THE GAUSSIAN PARTICLE FILTER FOR DIAGNOSIS OF NON-LINEAR SYSTEMS

Frank Hutter * **Richard Dearden** **

* *Technische Universität Darmstadt, Fachbereich Informatik, D-64289 Darmstadt, Germany, mail@fhutter.de*
** *RIACS / NASA Ames Research Center, M.S. 269-3 Moffett Field, CA 94035 USA,dearden@email.arc.nasa.gov*

Abstract: Fault diagnosis is a critical task for autonomous operation of systems such as spacecraft and planetary rovers, and must often be performed on-board. Unfortunately, these systems frequently also have relatively little computational power to devote to diagnosis. For this reason, algorithms for these applications must be extremely efficient, and preferably anytime. In this paper we introduce the *Gaussian particle filter* (GPF), an efficient variant on the particle filtering algorithm for non-linear hybrid systems. Each particle samples a discrete mode and approximates the continuous variables by a multivariate Gaussian that is updated at each time-step using an unscented Kalman filter. The algorithm is closely related to Rao-Blackwellized Particle Filtering and equally efficient, but is more broadly applicable. We show that GPF performs diagnosis faster than traditional particle filters, and with a much lower rate of incorrect diagnoses.

1. INTRODUCTION

Fault diagnosis is a critical task for autonomous operation of systems such as spacecraft and planetary rovers. The diagnosis problem is to determine the state of a system over time given a stream of observations of that system. A common approach to this problem is *model-based diagnosis* (de Kleer and Williams, 1987; de Kleer and Williams, 1989), in which the overall system state is represented as an assignment of a *mode* (a discrete state) to each component of the system. Such an assignment is a possible description of the current state of the system if the set of models associated with the modes is consistent with the observed sensor values. One example of such a system is Livingstone (Williams and Nayak, 1996), which flew on the Deep Space One spacecraft as part of the Remote Agent Experiment in May 1999. In Livingstone, diagnosis is performed by maintaining a candidate hypotheses (in other systems more than one hypothesis is kept) about the current state of each system component, and comparing the candidate's predicted behaviour with the system sensors. Traditional model-based diagnosis operates on discrete models only, and uses *monitors* to discretize continuous sensor readings.

For many applications, e.g. planetary rovers, the complex dynamics of the system make reasoning with a discrete model inadequate. This is because too fine a discretization is required to accurately model the system; because the monitors would need global sensor information to discretize a single sensor correctly; and because transient events must be diagnosed. To overcome this we need to reason directly with the continuous values we receive from sensors: Our model needs to be a hybrid system.

A hybrid system consists of a set of discrete *modes*, which represent fault states or operational modes of the system, and a set of continuous variables which model the continuous quantities that affect system behaviour. We will use the term *state* to refer to the combination of these, that is, a state is a mode plus a value for each continuous variable, while the *mode* of a system refers only to the discrete part of the state. In many cases, not all of the hybrid system will be observable. Therefore, we also have an observation function that defines the

likelihood of an observation given the mode and the values of the continuous variables. All these processes are inherently noisy, and the representation reflects this by explicitly including noise in the continuous values, and stochastic transitions between system modes. We describe our hybrid model in more detail in Section 2.

There are several challenges to overcome to produce an effective diagnosis algorithm for these types of hybrid models. The algorithm we present here attempts to make progress on all these problems, although primarily on the first four:

Very low prior fault probabilities: Diagnosis problems are particularly difficult for approximation algorithms based on sampling because of the low probabilities of transitions to fault states. This can lead to incorrect diagnoses because there are no samples in a fault state.

Restricted computational resources: For space applications, computation time is frequently at a premium, and on-board real-time diagnosis is often necessary. For this reason, diagnosis must be as efficient as possible.

Non-linear stochastic transitions and observations: Many algorithms are restricted to linear models with Gaussian noise. Our domains frequently behave non-linearly, so we would prefer an algorithm without this restriction.

Multimodal system behaviour: Even in a single discrete mode, the observations are often consistent with several values for the continuous variables, and so multi-modal distributions appear. For example, when a rover is commanded to accelerate, we are often uncertain about exactly when the command is executed. Different start times lead to different estimates of current speed, and hence a multi-modal distribution. Again, this is a problem for a number of algorithms, particularly for Kalman filters.

High dimensional state spaces: As the dimensionality of a problem grows, the number of samples required to accurately approximate the posterior distribution grows exponentially.

Computing exact diagnoses for a model such as the one we describe above is computationally intractible. To overcome this, a number of authors have proposed approximate inference algorithms (Verma *et al.*, 2001; Washington, 2000). The most general approach proposed is the particle filter (Isard and Blake, 1998; Doucet, 1998) which sequentially computes an approximation to the posterior probability distribution of the states of the system given the observations. The posterior distribution is approximated by a set of point samples or *particles*. We discuss particle filters in Section 2.2, for a much more extensive review see (Doucet *et al.*, 2001).

An increasingly commonly used variant on particle filters is the Rao-Blackwellized particle filter (RBPF) (de Freitas, 2002), which we describe in Section 2.3. RBPF is a much more computationally efficient variant of PF, but is only applicable to a restricted set of models. In this paper we present the *Gaussian particle filter*, an algorithm which maintains the considerable computational gains of RBPF, but can be used to diagnose the state of any hybrid system. We describe the Gaussian particle filter in Section 3, and demonstrate its effectiveness in Section 4.

2. HYBRID STATE ESTIMATION

Following a number of authors, we model the system to be diagnosed as a discrete-time probabilistic hybrid automaton (PHA). We refer the reader to (Hofbaur and Williams, 2002) for details of PHAs, but will use the following notation here:

- $Z = z_1, \ldots, z_n$ is the set of discrete modes the system can be in.
- $X = x_1, \ldots, x_m$ is the set of continuous state variables which capture the dynamic evolution of the automaton. We write $P(Z_0, X_0)$ for the prior distribution over Z and X.
- Y is the set of observable variables. We write $P(Y_t|z_t, x_t)$ for the distribution of observations in state (z_t, x_t).
- There is a transition function that specifies $P(Z_t|z_{t-1}, x_{t-1})$, the conditional probability distribution over modes at time t given that the system is in state (z, x) at $t-1$. In some systems, this is independent of the continuous variables: $P(Z_t|z_{t-1}, x_{t-1}) = P(Z_t|z_{t-1})$.
- We write $P(X_t|z_{t-1}, x_{t-1})$ for the distribution over X at time t given that the system is in state (z, x) at $t-1$.

We denote a hybrid state of the system by $s = (z, x)$, which consists of a discrete mode z, and an assignment to the state variables x.

Diagnosis of a hybrid system of this kind is determining, at each time-step, the *belief state* $P(S_t|y_{1:t})$, a distribution that, for each state s, gives the probability that s is the true state of the system, given the observations so far. In principle, belief state tracking is an easy task, which can be performed using the *forward pass* equation:

$$\begin{aligned} P(s_t|y_{1:t}) &= \alpha P(y_t|s_t) \int P(s_t|s_{t-1}) P(s_{t-1}|y_{1:t-1}) ds_{t-1} \\ &= \alpha P(y_t|z_t, x_t) \\ &\int P(x_t|z_t, x_{t-1}) P(z_t|z_{t-1}, x_{t-1}) P(s_{t-1}|y_{1:t-1}) ds_{t-1} \end{aligned}$$

where α is a normalizing constant. Unfortunately, computing the integral exactly is intractable in all but the smallest of problems, or in certain special cases. The most important special case is a unimodal linear model with Gaussian noise. This is solved optimally and efficiently by the Kalman filter (KF). We describe the KF below; then, we weaken the model restrictions and describe algorithms for more general models, such as Particle Filters and Rao-Blackwellized Particle Filters. We

(1) For N particles $p^{(i)}$, $i = 1, \ldots, N$, sample discrete modes $z_0^{(i)}$, from the prior $P(Z_0)$.
(2) For each particle $p^{(i)}$, sample $x_0^{(i)}$ from the prior $P(X_0|z_0^{(i)})$.
(3) for each time-step t do
  (a) For each particle $p^{(i)} = (z_{t-1}^{(i)}, x(i)_{t-1})$ do
    (i) Sample a new mode: $\hat{z}_t^{(i)} \sim P(Z_t|z_{t-1}^{(i)})$.
    (ii) Sample new continuous parameters: $\hat{x}_t^{(i)} \sim P(X_t|\hat{z}_t^{(i)}, x_{t-1}^{(i)})$.
    (iii) Compute the weight of particle $\hat{p}^{(i)}$: $w_t^{(i)} \leftarrow P(y_t|\hat{z}_t^{(i)}, \hat{x}_t^{(i)})$.
  (b) Resample N new samples $p^{(i)}$ where: $P(p^{(i)} = \hat{p}^{(k)}) \propto w_t^{(k)}$

Fig. 1. The particle filtering algorithm.

end with the most general problem for which we propose the Gaussian Particle Filter.

2.1 Kalman Filters

When the system we want to diagnose has only one discrete mode, linear transition and observation functions for the continuous parameters and Gaussian noise there exists a closed form solution to the tracking problem. In this case, the belief state is a multivariate Gaussian and can be computed incrementally using a *Kalman filter* (KF). At each time-step t the Kalman filtering algorithm updates sufficient statistics $(\mu_{t-1}, \Sigma_{t-1})$, prior mean and covariance of the continuous distribution, with the new observation y_t. We omit details and the Kalman equations here, and refer interested readers to (Grewal and Andrews, 1993).

The Kalman filter is an extremely efficient algorithm. However, in the case of non-linear transformations it does not apply; good approximations are achieved by the *extended Kalman filter* (EKF) and the *unscented Kalman filter* (UKF) with the UKF generally dominating the EKF (Wan and van der Merwe, 2000). Rather than using the standard Kalman filter update to compute the posterior distribution, the UKF performs the following: Given an m-dimensional continuous space, $2m+1$ *sigma points* are chosen based on the a-priori covariance (see (Wan and van der Merwe, 2000) for details). The non-linear system equation is then applied to each of the sigma points, and the a-posteriori distribution is approximated by a Gaussian whose mean and covariance are computed from the sigma points. This unscented Kalman filter update yields an approximation of the posterior whose error depends on how different the true posterior is from a Gaussian. For linear and quadratic transformations, the error is zero.

2.2 Particle Filters

While the success of the above approaches depend on how strongly the belief state resembles a multivariate Gaussian, the *particle filter* (PF) (Isard and Blake, 1998) is applicable regardless of the underlying model. A particle filter is a Markov chain Monte Carlo algorithm that approximates the belief state using a set of samples (particles), and keeps the distribution updated as new observations are made over time. The basic PF algorithm is shown in Figure 1. To update the belief distribution given a new observation, the algorithm operates in three steps as follows:

The Monte Carlo step: This step considers the evolution of the system over time. It uses the stochastic model of the system to generate a possible future state for each sample. In our hybrid model (and Figure 1), this is performed by sampling a discrete mode, and then the continuous state given the new mode.

The reweighting step: This corresponds to conditioning on the observations. Each sample is weighted by the likelihood of seeing the observations in the (updated) state represented by the sample. This step leads samples that predict the observations well to have high weight, and samples that are unlikely to generate the observations to have low weight.

The resampling step: To produce a uniformly weighted posterior, we then resample a set of uniformly weighted samples from the distribution represented by the weighted samples. In this resampling the probability that a new sample is a copy of a particular sample s is proportional to the weight of s, so high-weight samples may be replaced by several samples, and low-weight samples may disappear.

At any time t, the PF algorithm approximates the true posterior belief state given observations $y_{1:t}$ by a set of samples (or particles):

$$P(Z_t, X_t|y_{1:t}) \approx \hat{P}(Z_t, X_t|y_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} w_t^{(i)} \delta_{(Z_t,X_t)}((z_t^{(i)}, x_t^{(i)}))$$

where $w_t^{(i)}$, $z_t^{(i)}$ and $x_t^{(i)}$ are weight, discrete mode and continuous parameters of particle $p^{(i)}$ at time t, N is the number of samples, and $\delta_x(y)$ denotes the Dirac delta function.

Particle filters have a number of properties that make them a desirable approximation algorithm for diagnosis. As we said above, unlike the Kalman filter, they can be applied to non-linear models with arbitrary prior belief distributions. They are also *contract anytime* algorithms, meaning that if you specify in advance how much computation time is available, a PF algorithm can estimate a belief distribution in the available time—by changing the number of samples, you trade off computation time for the quality of the approximation. In fact, the computational requirements of a particle filter depend *only* on the number of samples, not on the complexity of the model.

Unfortunately, as we said in the introduction, diagnosis problems have some characteristics that

make standard particle filtering approaches less than ideal. In particular, on-board diagnosis for applications such as spacecraft and planetary rovers must be performed using very limited computational resources, and transitions to fault modes typically have very low probability of occurring. This second problem leads to a form of *sample impoverishment*, in which modes with a non-zero probability of being the actual state of the system contain no samples, and are therefore treated by the particle filter as having zero probability. This is particularly a problem for diagnosis, because these are exactly the states for which we are most interested in estimating the likelihood. There have been a few approaches to tackling this issue, most notably (Dearden and Clancy, 2002) and (Thrun *et al.*, 2001).

Another traditional problem of particle filters is that the number of samples needed to cope with high dimensional continuous state spaces is enormous. Especially in the case of high noise levels and widespread distributions, approximations via sampling do not yield good results. If it is possible to represent the continuous variables in a compact way, e.g. in the form of sufficient statistics, this generally helps by greatly reducing the number of particles needed. In the next section, we introduce one instance of this, the highly efficient Rao-Blackwellized Particle Filter which only samples the discrete modes and propagates sufficient statistics for the continuous variables.

2.3 Rao-Blackwellized Particle Filters

Recent work on *Rao-Blackwellized Particle Filtering* (RBPF) (de Freitas, 2002; Morales-Menendez *et al.*, 2002) has focused on combining PFs and KFs for tracking linear multimodal systems with Gaussian noise. In this kind of model, the belief state is a mixture of Gaussians. Rather than sampling a complete system state, in RBPF for hybrid systems, one combines a Particle Filter that samples the discrete modes z_t, and a Kalman Filter for each discrete mode $z_t \in Z$ that propagates sufficient statistics $(\mu_t^{(i)}, \Sigma_t^{(i)})$ for the continuous parameters x_t. The algorithm is shown in Figure 2. At each time-step t, first, the discrete mode is sampled according to the transition prior. Then, for each particle $p^{(i)}$ a Kalman filter is called to compute the prior mean $\hat{y}_{t|t-1}^{(i)}$ and covariance $\hat{S}_t^{(i)}$ of the observation and update the mean $\mu_t^{(i)}$ and covariance $\Sigma_t^{(i)}$ for the continuous parameters. The variable $\theta(z_t^{(i)})$ denotes the parameters of the Kalman Filter belonging to mode $z_t^{(i)}$. Finally, the particle weight is computed as the observation probability $P(y_t|\hat{y}_{t|t-1}^{(i)}, \hat{S}_t^{(i)})$ of y_t given the prior observation mean and covariance. As in regular Particle Filtering, a resampling step is necessary to prevent particle impoverishment.

As shown in (Morales-Menendez *et al.*, 2002), it is possible in Rao-Blackwellized Particle Filtering

(1) For N particles $p^{(i)}$, $i = 1, \ldots, N$, sample discrete modes $z_0^{(i)}$, from the prior $P(Z_0)$.
(2) For each particle $p^{(i)}$, set $\mu_0^{(i)}$ and $\Sigma_0^{(i)}$ to the prior mean and covariance in state $z_0^{(i)}$.
(3) For each time-step t do
  (a) For each $p^{(i)} = (z_{t-1}^{(i)}, \mu_{t-1}^{(i)}, \Sigma_{t-1}^{(i)})$ do
    (i) Sample a new mode: $\hat{z}_t^{(i)} \sim P(Z_t|z_{t-1}^{(i)})$.
    (ii) Perform Kalman update using parameters from mode $\hat{z}_t^{(i)}$:

$$(\hat{y}_{t|t-1}^{(i)}, \hat{S}_t^{(i)}, \hat{\mu}_t^{(i)}, \hat{\Sigma}_t^{(i)}) \leftarrow KF(\mu_{t-1}^{(i)}, \Sigma_{t-1}^{(i)}, y_t, \theta(z_t^{(i)})).$$

    (iii) Compute the weight of particle $\hat{p}^{(i)}$:

$$w_t^{(i)} \leftarrow P(y_t|\hat{y}_{t|t-1}^{(i)}, \hat{S}^{(i)}) = N(y_t; \hat{y}_{t|t-1}^{(i)}, \hat{S}^{(i)}).$$

  (b) Resample as in step 3.(b) of the PF algorithm (see Figure 1).

Fig. 2. The RBPF algorithm.

to sample the discrete modes directly from the posterior. It is also possible to resample *before* the transition according to the expected posterior weight distribution such that those particles get multiplied which are likely to transition to states of high confidence. These improvements result in an even more efficient algorithm called RBPF2 (Morales-Menendez *et al.*, 2002).

3. NON-LINEAR ESTIMATION

Since RBPF uses a KF for its continuous state estimation, it is restricted to linear problems with Gaussian noise. Many of the problems we are interested in do not have these properties. To overcome this, we propose the *Gaussian particle filter* (GPF). In general hybrid systems, there is no tractable closed-form solution for the continuous variables, so we cannot maintain sufficient statistics with every sample. It is however possible to propagate an approximation of the continuous variables. We sample the mode as usual and for every particle update a Gaussian approximation of the continuous parameters using an unscented Kalman filter. Since the unscented Kalman filter only approximates the true posterior distribution, the GPF is a biased estimator in non-linear models; however, by not sampling the continuous state, we greatly reduce the estimator's variance.

The GPF algorithm is very similar to the RBPF algorithm presented in Figure 2. In both of these algorithms particle $p^{(i)}$ represents the continuous variables with a multivariate Gaussian $N(\mu_t^{(i)}, \Sigma_t^{(i)})$. In the case of linear models and RBPF, this Gaussian is a sufficient statistic, in the case of non-linear models and GPF, it is an approximation. In the algorithm, the only change

is in line 3.(a)ii of Figure 2, which is replaced by:

> 3.a(ii) Perform an unscented Kalman update using parameters from mode $\hat{z}_t^{(i)}$:
>
> $$(\hat{y}_{t|t-1}^{(i)}, \hat{S}_t^{(i)}, \hat{\mu}_t^{(i)}, \hat{\Sigma}_t^{(i)}) \leftarrow UKF(\mu_t^{(i)}, \Sigma_t^{(i)}, y_t, \theta(z_t^{(i)}))$$

This change is due to the non-linearity of transition and/or observation function. A Kalman update is simply not possible, but a good approximation is achieved with an unscented Kalman filter. The approximation of continuous variables in the GPF is a mixture of Gaussians rather than the set of samples as in a PF. Since the expressive power of every particle is higher, fewer particles are needed to achieve the same approximation accuracy. This more than offsets the small additional computational cost per sample. Furthermore, this compact approximation is likely to scale smoothly with an increase in dimensionality.

Like RBPF, the GPF can be improved by sampling directly from the posterior distribution and resampling *before* the transition. We call the resulting algorithm GPF2 and detail it in Figure 3. For each particle, before actually sampling a discrete mode, we look at each possible mode m, update our approximations of the continuous parameters assuming we had sampled m, and compute the observation likelihood for those approximations. This and the transition prior give the posterior probability of transitioning to m. Then for each particle we sample a new discrete mode from the posterior we computed for it.

At each time-step t, for every particle $p^{(i)}$, first we enumerate each possible successor mode m, i.e. each mode $m \in Z$ such that $P(m|z_{t-1}^{(i)}) > 0$. For each m, we perform an unscented Kalman update, and compute analytically the observation likelihood $P(y_t|m, \mu_{t-1}^{(i)}, \Sigma_{t-1}^{(i)}) = P(y_t|y_{t|t-1}^{(i,m)}, S_t^{(i,m)})$. Then, we compute the unnormalized posterior probability $Post(i,m)$ of transitioning to mode m with particle $p^{(i)}$; this is given simply by the product of the transition prior to m and the observation likelihood in m. Next we compute the weight of each particle $\hat{p}^{(i)}$ as the sum of the posterior probabilities of it's successor modes and resample N particles according to this weight distribution. Note, that $Post(i,m)$, $\mu_t^{(i,m)}$ and $\Sigma_t^{(i,m)}$ also need to be resampled, i.e. when particle $p^{(i)}$ is sampled to be particle $\hat{p}^{(k)}$, then $Post(i,m) \leftarrow \widehat{Post}(k,m)$, $\mu_t^{(i,m)} \leftarrow \hat{\mu}_t^{(k,m)}$ and $\Sigma_t^{(i,m)} \leftarrow \hat{\Sigma}_t^{(k,m)}$ for all m.

Finally, for every particle $p^{(i)}$, a successor mode m is sampled according to the posterior probability; this mode is used as $z_t^{(i)}$; $\mu_t^{(i)}$ and $\Sigma_t^{(i)}$ are set to the already computed value $\mu_t^{(i,m)}$ and $\Sigma_t^{(i,m)}$.

GPF2 only differs from the RBPF2 algorithm in that it is calling an unscented Kalman filter update instead of a Kalman update due to the non-linear character of the transformations. It is a very efficient algorithm for state estimation on non-linear models with transition and observation functions that transform a Gaussian distribution to a distribution that's close to a Gaussian. Very low fault priors are handled especially gracefully by GPF2 since it samples the discrete modes from their true posterior distribution. When there is strong enough evidence the fault will be detected regardless of how low the prior is.

(1) For N particles $p^{(i)}$, $i = 1, \ldots, N$, sample discrete modes $z_0^{(i)}$, from the prior $P(Z_0)$.
(2) For each particle $p^{(i)}$, set $\mu_0^{(i)}$ and $\Sigma_0^{(i)}$ to the prior mean and covariance in state $z_0^{(i)}$.
(3) For each time-step t do
  (a) For each $p^{(i)} = (z_{t-1}^{(i)}, \mu_{t-1}^{(i)}, \Sigma_{t-1}^{(i)})$ do
    (i) For each possible successor mode $m \in succ(z_{t-1}^{(i)})$ do
      (A) Perform unscented Kalman update using parameters from mode m:

$$(\hat{y}_{t|t-1}^{(i,m)}, \hat{S}_t^{(i,m)}, \hat{\mu}_t^{(i,m)}, \hat{\Sigma}_t^{(i,m)}) \leftarrow UKF(\mu_{t-1}^{(i)}, \Sigma_{t-1}^{(i)}, y_t, \theta(m)).$$

      (B) Compute posterior probability of mode m as:

$$\widehat{Post}(i,m) \leftarrow P(m|z_{t-1}^{(i)}, y_t) = P(m|z_{t-1}^{(i)}) N(y_t; y_{t|t-1}^{(i,m)}, S_{t|t-1}^{(i,m)}).$$

    (ii) Compute the weight of particle $\hat{p}^{(i)}$: $w_t^{(i)} \leftarrow \sum_{m \in succ(z_{t-1}^{(i)})} \widehat{Post}(i,m)$
  (b) Resample as in step 2.(b) of the PF algorithm (see Figure 1) (also resample $Post$, μ_t and Σ_t).
  (c) For each particle $p^{(i)}$ do
    (i) Sample a new mode: $m \sim P(Z_t|z_{t-1}^{(i)}, y_t)$.
    (ii) Set $z_t^{(i)} \leftarrow m$, $\mu_t^{(i)} \leftarrow \mu_t^{(i,m)}$ and $\Sigma_t^{(i)} \leftarrow \Sigma_t^{(i,m)}$.

Fig. 3. The GPF2 algorithm.

4. EXPERIMENTS

We performed experiments on a simple model of the suspension system of the K-9 rover at NASA Ames Research Center. The model has six discrete modes and six continuous variables, two of which are observable. The continuous parameters follow non-linear trajectories in three of the modes. More details of the model can be found in the full version of the paper. Figure 4 shows the rate of state estimation errors for the GPF, GPF2 and traditional particle filters, as well as the unscented particle filter, discussed below.

The diagnosis of every filter is taken to be the maximum a posteriori (MAP) estimate for the discrete modes; we define a discrepancy between this MAP estimate and the real discrete mode as an error. Figure 4 shows the error

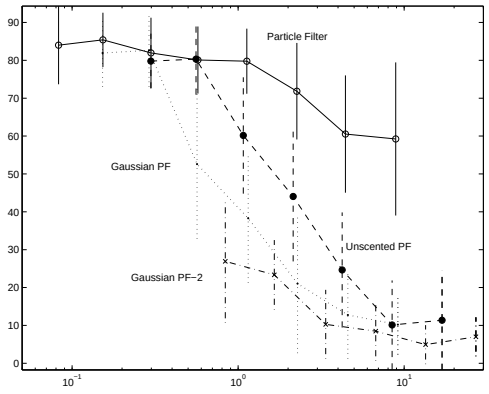

Fig. 4. Performance for the GPF, the GPF when sampling from the posterior, the UPF, and traditional particle filters. Estimation based on 25 runs.

rates ($\frac{\#diagnosis\ errors}{\#time\ \ steps}$) achieved by the algorithms with different numbers of samples; the x-axis is the CPU time the algorithms needed for the computation. The graph shows that GPF is a better approximation than PF given the same computing resources, particularly as the number of samples increases and the discrete states become adequately populated with samples. GPF2 is considerably slower per sample but its approximation is superior to PF or GPF.

We also compare our results with the *unscented particle filter* (UPF) of (van der Merwe *et al.*, 2001). The GPF and UPF have a number of similarities. Both use a set of particles each of which performs an unscented Kalman update at every time step. In UPF, the Kalman update approximation $N(mu_t, \Sigma_t)$ of the posterior is used as a proposal for the particle filter, in GPF this approximation is used as the filter result.

In our experiments there is little difference between the results of GPF and UPF. GPF is generally faster by a constant factor since it does not need to sample the continuous state, and the weight computation is faster. We would expect the UPF to yield better results when the shape of the posterior distribution is very different from a Gaussian and would expect the GPF to do better when there is a big posterior covariance Σ_t such that the sampling introduces high variance on the estimate. In this case, the UPF will need more particles to yield the same results. Since neither of these conditions applies in our domain, both algorithms show similar performance, with GPF being slightly faster.

We are currently testing these algorithms on larger problems, which should better show the performance gain from GPF over the standard particle filter. We anticipate that GPF will be very useful for on-board diagnosis on the K-9 rover where models can't be easily linearised. GPF2 is less clearly useful due to its large computational demands, but may be applicable where extra time can be made available for a critical diagnosis, or where the space of modes to sample from can be reduced using other techniques such as traditional diagnosis algorithms.

REFERENCES

de Freitas, Nando (2002). Rao-Blackwellised particle filtering for fault diagnosis. In: *IEEE Aerospace*.

de Kleer, Johann and Brian C. Williams (1987). Diagnosing multiple faults. In: *Readings in Nonmonotonic Reasoning* (Matthew L. Ginsberg, Ed.). pp. 372–388. Morgan Kaufmann.

de Kleer, Johann and Brian C. Williams (1989). Diagnosis with behavioral modes. In: *Proceedings of the 11th IJCAI*. pp. 1324–1330.

Dearden, Richard and Dan Clancy (2002). Particle filters for real-time fault detection in planetary rovers. In: *Proc. of DX-02*. pp. 1–6.

Doucet, A. (1998). On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310. Department of Engineering, Cambridge Univ.

Doucet, Arnaud, Freitas, Nando De and Gordon, Neil, Eds.) (2001). *Sequential Monte Carlo in Practice*. Springer-Verlag.

Grewal, M. S. and A. P. Andrews (1993). *Kalman Filtering: Theory and Practice*. Prentice Hall.

Hofbaur, Michael W. and Brian C. Williams (2002). Hybrid diagnosis with unknown behaviour modes. In: *Proc. of DX-02*. pp. 97–105.

Isard, M. and A. Blake (1998). CONDENSATION: Conditional density propagation for visual tracking. *Intl. Jnl. of Computer Vision*.

Morales-Menendez, Ruben, Nando de Freitas and David Poole (2002). Real-time monitoring of complex industrial processes with particle filters. In: *NIPS 15*.

Thrun, Sebastian, John Langford and Vandi Verma (2001). Risk sensitive particle filters. In: *NIPS 14*.

van der Merwe, Rudolph, Nando de Freitas, Arnaud Doucet and Eric Wan (2001). The unscented particle filter. In: *NIPS 13*.

Verma, V., J. Langford and R. Simmons (2001). Non-parametric fault identification for space rovers. In: *International Symposium on Artificial Intelligence and Robotics in Space (iSAIRAS)*.

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

Washington, Rich (2000). On-board real-time state and fault identification for rovers. In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation*.

Williams, Brian C. and P. Pandurang Nayak (1996). A model-based approach to reactive self-configuring systems. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference*. AAAI Press / The MIT Press. pp. 971–978.