دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر

ترکیب داده/ اطلاعات

تمرین سری چهارم

محمدحسین بدیعی

شماره دانشجویی 810199106

گرایش : کنترل – هوش مصنوعی و رباتیک

استاد : دکتر بهزاد مشیری

بهار 1399-1400

در ابتدا معادلاتِ حالت تحول سیستم را که در صورت سوال ذکر شده است در نظر می‌گیریم.

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{u\cos(x_1) - (M+m)g\sin(x_1) + ml(\cos(x_1)\sin(x_1))x_2^2}{ml\cos^2(x_1) - (M+m)l}$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{u + ml(\sin(x_1))x_2^2 - mg\cos(x_1)\sin(x_1)}{M + m - m\cos^2(x_1)}$$

و همچنین معادله خروجی‌ای که مساله داده است نیز به صورت زیر است.

$$y_1 = x_1$$

$$y_2 = x_3$$

حال طبق خواسته‌ی مسأله مدل گسسته‌ی این سیستم را به صورت زیر بدست می‌آوریم.

$$\frac{x_1(k) - x_1(k-1)}{T_s} = x_2(k-1)$$

$$\Rightarrow x_1(k) = x_2(k-1) * T_s + x_1(k-1)$$

$$\frac{x_2(k) - x_2(k-1)}{T_s} = \frac{u(k)\cos(x_1(k-1)) - (M+m)g\sin(x_1(k-1)) + ml(\cos(x_1(k-1))\sin(x_1(k-1)))x_2^2(k-1)}{ml\cos^2(x_1(k-1)) - (M+m)l}$$

$$\Rightarrow x_2(k) = \frac{u\cos(x_1(k-1)) - (M+m)g\sin(x_1(k-1)) + ml(\cos(x_1(k-1))\sin(x_1(k-1)))x_2^2(k-1)}{ml\cos^2(x_1(k-1)) - (M+m)l} * T_s + x_2(k-1)$$

$$\frac{x_3(k) - x_3(k-1)}{T_s} = x_4(k-1)$$

$$\Rightarrow x_3(k) = x_4(k-1) * T_s + x_3(k-1)$$

$$\frac{x_4(k) - x_4(k-1)}{T_s} = \frac{u(k) + ml(\sin(x_1(k-1)))x_2^2(k-1) - mg\cos(x_1(k-1))\sin(x_1(k-1))}{M + m - m\cos^2(x_1(k-1))}$$

$$\Rightarrow x_4(k) = \frac{u(k) + ml(\sin(x_1(k-1)))x_2^2(k-1) - mg\cos(x_1(k-1))\sin(x_1(k-1))}{M + m - m\cos^2(x_1(k-1))} * T_s + x_4(k-1)$$

$$y(k) = x_1(k) + x_3(k)$$

در نهایت معادلات به شکل زیر در خواهد آمد:

$$x(k)=\begin{bmatrix} x_2(k-1)*T_s+x_1(k-1) \\ \frac{u(k)\cos(x_1(k-1))-(M+m)g\sin(x_1(k-1))+ml(\cos(x_1(k-1))\sin(x_1(k-1)))x_2^2(k-1)}{ml\cos^2(x_1(k-1))-(M+m)l}*T_s+x_2(k-1) \\ x_4(k-1)*T_s+x_3(k-1) \\ \frac{u(k)+ml\sin(x_1(k-1))x_2^2(k-1)-mg\cos(x_1(k-1))\sin(x_1(k-1))}{M+m-m\cos^2(x_1(k-1))}*T_s+x_4(k-1) \end{bmatrix}$$

$$y(k)=\begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}x(k)$$

حال که گسسته‌سازی با موفقیت صورت گرفته است، نویزها را طبق خواسته مسأله اضافه کرده و مطابق با متغیرهای صورت سوال معادل‌سازی می‌کنیم.

$$x(k)=\begin{bmatrix} x_2(k-1)*T_s+x_1(k-1) \\ \frac{u(k)\cos(x_1(k-1))-(M+m)g\sin(x_1(k-1))+ml(\cos(x_1(k-1))\sin(x_1(k-1)))x_2^2(k-1)}{ml\cos^2(x_1(k-1))-(M+m)l}*T_s+x_2(k-1) \\ x_4(k-1)*T_s+x_3(k-1) \\ \frac{u(k)+ml\sin(x_1(k-1))x_2^2(k-1)-mg\cos(x_1(k-1))\sin(x_1(k-1))}{M+m-m\cos^2(x_1(k-1))}*T_s+x_4(k-1) \end{bmatrix}$$

$$+w(k-1) \qquad ,w\sim N(0,Q) \qquad ,Q=\begin{bmatrix}0.9&0&0&0\\0&0.8&0&0\\0&0&0.9&0\\0&0&0&0.8\end{bmatrix}$$

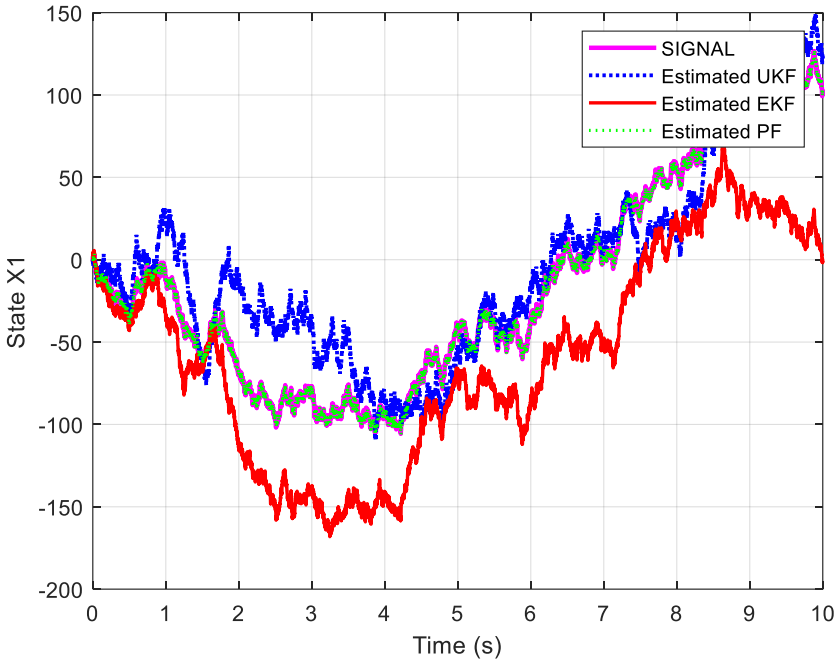
$$z(k)=\begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}x(k)+v(k) \qquad ,v\sim N(0,R) \qquad ,R=\begin{bmatrix}1.2&0\\0&0.8\end{bmatrix}$$

اکنون هر یک از فیلترها را در متلب پیاده‌سازی کرده و نتایج را نشان می‌دهیم.

در ابتدا از ما خواسته شده است که نتایج حاصل از اعمال فیلترها را بر روی سیستم ذکر شده در صورت سوال در چهار شکل نمایش دهیم.

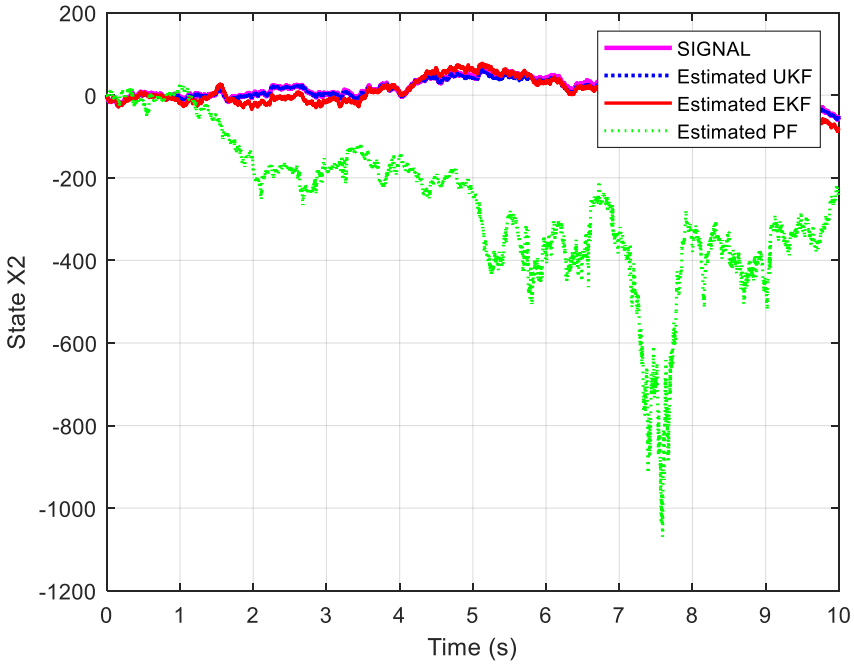
هر یک از این شکل‌ها سه سیگنالِ تخمین زده شده از سیستم را به همراه حالت اصلی نمایش می‌دهد به طوریکه شکل یک مربوط به حالتِ x1 از سیستم و شکل دوم برای حالت x2 از سیستم تا شکل چهارم که برای حالتِ x4 از سیستم می‌باشد.

حالت اولِ سیستم (X1)



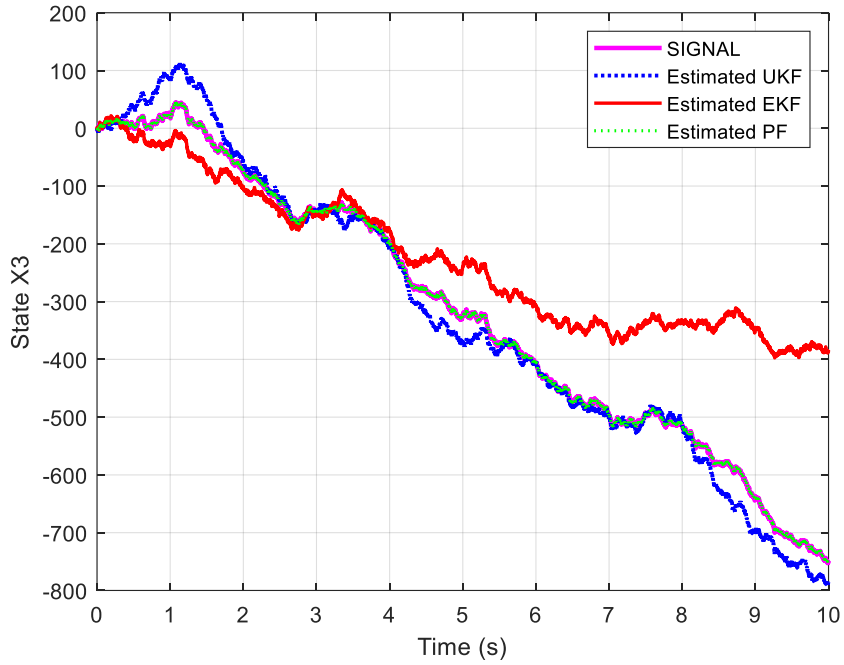
شکل 1 – ترسیم چهار سیگنال شامل حالت اولِ سیستم به همراهِ سه سیگنالِ بدست آمده از تخمین‌های EKF و UKF و PF

حالت دومِ سیستم (X2)



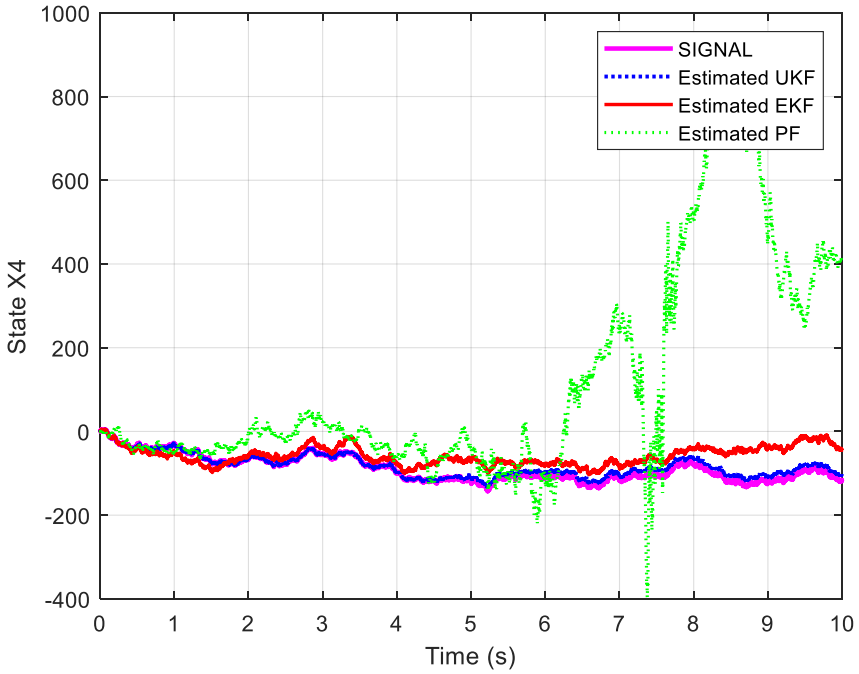
شکل 2 – ترسیم چهار سیگنال شامل حالت دومِ سیستم به همراهِ سه سیگنالِ بدست آمده از تخمین‌های EKF و UKF و PF

حالت سومِ سیستم (X3)



شکل 3 – ترسیم چهار سیگنال شامل حالت سومِ سیستم به همراهِ سه سیگنالِ بدست آمده از تخمین‌های EKF و UKF و PF

حالت چهارمِ سیستم (X4)



شکل 4 – ترسیم چهار سیگنال شامل حالت چهارمِ سیستم به همراهِ سه سیگنالِ بدست آمده از تخمین‌های EKF و UKF و PF

نتایج برای N=10001 مشاهده با Ts=0.001 می باشد. پس زمانِ شبیه سازیِ ما برابر با 10 sec شد.

اکنون می‌توانیم نسبت به خروجی‌هایی که از پیاده سازی در این بخش انجام دادیم، نتایج زیر را استنباط کنیم.

در ایتدا باید اشاره ای کنیم به سرعت ران شدنِ کدها در هر بخش. ما فایل ها را به صورت ماژولار نوشتیم و قادر بودیم از تک تکِ آنها به صورت جداگانه ران گرفته و الگوریتم ها را بررسی نماییم. در این حین سرعت ران شدنِ EKF از دو الگوریتمِ دیگر بالاتر بود. دلیلِ این امر را به راحتی با محاسبه‌ی complexity هر یک از الگوریتم ها در کد می‌تواند یافت. در EKF ما با کمترین حلقه‌های for نسبت به دو الگوریتمِ دیگر برخوردار بودیم و لذا می‌توان استدلال کرد که EKF کمترین runtime complexity را دارا می‌باشد.

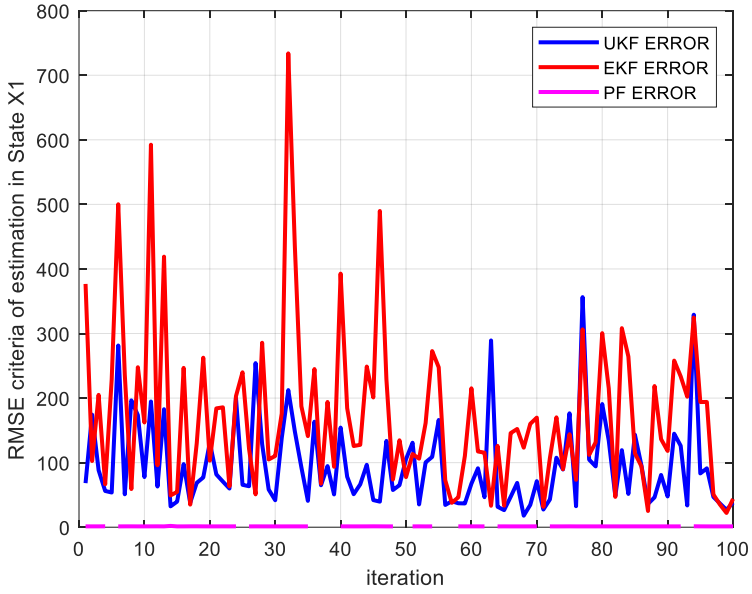
نکته قابل توجه بعدی بحثِ کاراییِ تخمین زدنِ هر یک از این الگوریتم‌ها در شرایط یکسان است. توجه می‌کنیم که UKF در مجموع تخمین‌گر مناسب تری برای تمامیِ حالات بوده است البته هر چند PF در حالاتی که سنسور قادر به اندازه‌گیری بوده (حالات اول و سوم) عملِ بسیار مطلوبی را از خود نشان داده است ولی در تخمینِ متغیرهای حالتِ دوم و چهارم اینگونه نیست. البته طبق تجربه و انجامِ آزمون و خطا در کد یافتم که عواملی مانندِ ذراتِ فیلتر تاثیرِ به سزایی از خود نشان می‌دهند ولی با این حال من بهترین عملکرد را برای مجموعِ چهار حالت در نظر گرفته و کاراییِ موثرِ UKF را بالاتر از بقیه می‌دانم. لذا همانطور که در موردِ runtime complexity ها بحث نمودیم، اینکه این فامتور (runtime complexity) برای UKF بالا باشد، آنچنان بی‌راه نیست و به گونه‌ای نشان‌دهنده‌ی عملکردِ خوبِ این فیلتر است. پس از فیلترِ UKF، عملکردِ EKF در مجموع برای تخمین بهتر است و در نهایت PF را در این رده بندی قرار می‌دهم. مشاهداتِ من نشان می‌دهد که PF نسبت به متغییرهایی که قابلِ اندازه‌گیری توسطِ سنسور نیستند، عملکردِ خوبی از خود بروز نخواهد داد و در شرایطِ نویزی برای یک سیستمِ تغییرپذیر مثل سیستمِ فوق، در صورتِ عدمِ اندازه‌گیریِ سنسوری از مجموعه متغیرها، فیلترخوبی نخواهد بود ولی اگر سنسورِ ما قادر به اندازه‌گیریِ تمامیِ حالاتِ سیستم باشد، بی شک این فیلتر یکی از فیلترهای مناسب برای تخمین است که این حقیقت را می‌توان از شکل 1 و شکل3 استنباط نمود.

در بخش آخر نیز ار ما خواسته شده است که Monte Carlo Simulation انجام دهیم.

طبق خواسته‌ی سوال دقیقا شکل‌ها را به گونه‌ای که ذکر شده برای معیارِ خطای RMSE و برای تک تکِ حالت‌های سیستم و برای هر سه الگوریتم رسم نمودیم. نتیجه به صورت زیر درآمد.

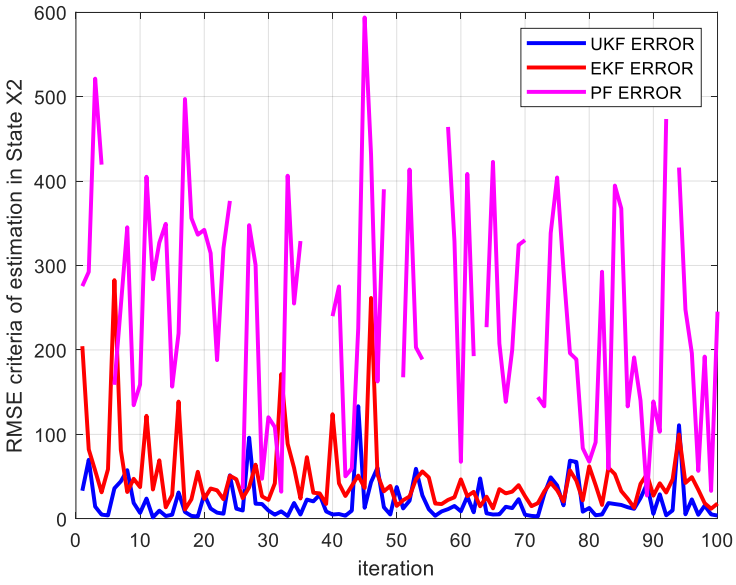
Monte Carlo Simulation

نتایج Monte Carlo Simulation برای حالت اولِ سیستم (X1)



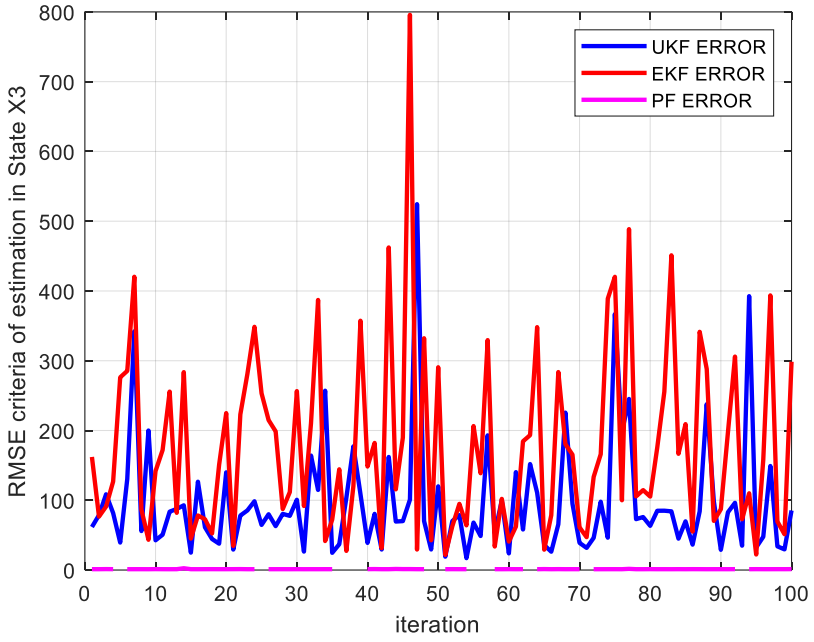
شکل 5 – نتایج Monte Carlo Simulation برای حالت اولِ سیستم (X1) با معیارِ خطای RMSE

نتایج Monte Carlo Simulation برای حالت دومِ سیستم (X2)



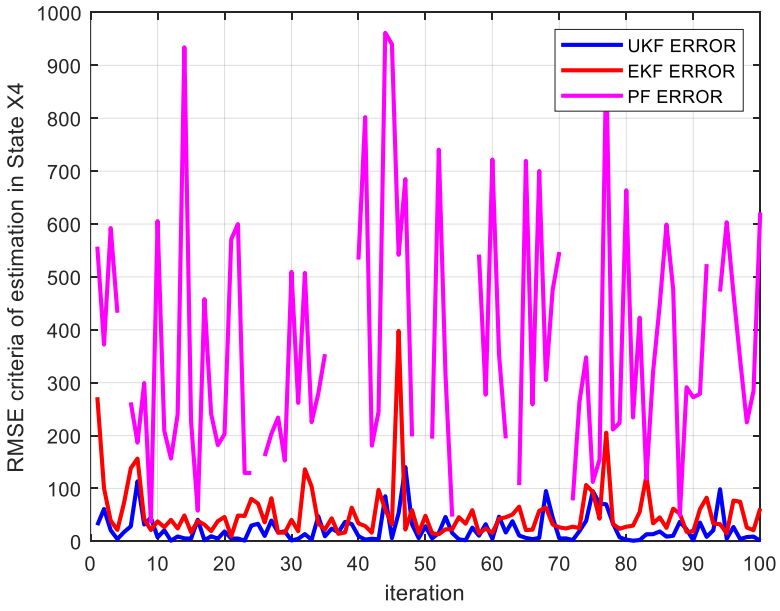
شکل 6 – نتایج Monte Carlo Simulation برای حالت دومِ سیستم (X2) با معیارِ خطای RMSE

نتایج Monte Carlo Simulation برای حالت سومِ سیستم (X3)



شکل 7 – نتایج Monte Carlo Simulation برای حالت سومِ سیستم (X3) با معیارِ خطای RMSE

نتایج Monte Carlo Simulation برای حالت چهارمِ سیستم (X4)



شکل 8 – نتایج Monte Carlo Simulation برای حالت چهارمِ سیستم (X4) با معیارِ خطای RMSE

در نهایت نتایجِ حاصل از شبیه‌سازیِ Monte Carlo با معیار خطای RMSE برای 100 ران مطابق با چهار شکلِ فوق می‌باشد که هر شکل ویژه‌ی یکی از حالاتِ سیستم است. در این شبیه‌سازی مشاهده می‌کنیم که خطا برای تخمینِ حالتِ یک سیستم توسط الگوریتم‌های UKF و EKF تقریبا برابر با 150 و برای فیلتر PF بسیار نزدیک به صفر است. این تفاسیر را برای حالتِ سومِ سیستم نیز مشاهده می‌کنیم. دلیلِ این امر، کارایی مطلوب توسطِ فیلترِ PF برای این دو حالت را اندازه‌گیریِ سینسوری در حالاتِ یک و سه از سیستم می‌دانم و لذا این فیلتر را بسیار وابسته به اندازه‌گیریِ حالات توسطِ سنسور برآورد می‌کنیم. چراکه همانطور که در این شبیه‌سازی نیز به گونه‌ای دیگر و با معیارِ خطا عملکردِ فیلترها را بررسی می‌کنیم، عملکردِ فیلترِ PF در حالاتِ دو و چهار که سنسور خروجی ای روی آنها به ما نمی‌دهد، چندان مناسب نیست.

برای حالاتِ دو و چهار از سیستم، اگرچه PF بدلیلِ فوق عملکردِ مطلوبی از خود نشان نمی‌دهد ولی فیلترهای UKF و EKF خطای RMSE را برای این سیستم حدود 20 (به طور متوسط) برآورد می‌کنند. هر چند که عملکردِ UKF نسبت به EKF نیز مطلوب‌تر است که در این مورد در بخشِ قبل که پیاده‌سازیِ فیلترها بود بحث نمودیم.

در انتها لازم به ذکر است که یازده فایل در قسمتِ Codes به صورتِ ماژولار پیاده سازی شده است که 9 فایل فانکشن هستند و دو فایل برای ران گرفتن و انجامِ شبیه سازی می‌باشند. لذا برای نمایش موارد خواسته شده در بخش قبل کافی است فایلِ Plot_EKF_UKF_PF.m را اجرا نمایید و برای نمایشِ شبیه‌سازیِ Monte Carlo Simulation کافی است که فایلِ Plot_Error.m را اجرا نمایید.

با تشکر - بدیعی