Robotics Beta

Difference between Rao-Blackwellized particle filters and regular ones

Asked 7 years, 4 months ago Active 4 years ago Viewed 7k times

13

From what I've read so far, it seems that a *Rao-Blackwellized* particle filter is just a normal particle filter used after marginalizing a variable from:

$$p(r_t, s_t | y^t)$$

I'm not really sure about that conclusion, so I would like to know the precise differences between these two types of filters. Thanks in advance.

5

slam particle-filter

Share Improve this question Follow

edited Oct 6 '15 at 12:48
agold
260 1 2 12

asked Jan 6 '14 at 22:15
Ash
360 4 11

are you asking how Rao-Blackwellized particle filter works in slam ? what do you mean by regular filters ? – nayab Feb 20 '14 at 9:53

2 Answers

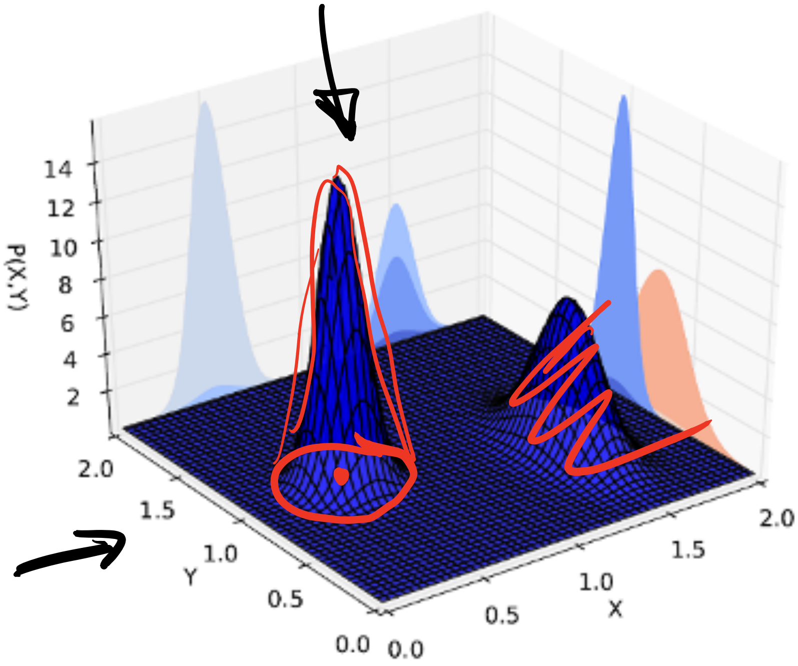
Active Oldest Votes

14

The Rao-Blackwellized Particle Filter (RBPF) as you say in your question performs a marginalization of the probability distribution of your state space.

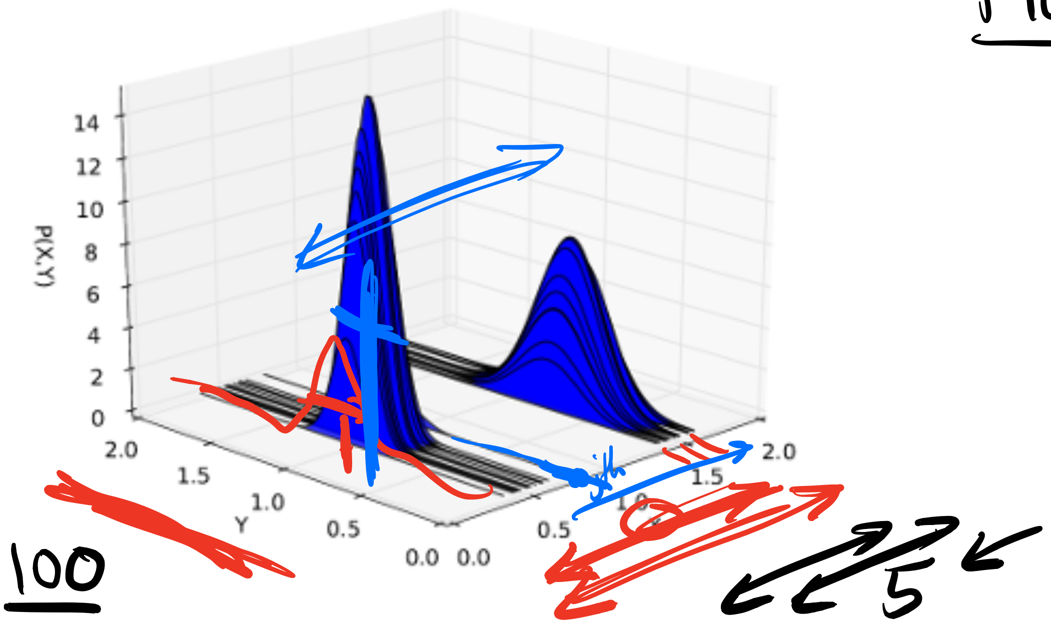
The particle filter uses sampling to represent the multivariate probability distribution of your state space. Using samples to represent a distribution is firstly only an approximation, and secondly not very efficient in most cases. The higher the dimension of the state, the more particles you require. One trick introduced by Doucet et al. is to marginalize out a subset of the state space, which can be handled in a more efficient way by using a Gaussian representation.

These figures from my thesis might help to visualize the concept. Let's say you have a joint distribution over X and Y



instead of sampling jointly over X and Y, we can marginalize Y and represent it as a Gaussian distribution. In this way we just need to sample over X,

$\boxed{\int 105} \rightarrow$



which is much more efficient. Note that each sample now represent a distribution over Y.

This marginalization is very popular in SLAM. The reason is that jointly sampling over position and map is impractical. The RBPF idea has been made popular in FastSLAM, realizing that marginalizing the maps from the joint distribution makes the problem tractable. As in the example above, each particle there represents a pose, correspondences and a map. So there is one map per particle.

So the difference between an RBPF and a regular particle filter is that the RBPF samples over a subspace of the probability distribution of the state, and represents the rest using a different statistic. I have another related question on the math background of the Rao-Blackwellization part.

Share Improve this answer Follow

edited Apr 13 '17 at 12:44
Community ♦
1

answered Aug 25 '14 at 7:00
Jakob
2,914 14 32

-1

Page 151 should help,

https://users.aalto.fi/~ssarkka/pub/cup_book_online_20131111.pdf

where u is a sample (particle without weight)

Share Improve this answer Follow

answered Apr 27 '17 at 5:44
anon
11

Nice explanation in the book. For the answer it is always good to at least summarize the content of the link, as it may not be available forever. – Jakob Apr 27 '17 at 9:31

can be an effective approach to estimate systems that involve highly non-Gaussian models.

4.10.3 Rao-Blackwellized Particle Filter

The dynamic model in Equation (4.162) represents a generic nonlinear model and the noise terms may even be allowed to be non-Gaussian in the general particle filter. We have seen in §4.10.1 that if additive Gaussian noise is employed, then an optimal particle filter can be used. A logical extension is to provide a more general model that may be broken up into purely nonlinear aspects and conditionally linear-Gaussian aspects. Several applications, such as ones that involve positioning, navigation, and tracking,[51] fall into this category. A Rao-Blackwellized particle filter[52] (RBPF) exploits this structure by marginalizing out the conditional linear parts and estimating them using exact filters, such as the Kalman filter.

The RBPF assumes that the state vector is decomposed into $\mathbf{x}_k = [\mathbf{x}_{1k}^T\ \mathbf{x}_{2k}^T]^T$ where

Dynamics

mean

covariance

nonlinear

particles sampling

$$\mathbf{x}_{1k+1} = \boxed{\mathbf{f}(\mathbf{x}_{1k}, \mathbf{w}_{1k})} \tag{4.187a}$$

linear

LTV

$$\mathbf{x}_{2k+1} = \Phi_k(\mathbf{x}_{1k})\mathbf{x}_{2k} + \Gamma_k(\mathbf{x}_{1k})\mathbf{u}_k + \Upsilon_k(\mathbf{x}_{1k})\mathbf{w}_{2k}, \quad \mathbf{w}_{2k} \sim N(\mathbf{0}, Q_k) \tag{4.187b}$$

meas

$$\tilde{\mathbf{y}}_k = H_k(\mathbf{x}_{1k})\mathbf{x}_{2k} + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, R_k) \tag{4.187c}$$

Note that $\mathbf{w}_{1k}$ need not be Gaussian but $\mathbf{w}_{2k}$ and $\mathbf{v}_k$ are assumed to be zero-mean and Gaussian. The system matrices for $\mathbf{x}_{2k}$, such as Φ_k, Γ_k, Υ_k, etc., can be functions of $\mathbf{x}_{1k}$ in this formulation. From this point forward we will drop the explicit notation used in Equation (4.187) that shows this dependence. In the RBPF we must be able to sample from the distribution $p(\mathbf{x}_{1k+1}|\mathbf{x}_{1k})$ and hence it is usually assumed that Equation (4.187a) has the form $\mathbf{x}_{1k+1} = \mathbf{f}(\mathbf{x}_{1k}) + \mathbf{w}_{1k}$. The basic concept of the RBPF is to employ a Kalman filter on a set of particles to the conditional linear model given by Equations (4.187b) and (4.187c). The Kalman filter alone cannot be used because of the nonlinearities given by the model in Equation (4.187a).

A good derivation of the RBPF is provided in Ref. [53], which is shown here. In the BF the importance function $q(\mathbf{x}_{k+1}|\mathbf{X}_k^{(j)}, \tilde{\mathbf{Y}}_{k+1})$ is chosen as the prior pdf $\boxed{p(\mathbf{x}_{k+1}|\mathbf{x}_k^{(j)})}$. Assuming that $\mathbf{x}_{1k+1}$ is independent of $\mathbf{x}_{2k}$, conditioned upon $\mathbf{x}_{1k}$, the weight update is then given by

$$\boxed{w_{k+1}^{(j)} = w_k^{(j)} p(\tilde{\mathbf{y}}_{k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_k)} \tag{4.188}$$

where $\mathbf{X}_{1k+1}^{(j)} = \{\mathbf{x}_{10}^{(j)}, \mathbf{x}_{11}^{(j)}, \dots, \mathbf{x}_{1k+1}^{(j)}\}$ and

$$p(\tilde{\mathbf{y}}_{k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_k) = \int p(\tilde{\mathbf{y}}_{k+1}|\mathbf{x}_{2k+1}, \mathbf{x}_{1k+1}^{(j)})\, p(\mathbf{x}_{2k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_k)\, d\mathbf{x}_{2k+1} \tag{4.189}$$

x_{1k}: particle states

x_{2k}: linear Gaussian states

N particles

for ea. particle j^{th}

$\vec{x}_{1k}^{(j)}, \ x_{2k}^{(j)}, \ P_{2k}^{(j)}$

mean

From Equation (4.187c) we have

$$p(\tilde{\mathbf{y}}_{k+1}|\mathbf{x}_{2k+1}, \mathbf{x}_{1k+1}^{(j)}) = N(\tilde{\mathbf{y}}_{k+1}|H_{k+1}^{(j)}\mathbf{x}_{2k+1}, R_{k+1}) \tag{4.190}$$

The distribution $p(\mathbf{x}_{2k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_k)$ is given by

$$p(\mathbf{x}_{2k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_k) = \int p(\mathbf{x}_{2k+1}|\mathbf{x}_{2k}, \mathbf{x}_{1k+1}^{(j)})p(\mathbf{x}_{2k}|\mathbf{X}_{1k}^{(j)}, \tilde{\mathbf{Y}}_k)\, d\mathbf{x}_{2k} \tag{4.191}$$

From Equation (4.187b) we have

$$p(\mathbf{x}_{2k+1}|\mathbf{x}_{2k}, \mathbf{x}_{1k+1}^{(j)}) = N(\mathbf{x}_{2k+1}|\Phi_k^{(j)}\mathbf{x}_{2k} + \Gamma_k^{(j)}\mathbf{u}_k, \Upsilon_k^{(j)}Q_k\Upsilon_k^{(j)T}) \tag{4.192}$$

According to the RBPF approach, we are given the distribution $p(\mathbf{x}_{2k}|\mathbf{X}_{1k}^{(j)}, \tilde{\mathbf{Y}}_k)$, which is precisely the one that we are updating on-line. Consistent with the Gaussian nature of the problem setup, this distribution is itself Gaussian, which in fact is the *a priori* distribution of the state in the Kalman filter equations. This allows us to write

$$p(\mathbf{x}_{2k}|\mathbf{X}_{1k}^{(j)}, \tilde{\mathbf{Y}}_k) = N(\mathbf{x}_{2k}|\mathbf{x}_{2k}^{(j)}, P_{2k}^{(j)}) \tag{4.193}$$

In the derivation of the Kalman filter, although not explicitly shown, the following identity has been used for a distribution $N(\mathbf{x}|\mathbf{a}, S)$, which is a Gaussian distribution with mean $\mathbf{a}$ and covariance S:

$$\int N(\mathbf{x}|A\mathbf{a}, S)N(\mathbf{a}|\mathbf{y}, P)\, d\mathbf{a} = N(\mathbf{x}|\mathbf{n}, U) \tag{4.194}$$

where $U = APA^T + S$ and $\mathbf{n} = A\mathbf{y}$. Identifying Equation (4.194) to Equation (4.191) with the integrand terms given by Equations (4.192) and (4.193), we now have

$$p(\mathbf{x}_{2k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_k) = N(\mathbf{x}_{2k+1}|\mathbf{x}_{2k+1}^{-(j)}, P_{2k+1}^{-(j)}) \tag{4.195}$$

where

$$\mathbf{x}_{2k+1}^{-(j)} \equiv \Phi_k^{(j)}\mathbf{x}_{2k}^{(j)} + \Gamma_k^{(j)}\mathbf{u}_k \tag{4.196a}$$

$$P_{2k+1}^{-(j)} \equiv \Phi_k^{(j)}P_{2k+1}^{(j)}\Phi_k^{(j)T} + \Upsilon_k^{(j)}Q_k\Upsilon_k^{(j)T} \tag{4.196b}$$

We again make use of Equation (4.194). But this time we apply Equation (4.189) using Equations (4.190) and (4.195) to obtain

$$p(\tilde{\mathbf{y}}_{k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_k) = N(\tilde{\mathbf{y}}_{k+1}|\mathbf{y}_{k+1}^{(j)}, E_{k+1}^{-(j)}) \tag{4.197}$$

where

$$\mathbf{y}_{k+1}^{(j)} \equiv H_{k+1}^{(j)}\mathbf{x}_{2k+1}^{(j)} \tag{4.198a}$$

$$E_{k+1}^{-(j)} \equiv H_{k+1}^{(j)}P_{2k+1}^{-(j)}H_{k+1}^{(j)T} + R_k \tag{4.198b}$$

The remaining derivation follows the pattern of the Kalman update equations derivation in §3.3.1, with

$$p(\mathbf{x}_{2k+1}|\mathbf{X}_{1k+1}^{(j)}, \tilde{\mathbf{Y}}_{k+1}) = N(\mathbf{x}_{2k+1}|\mathbf{x}_{2k+1}^{(j)}, P_{2k+1}^{(j)}) \tag{4.199}$$

This leads to

$$\mathbf{x}_{2k+1}^{(j)} = \mathbf{x}_{2k}^{-(j)} + K_{k+1}^{(j)}\left[\tilde{\mathbf{y}}_{k+1} - \mathbf{y}_{k+1}^{(j)}\right] \tag{4.200a}$$

$$P_{2k+1}^{(j)} = \left[I - K_{k+1}^{(j)} H_{k+1}^{(j)}\right] P_{2k+1}^{-(j)} \tag{4.200b}$$

where $K_{k+1}^{(j)} = P_{2k+1}^{-(j)} H_{k+1}^{(j)T}\left(E_{k+1}^{-(j)}\right)^{-1}$. We can now make the identification $P_{2k+1}^{+(j)} \equiv P_{2k+1}^{(j)}$ and $\mathbf{x}_{2k+1}^{+(j)} \equiv \mathbf{x}_{2k+1}^{(j)}$ to maintain consistent notation with the Kalman filter.

At each time instant a set of N particles is developed for $\mathbf{x}_{1k}^{(j)}$, $\mathbf{x}_{2k}^{(j)}$, and $P_{2k}^{(j)}$, which is the covariance of $\mathbf{x}_{2k}^{(j)}$ given the set $\mathbf{X}_{1k}^{(j)} = \{\mathbf{x}_{10}^{(j)}, \mathbf{x}_{11}^{(j)}, \ldots, \mathbf{x}_{1k}^{(j)}\}$. The samples $\mathbf{x}_{1k}^{(j)}$ are drawn from $p(\mathbf{x}_{1k+1}|\mathbf{x}_{1k}^{(j)})$. An initial set of samples $\mathbf{x}_{20}^{(j)}$ can be drawn from an initial estimate, denoted by $\hat{\mathbf{x}}_{20}$, and covariance P_{20}, and we can set $P_{20}^{(j)} = P_{20}$ for every ith particle. However, different $P_{20}^{(j)}$ can be chosen if desired. At each time instant, perform the following steps: importance function

draw n samples for x_{1k} components

- Draw $\mathbf{x}_{1k+1}^{(j)} \sim p(\mathbf{x}_{1k+1}|\mathbf{x}_{1k}^{(j)})$ for $j = 1, 2, \ldots, N$.
- Perform a Kalman propagation for each particle $j = 1, 2, \ldots, N$

Kalman filter eqns for $x_{2k}^{(j)}$

$\Phi_k^{(j)} = \Phi_k(x_{1k}^{(j)})$

$H_{k+1}^{(j)} = H_{k+1}(x_{1k}^{(j)})$

$$\boxed{\begin{aligned} \mathbf{x}_{2k+1}^{-(j)} &= \Phi_k^{(j)}\mathbf{x}_{2k}^{+(j)} + \Gamma_k^{(j)}\mathbf{u}_k \\ P_{2k+1}^{-(j)} &= \Phi_k^{(j)} P_{2k}^{+(j)} \Phi_k^{(j)T} + \Upsilon_k^{(j)} Q_k \Upsilon_k^{(j)T} \end{aligned}} \tag{4.201a, 4.201b}$$

- Update the weights for each particle $j = 1, 2, \ldots, N$

$$\boxed{\begin{aligned} w_{k+1}^{(j)} &= w_k^{(j)} \frac{1}{\det\left[2\pi E_{k+1}^{-(j)}\right]^{1/2}} \exp\left[-\frac{1}{2}\mathbf{e}_{k+1}^{-(j)T}\left(E_{k+1}^{-(j)}\right)^{-1}\mathbf{e}_{k+1}^{-(j)}\right] \\ w_{k+1}^{(j)} &\leftarrow \frac{w_{k+1}^{(j)}}{\sum_{j=1}^{N} w_{k+1}^{(j)}} \end{aligned}} \tag{4.202a, 4.202b}$$

likelihood function for $\tilde{y}_{k+1}$ (normal dist.)

renormalize weights

where $\mathbf{e}_{k+1}^{-(j)} \equiv \tilde{\mathbf{y}}_{k+1} - H_{k+1}^{(j)}\mathbf{x}_{2k+1}^{-(j)}$ and $E_{k+1}^{-(j)} \equiv H_{k+1}^{(j)} P_{2k+1}^{-(j)} H_{k+1}^{(j)T} + R_{k+1}$.

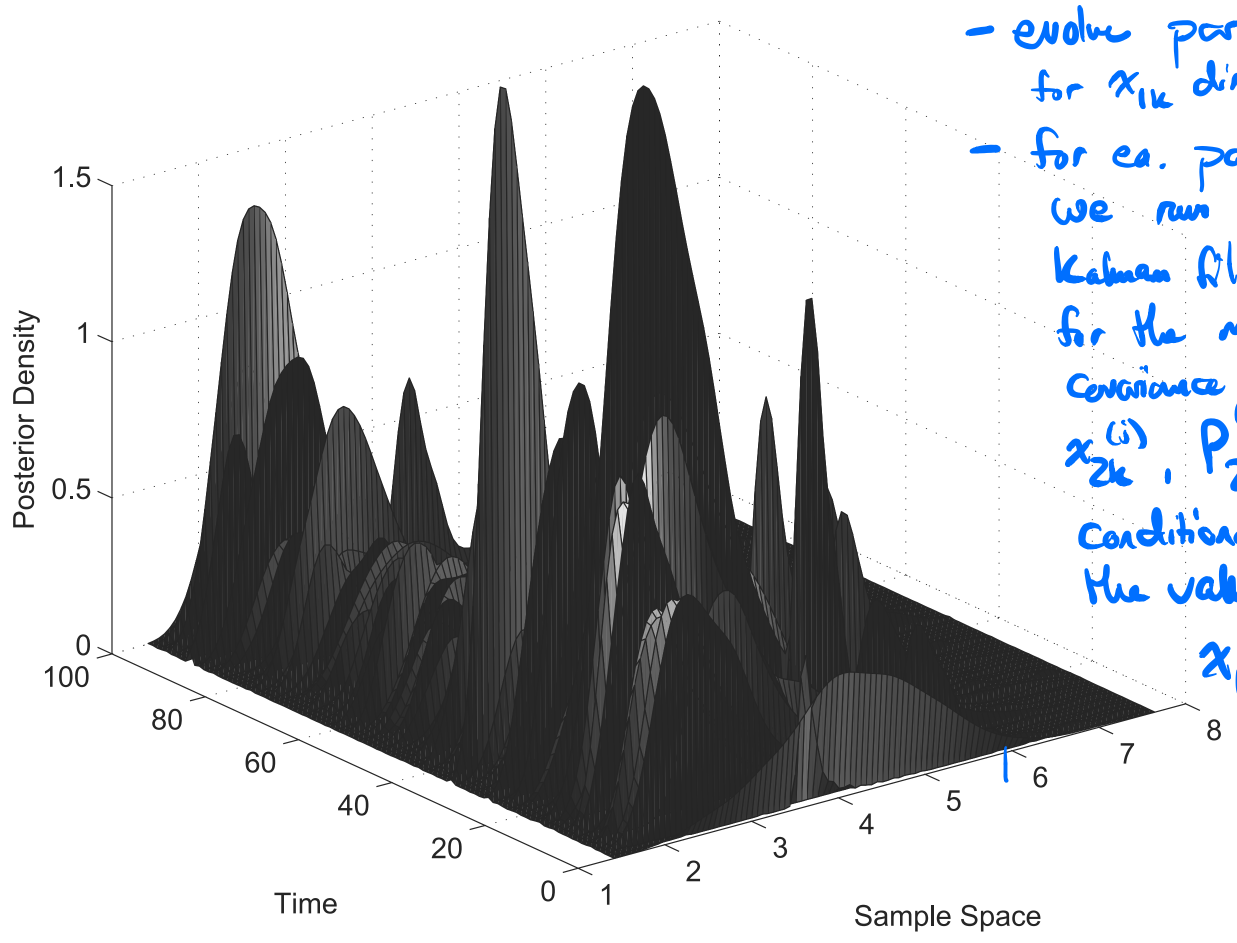
- Compute the Kalman gain for each particle $j = 1, 2, \ldots, N$

meas

what we expect the meas to be.

$$\boxed{K_{k+1}^{(j)} = P_{2k+1}^{-(j)} H_{k+1}^{(j)T}\left(E_{k+1}^{-(j)}\right)^{-1}} \tag{4.203}$$

separate Kalman gain for ea. particle



Summarize:

- evolve particles for x_{1k} dimensions
- for ea. particle we run a separate Kalman filter for the mean & covariance of $x_{2k}^{(i)}$, $P_{2k}^{(i)}$ conditioned on the value of $x_{1k}^{(i)}$

Figure 4.15: Posterior Density

- Perform a Kalman update for each particle $j = 1, 2, \ldots, N$

using meas to update estimate for $x_{2k}^{(i)}$

$$\mathbf{x}_{2k+1}^{+(j)} = \mathbf{x}_{2k+1}^{-(j)} + K_{k+1}^{(j)} \left[\tilde{\mathbf{y}}_{k+1} - H_{k+1}^{(j)} \mathbf{x}_{2k+1}^{(j)} \right] \tag{4.204a}$$

$$P_{2k+1}^{+(j)} = \left[I - K_{k+1}^{(j)} H_{k+1}^{(j)} \right] P_{2k+1}^{-(j)} \tag{4.204b}$$

State estimates and the state covariance can be computed using

$$\hat{\mathbf{x}}_k \approx \sum_{j=1}^{N} w_k^{(j)} \mathbf{x}_k^{(j)} \tag{4.205a}$$

$$P_k \approx \sum_{j=1}^{N} w_k^{(j)} \left\{ \tilde{\mathbf{x}}_k^{(j)} \tilde{\mathbf{x}}_k^{(j)T} + \begin{bmatrix} 0_{n_1 \times n_1} & 0_{n_1 \times n_2} \\ 0_{n_2 \times n_1} & \boxed{P_{2k}^{+(j)}} \end{bmatrix} \right\} \tag{4.205b}$$

$$\tilde{\mathbf{x}}_k^{(j)} = \mathbf{x}_k^{(j)} - \hat{\mathbf{x}}_k \tag{4.205c}$$

where $\mathbf{x}_k^{(j)T} = \left[\mathbf{x}_{1k}^{(j)T} \ \ \mathbf{x}_{2k}^{+(j)T} \right]^T$, n_1 is the length of $\mathbf{x}_1$, and n_2 is the length of $\mathbf{x}_2$.

Resampling and roughening can also be done as needed. The RBPF appears to be computationally expensive because a Kalman filter is executed on each particle.

Also, $\mathbf{x}_{1k+1}^{(j)}$ particles must be drawn at each time step. The main advantage of the RBPF is that *fewer* particles are typically needed than for a full filter, such as the BF. Thus, depending on the system at hand, the RBPF may in fact be more computationally efficient than the BF due to the reduction in the number of required particles. This issue is of course application dependent. One must weigh whether or not a RBPF provides the computational advantages over a standard PF while providing the desired accuracy for the particular application at hand.

Example 4.13: In this example the RBPF is used to estimate the states of a finite impulse response (FIR) filter.[53] The truth model is generated using the following:

$$\begin{aligned}
x_{1k+1} &= \cos(x_{1k}) + \sin(x_{1k}) + w_{1k}\\
x_{2k+1} &= x_{2k} + w_{2k}\\
\tilde{y}_k &= x_{1k}x_{2k} + v_k
\end{aligned}$$

where w_{1k} and w_{2k} are zero-mean Gaussian noise processes with variances given by 0.09 and 0.04, respectively, and v_k is a zero-mean Gaussian noise process with variance given by 0.01. The true states are initialized with $x_{10} = 1$ and $x_{20} = 2$, and 100 synthetic measurements are generated. In this example $\mathbf{f}(\mathbf{x}_{1k}) = \cos(x_{1k}) + \sin(x_{1k})$, $\Phi_k = 1$, and $H_l = x_{1k}$.

The particles for x_1 are generated using a Gaussian distribution with mean given by $\cos(x_{10}) + \sin(x_{10})$ and variance given by 0.09. The particles for x_2 are generated using a Gaussian distribution with mean 0 and variance 1. Note that there is a fairly large error in the mean estimate for x_2 at the initial time and $P_{20} = 1$ is used to compensate for this error. A total of 500 particles is used.

Resampling is done at each time-step using systematic resampling, but no roughening is done. A plot of the posterior pdfs for the second state as they evolve over time is shown in Figure 4.15. This shows that the posterior pdf is qualitatively well approximated by a Gaussian function since only one peak exists. A plot of the errors and 3σ boundaries for the second state is shown in Figure 4.16. The errors are clearly within their respective 3σ boundaries, which indicates that the RBPF is functioning consistently.

$$x_{1k+1} = \underline{x_{1k}} + \Delta t\, x_{2k}$$

4.10.4 Navigation Using a Rao-Blackwellized Particle Filter

We now consider another form of an RBPF, where the system can be partitioned into linear and nonlinear parts that are coupled:

linear meas eqn for $x_{1k+1} - f(x_{1k})$

linear dynamics

$$\mathbf{x}_{1k+1} = \underline{\mathbf{f}(\mathbf{x}_{1k})} + \underline{\Phi_{1k}\mathbf{x}_{2k}} + \Upsilon_{1k}\mathbf{w}_{1k} \qquad \boxed{(4.206a)}$$

$$\mathbf{x}_{2k+1} = \Phi_{2k}\mathbf{x}_{2k} + \Upsilon_{2k}\mathbf{w}_{2k} \qquad (4.206b)$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}\boxed{(\mathbf{x}_{1k})} + \mathbf{v}_k \qquad (4.206c)$$

2 meas eqns:

→ ① $\tilde{y}_k = h(x_{1k}) + v_k$ → meas for x_{1k} → used to update weights

→ ② new output: $x_{1k+1} - f(x_{1k}) = \Phi_{1k}\, x_{2k} + \Upsilon_{1k}\, w_{1k}$

→ from x_{2k} ↗ new meas.

used in Kalman filter update different for ea. particle

Ex.

x_{1k} planar position

x_{2k} velocity. IMU states. } Pinson Model for IMU's Gyros } → attitudes

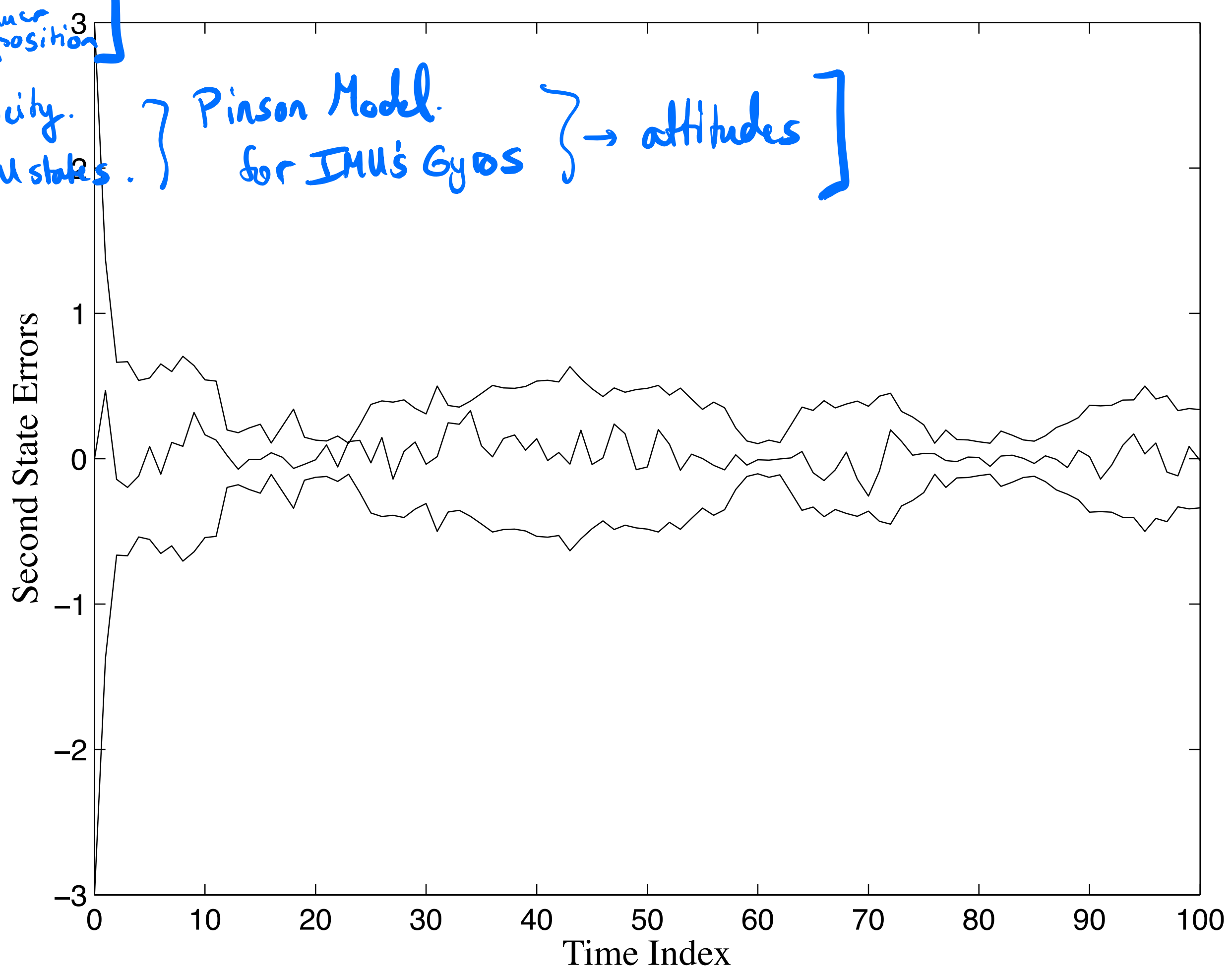


Figure 4.16: State Estimate Errors for x_2

Here it is assumed that $\mathbf{w}_{1k}$ and $\mathbf{w}_{2k}$ are zero-mean Gaussian noise processes that may be correlated, so that

$$\mathbf{w}_k \equiv \begin{bmatrix} \mathbf{w}_{1k} \\ \mathbf{w}_{2k} \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} Q_{1k} & \boxed{Q_{12k}} \\ \boxed{Q_{12k}^T} & Q_{2k} \end{bmatrix} \right) \tag{4.207}$$

The pdf of $\mathbf{x}_{20}$ is assumed to be Gaussian with known mean and covariance given by P_{20}. The pdfs for $\mathbf{x}_{10}$ and $\mathbf{v}_k$ are arbitrary but in most cases the pdf of $\mathbf{v}_k$ is Gaussian, with $\mathbf{v}_k \sim N(\mathbf{0}, R_k)$.

Note that several navigation-type problems fall into the category of models given by Equation (4.206), where $\mathbf{x}_1$ typically denotes position states and $\mathbf{x}_2$ denotes velocity states, respectively.[51] Hence, we call the ensuing particle filter the *navigation RBPF*. Reference [54] provides a derivation of the RBPF for this case, which is shown here. Using Bayes' rule on $p(\mathbf{X}_{1k}, \mathbf{x}_{2k} | \tilde{\mathbf{Y}}_k)$ gives

$$p(\mathbf{X}_{1k}, \mathbf{x}_{2k} | \tilde{\mathbf{Y}}_k) = p(\mathbf{x}_{2k} | \mathbf{X}_{1k}, \tilde{\mathbf{Y}}_k) p(\mathbf{X}_{1k} | \tilde{\mathbf{Y}}_k) \tag{4.208}$$

Because the measurements, $\mathbf{Y}_k$, are conditionally independent of $\mathbf{X}_{1k}$, then the pdf $p(\mathbf{x}_{2k} | \mathbf{X}_{1k}, \tilde{\mathbf{Y}}_k)$ can be rewritten as

$$p(\mathbf{x}_{2k} | \mathbf{X}_{1k}, \tilde{\mathbf{Y}}_k) = p(\mathbf{x}_{2k} | \mathbf{X}_{1k}) \tag{4.209}$$

Consider the following system: Run a Kalman filter on x_2

$$\mathbf{x}_{2k+1} = \Phi_{2k}\mathbf{x}_{2k} + \Upsilon_{2k}\boxed{\mathbf{w}_{2k}} \quad \text{"dynamics for } x_2\text{"} \tag{4.210a}$$

$$x_{1k+1} - f(x_{1k}) = \mathbf{z}_k = \Phi_{1k}\mathbf{x}_{2k} + \Upsilon_{1k}\boxed{\mathbf{w}_{1k}} \quad \text{"meas for } z_k\text{"} \tag{4.210b}$$

where $\mathbf{z}_k \equiv \mathbf{x}_{1k+1} - \mathbf{f}(\mathbf{x}_{1k})$. A Kalman filter can now be applied to Equation (4.210). Then, we have

$$p(\mathbf{x}_{2k}|\mathbf{X}_{1k}) = N(\mathbf{x}_{2k}^-, P_{2k}^-) \tag{4.211}$$

where $\mathbf{x}_{2k}^-$ and P_{2k}^- come from the Kalman filter. Due to the term Q_{12k}, a correlated Kalman filter must be employed. We replace $\mathbf{w}_{2k}$ with

$$\bar{\mathbf{w}}_{2k} = \mathbf{w}_{2k} - Q_{12k}^T Q_{1k}^{-1}\mathbf{w}_{1k} \tag{4.212}$$

Then, the state equation for $\mathbf{x}_{2k}$ becomes

$$\mathbf{x}_{2k+1} = (\Phi_{2k} - C_k\Phi_{1k})\mathbf{x}_{2k} + \Upsilon_{2k}\bar{\mathbf{w}}_{2k} + C_k[\mathbf{x}_{1k+1} - \mathbf{f}(\mathbf{x}_{1k})] \tag{4.213}$$

where

$$C_k = \Upsilon_{2k}Q_{12k}^T Q_{1k}^{-1}(\Upsilon_{1k}^T\Upsilon_{1k})^{-1}\Upsilon_{1k}^T \tag{4.214}$$

We can write $p(\mathbf{X}_{1k}|\tilde{\mathbf{Y}}_k)$ recursively by repeated use of Bayes' rule, according to

$$p(\mathbf{X}_{1k}|\tilde{\mathbf{Y}}_k) = \frac{p(\tilde{\mathbf{y}}_k|\mathbf{x}_{1k})p(\mathbf{x}_{1k}|\mathbf{X}_{1k-1})}{p(\tilde{\mathbf{y}}_k|\tilde{\mathbf{Y}}_{k-1})}p(\mathbf{X}_{1k-1}|\tilde{\mathbf{Y}}_{k-1}) \tag{4.215}$$

Due to the nonlinear state equation for $\mathbf{x}_{1k}$, a PF is employed to solve Equation (4.215). The weights are represented by the likelihood $p(\tilde{\mathbf{y}}_k|\mathbf{x}_{1k}^{(j)})$. The particles are sampled from $p(\mathbf{x}_{1k+1}^{(j)}|\mathbf{X}_{1k}^{(j)})$. Using the state equation for $\mathbf{x}_{1k}$ from Equation (4.206a) together with Equation (4.211) we have

$$p(\mathbf{x}_{1k+1}^{(j)}|\mathbf{X}_{1k}^{(j)}) = N(\mathbf{f}(\mathbf{x}_{1k}^{(j)}) + \Phi_{1k}\mathbf{x}_{2k}^{-(j)}, \Phi_{1k}P_{2k}^-\Phi_{1k}^T + \Upsilon_{1k}Q_{1k}\Upsilon_{1k}^T) \tag{4.216}$$

Note that the covariances for all the particles are the same, so only one P_{2k}^- needs to be employed.

A summary of the navigation RBPF is now provided.[54] The first step is to generate the $\mathbf{x}_{10}$ particles from $p(\mathbf{x}_{10})$ and set the weights, $w_k^{(j)}$, all equal to $1/N$. The Kalman filters are initialized with $\mathbf{x}_{20}^{-(j)}$ using an initial condition for $\mathbf{x}_{20}$ as the mean and the P_{20} as the covariance. Here, we assume that the measurement noise is zero-mean Gaussian. At each time instant perform the following:

- Update the weights for each particle $j = 1, 2, \ldots, N$

1

$$w_k^{(i+1)} = w_k^{(j)}\exp\left[-\frac{1}{2}\left(\tilde{\mathbf{y}}_k - \mathbf{y}_k^{(j)}\right)R_k^{-1}\left(\tilde{\mathbf{y}}_k - \mathbf{y}_k^{(j)}\right)^T\right] \tag{4.217a}$$

$$w_{k+1}^{(j)} \leftarrow \frac{w_{k+1}^{(j)}}{\sum_{j=1}^N w_{k+1}^{(j)}} \quad \text{renormalize.} \tag{4.217b}$$

likelihood function for $\tilde{y}_k$ Gaussian where $\mathbf{y}_k^{(j)} \equiv \mathbf{h}(\mathbf{x}_{1k}^{(j)})$.

- Resample $\mathbf{x}_{1k}^{(j)}$ if needed.
- Propagate the particles for each particle $j = 1, 2, \ldots, N$

from 4.206(a)

particle propagation for $x_{1k}^{(j)}$ →

$$\boxed{\mathbf{x}_{1k+1}^{(j)} \sim N(\mathbf{f}(\mathbf{x}_{1k}^{(j)}) + \Phi_{1k}\mathbf{x}_{2k}^{-(j)}, \Phi_{1k}P_{2k}^{-}\Phi_{1k}^T + \Upsilon_{1k}Q_{1k}\Upsilon_{1k}^T)} \tag{4.218}$$

*

$x_{1k+1}^{(j)}$ is drawn from the distribution *

- Compute the Kalman gain

$$\boxed{K_k = P_{2k}^{-}\Phi_{1k}^T[\Phi_{1k}^T P_{2k}^{-}\Phi_{1k}^T + \Upsilon_{1k}Q_{1k}\Upsilon_{1k}^T]^{-1}} \tag{4.219}$$

- Update the Kalman filters for each particle $j = 1, 2, \ldots, N$

meas update

2

mean value for $x_{2k}^{(j)}$ for ea particle j

$$\mathbf{x}_{2k}^{+(j)} = \mathbf{x}_{2k}^{-(j)} + K_k\left[\boxed{\mathbf{x}_{1k+1}^{(j)} - \mathbf{f}(\mathbf{x}_{1k}^{(j)})} - \Phi_{1k}\mathbf{x}_{2k}^{-(j)}\right] \tag{4.220a}$$

$$P_{2k}^{+} = [I - K_k\Phi_{1k}]P_{2k}^{-} \tag{4.220b}$$

- Propagate the Kalman filters for each particle $j = 1, 2, \ldots, N$

propagation update

$$\mathbf{x}_{2k+1}^{-(j)} = D_k\mathbf{x}_{2k}^{+(j)} + C_k\left[\mathbf{x}_{1k+1}^{(j)} - \mathbf{f}(\mathbf{x}_{1k}^{(j)})\right] \tag{4.221a}$$

$$P_{2k+1}^{-} = D_k P_{2k}^{+} D_k^T + \Upsilon_{2k}\bar{Q}_{2k}\Upsilon_{2k}^T \tag{4.221b}$$

where

Correlated Noise KF: sec 3.3.6 p. 158

$$\bar{Q}_{2k} = Q_{2k} - Q_{12k}^T Q_{1k}^{-1} Q_{12k} \tag{4.222a}$$

$$C_k = \Upsilon_{2k}Q_{12k}^T Q_{1k}^{-1}(\Upsilon_{1k}^T\Upsilon_{1k})^{-1}\Upsilon_{1k}^T \tag{4.222b}$$

$$D_k = \Phi_{2k} - C_k\Phi_{1k} \tag{4.222c}$$

State estimates and the state covariance can be computed using

weighted sums to get state estimate & estimate covariance

$$\hat{\mathbf{x}}_k \approx \sum_{j=1}^{N} w_k^{(j)}\mathbf{x}_k^{(j)} \tag{4.223a}$$

$$P_k \approx \begin{bmatrix} 0_{n_1\times n_1} & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & P_{2k}^{+} \end{bmatrix} + \sum_{j=1}^{N} w_k^{(j)}\tilde{\mathbf{x}}_k^{(j)}\tilde{\mathbf{x}}_k^{(j)T} \tag{4.223b}$$

$$\tilde{\mathbf{x}}_k^{(j)} = \mathbf{x}_k^{(j)} - \hat{\mathbf{x}}_k \tag{4.223c}$$

where $\mathbf{x}_k^{(j)} = \begin{bmatrix}\mathbf{x}_{1k}^{(j)T} & \mathbf{x}_{2k}^{+(j)T}\end{bmatrix}^T$, n_1 is the length of $\mathbf{x}_1$, and n_2 is the length of $\mathbf{x}_2$.

Example 4.14: In this example the navigation RBPF is used to track an unknown object's position and velocity using a set of two range measurements. The states of the unknown object are its planar position and associated velocity. The truth model is generated using the following:

$$x_k = \begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix} \begin{matrix} \rightarrow \text{planar position} \\ \rightarrow \text{planar velocity.} \end{matrix}$$

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k \rightarrow \text{constant velocity}$$

where Δt is the sampling interval, which is set to 0.1 seconds, and $\mathbf{x}_k = [x_{1k}\ x_{2k}\ x_{3k}\ x_{4k}]^T$. The final time of the simulation run is 240 minutes. The covariance of $\mathbf{w}_k$ is given by

$$Q_k = q \begin{bmatrix} (\Delta t^3/3)I_{2\times 2} & (\Delta t^2/2)I_{2\times 2} \\ \\ (\Delta t^2/2)I_{2\times 2} & \Delta t I_{2\times 2} \end{bmatrix}$$

where $I_{2\times 2}$ is a 2×2 identity matrix. For simulation purposes we set $q = 1 \times 10^{-10}$. The initial condition is given by $\mathbf{x}_0 = [15\ 15\ 0\ 0]^T$. All units are in kilometers and seconds. Two range measurements are provided at each time. The measurement model is given by

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} [(X_{1k} - x_{1k})^2 + (Y_{1k} - x_{2k})^2]^{1/2} \\ [(X_{2k} - x_{1k})^2 + (Y_{2k} - x_{2k})^2]^{1/2} \end{bmatrix} + \mathbf{v}_k$$

where (X_{1k}, Y_{1k}) and (X_{2k}, Y_{2k}) represent two vehicles with radar sensors. For the simulation X_{1k} varies linearly from -5 km to 30 km over the 240 minute time run and Y_{1k} is set to zero for the entire time. Also, $X_{2k} = 10\cos(0.001\,t_k)$ and $Y_{2k} = 30\sin(0.005\,t_k)$. Synthetic measurements are generated using zero-mean Gaussian noise with covariance $R_k = 0.01 I_{2\times 2}$ for $\mathbf{v}_k$.

For the navigation RBPF a total of 500 particles is used. The state vector is decomposed into the first two states and last two states. Initial particles are generated using zero-mean Gaussian noise for both $\mathbf{x}_{10}$ and $\mathbf{x}_{20}$. The covariance for $\mathbf{x}_{10}$ is given by $64 I_{2\times 2}$ and the covariance for $\mathbf{x}_{20}$ is given by $P_{20} = 0.001 I_{2\times 2}$. The various quantities used in Equation (4.206) are given by

$$\mathbf{f}(\mathbf{x}_{1k}) = \mathbf{x}_{1k}, \quad \Phi_{1k} = \Delta t I_{2\times 2}, \quad \Phi_{2k} = I_{2\times 2}$$
$$\Upsilon_{1k} = \Upsilon_{2k} = I_{2\times 2}$$
$$Q_{1k} = (\Delta t^3/3) I_{2\times 2}, \quad Q_{2k} = \Delta t I_{2\times 2}, \quad Q_{12k} = (\Delta t^2/2) I_{2\times 2}$$

The navigation RBPF can now be executed with the aforementioned values. Resampling is done at each time step using systematic resampling, but no roughening is done. State estimates and covariances are computed using Equation (4.223). A plot of the errors for the first state along with the respective 3σ boundaries is shown in Figure 4.17. This indicates that the navigation RBPF is working properly.

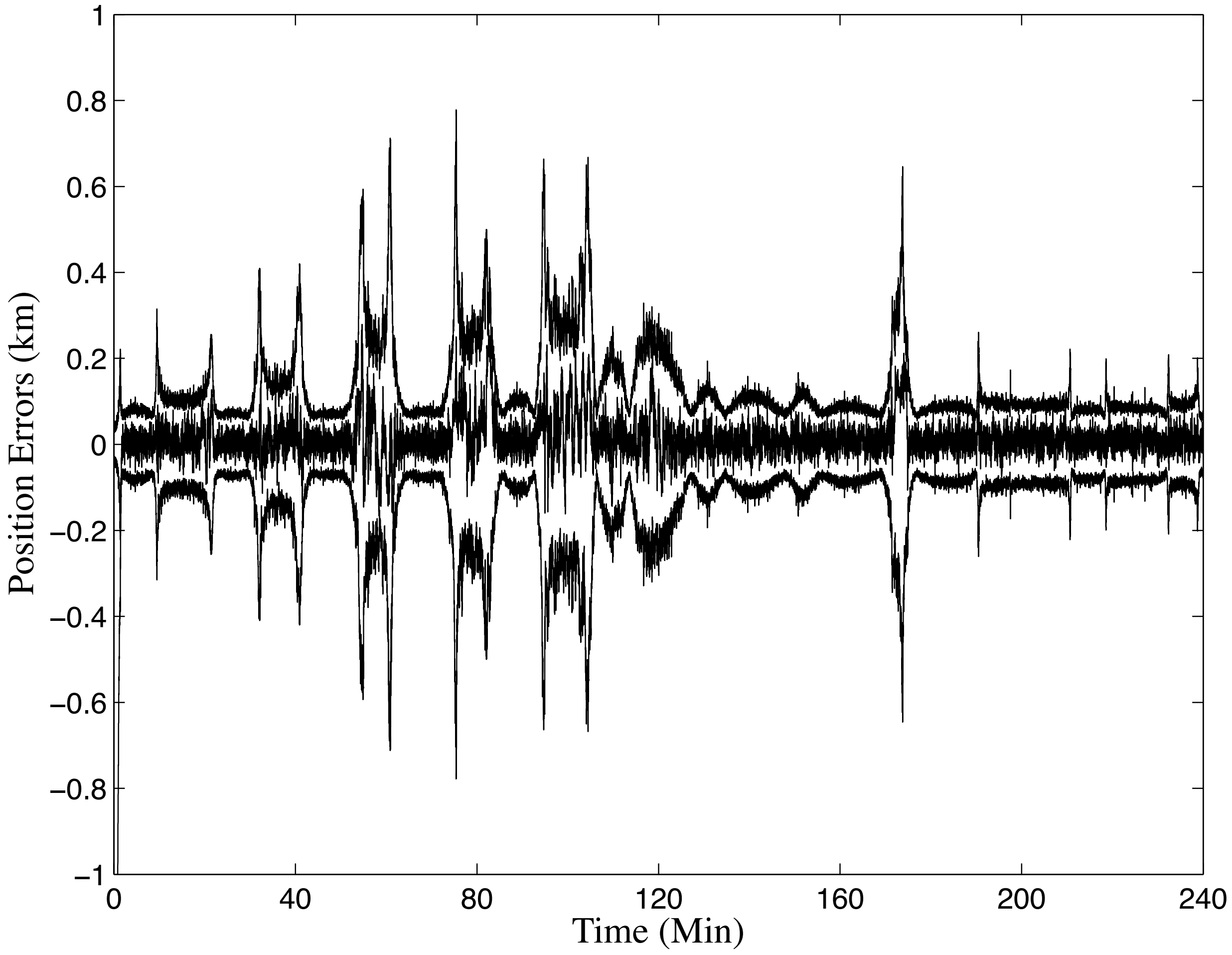


Figure 4.17: State Estimate Errors for x_1

4.11 Error Analysis

The optimality of the Kalman filter hinges on many factors. First, although precise knowledge of the process noise and measurement inputs is not required, we must have accurate knowledge of their respective covariance values. When these covariances are not well known then the methods in §4.6 can be applied to estimate them on-line. Also, errors in the assumed model may be present. Determining these errors is usually a formidable task. This section shows an analysis of how the error covariance of the nominal system is changed with the aforementioned errors. This new covariance can be used to assess the performance of the nominal Kalman filter given bounds on the model and noise quantities, which may provide insight to filter performance and sensitivity to various errors. The development in this section is based on continuous-time models and measurements. Also, in this section we eliminate the explicit dependence on time for notational brevity. Consider the following nominal system, which will be used to derive the Kalman filter: