau(i,j)$
    end for
    Compute factored weight: $w_{t-1}^{(c,i)} = (w_0)_{t-1}^{(c,i)} \sum_j w_t^{(c,j)}$
  end for
end for

for all aggregate factors $*$ do
  Step 2: Join (if necessary, for aggregate factors that are not basic factors)
  Equijoin the discrete parts of factored particles: $\{\tilde{z}_{t-1}^{(*,i)}\}_{i=1}^M = \Join_c \mathbf{z}_{t-1}^{(c)}$
  for $i = 1$ to $M$ do
    let $\{i_c\}$ denote the indices to the factored particles that comprise $\tilde{z}^{(*,i)}$
    Multiply the continuous particles: $\mathcal{N}(\tilde{\mu}_{t-1}^{(*,i)}, \tilde{\Sigma}_{t-1}^{(*,i)}) = \prod_c \mathcal{N}(\mu_{t-1}^{(c,i_c)}, \Sigma_{t-1}^{(c,i_c)})$
    Compute importance weight: $\tilde{w}_{t-1}^{(*,i)} = \prod_c w_{t-1}^{(c,i_c)}$
  end for

  Step 3: Resampling
  $\{\tilde{s}_{t-1}^{(*,i)}\}_{i=1}^N \leftarrow$ Sample with replacement $\{\tilde{s}_{t-1}^{(*,i)}\}_{i=1}^M$ according to $\{\tilde{w}_t^{(*,i)}\}_{i=1}^M$

  Step 4: Global dynamics propagation
  Compute global transition probabilities:
    $T(i,j) = \sum_{\tilde{X}_{t-1}^*} p(\tilde{Z}_t^* = j | \tilde{z}_{t-1}^{(*,i)}, \tilde{X}_{t-1}^*) p(g(\tilde{X}_{t-1}^*) | \tilde{z}_{0:t-1}^{(*,i)}, y_{1:t-1})$
    where $\tilde{X}_{t-1}^* \sim \mathcal{N}(\tilde{\mu}_{t-1}^{(*,i)}, \tilde{\Sigma}_{t-1}^{(*,i)})$
  Compute approximate posterior probabilities:
    $\hat{p}(\tilde{Z}_t^* = j | \tilde{z}_{0:t-1}^{(*,i)}, y_{1:t-1}) = \hat{L}_t^{(*,j)} T(i,j)$ where $\hat{L}_t^{(*,j)} \equiv \prod_c L_t^{(c,j_c)}$
  $\{\tilde{z}_t^{(*,i)}\}_{i=1}^N \leftarrow$ Generate $N$ samples from $\hat{p}(\tilde{Z}_t^* | \tilde{z}_{0:t-1}^*, y_{1:t-1})$
  for $i = 1$ to $N$ do
    $(\tilde{\mu}_t^{(*,i)}, \tilde{\Sigma}_t^{(*,i)}, \hat{y}_t^{(*,i)}, S_t^{(*,i)}) \leftarrow$ UKF$(\tilde{\mu}_{t-1}^{(*,i)}, \tilde{\Sigma}_{t-1}^{(*,i)}, y_t, \Phi(\tilde{z}_t^{(*,i)}))$
    Compute true likelihood: $L_t^{(*,i)} = \mathcal{N}(y_t; \hat{y}_t^{(*,i)}, S_t^{(*,i)})$
    Compute sampling bias: $(\tilde{w}_0)_t^{(*,i)} = \frac{p(\tilde{z}_t^{(*,i)} | \tilde{z}_{0:t-1}^*, y_{1:t})}{\hat{p}(\tilde{z}_t^{(*,i)} | \tilde{z}_{0:t-1}^*, y_{1:t})}$
  end for

  Step 5: Projection (if necessary, for aggregate factors that are not basic factors)
  Project particles:   $z_t^{(c,i)} = \pi\ \tilde{z}_t^{(*,i)}$;   $\mu_t^{(c,i)} = \pi\ \tilde{\mu}_t^{(*,i)}$;   $\Sigma_t^{(c,i)} = \pi\ \tilde{\Sigma}_t^{(*,i)}$
  Compute factored prior weight: $(w_0)_t^{(c,i)} = \sqrt[n^*]{(\tilde{w}_0)_t^{(*,i)}}$
end for
```

[1] When g^c take on the form of linear inequalities, there are efficient sampling-based algorithms [Genz, 2002] that can accurately estimate the multi-normal cdf $p(g^c(X_{t-1}^c) | z_{0:t-1}^{(c,i)}, y_{1:t-1})$.

4.1 Factored look-ahead

The goal is to propagate each factored particle $s_{t-1}^{(c,i)}$ forward in time and assess how well its successor states $\{s_t^{(c,j)}\}$ can predict the evidence at the next time slice. For each factored particle $s_{t-1}^{(c,i)}$, we enumerate all possible successor states $\{z_t^{(c,j)}\}$ that are reachable from $z_{t-1}^{(c,i)}$. Conditional on $z_t^{(c,j)}$, $x_{t-1}^{(c,i)}$ evolves according to local dynamics specified by $z_t^{(c,j)}$. The local dynamics is governed by a set of equations that involve only the variables in the factor. We have a systematic method for projecting the global dynamics onto a factor to obtain local dynamics. The local dynamics depends on expectations of variables outside the factor. Details will be presented in Section 4.6.

We propagate the local dynamics using the unscented Kalman filter (UKF) [Wan and van der Merwe, 2000], a variant of the Kalman filter. The Kalman filter [Grewal and Andrews, 2001] is an efficient, recursive method that finds the least-square estimate of the state of a linear stochastic process with Gaussian noise. UKF extends the Kalman filter by appling the unscented transformation to handle nonlinearities in the system model. UKF generates an estimate $\hat{x}_t$ of the continuous state x_t, and generates from that an estimate $\hat{y}_t$ of the observation. It also generates Σ_t, the covariance of the state estimate error $x_t - \hat{x}_t$, and S_t, the covariance of $y_t - \hat{y}_t$. Given these estimates, the likelihood of the observation y_t is given by a normal distribution with mean $\hat{y}_t$ and covariance S_t, i.e. $L_t = \mathcal{N}(y_t; \hat{y}_t, S_t)$.

The local dynamics propagation process generates a factored likelihood for each successor state $s^{(c,j)}$. The factored likelihood, $L_t^{(c,j)} \equiv \mathcal{N}(y_t^c; \hat{y}^{(c,j)}, S_t^{(c,j)})$, is used to compute the importance weight:

$$w_t^{(c,j)} = L_t^{(c,j)} p(z_t^{(c,j)} | z_{0:t-1}^{(c,i)}, y_{1:t-1}) \tag{7}$$

The weights of the successor states are summed together, i.e. $w_{t-1}^{(c,i)} = \sum_j w_t^{(c,j)}$, to derive the weight for the original factored particle $s_{t-1}^{(c,i)}$. The new weights $\{w_{t-1}^{(c,i)}\}$ approximate the posterior distribution of the factor. Compared to weights that are generated without look-ahead, these weights better reflect the system state because they encapsulate the extra information given by the observation at time t.

4.2 Join

If there is sufficient correlation between factors, then it is necessary to take into account these interactions during belief update. If necessary, the join step augments these factors with the correlated variables that lie outside their domains. Let Z_{t-1}^d and X_{t-1}^d denote the variables from factor d that are to be joined with factor c. We compute the equijoin of $\{z_{t-1}^{(c,i)}\}_{i=1}^N$ with $\{z_{t-1}^{(d,i)}\}_{i=1}^N$ to generate a larger set of discrete samples $\{\tilde{z}_{t-1}^{(*,i)}\}_{i=1}^M$. In essence, the equijoin operation produces a new set of factored particles, where each new particle is a sample of $Z_{t-1}^c \cup Z_{t-1}^d$, in which the values appearing in $Z_{t-1}^c \cap Z_{t-1}^d$ are forced to be equal. (For details, please refer to [Ng *et al.*, 2002], where we also discussed a sample-join operation that approximates the equijoin operation, for when the equijoin is too expensive to compute).

For each $\tilde{z}_{t-1}^{(*,i)}$, the corresponding $\tilde{\mu}_{t-1}^{(*,i)}$ is a vector that combines the elements of $\mu_{t-1}^{(c,i_1)}$ and $\mu_{t-1}^{(d,i_2)}$, where i_1 and i_2 refer to the indices of the factored particles that comprise $\tilde{z}_{t-1}^{(*,i)}$. For $X \in X_{t-1}^c \cap X_{t-1}^d$, the product of the marginal distributions over X in $\mathcal{N}(\mu_{t-1}^{(c,i_1)}, \mu_{t-1}^{(c,i_1)})$ and $\mathcal{N}(\mu_{t-1}^{(d,i_2)}, \mu_{t-1}^{(d,i_2)})$ is computed, and the resulting mean and covariance are used for $(\tilde{\mu}_{t-1}^{(*,i)}, \tilde{\Sigma}_{t-1}^{(*,i)})$.[2]

[2] This product can be easily computed for Gaussian distributions. The resulting mean μ and

The importance weight of $\tilde{s}_{t-1}^{(*,i)}$ is the product of the corresponding importance weights from its constituent factored particles, i.e. $\tilde{w}_{t-1}^{(*,i)} = w_{t-1}^{(c,i_1)} \times w_{t-1}^{(d,i_2)}$. For factors with overlapping variables, the join step serves to synchronize the information in these shared variables.

The key to performing asynchronous joins is that we join factors together only when necessary. At a given time, there may be less than K factors because some basic factors have joined to form larger, *aggregate* factors. Note that the aggregate factors may still be much smaller than the full global state, since only a few factors may be joined together at a particular time step. At a subsequent time step, other factors may be joined together, thereby achieving that in the long run information is propagated throughout the system.

4.3 Resampling

The aggregate factors are individually resampled. For each aggregate factor, N particles are sampled with replacement from $\{\tilde{s}_{t-1}^{(*,i)}\}_{i=1}^{M}$, where the probability of choosing $\tilde{s}_{t-1}^{(*,i)}$ is proportional to $\tilde{w}_t^{(*,i)}$.

4.4 Dynamics propagation

For high-dimensional hybrid systems, full dynamics propagation is an expensive procedure, especially in light of autonomous transitions that require evaluating the discrete transition probabilities at each time step. Instead of full dynamics propagation, we propagate only the lower-dimensional dynamics for aggregate factors because the aggregate factors are assumed nearly independent. From the factored look-ahead step, we have already computed the factored transition prior probabilities and the factored likelihoods for each basic factor. So whenever viable, we will reuse these factored quantities to approximate the aggregate transition prior and the aggregate likelihood.

To propagate $\tilde{\mathbf{s}}_{t-1}^{*}$, the first step is to propagate the discrete samples. To do so, one must compute the proposal distribution. A popular choice for the proposal distribution is the transition prior $p(\tilde{Z}_t^{*}|\tilde{z}_{0:t-1}^{(*,i)}, y_{1:t-1})$, given by (4). Since the factored transition priors are already computed in factored look-ahead, it may be possible to approximate the aggregate transition prior as the product of the factored priors obtained earlier. However, this is not advisable. Take as example the rover, where the left part corresponds to being at a high slope and the right corresponds to being in a deep hole. When combined, this rover configuration is highly unlikely. Because factored transitions may lead to unlikely global transitions, it is best to compute these transition prior probabilities exactly for each aggregate factor.

On the other hand, the aggregate likelihood can be approximated as the product of the factored likelihoods without introducing too much error. Like the assumption that justifies the factored representation of the belief state, the assumption that the likelihood can be well approximated by the product of factored likelihoods depends on a sensible partition of the system into weakly interacting subsystems. Unlike the transition prior probabilities, it is more reasonable to approximate the likelihood from the factored likelihoods for two reasons: The first is that local observations do not contradict each other, since they are part of the same global observation. As a result, factored likelihoods will probably not correspond to a contradictory aggregate likelihood. The second reason is that most real-world systems are often observed by local sensors. For these systems, there is no global observation with which to calculate a likelihood other than piecing together the likelihoods from local evidence. Thus, using local observations matches reality. In our rover example,

covariance Σ for the product of two Gaussian distributions, $\mathcal{N}(\mu_1, \Sigma_1) \times \mathcal{N}(\mu_2, \Sigma_2)$, are given by $\mu = \Sigma\,(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)$ and $\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$.

the observed variables are the bogey angles and the rocker angles from the left and right sides. Since the observations are localized to either the left or the right, the rover satisfies the assumption that it is observed only locally and that its likelihood can be approximated by the product of the left and right likelihoods.

Using this approximation, we can improve upon using the transition prior as the proposal distribution by using an approximately optimal proposal distribution:

$$\hat{p}(\tilde{Z}_t^* = \tilde{z}_t^{(*,j)} | \tilde{z}_{0:t-1}^{(*,i)}, y_{1:t}) = \hat{L}_t^{(*,j)} p(\tilde{z}_t^{(*,j)} | \tilde{z}_{0:t-1}^{(*,i)}, y_{1:t-1}) \tag{8}$$

where $\hat{L}_t^{(*,j)} \equiv \prod_c L_t^{(c,j_c)}$ is the product of the factored likelihoods from the constituent samples that form the aggregate factored particle. This is an approximation to the posterior distribution, which is also the optimal proposal distribution:

$$p(\tilde{Z}_t^* = \tilde{z}_t^{(*,j)} | \tilde{z}_{0:t-1}^{(*,i)}, y_{1:t}) = L_t^{(*,j)} p(\tilde{z}_t^{(*,j)} | \tilde{z}_{0:t-1}^{(*,i)}, y_{1:t-1}) \tag{9}$$

The reason that it is impractical to compute the true likelihood is because that would require performing UKF for all the aggregate discrete states.

After $\{\tilde{z}_t^{(*,i)}\}_{i=1}^N$ are generated from (8), the continuous variables are propagated by applying UKF on the aggregate dynamcs to get $\{(\tilde{\mu}_t^{(*,i)}, \tilde{\Sigma}_t^{(*,i)})\}_{i=1}^N$. The actual likelihoods are computed as well. The reason that this can be performed efficiently is because we only need to perform UKF once for the sampled state, not for all the discrete states. The actual likelihoods are used in the importance weights to correct for having sampled from an approximate posterior distribution:

$$(\tilde{w}_0)_t^{(*,i)} = \frac{p(\tilde{z}_t^{(*,i)} | \tilde{z}_{0:t-1}^*, y_{1:t})}{\hat{p}(\tilde{z}_t^{(*,i)} | \tilde{z}_{0:t-1}^*, y_{1:t})} \tag{10}$$

Since the particles were uniformly sampled from $\hat{p}(\tilde{z}_t^{(*,i)} | \tilde{z}_{0:t-1}, y_{1:t})$, the weights in (10) serve to "shift" the distribution mass of the particles to reflect the correct distribution $p(\tilde{z}_t^{(*,i)} | \tilde{z}_{0:t-1}, y_{1:t})$. Note that there is no double-counting of evidence because the effect of the initial conditioning is cancelled out by the denominator in (10).

4.5 Projection

To maintain the factored representation of the belief state, each particle $\tilde{s}_t^{(*,i)}$ is projected back as sets of samples over the basic factors. This is done by projecting $\{\tilde{z}_t^{(*,i)}\}_{i=1}^N$ onto the factored discrete domain and then projecting $\{\tilde{\mu}_t^{(*,i)}, \tilde{\Sigma}_t^{(*,i)}\}_{i=1}^N$ onto the factored continuous domain. This has the effect of zero-ing out all the statistics that are irrelevant to the factor. The factored particles inherit the weight of its parent aggregate factor, $(w_0)_t^{(c,i)} = \sqrt[n^*]{(\tilde{w}_0)_t^{(*,i)}}$, where n^* is the number of basic factors that have been joined to form the aggregate factor. Since the weight of the aggregate factor was originally obtained from multiplying the weights of the basic factors, taking the $(n^*)^{\text{th}}$ root of the aggregate weight is an inverse operation that gets us back to the basic factored weights. The weight $(w_0)_t^{(c,i)}$ is then multiplied into $w_t^{(c,i)}$ during factored look-ahead in the next iteration.

Effect of join and projection In probabilistic terms, the projection step marginalizes the distribution represented by the joined particles to the local distributions over the factors. The join step is equivalent to taking the product of these marginal distributions, as represented by the sets of the factored particles. Thus the combination of these two steps is equivalent to taking the product of marginals of the original distribution. This has the effect of de-correlating the dependencies between factors, while maintaining the same marginal distributions over the factors as the original distribution. This is key to preserving the tractability of the belief state representation over time.

4.6 Dynamics projection

In HYBRID-FP, local dynamics models are needed by UKF to propagate the continuous variables in factored look-ahead. Here, we briefly summarize the method by which we project the dynamics of a hybrid system to those of its constituent factors. The method places no constraints on the form of the dynamics equation and does not require any unwieldy linearization of the dynamics equation. Generally speaking, the method derives the local dynamics equation by computing the expectation of the global dynamics equations over the non-local variables, conditional upon the value of the local variables. The result is an "expected" equation that incorporates probabilistic information about the non-cluster variables in the projected dynamics.

First, replace each global guard condition G with a local guard condition that consists only of the guard events mentioning the local variables X_t^c. Then, we derive the local dynamics equations that govern the evolution of X_t^c. For a given guard condition G, assume that the global state X_t evolves according to $X_t = f_G(X_{t-1})$. Let $X_t^{\bar{c}}$ denote variables outside of factor c. First, we select the global dynamics equations whose guard condition G contains G^c, the local guard event of interest. Let $G^{\bar{c}} \equiv G \backslash G^c$ denote the guard events in G that involve only $X_t^{\bar{c}}$. Then we replace $X_t^{\bar{c}}$ with the conditional expected value $\mathrm{E}[X_{t-1}^{\bar{c}} | X_{t-1}^c]$ to insure that the equations no longer depend explicitly on $X_{t-1}^{\bar{c}}$. Then we weight each of these equations by the conditional probability $\Pr(G^{\bar{c}} | X_{t-1}^c)$ and take the sum of these expressions. For the local guard G^c, the projected equation over the variables X_t^c is given by:

$$X_t^c = \sum_G \Pr(G^{\bar{c}} | X_{t-1}^c) \cdot f_G(X_{t-1}^c, \mathrm{E}[X_{t-1}^{\bar{c}} | X_{t-1}^c])$$

where the sum is taken over G, the set of global guard conditions that contains the guard events G^c.

This procedure is best illustrated with an example. Let's consider a simple model consisting of two tanks. Depending on on the water levels, water will flow from one tank to another.

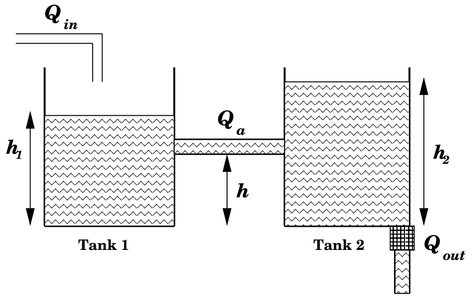

Figure 5: A simple two-tank system

There are four discrete states, where each corresponds to the system state when a unique set of guard events is satisfied.

$$z_t = \begin{cases} 0 & \text{if} \quad h_1 < h,\ h_2 < h \\ 1 & \text{if} \quad h_1 \geq h,\ h_2 < h \\ 2 & \text{if} \quad h_1 < h,\ h2 \geq h \\ 3 & \text{if} \quad h_1 \geq h,\ h_2 \geq h \end{cases}$$

The continuous variables are h_1 and h_2. Depending on z_t, the inter-tank flow Q_a exhibits different behavior. The system equations are given as follows:

$$Q_{in} = C_{in} u(t)$$

$$Q_{out} = C_{out} \sqrt{\rho g h_2} + \xi(t)$$

$$Q_a(h_1, h_2) = \begin{cases} 0, & \text{if } z_t = 0 \\ C_a \sqrt{\rho g (h_1 - h)} & \text{if } z_t = 1 \\ -C_a \sqrt{\rho g (h_2 - h)}, & \text{if } z_t = 2 \\ \text{sign}(h_1 - h_2) C_a \sqrt{\rho g |h_1 - h_2|}, & \text{if } z_t = 3 \end{cases}$$

$$\begin{aligned} \dot{h}_1 &= \frac{1}{A}(Q_{in} - Q_a(h_1, h_2)) \\ \dot{h}_2 &= \frac{1}{A}(Q_a(h_1, h_2) - Q_{out}) \end{aligned} \tag{11}$$

where C_{in}, C_{out} and C_a are linear gains; A is the area of the tank's cross-section; ρ is the density of water and g is the acceleration of gravity.

To derive the dynamics model local to Tank 1, we note that only h_1 is relevant to this tank. Therefore, the local guard conditions for this subsystem are those that involve h_1, namely ($h_1 < h$) and ($h_1 \geq h$). For the discrete state corresponding to $h_1 < h$, the system evolves according to the following:

$$\begin{aligned} Q_a(h_1) &= \text{Pr}(h_2 < h | h_1) \cdot \text{E}[0 | h_1] + \text{Pr}(h_2 \geq h | h_1) \cdot \text{E}[-C_a \sqrt{\rho g (h_2 - h)} | h_1] \\ &= -\text{Pr}(h_2 \geq h | h_1) \cdot C_a \sqrt{\rho g (\text{E}[h_2 | h_1] - h)} \end{aligned}$$

4.7 Further approximations

The two most expensive operations in HYBRID-FP are the join step and the computation of discrete transition probabilities in the dynamics propagation step. For systems with an extremely large state space, these two operations may take a long time to compute. As a result, we present approximations to these two operations, thus allowing for a wider applicability of the HYBRID-FP algorithm.

Approximate join This approximation is aimed at optimizing the join operation by replacing equijoin with a new operation called *sample-join*. This idea, introduced in [Ng *et al.*, 2002], uses an importance sampling method for sampling from the product of marginal distributions. The aggregate particle is instantiated incrementally as follows: The method samples a consistent particle from a basic factor and multiplies the weight of the aggregate particle by the fraction of consistent particles from the factor. The weighting compensates for the fact that the aggregate particle is sampled from subsets of consistent factored particles instead of being generated from an exact equijoin of the factored particles.

***Sparse Mode Transition Probabilities* (SMP)** Instead of calculating the transition probabilities for all the discrete aggregate states, this approximation calculates the transition probabilities for only a sampled subset of the states. (The probabilities for any non-sampled states is 0). The sampled subset of states is constructed from samples of factored states chosen according to the factored transition probabilities (computed during factored lookahead). Then, the transition probabilities are computed only for this sampled subset, instead of for all the aggregate discrete states. In essence, when computational resources are limited, this approximation allows one to consider only the most likely aggregate sucessor states for dynamics propagation.

***Factored Mode Propagation* (FMP)** Recall our justification against approximating the discrete transition prior in (4) as the product of factored priors from factored look-ahead. Since local transitions may lead to unlikely global transitions, this approach was not advisable in general models. However, if all discrete dynamics are self-contained with the factors (i.e. there are no interaction involving discrete variables between the factors), then local transitions are completely independent and will not lead to an unlikely global transition when combined. Thus, for these models with this particular structure, it is possible to approximate the aggregate transition prior probabilities as the product of the factored priors. In essence, this is the same as being able to propagate the discrete variables in the factored state space instead of the aggregate state space. But since the continuous variables from different factors may be correlated, we still need to propagate the continuous dynamics in the aggregate state space.

The approximation works as follows: For each factored particle, we propagate $z_{t-1}^{(c,i)}$ by sampling from the factored importance weights $\{w_{t-1}^{(c,j)}\}$. These weights, which were computed during factored look-ahead, reflect the posterior distribution of the factored discrete state. The factored particles $\{z_t^{(c,i)}\}_{i=1}^N$ are then joined, if necessary, and resampled. The dynamics propagation step now involves only the propagation of the continuous variables in the aggregate factor. As a result of having generated the aggregate discrete samples $\{\tilde{z}_t^{(*,i)}\}_{i=1}^N$ directly from the factored samples, the weights in (10) are now given as:

$$(w_0)_t^{(*,i)} \quad = \quad \frac{p(\tilde{z}_t^{(*,i)}|\tilde{z}_{0:t-1}^*, y_{1:t})}{\hat{p}(\tilde{z}_t^{(*,i)}|\tilde{z}_{0:t-1}^*, y_{1:t})} = \frac{L_t^{(*,i)} p\left(\tilde{z}_t^{(*,i)}|\tilde{z}_{0:t-1}^*, y_{1:t-1}\right)}{\left(\prod_c w_t^{(c,i_c)} \text{ s. t. } \tilde{z}_t^{(*,i)} = \ \bowtie z_t^{(c,i_c)}\right)}$$

4.8 Sources of approximation

There are three main sources of approximation in HYBRID-FP: The first stems from the factored particles representation, which assumes that the belief state can be approximated as the product of factored belief states. The second is the factored look-ahead prediction, which approximates the effect of full look-ahead prediction. The third is the assumption that the aggregate likelihood can be approximated from the factored likelihoods, which is used to justify sampling z_t from an approximate proposal distribution. As explained, these approximations are necessary for making inference tractable for large hybrid systems. Often, if the factors are chosen to reflect weakly-interacting subsystems, these approximations do not introduce too much error to the belief state.

5 Experimental results

5.1 Early results without asynchronous sampling

We present early results that assessed the performance of basic HYBRID-FP algorithm on models that operate on a single time granularity. In these experiments, asynchronous inference was not performed because all factors operate on the same time scale. The two simulated domains that we used were an adaptation of the predator-prey model and a coupled tanks model. The models had much correlation between the factors so all factors were joined at every iteration of the HYBRID-FP algorithm.

Both the simulations and the experiments were implemented in Matlab. The experiments were run on shared Linux computers, where each computer has dual 2.4GHz XEON processors with 512KB cache and 2.5GB of memory. For both experiments, the maximum a posteriori (MAP) estimate is chosen as the most likely hypothesis to the unknown state.

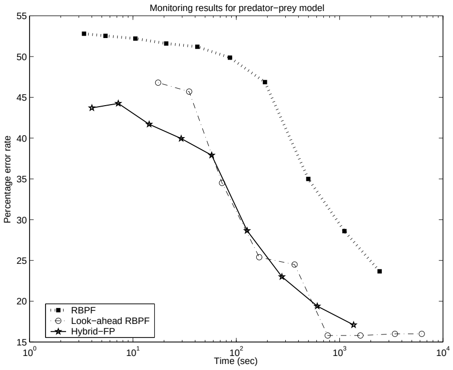

Figure 6: Predator-prey results.

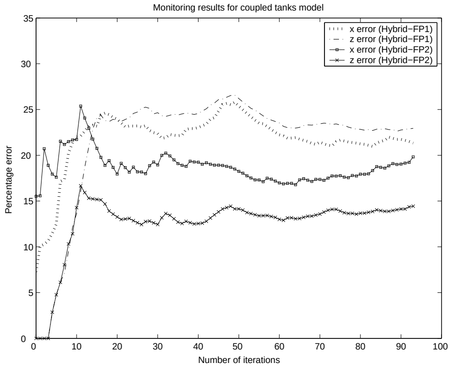

Figure 7: Coupled tanks results.

The predator-prey model contains four species divided into two groups. Each group contains two species that will cooperate or compete with each other for resources. At any time, each group takes on the role of prey or predator, depending on its population relative to the other group. The system dynamics is an extension of the Lotka-Volterra [Lotka, 1925] two-species model, where we have augmented the model with switching predator/prey roles between the group and cooperative/competitive behaviors within the group.

The continuous state $[x_1, x_2, x_3, x_4]$ represents the populations of the four species. The discrete state $[z_1, z_2, z_3]$ represents the species' modes of behavior where z_1 and z_2 denote the cooperative/competitive state of group 1 and group 2 respectively and z_3 denotes the assignment of predator/prey roles. The observation $[y_1, y_2]$ is a noisy measurement of the group populations. We have tailored this domain to have a sufficiently small state space so that it would be computationally feasible for RBPF and look-ahead RBPF. Since this model's discrete state space is quite small, HYBRID-FP provides only modest computational savings. Despite its size, this model is still an interesting challenge for HYBRID-FP because the dynamics between the two groups is quite coupled and does not offer the weakly-interacting structure that would be optimal for the factored representation. Yet, despite the approximations in HYBRID-FP, its accuracy comes quite close to look-ahead RBPF in Figure 6. The error rate is defined as the percentage of time steps in which the MAP estimate differed from actual state.

Our second experimental domain is a simulated coupled tanks model consisting of twelve tanks arranged in three ring-structured components of four tanks each, as shown in Figure 8. Input flow is introduced into the system through Tanks 1 and 12. Observations come from noisy measurements of the output flow of Tanks 3, 7 and 11. The number next to each edge denotes the height of the connecting pipe. The system is clustered into three subsystems, as shown by the dotted lines. Weak interactions are simulated by setting the inter-subsystem pipe levels higher than those of the intra-system pipes.

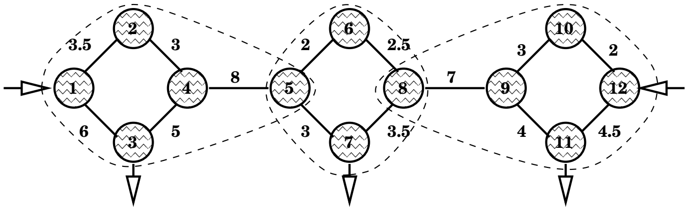

Figure 8: Coupled tanks model.

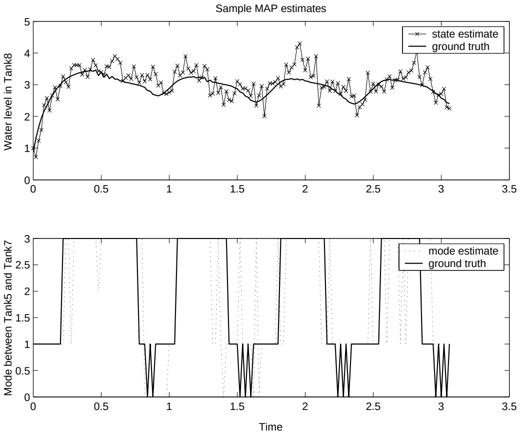

Figure 9: Sample tracking results for coupled tanks domain.

The model consists of twelve continuous variables, that correspond to the water levels in the tanks, and fourteen 4-valued discrete variables, that correspond to the behavior of inter-tank flow of each connecting pipe.[3] Each tank i is characterized by a continuous variable h_i that represents the water level and a set of discrete modes that represents the flux of water transferred between neighboring tanks through connecting pipes. The system equations for each tank is similar in form to those given in Equation (11). The main difference is that the inter-tank flow Q_a now incorporates contributions from all the neighboring tanks,

$$Q_a(h_i, h_{\mathcal{T}_i}) = \sum_{j \in \mathcal{T}_i} Q_a(h_i, h_j)$$

where $\mathcal{T}_i$ represents the set of neighboring tanks to tank i.

Besides the challenge of autonomous transitions, the sheer size of the discrete state space – 4^{14} states – makes this domain impossible to monitor with RBPF or look-ahead RBPF within Matlab's capabilities. As a result, we present results from two approximate versions of HYBRID-FP. In the FMP version, we treat the sets of factored discrete variables as completely independent and propagate the discrete states entirely in the factored space. The SMP version propagates the discrete states using a sparse transition distribution based on sampled states.

We ran the two algorithms on the same simulated data with the same initial condition for 100 time steps that span about 3 time units of the simulated data. In Figure 9, we plot a small representative subset of the tracked results against the ground truth from a typical run of FMP with 50 particles.

Figure 7 shows the relative error of FMP and SMP averaged over time. In theory, SMP should outperform FMP because FMP considers the factored states as completely independent while SMP still takes into account the correlation between the factor states, even if only for a subset of the aggregate states. In fact, one can interpret FMP as a special case of SMP, in which the sample set of aggregate states is of size 1 and consists only of the particle that resulted from joining the discrete states sampled from the factored probabilities. The fact that SMP outperforms FMP experimentally confirms this observation that FMP is an approximation of SMP.

[3] [Lerner *et al.*, 2000] describes a hybrid monitoring approach that combines BK with state collapsing and pruning for CLGs. The authors experimented on a similar tank model (consisting of 5 tanks linked in parallel) that does not exhibit autonomous transitions.

Tank	FMP50	FMP20	SMP20
1	0.1481	0.1777	0.1203
2	0.1931	0.2950	0.1574
3	0.1296	0.1139	0.1180
4	0.2729	0.3644	0.1603
5	0.2661	0.2513	0.3445
6	0.4304	0.4659	0.4271
7	0.0985	0.0897	0.1041
8	0.2293	0.1799	0.2717
9	0.1616	0.1418	0.1023
10	0.3286	0.3119	0.2321
11	0.0623	0.0589	0.0522
12	0.0958	0.1100	0.0685

Figure 10: Relative errors for each tank

Table 10 displays the relative errors for each tank. The number next to FMP or SMP denotes the number of particles used in the experiments. Tank 6 was especially difficult due to the oscillatory dynamics of h_6. Almost all of Tank 6's discrete states were identified correctly so the tracking error was mainly due to UKF, which was responsible for the continuous state tracking.

In general, the two approximations of HYBRID-FP were able to track the belief state quite well, with an average relative error of 20%, using only 20 particles. For a hybrid system with autonomous transitions and such a large discrete state space, this result is encouraging. From the experiments, we observe that tracking error usually occurs along the boundaries of state transitions, which often lead to incorrect discrete state estimates. When the discrete state is misdiagnosed, error is introduced in the continuous state estimate because the continuous state is incorrectly propagated using the wrong dynamics model. Subsequently, this affects the estimation of the transition probabilities for the next mode, due to the effect of autonomous transitions. Intuitively, this should mean that the degradation in accuracy would be perpetuated over time. Instead, we observe that recovery from these misdiagnoses was relatively fast, due to the benefit of look-ahead prediction.

5.2 Results of asychronous sampling on the rover model

We evaluate the performance of HYBRID-FP with look-ahead on the rover model. Both the simulations and the experiments were implemented in Matlab. The experiments were run on a Windows XP machine with an AMD Athlon XP-M 2000+ processor and 480 Mb of ram.

The rover model is partitioned into three basic factors. The background factor is parent to the speed variable that appears in both the left and the right factors. In addition to the speed variable, the left and the right factors share the rover roll angle. The roll angle is a measure of the correlation between the left and the right factors. Large changes in the roll signify strong coupling between the two factors and this correlation needs to be taken into account. At every time step, the left and the right factors are updated independently or as an aggregate factor, depending on whether the roll angle has changed significantly since the two factors were last joined. At every 25th time step, we reason about the background factor and propagate its effect on the speed variable to the left and right factors.

In Figure 11, we compare the two versions of HYBRID-FP (with and without the look-ahead) and also compare the look-ahead version against look-ahead RBPF. The tracking error is taken to be the difference between the simulated ground truth and the average of

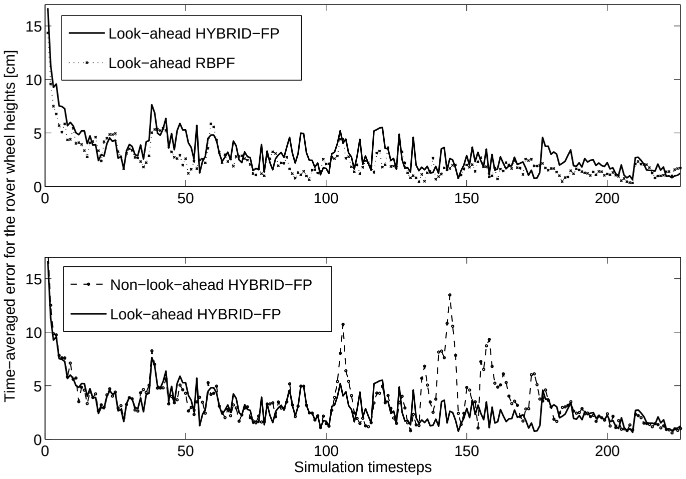

Figure 11: Comparisons against look-ahead HYBRID-FP

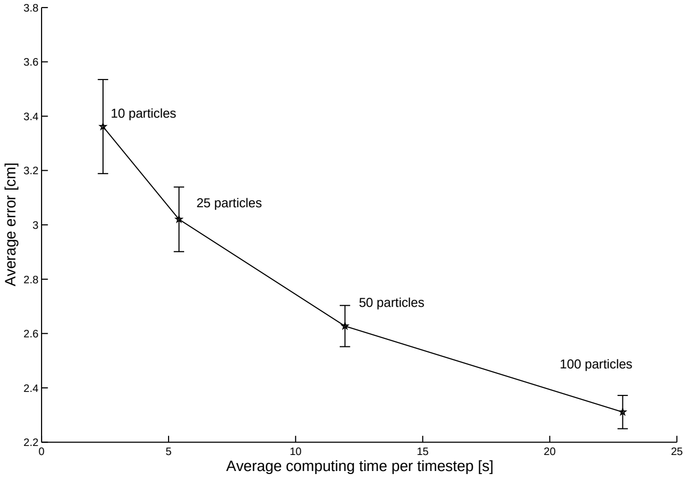

Figure 12: Results with varying number of particles

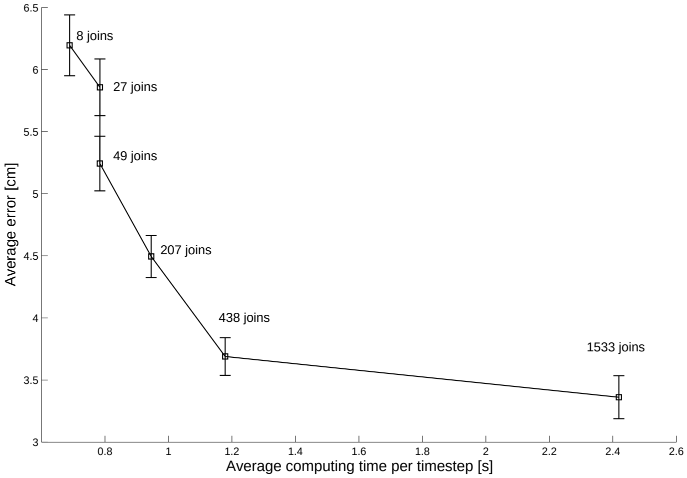

Figure 13: Results with varying join frequencies

the particles. All algorithms were run with 10 particles. We see on the top plot of Figure 11 that the performance of look-ahead HYBRID-FP closely approximates that of look-ahead RBPF, which confirms the soundness of the approximations taken in our algorithm. For a small improvement in monitoring accuracy, RBPF takes about 35 times as long as HYBRID-FP to complete one iteration of inference. The reason is due to the fact that RBPF has to perform global look-ahead over the full space of 2304 discrete states while HYBRID-FP needs to only consider 52 states during factored look-ahead. The bottom plot of Figure 11 shows the impact of look-ahead in HYBRID-FP. Not surprisingly, we see that the look-ahead version of HYBRID-FP performs better because factored look-ahead allows the particles to be propagated using an approximation to the optimal proposal distribution while in the other version, particles are propagated using the transition prior distribution. The discrepancy in performance is especially apparent when multiple wheels become stuck because the look-ahead importance weights are able to better capture this unlikely scenario.

In Figures 12 and 13, we report the monitoring accuracy and runtime for different parametrizations of the look-ahead HYBRID-FP experiments. We ran experiments with look-ahead HYBRID-FP using different numbers of particles and using different threshold values that affect the frequency of joins among the factors. Figure 12 shows the results for different numbers of particles, while holding the join threshold fixed. The number next to each data point corresponds to the number of particles used. We see that as the number of particles increase, the monitoring accuracy is improved but at the expense of increased runtime. In Figure 13, we show the results for varying the threshold for the change in roll angle, which is used as a heuristic measure to gauge the amount of interaction between the left and the right factors. Setting this threshold to a low value has the effect of increasing the frequency of joins. The number next to each data point denotes the number joins that occurred during sample runs with 10 particles. We see that beyond a certain number of joins, the added benefit of each join becomes marginal. This is because we are joining the factors even when their correlation is small.

6 Conclusion

This work presents an extension of factored particles that allows for efficient monitoring of complex hybrid systems, including those that exhibit autonomous transitions and those that operate at multiple time granularities. By leveraging the factored particle representation, HYBRID-FP is able to reason about each factor independently at its own time granularity and join these factors only when the factors are sufficiently correlated. This feature of asynchonous inference, along with the novel idea of factored look-ahead, makes HYBRID-FP an efficient and versatile method for resource-bounded inference in large hybrid systems.

Currently, this work assumes that the factors evolve at known fixed time granularities and that the system dynamics are specified only for the set of known time granularities. To handle systems with degradable or renewable components, whose time granularities may vary over time, time granularities will need to be learned on the fly. In addition, one cannot use the same dynamics model for different time granularities. Since time plays an important role that affects how systems may change from one time point to another, one must find an efficient way to "scale" the system dynamics to different time granularities. For future work, we plan to investigate the learning of time granularities and the automatic recomputation of the dynamics model for different time granularities.

Acknowledgments

This work was supported by the NASA IS/AR grant NCC2-1236 and the NASA Ames/RIACS 2003 Summer Student Research Program. We also would like to thank Frank Hutter for his mathematical insights and discussions during the formative stage of

this work.

References

[Boyen and Koller, 1998] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proc. of the 14th UAI*, 1998.

[de Freitas *et al.*, 2004] N. de Freitas, R. Dearden, F. Hutter, R. Morales-Mendez, J. Mutch, and D. Poole. Diagnosis by a waiter and a mars explorer. In *Proc. of IEEE, Special Issue on Sequential State Estimation*, 2004.

[Doucet *et al.*, 2000a] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proc. of the 16th UAI*, 2000.

[Doucet *et al.*, 2000b] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 2000.

[Friedman *et al.*, 1998] N. Friedman, D. Koller, , and A. Pfeffer. Structured representation of complex stochastic systems. In *Proc. of the 15th AAAI*, 1998.

[Funiak and Williams, 2003] S. Funiak and B. Williams. Multi-modal particle filtering for hybrid systems with autonomous mode transitions. In *Proc. of the 14th Intl. Workshop on Principles of Diagnosis*, 2003.

[Genz, 2002] A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 2002.

[Grewal and Andrews, 2001] M. Grewal and A. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley-Interscience, 2001.

[Hacot, 1998] H. Hacot. Analysis and traction control of a rocker-bogie planetary rover. Master's thesis, MIT, 1998.

[Koutsoukos *et al.*, 2002] X. Koutsoukos, J. Kurien, and F. Zhao. Monitoring and diagnosis of hybrid systems using particle filtering methods. In *Proc. of the 15th Intl Symposium on MTNS*, 2002.

[Kwok *et al.*, 2004] C. Kwok, D. Fox, and M. Meila. Real-time particle filter. In *Proc. of the IEEE*, 2004.

[Lerner and Parr, 2001] U. Lerner and R. Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proc. of the 17th UAI*, 2001.

[Lerner *et al.*, 2000] U. Lerner, R. Parr, D. Koller, and G. Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *Proc. of the 17th AAAI*, 2000.

[Lerner *et al.*, 2002] U. Lerner, B. Moses, M. Scott, S. McIlraith, and D. Koller. Monitoring a complex physical system using a hybrid dynamic bayes net. In *Proc. of the 18th UAI*, 2002.

[Lerner, 2002] U. Lerner. *Hybrid Bayesian Networks for Reasoning About Complex Systems*. PhD thesis, Stanford University, 2002.

[Lotka, 1925] A. Lotka. *Elements of Physical Biology*. Williams and Wilkins Co., 1925.

[Ng *et al.*, 2002] B. Ng, L. Peshkin, and A. Pfeffer. Factored particles for scalable monitoring. In *Proc. of the 18th UAI*, 2002.

[Verma *et al.*, 2003] V. Verma, S. Thrun, and R. Simmons. Variable resolution particle filter. In *Proc. of the 18th IJCAI*, 2003.

[Wan and van der Merwe, 2000] E. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *Proc. of IEEE Symposium*, 2000.

[Willeke and Dearden, 2004] T. Willeke and R. Dearden. Building hybrid rover models: Lessons learned. In *Proc. of the 15th Intl. Workshop on Principles of Diagnosis*, 2004.