Performance Comparison of Nonlinear Estimation Methods

Brendan Drew, Norman Facas, Nick Hunter

Introduction

When choosing a state estimation approach, several competing concerns must be balanced: computational effort, system noise, system non-linearity, estimator accuracy, and robustness all form a trade space that must be carefully considered before choosing a particular approach. In this project, we explored the performance of several different state estimation problems on an exemplar system consisting of two masses coupled by (nonlinear) springs and dampers. The system was chosen due to its ability to model the dynamics of a large variety of physical systems and relative simplicity. We used four different metrics to evaluate the performance of each estimator under a wide variety of degree of noise and degree of nonlinearity.

System Model

The system states are the displacements and velocities of each mass. The net forces acting on each mass are modeled by the following equations:

$$
\begin{aligned}
F_{s1} &= -k_1 x_1 + \varepsilon_1 x_1^3 \\
F_{d1} &= -\delta_1 \dot{x}_1 \\
F_{s2} &= -k_2 (x_2 - x_1) + \varepsilon_2 (x_2 - x_1)^3 \\
F_{d2} &= -\delta_1 (\dot{x}_2 - \dot{x}_1) \\
F_{m1}^{net} &= F_{s1} + F_{d1} - F_{s2} - F_{d2} \\
F_{m2}^{net} &= F_{s2} + F_{d2} + F_{ext}
\end{aligned}
$$

Experimental Setup

All experiments were performed using Simulink. There are four models that are run: a calibration model, and three models corresponding to the Luenberger Observer, EKF, and UKF respectively. A master script loads parameters, running each model in turn and gathering statistics. Source code is available on the attached CD.

Metrics

Four metrics were chosen. The first two (median RMSE and maximum RMSE) focus on average case and worst case estimator error respectively. These metrics are appropriate for situations where computational effort is not a direct concern. In real systems, hardware constraints will play an important role in which estimators are computationally feasible. To that end, our second set of metrics (median MSE x average computational time and maximum MSE x average computational time) are designed to capture the idea of "bang for the buck" in the average and worst case scenarios.

Methods and Estimator Design

Five common estimators were selected for evaluation: a Luenberger observer with LQR-designed gains (steady-state Kalman Filter gain), a Kalman filter, an Extended Kalman Filter, an Unscented Kalman Filter, and a Particle Filter.

Extended Kalman Filter

The extended Kalman was also applied to the system, see Equation 1. The extended Kalman filter is an extension of the Kalman filter. Instead of linearizing the system about one point it linearizes the data based on the current state. It was not possible to directly convert the nonlinear continuous equations into nonlinear discrete equations; rather the data was linearized in continuous time and then converted to discrete equation. The other difference from the Kalman filter is that the extended Kalman filter updates the state based on the nonlinear equation rather then the linerazied equations. To update the state the data had to be simulated in continuous time since over the discrete time period since it was not possible to have non linear discrete equations.

$$
\begin{aligned}
&\hat{x}_k^{(-)} = f(\hat{x}_{k-1}^{(+)}, u_k) \\
&\phi_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k^{(-)}} \\
&P_k^{(-)} = \phi_{k-1} P_{k-1}^{(+)} \phi_{k-1}{}^T + G_{k-1} Q_{k-1} G_{k-1}{}^T \\
&\overline{K}_k = P_k^{(-)} H_k{}^T \left(H_k P_k^{(-)} H_k{}^T + R\right)^{-1} \\
&\hat{x}_k^{(+)} = \hat{x}_k^{(-)} + \overline{K}_k \left(z_k - H_k \hat{x}_k^{(-)}\right) \\
&P_k^{(+)} = P_k^{(-)} - \overline{K}_k H_k P_k^{(-)}
\end{aligned}
\qquad (1)
$$

Unscented Kalman Filter

We implemented the symmetric UKF algorithm presented in [Simon]:

For the time update, generate $\chi_i^{(-)}$ using the SVD of $P_{k-1}^{(-)}$: $\chi_i^{(-)} = \sqrt{n\sigma_i}\mathbf{v}_i$, $\chi_{2i}^{(-)} = -\sqrt{n\sigma_i}\mathbf{v}_i$ where n is the dimension of the space, σ_i is the i[th] singular value that has been *regularized* (if $0 \le \sigma_i < \tau$, set $\sigma_i = \tau$), and $\mathbf{v}_i$ is the i[th] column of the V matrix and $\chi_0^{(-)} = 0$. Geometrically, these sigma points correspond to the center of the distribution and steps along each of the covariance's eigen-axes. The weights w_i are

$$
w_i = \begin{cases} \dfrac{\kappa}{2(n+\kappa)}, \text{ if } i = 0 \\ \dfrac{1}{2(n+\kappa)}, \text{ otherwise} \end{cases}
$$

where $\kappa = 3 - n$ has been chosen to minimize distortion of higher order moments of the distribution. The previous mean is added to the sigma points, which are then propagated through the system dynamics and then used to empirically estimate the prior mean and state covariance.

$$
\begin{aligned}
&\hat{x}_k^{(-)} = \sum_i w_i f(\hat{x}_{k-1}^{(+)} + \chi_i^{(-)}, u_k) \\
&P_k^{(-)} = \sum_i w_i \left(f(\hat{x}_{k-1}^{(+)} + \chi_i^{(-)}, u_k) - \hat{x}_k^{(-)}\right)\left(f(\hat{x}_{k-1}^{(+)} + \chi_i^{(-)}, u_k) - \hat{x}_k^{(-)}\right)^T + Q_{k-1}
\end{aligned}
$$

During the measurement update, a new set of sigma points based on the new prior covariance are generated. The new sigma points are propagated through the output equation (linear in our case) and used to empirically estimate the output covariance, the cross covariance, and the Kalman gain.

$$
\begin{aligned}
&P_y = \sum_i w_i \left(H\left(\hat{x}_{k-1}^{(+)} + \chi_i^{(+)}\right) - \hat{y}_k\right)\left(H\left(\hat{x}_{k-1}^{(+)} + \chi_i^{(+)}\right) - \hat{y}_k\right)^T \\
&P_{xy} = \sum_i w_i \left(f(\hat{x}_{k-1}^{(+)} + \chi_i^{(-)}, u_k) - \hat{x}_k^{(-)}\right)\left(H\left(\hat{x}_{k-1}^{(+)} + \chi_i^{(+)}\right) - \hat{y}_k\right)^T \\
&\overline{K}_k = P_{xy} P_y^{-1}
\end{aligned}
$$

Finally, the posterior estimates of state mean and covariance are generated by applying the Kalman gain to each sigma point and computing weighted sample expectations.

$$\hat{x}_k^{(+)} = \sum_i w_i \left(\hat{x}_{k-1}^{(-)} + \chi_i^{(+)} + \overline{K}_k \left(\hat{y}_k - H\left(\hat{x}_{k-1}^{(+)} + \chi_i^{(+)}\right)\right)\right)$$

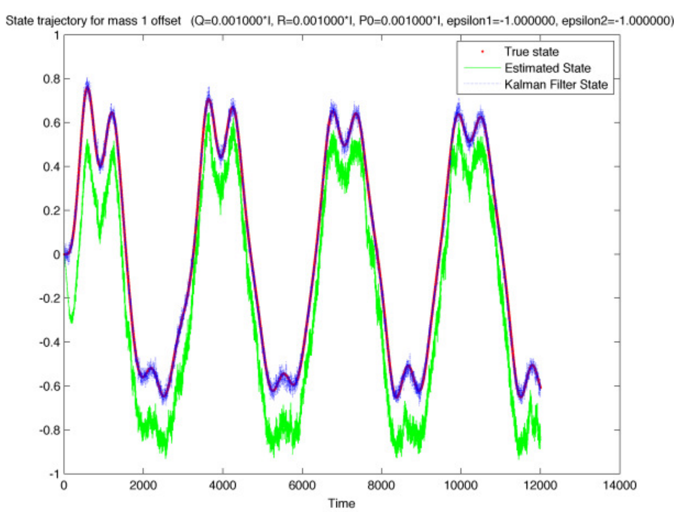
$$P_k^{(+)} = \sum_i w_i \left(\hat{x}_{k-1}^{(-)} + \chi_i^{(+)} + \overline{K}_k \left(\hat{y}_k - H\left(\hat{x}_{k-1}^{(+)} + \chi_i^{(+)}\right)\right) - \hat{x}_k^{(+)}\right)\left(\hat{x}_{k-1}^{(-)} + \chi_i^{(+)} + \overline{K}_k \left(\hat{y}_k - H\left(\hat{x}_{k-1}^{(+)} + \chi_i^{(+)}\right)\right) - \hat{x}_k^{(+)}\right)^T$$

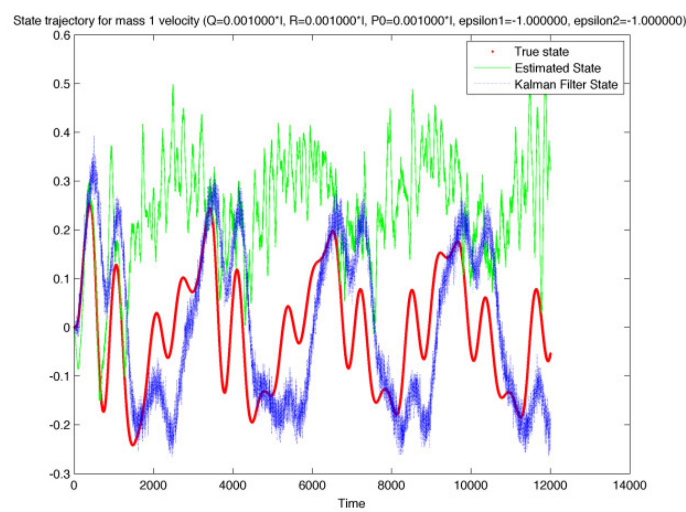
Particle Filter

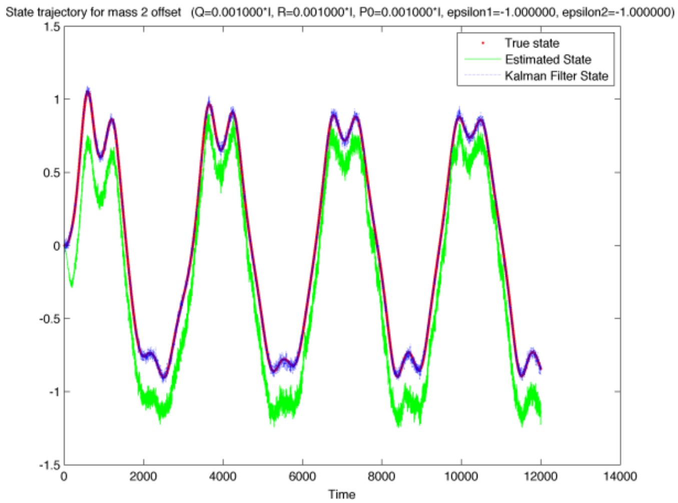
Our final estimation method was an implementation of a SIR (sample, importance, resample) particle filter with roughening to prevent sample impoverishment, following the algorithm in [Simon].

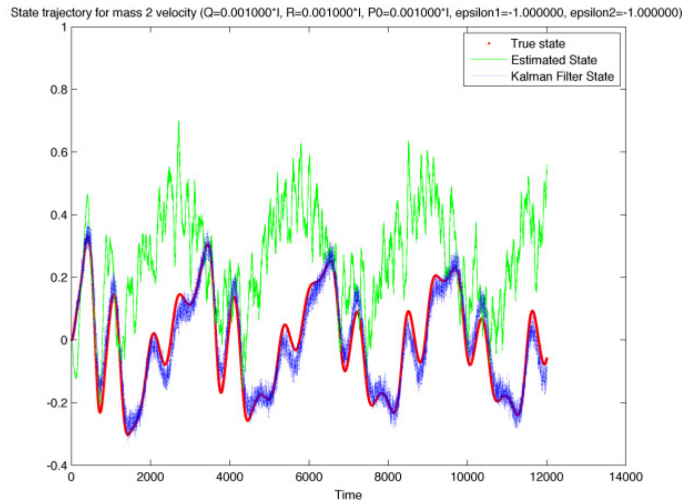
An initial population X^k of particles is sampled from the prior distribution $P^k[x_k \mid x_{1:k-1}, y_{1:k-1}]$. Each of these particles are propagated using the system dynamics which were approximated by Euler's method. For each particle, we computed the particle's importance weight $q_k^i = P\left[\hat{y}_k \mid X_i^k\right]$. The sample weights are normalized to have unit sum. We then resample the particles with replacement, each particle being chosen with likelihood equal to its importance weight. The new population of particles is asymptotically distributed as $P^k[x_k \mid x_{1:k-1}, y_{1:k}]$. These samples are then roughened by adding white noise which prevents sample impoverishment (i.e. lack of any population diversity in the particles). The output state was computed as the sample mean of the posterior particle distribution.

Unfortunately, the particle filter proved to be computationally untractable, largely due to implementation issues. The filter was implemented as a mixture of a C++ Simulink™ block that provided the core algorithm and Matlab™ m-files that implemented the specific system dynamics and output likelihood calculations. As a consequence, each iteration of the algorithm required O(2n) Matlab function calls, resulting in a ~38 hr run-time for a single test scenario with 50,000 particles. As the figures below demonstrate, 50,000 particles were not sufficient to accurately capture the unmeasured system dynamics.





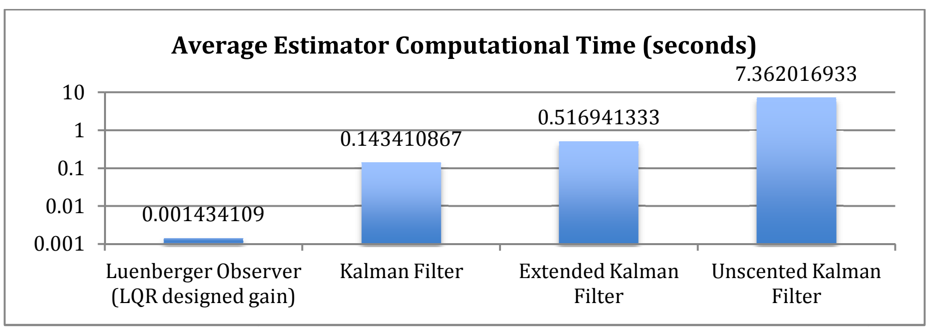




Results and Analysis

Full simulation results are presented in Appendix A. The tables below summarize best performance under each of the four metrics we chose, with the best highlighted in gray. For the "bang for the buck" tests, the LQR designed Luenberger observer was not considered for recommendation outside of the linear case. In all cases, lower scores indicate better performance with the best performer highlighted in gray.

Computational Effort



The UKF is, on average, 50x slower than the KF and 14x slower than the EKF.

Median RMSE Metric

	Linear				Negligible Nonlinearity				Low Nonlinearity				Moderate Nonlinearity				Large Nonlinearity			
Noise	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF
Low	0.001	0.001	0.002	0.002	0.00131	0.001	0.002	0.002	0.03763	0.037	0.003	0.002	0.137	0.136	0.002	0.001	0.207	0.205	0.002	0.001
Medium	0.029	0.028	0.016	0.002	0.02854	0.028	0.016	0.002	0.0518	0.051	0.017	0.002	0.134	0.133	0.017	0.001	0.207	0.205	0.017	0.001
High	0.901	0.896	0.509	0.027	0.90178	0.896	0.507	0.027	0.90479	0.901	0.513	0.128	0.938	0.932	0.54	0.131	0.909	0.904	0.577	0.312

For the median RMSE metric the UKF is the clear winner, dramatically outperforming EKF and KF. Surprisingly, the UKF outperforms the EKF and KF even for the purely linear system. For increasing non-linearity, the UKF typically out-performs the KF by a factor of between 3 and 200, depending on noise.

Maximum RMSE Metric

	Linear				Negligible Nonlinearity				Low Nonlinearity				Moderate Nonlinearity				Large Nonlinearity			
Noise	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF
Low	0.005	0.005	0.002	0.003	0.00714	0.007	0.002	0.003	0.14943	0.148	0.002	0.004	0.334	0.332	0.002	0.003	0.454	0.449	0.002	0.003
Medium	0.177	0.177	0.009	0.004	0.17889	0.178	0.009	0.004	0.28939	0.288	0.009	0.004	0.439	0.436	0.009	0.003	0.507	0.502	0.009	0.003
High	5.616	5.596	0.262	5.48	5.61841	5.599	0.263	5.487	5.77437	5.754	0.271	4.97	5.713	5.693	0.286	7.905	5.629	5.606	0.322	19.07

For maximum MSE error, the EKF and UKF generally trade off. This is partly due to greatly increased range of UKF error for large noise, although the variance is still less than that of the EKF. For applications that require strong bounds on maximum error, the EKF appears, on balance, to be the better choice for this system.

Combined Median MSE and Computational Effort

	Linear				Negligible Nonlinearity				Low Nonlinearity				Moderate Nonlinearity				Large Nonlinearity			
Noise	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF
Low	2E-09	2E-07	3E-06	2E-05	2E-09	2E-07	3E-06	2E-05	2E-06	0.0002	5E-06	2E-05	3E-05	0.0026	3E-06	2E-05	6E-05	0.006	3E-06	1E-05
Medium	1E-06	0.0001	0.0001	2E-05	1E-06	0.0001	0.0001	2E-05	4E-06	0.0004	0.0001	2E-05	3E-05	0.0025	0.0001	2E-05	6E-05	0.006	0.0001	1E-05
High	0.0012	0.1151	0.1338	0.0054	0.0012	0.1152	0.133	0.0053	0.0012	0.1163	0.1361	0.12	0.0013	0.1245	0.1506	0.1262	0.0012	0.1172	0.1719	0.716

Incorporating both mean square error and average computation time gives the LQR designed Luenberger observer a dramatic advantage. As a consequence, the observer was only considered for low noise, and no or negligible nonlinearity cases. In this scenario, we can see that despite the dramatically increased computational complexity, the UKF tends to offer the best “bang for the buck” in a wide variety of cases.

Combined Maximum MSE and Computational Effort

	Linear				Negligible Nonlinearity				Low Nonlinearity				Moderate Nonlinearity				Large Nonlinearity			
Noise	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF	LQR	KF	EKF	UKF
Low	4E-08	4E-06	1E-06	8E-05	7E-08	7E-06	1E-06	8E-05	3E-05	0.003	2E-06	1E-04	2E-04	0.016	1E-06	6E-05	3E-04	0.029	2E-06	6E-05
Mediur	5E-05	0.004	4E-05	1E-04	5E-05	0.005	4E-05	1E-04	1E-04	0.012	4E-05	1E-04	3E-04	0.027	4E-05	6E-05	4E-04	0.036	4E-05	6E-05
High	0.045	4.491	0.035	221.1	0.045	4.495	0.036	221.6	0.048	4.748	0.038	181.9	0.047	4.648	0.042	460.1	0.045	4.506	0.054	2678

A running theme of the combined metrics is the fact that the almost trivial computation required to implement the LQR designed Luenberger observer gives it a dramatic advantage. Again, the observer was only considered for the low noise and minimal non-linearity cases. For this pessimistic scenario, the EKF dominates, in large part due to the greatly increased range of UKF mean square errors in greater higher noise.

Conclusions and Future Work

The ideal choice of estimator is extremely application dependent. We examined a variety of scenarios and showed that the EKF tends to give better worst cast performance in the presence of increased noise while the UKF is generally the best performing estimator in terms of median root mean square error. Our particle filter implementation, which we initially expected would out perform all the other approaches in terms of accuracy, proved to be computationally infeasible.

In terms of future work, there are several relatively straight forward optimizations to the particle filter that could be made, including vectorizing the Matlab portions and applying the Mex compiler to produce native code. If an improvement of 3-4 orders of magnitude were possible, we could justify repeating the experiments with a suitably large number of particles to get reasonable accuracy.

Works Cited

[Simon] Dan Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches, Wiley Interscience, 2006.

Appendix A: Simulation Results

Raw Data

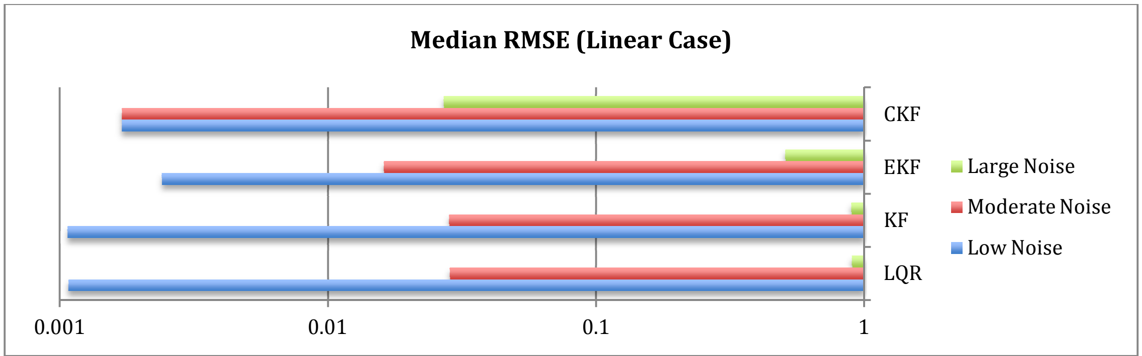
Experiment			LQR Luenberger Observer						
Noise	Nonlinearity	Calibration Time	Time (seconds)	Minimum MSE	25th Percentile MSE	Median MSE	75th Percentil MSE	Maximum MSE	Variance
1E-06	0	1.53486	1.33627	0	0.000731	0.00108	0.001635	0.005426	0.000001
1E-06	-0.001	1.31017	1.336348	0	0.000966	0.00131	0.001884	0.007142	0.000001
1E-06	-0.1	1.3058	1.336339	0	0.01284	0.03763	0.062048	0.149434	0.001132
1E-06	-1	1.33453	1.336851	0	0.036685	0.13704	0.184306	0.334227	0.00634
1E-06	-10	1.36127	1.336397	0	0.108631	0.20723	0.261071	0.453527	0.012876
0.001	0	1.31099	1.336586	0	0.013746	0.02855	0.047785	0.177395	0.00062
0.001	-0.001	1.31086	1.336998	0	0.013745	0.02854	0.047784	0.178888	0.000619
0.001	-0.1	1.33538	1.336699	0	0.032852	0.0518	0.078303	0.289389	0.001202
0.001	-1	1.30561	1.336932	0	0.059501	0.13394	0.186762	0.438684	0.005662
0.001	-10	1.33241	1.337234	0	0.114876	0.20677	0.268951	0.506704	0.012282
1	0	1.31268	1.336557	0	0.434246	0.9011	1.510988	5.615642	0.620456
1	-0.001	1.33083	1.336892	0	0.434092	0.90178	1.511616	5.618408	0.620501
1	-0.1	1.30438	1.336806	0	0.444378	0.90479	1.520198	5.774374	0.617448
1	-1	1.34238	1.336463	0	0.50967	0.93795	1.553332	5.71281	0.60869
1	-10	1.29717	1.337457	0	0.469658	0.90861	1.522756	5.628762	0.609031

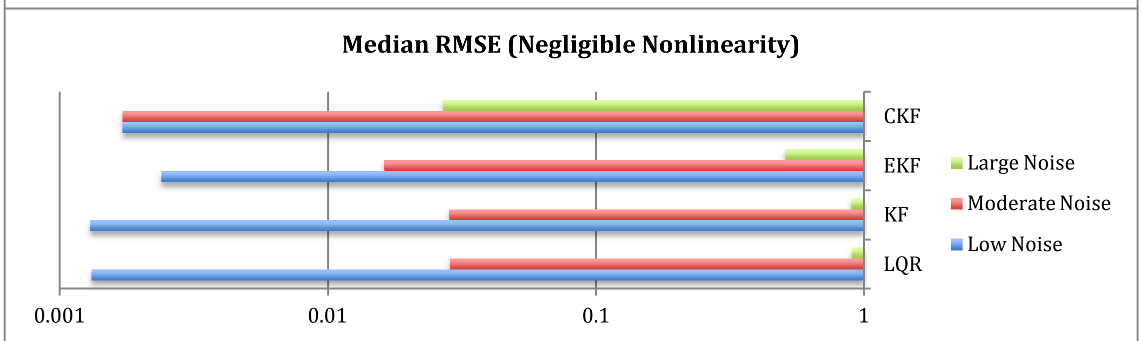
Experiment			Kalman Filter						
Noise	Nonlinearity	Calibration Time	Time (seconds)	Minimum MSE	25th Percentile MSE	Median MSE	75th Percentil MSE	Maximum MSE	Variance
1E-06	0	1.53486	1.433469	0.000022	0.000737	0.00107	0.001645	0.005458	0.000001
1E-06	-0.001	1.31017	1.441307	0.000056	0.000934	0.0013	0.001868	0.007139	0.000001
1E-06	-0.1	1.3058	1.440373	0.000056	0.012444	0.03717	0.061627	0.148322	0.001124
1E-06	-1	1.33453	1.491636	0.000056	0.035986	0.13587	0.18257	0.331713	0.006265
1E-06	-10	1.36127	1.446189	0.000056	0.106948	0.20521	0.258578	0.449071	0.012679
0.001	0	1.31099	1.465127	0.000336	0.013692	0.02834	0.047453	0.176821	0.000614
0.001	-0.001	1.31086	1.506287	0.000391	0.013677	0.02834	0.047478	0.178294	0.000614
0.001	-0.1	1.33538	1.476393	0.000596	0.03254	0.05143	0.077813	0.28775	0.001189
0.001	-1	1.30561	1.499739	0.000596	0.058645	0.1328	0.185162	0.43571	0.005585
0.001	-10	1.33241	1.529879	0.000596	0.114049	0.2047	0.266586	0.502232	0.012092
1	0	1.31268	1.462218	0.011561	0.432827	0.89595	1.503712	5.596087	0.614267
1	-0.001	1.33083	1.495676	0.009378	0.432264	0.89637	1.503631	5.598831	0.614313
1	-0.1	1.30438	1.487147	0.01659	0.442106	0.90052	1.511086	5.753774	0.611288
1	-1	1.34238	1.452819	0.016589	0.507429	0.93165	1.544646	5.692826	0.602782
1	-10	1.29717	1.552221	0.01508	0.467616	0.90412	1.515557	5.605583	0.603118

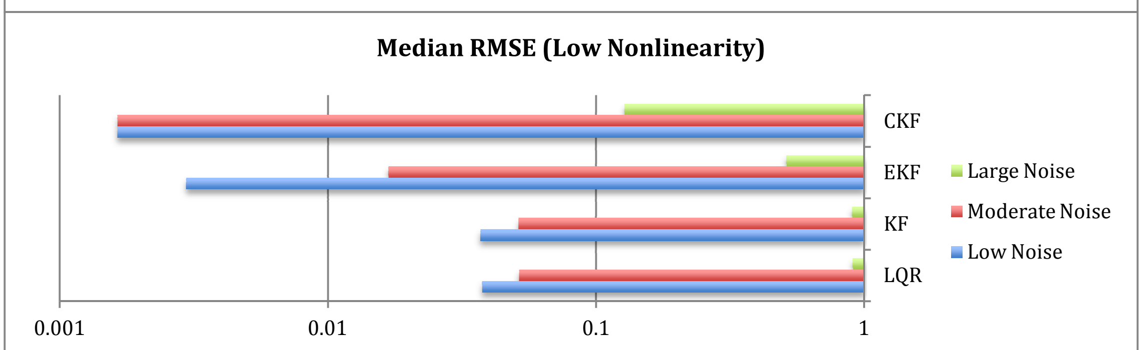
Experiment			EKF						
Noise	Nonlinearity	Calibration Time	Time (seconds)	Minimum MSE	25th Percentile MSE	Median MSE	75th Percentil MSE	Maximum MSE	Variance
1E-06	0	1.53486	1.787389	0.000109	0.001652	0.0024	0.003206	0.006435	0.000001
1E-06	-0.001	1.31017	1.851913	0.000109	0.001669	0.0024	0.003186	0.006396	0.000001
1E-06	-0.1	1.3058	1.954365	0.000109	0.001798	0.00296	0.004023	0.00678	0.000002
1E-06	-1	1.33453	1.830802	0.000067	0.001569	0.00222	0.002995	0.006932	0.000001
1E-06	-10	1.36127	1.806464	0.000109	0.001711	0.00243	0.003165	0.007758	0.000001
0.001	0	1.31099	1.804174	0.000116	0.008524	0.01621	0.02682	0.087872	0.000185
0.001	-0.001	1.31086	1.839575	0.00023	0.008537	0.01622	0.026852	0.088032	0.000185
0.001	-0.1	1.33538	1.873387	0.000087	0.009252	0.01683	0.027183	0.096942	0.000182
0.001	-1	1.30561	1.858508	0.000284	0.008965	0.0167	0.027346	0.098298	0.000184
0.001	-10	1.33241	1.853511	0.000298	0.009285	0.0168	0.026821	0.0867	0.000173
1	0	1.31268	1.811692	0.003015	0.261887	0.50874	0.8553	2.839678	0.190184
1	-0.001	1.33083	1.94174	0.003761	0.263009	0.50719	0.855561	2.844258	0.190225
1	-0.1	1.30438	1.83675	0.003027	0.270975	0.51318	0.866417	2.876667	0.18975
1	-1	1.34238	1.876772	0.006917	0.286219	0.53984	0.866889	2.996261	0.191178
1	-10	1.29717	1.856395	0.016352	0.321936	0.57666	0.908106	2.861806	0.183327

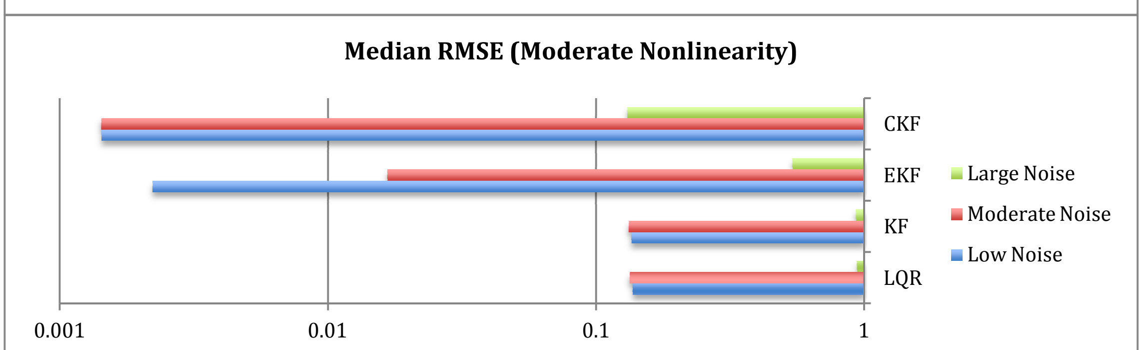
Experiment			UKF						
Noise	Nonlinearity	Calibration Time	Time (seconds)	Minimum MSE	25th Percentile MSE	Median MSE	75th Percentil MSE	Maximum MSE	Variance
1E-06	0	1.53486	8.642361	0	0.001124	0.00171	0.002347	0.003263	0.000001
1E-06	-0.001	1.31017	8.913946	0	0.001139	0.00172	0.002327	0.003261	0.000001
1E-06	-0.1	1.3058	8.913946	0	0.000806	0.00164	0.002677	0.003723	0.000001
1E-06	-1	1.33453	8.517694	0	0.000864	0.00143	0.001749	0.002767	0
1E-06	-10	1.36127	8.389935	0	0.00101	0.00128	0.001509	0.00289	0
0.001	0	1.31099	8.57394	0.00006	0.001127	0.00171	0.002346	0.003604	0.000001
0.001	-0.001	1.31086	8.721397	0.00006	0.001142	0.00172	0.002327	0.003601	0.000001
0.001	-0.1	1.33538	8.511662	0.00006	0.000807	0.00164	0.002676	0.003747	0.000001
0.001	-1	1.30561	8.709157	0.00006	0.000867	0.00143	0.00175	0.002854	0
0.001	-10	1.33241	8.752783	0.00006	0.001012	0.00128	0.00151	0.002925	0
1	0	1.31268	8.762593	0.000068	0.004127	0.02708	0.171716	5.479637	0.14392
1	-0.001	1.33083	8.718774	0.000083	0.00418	0.02681	0.172049	5.486781	0.143928
1	-0.1	1.30438	8.974281	0.005904	0.046794	0.1277	0.302166	4.970234	0.156403
1	-1	1.34238	8.649669	0.007047	0.049403	0.13092	0.574709	7.905448	0.46032
1	-10	1.29717	8.707433	0.03376	0.082486	0.31186	0.939726	19.07172	1.353962

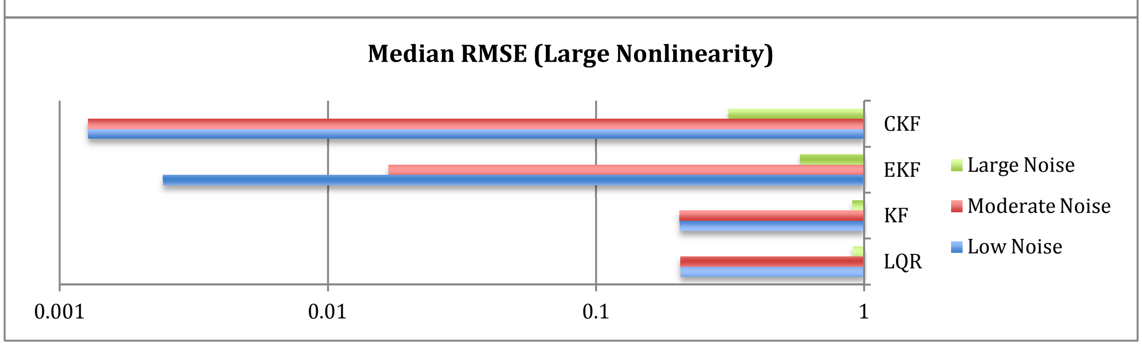
Median RMSE


Median RMSE (Linear Case)
CKF
EKF
KF
LQR
Large Noise
Moderate Noise
Low Noise
0.001
0.01
0.1
1

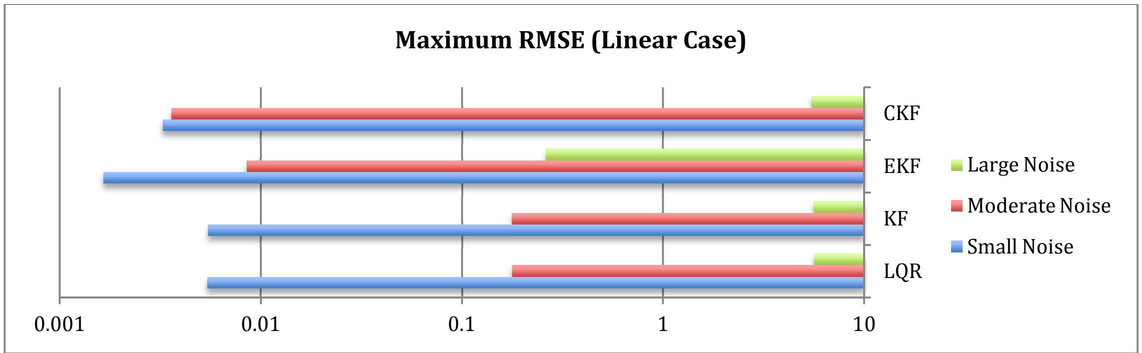

Median RMSE (Negligible Nonlinearity)
CKF
EKF
KF
LQR
Large Noise
Moderate Noise
Low Noise
0.001
0.01
0.1
1

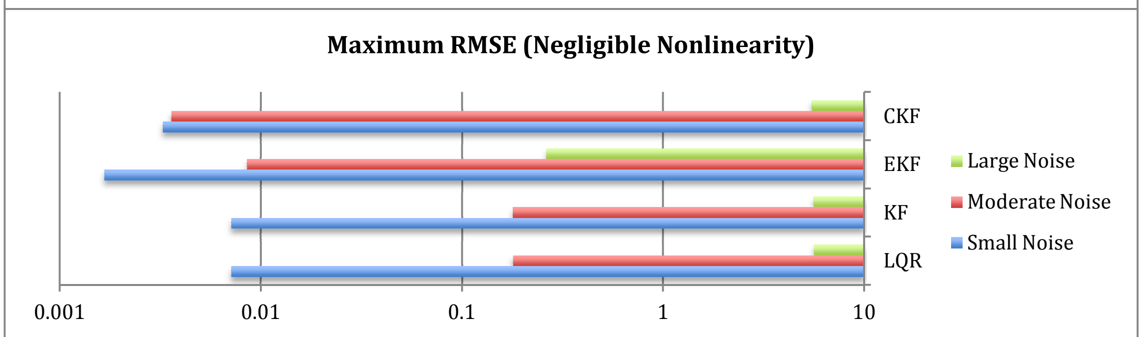

Median RMSE (Low Nonlinearity)
CKF
EKF
KF
LQR
Large Noise
Moderate Noise
Low Noise
0.001
0.01
0.1
1

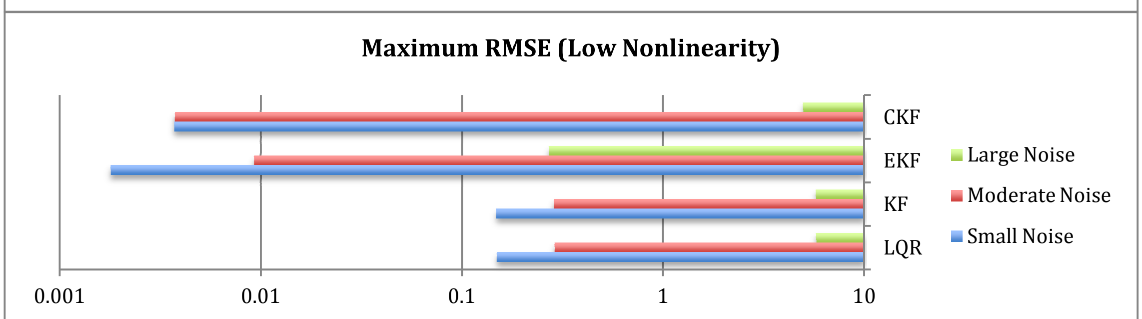

Median RMSE (Moderate Nonlinearity)
CKF
EKF
KF
LQR
Large Noise
Moderate Noise
Low Noise
0.001
0.01
0.1
1

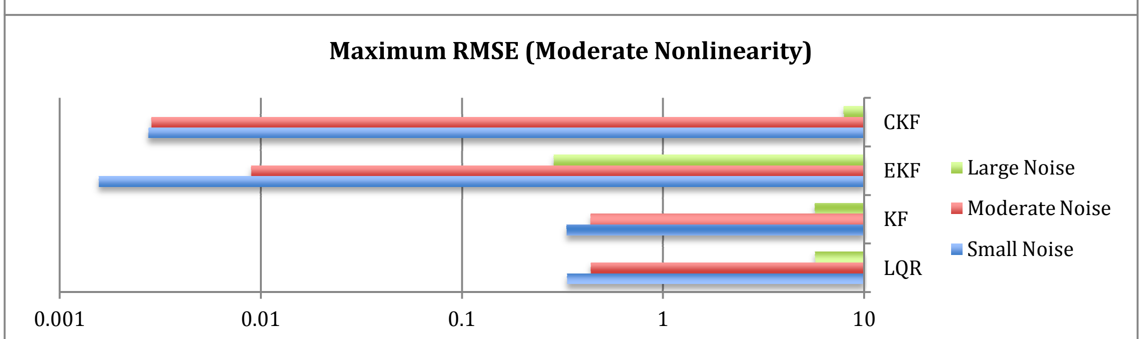

Median RMSE (Large Nonlinearity)
CKF
EKF
KF
LQR
Large Noise
Moderate Noise
Low Noise
0.001
0.01
0.1
1

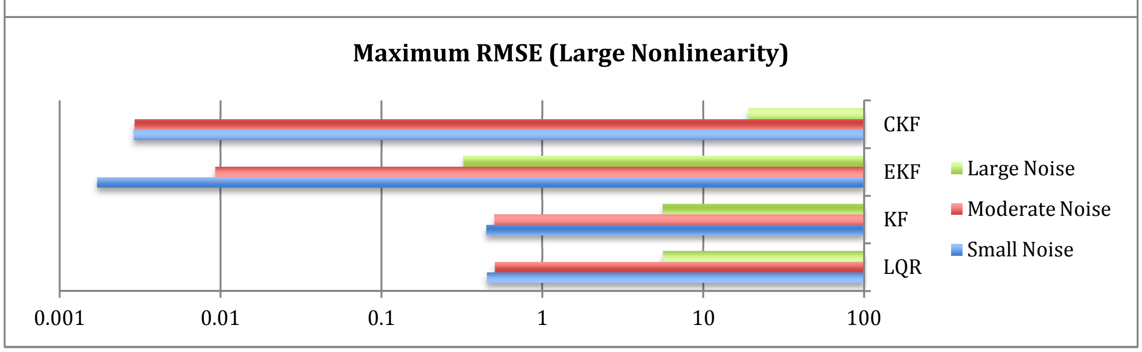
Maximum RMSE



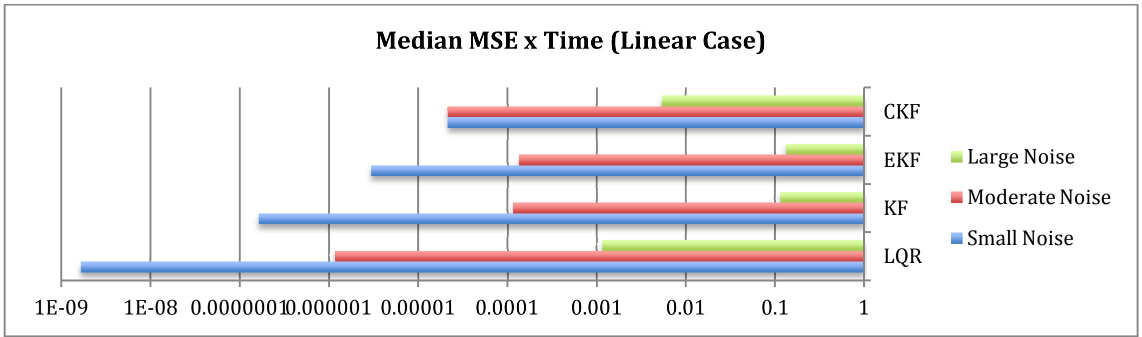


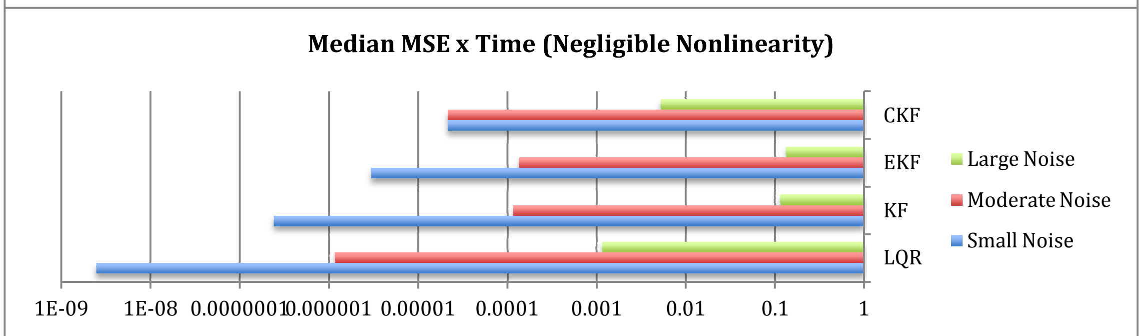


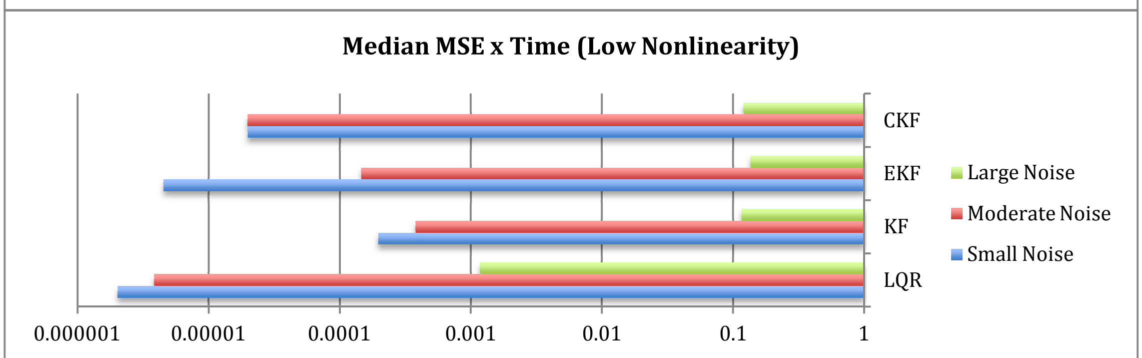


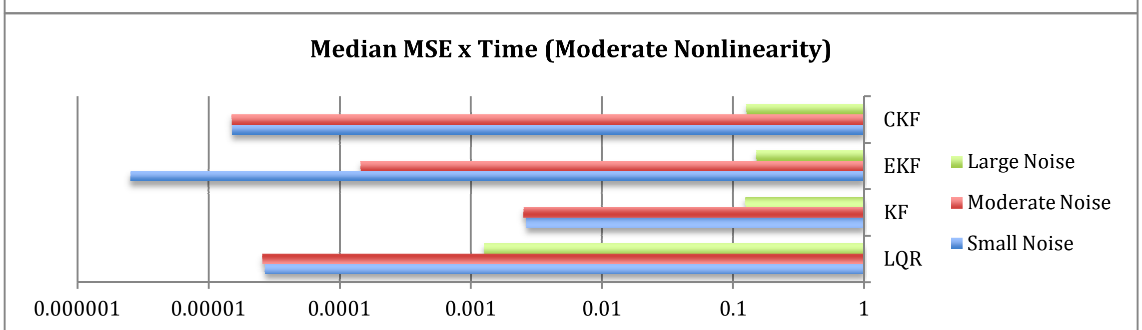


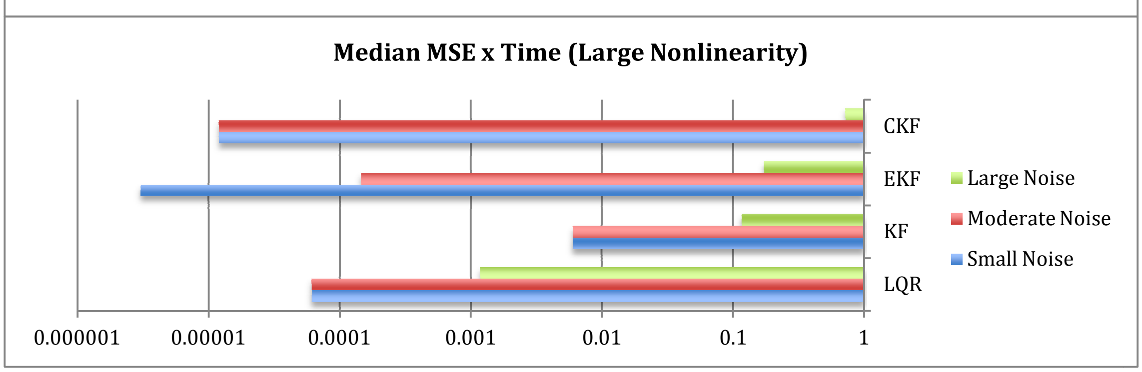
Combined Median RMSE and Time











Combined Maximum RMSE and Time