

## Square Root IEKF Derivation

Let the dynamics model be given by

$$x(k+1) = f[k, x(k), u(k), v(k)]$$

and the measurement model be given by

$$z(k) = h[k, x(k)] + w(k)$$

where  $v(k)$  and  $w(k)$  are white noise with  $v(k) \sim N(0, Q(k))$ , and  $w(k) \sim N(0, R(k))$ , and  $E[v(k)w(j)^T] = 0 \forall k, j$ .

We assume we have  $\hat{x}(k)$  and  $P(k)$ .

Expand the dynamics eqn. about  $x(k) = \hat{x}(k)$  and  $v(k) = 0$  to get

$$x(k+1) = f[k, \hat{x}(k), u(k), 0] + F(k)[x(k) - \hat{x}(k)] + \Gamma(k)v(k) + \text{HOT}$$

$$\text{where } F(k) = \frac{\partial f}{\partial x(k)} \Big|_{\hat{x}(k), u(k), 0} \quad \text{and} \quad \Gamma(k) = \frac{\partial f}{\partial v(k)} \Big|_{\hat{x}(k), u(k), 0}$$

Let  $\bar{x}(k+1) = f[k, \hat{x}(k), u(k), 0]$ . Drop higher order terms (HOT).

$$x(k+1) = \bar{x}(k+1) + F(k)x(k) - F(k)\hat{x}(k) + \Gamma(k)v(k)$$

$$\text{and } x(k) = F(k) \left[ x(k+1) - \Gamma(k)v(k) + F(k)\hat{x}(k) - \bar{x}(k+1) \right]$$

## SR Information Equations : (aka Data Equations)

$$(\text{prior state}) \quad \bar{z}_x(k) = R_{xx}(k)x(k) + w_x(k), \quad w_x(k) \sim (0, I)$$

$$(\text{prior proc. noise}) \quad \bar{z}_v(k) = 0 = R_{vv}(k)v(k) + w_v(k), \quad w_v(k) \sim (0, I)$$

$$\text{where } R_{vv}^T(k)R_{vv}(k) = Q^{-1}(k)$$

$$\bar{z}_x(k) = R_{xx}(k)\hat{x}(k), \quad \text{and } R_{xx}^T(k)R_{xx}(k) = P(k)$$

Normalize the measurement equation:

Let  $R_a^T(k) R_a(k) = R(k)$ . Then normalize so that

$$z_a(k) = R_a^{-T}(k) z(k), \quad h_a[k, x(k)] = R_a^{-T}(k) h[k, x(k)] \\ w_a(k) = R_a^{-T}(k) w(k)$$

$$\text{Thus, } z_a(k) = h_a[k, x(k)] + w_a(k).$$

### Steps for Iterated-Extended SRF:

Begin with the cost function:

$$J = \|R_{xx}(k)x(k) - \bar{z}_x(k)\|^2 + \|R_v v(k)\|^2 + \|z_a(k+1) - h_a[k+1, x(k+1)]\|^2$$

Set  $k=0$  (assuming first measurement is  $z(1)$ ).

Propagation Step: Note that  $\bar{z}_x(k) = R_{xx}(k)\bar{x}(k)$ .

(a) Use appx. inverse dynamics to eliminate  $x(k)$ :

$$J \approx \left\| \begin{bmatrix} R_{vv}(k) & 0 \\ -R_{xx}^{(k)} F(k)^{-1} & R_{xx}(k) F(k)^{-1} \end{bmatrix} \begin{bmatrix} v(k) \\ x(k+1) \end{bmatrix} - \begin{bmatrix} 0 \\ R_{xx}(k) F(k)^{-1} \bar{x}(k) \end{bmatrix} \right\|^2 + \|z_a(k+1) - h_a[k+1, x(k+1)]\|^2$$

(b) Perform usual QR factorization and multiplication by  $Q^T$  to get

$$= \left\| \begin{bmatrix} \bar{R}_{vv}(k) & \bar{R}_{vx}(k+1) \\ 0 & \bar{R}_{xx}(k+1) \end{bmatrix} \begin{bmatrix} v(k) \\ x(k+1) \end{bmatrix} - \begin{bmatrix} \bar{z}_v(k) \\ \bar{z}_x(k+1) \end{bmatrix} \right\|^2 + \|z_a(k+1) - h_a[k+1, x(k+1)]\|^2$$

$$(c) \text{ Let } v(k) = \bar{R}_{vv}^{-1}(k) [\bar{z}_v(k) - \bar{R}_{vx}(k+1) x(k+1)]$$

Substitute into cost func, thereby eliminating component involving  $v(k)$ .

Subtle point: Because of the way  $\bar{x}(k+1)$  is incorporated in step (a), the propagation step enforces the condition

$$\bar{x}(k+1) = \bar{R}_{xx}^{-1}(k+1) \bar{z}_x(k+1)$$

from nonlinear dynamics propagation

$$J = \| \bar{R}_{xx}(k+1) x(k+1) - \bar{z}_x(k+1) \|^2 + \| z_a(k+1) - h_a[k+1, x(k+1)] \|^2$$

### UPDATE STEP:

(d) Set  $i=0$

(e) Let  $\hat{x}^i(k+1) = \bar{x}(k+1) = f[k, \hat{x}(k), u(k), 0]$ .

(f) Expand the measurement equation as:

$$z_a(k+1) = \hat{z}_a^i(k+1) + H_a^i(k+1)[x(k+1) - \hat{x}^i(k+1)] + w_a(k+1) + \text{HOT}$$

where  $\hat{z}_a^i(k+1) = h_a[k+1, \hat{x}^i(k+1)]$ ,  $H_a^i(k+1) = \frac{\partial h_a}{\partial x(k+1)} \Big|_{\hat{x}^i(k+1)}$

(g) Drop HOT, substitute for the measurement equation and stack vector norms to get

$$J = \left\| \begin{bmatrix} \bar{R}_{xx}(k+1) \\ H_a^i(k+1) \end{bmatrix} x(k+1) - \begin{bmatrix} \bar{z}_x(k+1) \\ z_a(k+1) - \hat{z}_a^i(k+1) + H_a^i(k+1) \hat{x}^i(k+1) \end{bmatrix} \right\|^2$$

(h) Perform usual QR factorization and mult. by  $Q^T$  to get

$$J = \left\| \begin{bmatrix} R_{xx}^i(k+1) \\ 0 \end{bmatrix} x(k+1) - \begin{bmatrix} \bar{z}_x^i(k+1) \\ z_r^i(k+1) \end{bmatrix} \right\|^2$$

$$= \| R_{xx}^i(k+1) x(k+1) - \bar{z}_x^i(k+1) \|^2 + \| z_r^i(k+1) \|^2$$

(i) If this is EKF (not IEKF), let  $R_{xx}(k+1) = R_{xx}^i(k+1)$ ,  $\bar{z}_x(k+1) = \bar{z}_x^i(k+1)$ ,  $k \leftarrow k+1$ , and go to (a). Repeat until all data are exhausted.

If this is IEKF, goto (j).

(j)  $i \leftarrow i + 1$

(k) Set  $\hat{x}^i(k+1) = \left[ R_{xx}^{(i-1)}(k+1) \right]^{-1} z_x^{i-1}(k+1)$

(l) If  $\|\hat{x}^i(k+1) - \hat{x}^{i-1}(k+1)\| < \epsilon$  for some previously-chosen small  $\epsilon > 0$ , then increment  $k$  by 1 and goto (a).

Repeat until all data are exhausted. If  $\|\hat{x}^i(k+1) - \hat{x}^{i-1}(k+1)\| \geq \epsilon$ , then goto (f).

Note: The derivations of the square root versions of the IEKF is arguably easier than the derivations of the non-square-root IEKF.

$$\textcircled{1} \quad y = h^{-1}(z) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + w_c$$

Let  $\bar{z} = h(x_1, x_2)$ . Expand  $y$  in Taylor series about  $\bar{z}$ :

$$y \approx h^{-1}(\bar{z}) + \left[ \frac{\partial h^{-1}}{\partial z} \Big|_{\bar{z}} \right] \underbrace{(z - \bar{z})}_w$$

Note that  $h^{-1}(z) = \begin{bmatrix} z_1 \cos z_2 \\ z_1 \sin z_2 \end{bmatrix}$

$$h^{-1}(\bar{z}) = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \bar{z} = \begin{bmatrix} \bar{p} \\ \theta \end{bmatrix}$$

Thus

$$y \approx \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} w}_w$$

$$\textcircled{2} \quad R_c = E[W_c W_c^T] = A E[W W^T] A^T = A R A^T, \text{ where}$$

$$R = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$$

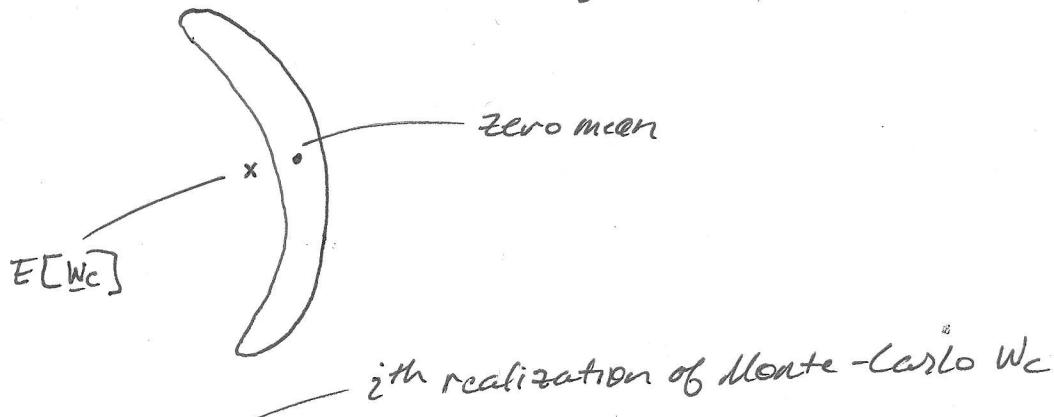
$$R_c = \frac{1}{2} \begin{bmatrix} \sigma_p^2 + 10^6 \sigma_\theta^2 & \sigma_p^2 - 10^6 \sigma_\theta^2 \\ \sigma_p^2 - 10^6 \sigma_\theta^2 & \sigma_p^2 + 10^6 \sigma_\theta^2 \end{bmatrix}$$

For  $\sigma_p = 10^5$ ,  $\theta_0 = 45^\circ$ ,

~~10^6 10^6~~

$$\textcircled{3} \quad y_{avg} \approx \begin{bmatrix} 70708.3 \\ 70707.1 \end{bmatrix}$$

④ For  $N=100$ , one realization gives  $E[\underline{w}_c] \approx \begin{bmatrix} -40 \\ 40 \end{bmatrix}$ , which is close to zero mean in a relative sense. Note that if  $\sigma_0$  were much greater than  $0.5^\circ$ ,  $E[\underline{w}_c]$  would depart significantly from zero mean: Banana-shaped density:



⑤  $E = \frac{1}{N} \sum_{i=1}^N \underline{w}_c^T R_c^{-1} \underline{w}_c$ . Then  $E \sim \chi^2_2$ .

From  $N=100$  Monte-Carlo runs,  $E = 1.9168205$ .

Bounds for  $P_F = 0.01$  are

$$r_1 = 1.522$$

$$r_2 = 2.553$$

Thus,  $R_c$  can be accepted as representative of the empirical covariance of  $\underline{w}_c$ . But if  $\sigma_0$  were much larger than  $0.5^\circ$ ,  $E \gg 2$  due to nonzero mean of  $\underline{w}_c$ , and test would ~~reject~~ reject  $R_c$ .

## 1. SOLUTION TO THE MULTIPLE-MODEL SOLAR PANEL DEPLOYMENT PROBLEM

See Figs. 1 and 2 and the attached code. A reasonable mode probability lower bound is  $\mu_{\min} = 2e - 3$ .

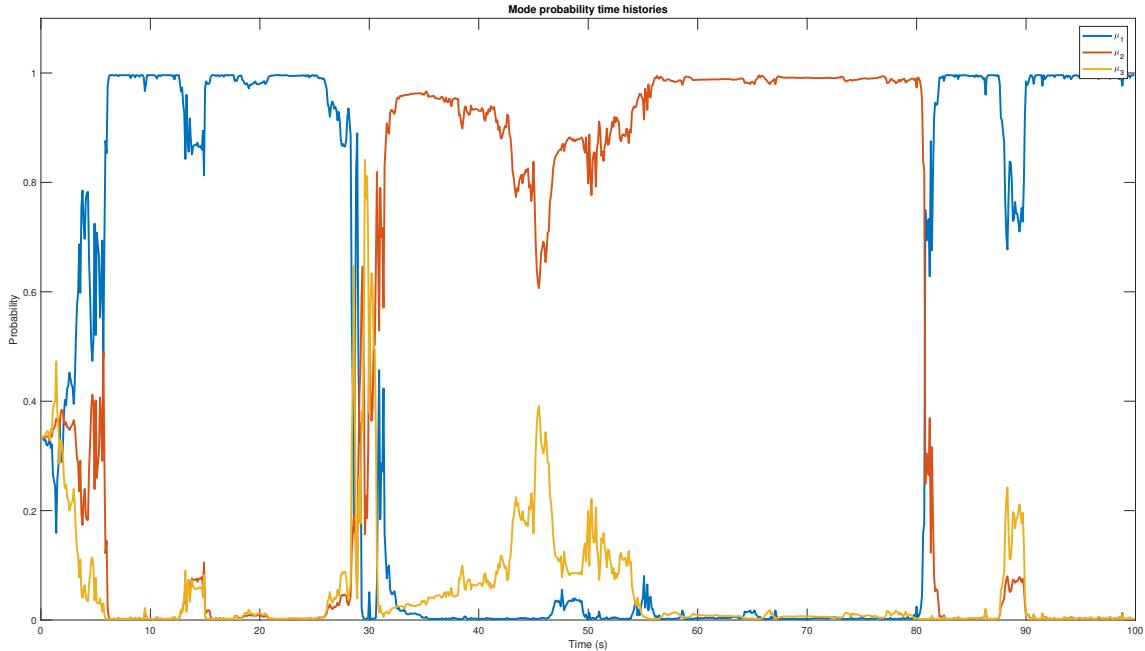


FIGURE 1. Mode probability time histories for three possible modes for a mode probability lower bound  $\mu_{\min} = 2e - 3$ .

- (a) What effect does changing the mode probability lower bound have on filter consistency?

Recall that, in the context of filtering, consistency is defined as in the lecture on Kalman filtering consistency (and in Bar Shalom, Section 5.4.2). To be consistent, a filter must be (1) unbiased, and (2) covariance-matching in the sense that  $E[\tilde{\mathbf{x}}(k)\tilde{\mathbf{x}}(k)^T]$  matches the state error covariance  $P(k)$  produced by the filter. Only a qualitative assessment of consistency is required here. One can see from Fig. 2 that for  $\mu_{\min} = 0.001$  the state errors appear unbiased and contained within the 1-sigma bounds approximately as would be expected. Thus,  $\mu_{\min} = 0.001$  seems to be a reasonable choice.

As  $\mu_{\min}$  is adjusted significantly above or below 0.15, the state errors remain unbiased but the filter's "covariance matching" degrades. As  $\mu_{\min}$  is increased, one finds that the filter-produced error covariance  $P(k)$  becomes more volatile and slightly pessimistic (too large) compared with the actual error covariance. In contrast, as  $\mu_{\min}$  is decreased the filter-produced error covariance  $P(k)$  becomes unresponsive (it approaches a flat line after the initial transient) and optimistic (too small) compared with the actual error covariance.

One could select  $\mu_{\min}$  rigorously by finding the value of  $\mu_{\min}$  that leads to  $E[\tilde{\mathbf{x}}(k)\tilde{\mathbf{x}}(k)^T]$  most closely matching the filter-produced error covariance  $P(k)$ . This would amount to a hypothesis test involving statistics of the form  $\epsilon(k) = \tilde{\mathbf{x}}^T(k)P^{-1}(k)\tilde{\mathbf{x}}(k)$ .

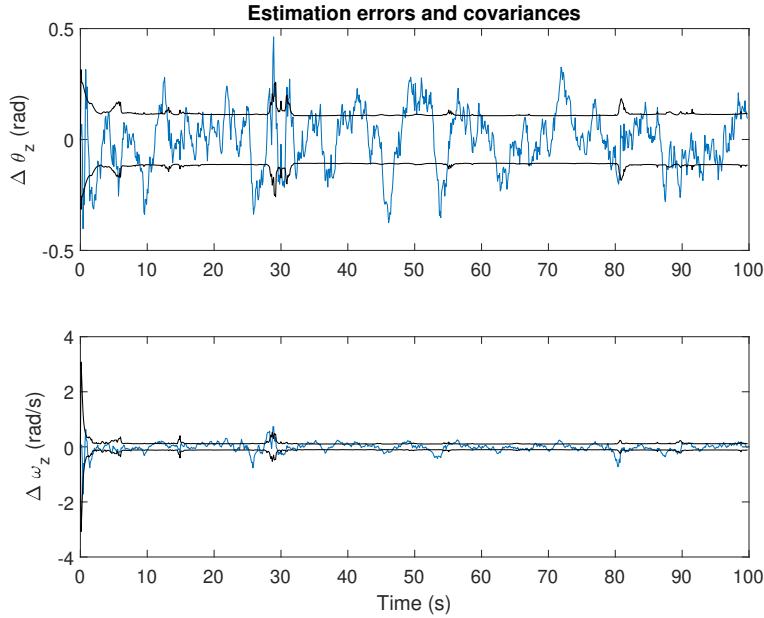


FIGURE 2. Error time histories for  $\theta_z$  (top panel) and  $\omega_z$  (bottom panel) for a mode probability lower bound  $\mu_{\min} = 2e-3$ .

- (b) Develop an algorithm for finding the optimum (consistency-maximizing) value of  $\mu_{\min}$ . Give a step-by-step description of your algorithm and also give the resulting optimum value of  $\mu_{\min}$ .

Assuming a truth-model simulator is available, as in the present case, the algorithm is a straightforward search over the range of reasonable  $\mu_{\min}$  for the value that yields best filter consistency as determined by the value of

$$\bar{\epsilon}_k \triangleq \frac{1}{N} \sum_{i=1}^N \epsilon^i(k)$$

where  $N$  denotes the number of independent Monte-Carlo runs. One could average over all  $\{\bar{\epsilon}_k : k = 1, 2, \dots, K\}$  from each run to produce a single statistic, if desired. Since  $\bar{\epsilon}_k \sim \chi_{n_x}^2$ , it follows that as  $N \rightarrow \infty$ ,  $\bar{\epsilon}_k \rightarrow n_x$ , which in this case is  $n_x = 2$ . The optimal value of  $\mu_{\min}$  is the one that puts  $\bar{\epsilon}_k$  (or its average over all  $k$ ) closest to  $n_x$  for a large  $N$ . The consistency-maximizing value is close to  $\mu_{\min} = 0.15$ . Note that this is *not* the optimal value if the goal is to minimize the empirical error covariance, it is simply the consistency-maximizing value. The optimum value for minimizing the empirical error covariance is approximately  $\mu_{\min} = 0.002$

- (c) What factors should be taken into account in the selection of the mode probability lower bound?

The static multiple-model filter already optimally accounts for the assumed “distance” between possible modes, the intensity of the process noise (as parameterized by  $Q(k)$ ), and the intensity of the measurement noise (as parameterized by  $R(k)$ ). The only purpose of the ad-hoc introduction of  $\mu_{\min} > 0$  was to allow the static multiple model estimator to handle dynamic model switching. The driving consideration for choice of  $\mu_{\min}$  is, therefore, the expected frequency of model transitions. As the expected time between model transitions goes to infinity, we have again the static case and  $\mu_{\min}$  should

approach zero. The assumed speed of each transition (i.e., whether instantaneous or gradual) also plays a role in the choice of  $\mu_{\min}$ .

- (d) What is the apparent mode time history (i.e., what modes were in effect and over what intervals)?

- 0 - 28 seconds :  $\mu = \mu_1$
- 28 - 80 seconds :  $\mu = \mu_2$
- 80 - 100 seconds :  $\mu = \mu_1$

Notice that there is no dead time between mode transitions—no “limbo” or indeterminate mode—because it was assumed that the transitions are instantaneous.

- (e) Suppose it were known that the faulty boom system tended to slip preferentially into one mode. How could the filter be modified to account for this?

Within the framework of our ad-hoc adaptation of the static multiple-model filter, one could assign a higher mode probability lower bound to the preferred mode than to the other two modes to indicate the systems tendency to slip into the preferred mode.

The proper way to handle this case (not ad-hoc) would be apply the dynamic multiple model estimator with defined transition probabilities from one mode to the others that was discussed in lecture.

Case i:

Linearized mean and covariance

75.895844641347608  
-3.977532674463731  
1.002080228655848 0.039693127323976  
0.039693127323976 1.757389988032180

Unscented mean and covariance

75.884285032749176  
-3.976926862262189  
1.002347478188725 0.039679121390983  
0.039679121390981 1.757390721692449

True mean and covariance

75.887372288526080  
-3.972265335810377  
1.011918239501008 0.041730202228521  
0.041730202228521 1.747572705912856

Case ii:

Linearized mean and covariance

75.895844641347608  
-3.977532674463731  
1.0e+02 \*  
0.020815991663193 0.206381415338717  
0.206381415338717 3.947991995884870

Unscented mean and covariance

73.294935073703527  
-3.841224781004712  
1.0e+02 \*  
0.156110670950729 0.199290912253928  
0.199290912253200 3.948363412513027

True mean and covariance

73.350814552954802  
-3.795155187502251  
1.0e+02 \*  
0.146659221048651 0.183039449437687  
0.183039449437687 3.680864195250434

## Sigma Point Filter and Particle Filter Answers

### I. Random number generation.

a.  $x = A \cdot z + b$

$$E[x] = E[A \cdot z + b] = A \cdot E[z] + E[b] = b$$

$$\begin{aligned} E[(x - E[x])(x - E[x])^T] &= E[(A \cdot z + b - b)(A \cdot z + b - b)^T] \\ &= E[(A \cdot z)(A \cdot z)^T] = A \cdot E[zz^T] \cdot A^T = A \cdot A^T \end{aligned}$$

b.  $b = \bar{x}$

$$A \cdot A^T = P_{xx}$$

$$A = chol(P_{xx})^T$$

c. See Matlab function *mvnrnd.m*.

### II. The unscented transform.

a. The linearization:  $x \approx \bar{r} \cdot \cos(\bar{\theta}) + \cos(\bar{\theta}) \cdot (r - \bar{r}) - \bar{r} \cdot \sin(\bar{\theta}) \cdot (\theta - \bar{\theta})$   
 $y \approx \bar{r} \cdot \sin(\bar{\theta}) + \sin(\bar{\theta}) \cdot (r - \bar{r}) + \bar{r} \cdot \cos(\bar{\theta}) \cdot (\theta - \bar{\theta})$

Take the expectation of the linearization:  
 $E[x] \approx \bar{r} \cdot \cos(\bar{\theta})$   
 $E[y] \approx \bar{r} \cdot \sin(\bar{\theta})$

Evaluate the second central moment to get the covariance:

$$\begin{aligned} E\left[\begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix} \begin{pmatrix} x - \bar{x} & y - \bar{y} \end{pmatrix}^T\right] &\approx \\ \begin{bmatrix} \cos(\bar{\theta}) & -\bar{r} \cdot \sin(\bar{\theta}) \\ \sin(\bar{\theta}) & \bar{r} \cdot \cos(\bar{\theta}) \end{bmatrix} \cdot \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \cdot \begin{bmatrix} \cos(\bar{\theta}) & -\bar{r} \cdot \sin(\bar{\theta}) \\ \sin(\bar{\theta}) & \bar{r} \cdot \cos(\bar{\theta}) \end{bmatrix}^T \end{aligned}$$

- b. See *problem2.m*. For part i) you'll find that the linearization is decent, since the range and angular uncertainty are small. For part ii) you'll find that the linearization has a substantial bias (more than 1-sigma in the range), and that the variance estimate in x is optimistic (much smaller than in reality).
- c. See *problem2.m*. For part i) you'll find the unscented transform performs about as well as the linearized transform. For part ii) you'll find the unscented transform predicts a mean much closer to the true mean, and its covariance matrix is not optimistic.

III. The sigma point filter.

- a. See *problem3.m*.
- b. See *problem3.m*.
  - i. The parameter  $\alpha$  has the largest effect on filter performance. Small values (0.01 and below) produce stable results and consistent estimates, mostly thanks to the stability of the measurements chosen. Larger values scale the sigma points too much, and may cause the rectangle estimate to collapse to a zero-width car (which happens to be another local maximum in the estimate's likelihood). The parameter  $\beta$  has smaller effects. Everything looks fine for values near 2, but the filter has a hard time converging to the correct car width at a value of 0. The filter also has a hard time tracking changes in the car's maneuvers for large values. Evidently the quartic term of the Taylor expansion is minimized near 2 for this system... As far as I can tell, the value of  $\kappa$  hardly influences the estimation results at all.
  - ii. What I'm trying to show here is that the information we have about the car's states (in particular its shape) are directly related to the maneuvers we see it perform. The car starts out moving directly toward the LIDAR, so we have almost no information about its length. When it turns, we suddenly get a lot more information. The SPF state covariance matrix reflects that.
  - iii. The primary difficulty of implementing this as an EKF is the measurement Jacobian: since the measurements are all generated via raytracing, the closed-form Jacobian doesn't look pretty at all.
  - iv. The bearing-bearing-range measurement is one of the few sets of measurements that remain stable and continuous over all orientations of the target car. The derivative of maximum range, for example, is undefined as a new side of the car becomes visible.

IV. The (bootstrap) particle filter.

- a. See *problem4.m*.
- b. See *problem4.m*.
  - i. The estimate shouldn't change much if you run it with 1000 particles (or at least, that's what I've seen). Fewer particles will most likely be inadequate to describe the entire state space fully, so it's likely that certain areas of the state space may not get sampled (sometimes, the particles might not be "near enough" to the correct state to let the filter converge to the right answer). Ideally, you'd want the MMSE estimate to be fairly stable, so having more particles is preferable.

- ii. This particular estimation problem is very tough because the information is extremely weak. When multiple clusters persist it means the posterior state density is multimodal: that is, multiple states are consistent with the set of sonar measurements.
- iii. The EKF has no shot at this problem. Even if you spent the time working out the Jacobians for the measurement function, the EKF would still have to make hard decisions about which three beacons the robot was measuring at each time interval, and any single wrong decision would likely make the filter diverge. The particle filter, on the other hand, effectively uses its particle set to maintain multiple hypotheses about which beacons it has seen. Particles that are consistent with the data stick around, and those that aren't get sampled out.