Bayesian Estimation
Classical and Particle Filtering Methods

N. Tapia Rivas

Department of Electrical Engineering
University of Chile

January, 2017

Outline

Outline

Bayesian Estimation

The main objective in Bayesian Estimation is to estimate the underlying probability distribution of a random signal X (state) given noisy uncertain measurement data Y, in order to perform statistical inferences such as the estimation of its current value.

$$\hat{P}(X|Y) \Rightarrow \hat{X}$$

Using Bayes' rule:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

- $P(Y|X)$ is the *likelihood*.
- $P(X)$ is the *prior* distribution.
- $P(Y)$ is the *evidence*.

So, we would like to estimate de *posterior* distribution combining the *prior* information and the measurements through the *likelihood*.

Bayesian Estimation

In Bayesian Signal Processing we are interested in a stochastic process $X_t = \{x(t)|t \in \mathbb{N}\}$. In these cases, generally we can assume a Markov Chain of first order in the states, and that the measurement depends only on the current state. So, we can have a Gauss-Markov state-space model of the process

$$x(t) = a[x(t-1)] + b[u(t-1)] + w(t-1)$$

and of the measurement

$$y(t) = c[x(t)] + v(t)$$

Where $w \sim \mathcal{N}(0, R_{ww}(t-1))$, $v \sim \mathcal{N}(0, R_{vv}(t-1))$ are the process and measurement noises, $u(t)$ is the input, and $a(\cdot), b(\cdot), c(\cdot)$ are some known functions.

Bayesian Estimation

- Analytic approach: Find analytic solutions for the posterior and perform statistic inferences based on integrals.

$$\mathbb{E}(f(x)) = \int f(x)P(x|y)dx$$

- Monte Carlo approach: Find analytic or approximate solutions for the posterior and perform statistic inferences based on sampling.

$$\mathbb{E}(f(x)) \approx \hat{f} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Outline

Kalman Filter (KF)

The Kalman Filter is a recursive analytic solution assuming linear model and gaussian noise. So, $a[x(t)] = A(t)x(t)$ and similar to the other functions. It is optimum in the sense of the MMSE. Due to linearity and gaussian noises, every distribution involved is also gaussian. So, the posterior has a simple closed formed and the estimation is performed finding the MAP.

Kalman Filter (KF)

Prediction

$$x(t|t-1) = A(t-1)x(t-1|t-1) + B(t-1)u(t-1)$$
$$P(t|t-1) = A(t-1)P(t-1|t-1)A'(t-1) + R_{ww}(t-1)$$

Innovation

$$e(t) = y(t) - y(t|t-1) = y(t) - C(t)x(t|t-1)$$
$$R_{ee}(t) = C(t)P(t|t-1)C'(t) + R_{vv}(t-1)$$

Update

$$K(t) = P(t|t-1)C'(t)R_{ee}^{-1}(t)$$
$$x(t|t) = x(t|t-1) + K(t)e(t)$$
$$P(t|t) = (I - K(t)C'(t))P(t|t-1)$$

Kalman Filter (KF)

Extended Version

We can extend this ideas to a nonlinear problem with gaussian noise and solving it approximately. This is achieved approximating the nonlinear functions

$$a[x(t-1)] \to \frac{da[x(t-1)]}{dx(t-1)}\Big|_{x=x(t|t-1)}$$

$$c[x(t)] \to \frac{dc[x(t)]}{dx(t)}\Big|_{x=x(t|t-1)}$$

While the prediction step for the state is made using the exact model, the subsequent steps are made in the same form as in the linear case, substituting the matrices $A(t)$ and $C(t)$ for its Jacobians.
The Extended Kalman Filter (EKF), while useful for approximate a nonlinear problem, implicitly assume that the posterior distribution is *unimodal*. So, for complex problems, highly nonlinear, it becomes insufficient.

Outline

Sampling

For the Monte Carlo approach we need to sample for a certain distribution. This can be achieved in several ways. Here we discuss Uniform Sampling and the Importance Sampling.

Sampling

Uniform Sampling

The CDF $P(X \leq x)$ of every distribution $p_x(x)$ is a *uniformly distributed random variable*, in the interval $(0, 1)$. This allows sampling from the CDF using uniform sampling as a intermediate step, and then performed the inverse CDF to find the corresponding x value. That wat, we get a sample from $p_x(x)$.

Sampling
Importance Sampling

To generate samples from $p_x(x)$, we can sample for an arbitrary proposal distribution $q_x(x)$, named *importance sampling distribution*, and then weight the samples accordingly.

$$\mathbb{E}_p\{f(x)\} = \int f(x)p(x)dx = \int f(x)\left(\frac{p(x)}{q(x)}\right)q(x)dx$$

So, we have an approximation of the distribution weighting the samples

$$\hat{p}(x) = \sum_{i=1}^{N} \mathcal{W}_i \delta(x - x_i)$$

Where $\mathcal{W}_i$ is a normalized weight proportional to $p(x_i)/q(x_i)$.

Sampling

Importance Sampling

In an iterative framework for obtaining the posterior, the weights can be updated using the recursion

$$W(t) = W(t-1) \cdot \frac{p(y(t)|x(t)) \cdot p(x(t)|x(t-1))}{q(x(t)|x(t-1), y(t))}$$

If the prior distribution is used as the importance distribution, then the recursion simplifies to

$$W(t) = W(t-1) \cdot p(y(t)|x(t))$$

Outline

Particle Filter (PF)

Concept

Particle Filters are Bayesian processors that estimate the posterior sequentially using Monte Carlo approach. That is, it uses N_p samples (particles) from the posterior, sampled using the importance sampling technique, and approximate the posterior with the set of particles and weights to perform statistics inferences. These particles are sequentially evolved through the process model to track the posterior in each time step.

$$\hat{p}(x(t)|y(t)) = \sum_{i=1}^{N} \mathcal{W}_i(t)\delta(x(t) - x_i(t))$$

Particle Filter (PF)
Degeneracy

As the weigths are updated during the filtering process, it is inevitable that their variance will increase and most of its values will be concentrated in a small number of particles. This is called the *degeneracy* problem, and it can be solved by performing a *resampling* step after the update.
This resampling can be performed whenever the degeneracy of the weights has achieved some threshold. For example, by evaluating

$$N_{eff}(t) = \frac{1}{\sum_{i=1}^{N_p} W_i^2(t)} \leq N_{thresh}$$

Where N_{eff} is the effective number of particles at time t.
The resampling replaces the old particles with new ones, obtained from the old ones using their weights as their probability of chosing them with replacement.

Particle Filter (PF)

Bootstrap PF

A very common filter is the Bootstrap PF, which uses the prior distribution as its importance sampling and performs resampling at every time step. So, the samples are drawn from the process model (prior), and the update is performed using the measurement model (likelihood).

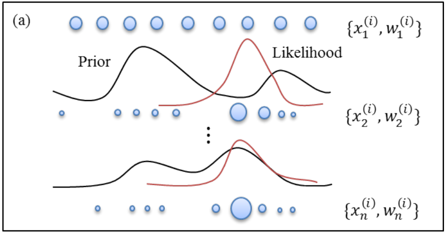

Figure 1: Bootstrap Particle Filter.

Outline

SOC in Li-Ion Batteries

In batteries, due to their several applications, it is of interest to perform some management in order to provide real-time information about its health and to make some decisions about the its future usage. To accomplish these goals, one of the main important variables is the SOC. The SOC (State of Charge) represents a percentage of the maximum amount of energy that can be stored in the battery. That is, $SOC = 1$ means a fully-charged battery, while $SOC = 0$ means a completely discharged one.

SOC in Li-Ion Batteries

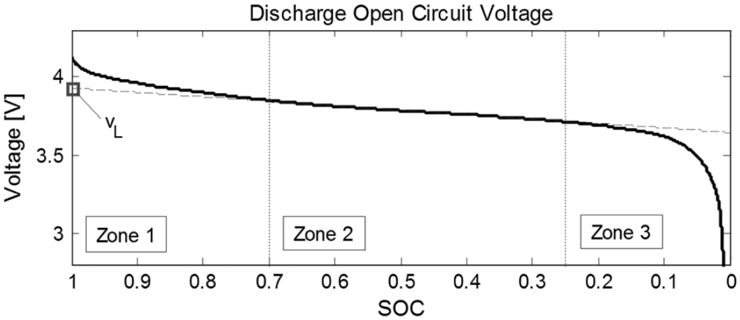

Figure 2: Li-Ion battery discharge open circuit voltage (OCV) as a function of SOC.

SOC in Li-Ion Batteries

We can model the voltage with a discrete-time empirical characterization given by the state transition and measurement models

$$x_1(t) = x_1(t-1) + w_1(t-1) \tag{1}$$

$$x_2(t) = x_2(t-1) - \frac{1}{E_{crit}} v(t-1)u(t-1)\Delta T + w_2(t-1) \tag{2}$$

$$y(t) = v(t) + \eta(t-1) \tag{3}$$

Where

$$\begin{aligned} v(t) = v_L + (v_0 - v_L)e^{\gamma(x_2(t)-1)} + \alpha v_L(x_2(t)-1) \\ + (1-\alpha)v_L(e^{-\beta} - e^{-\beta\sqrt{x_2(t)}}) - u(t)x_1(t) \end{aligned} \tag{4}$$

In this model, $v(t)$ represents the true voltage, $u(t)$ the current as an input, $x_1(t)$ the internal impedance, and $x_2(t)$ the SOC. The initial condition for $x_1(t)$ is the estimated absolute value for the impedance Z_p.

Outline

KF: Basic Linear Example

State-Space Model

Suppose we have a linear system whose discrete-time ($\Delta T = 100\text{ms}$) model is given by:

$$x(t) = 0.97x(t-1) + 100u(t-1) + w(t-1) \quad (5)$$
$$y(t) = 2x(t) + v(t) \quad (6)$$

Where $w \sim \mathcal{N}(0, R_{ww})$, $v \sim \mathcal{N}(0, R_{vv})$ are AWGN. The initial condition is $x(0) = 2.5$, the input is $u(t) = 300e-6$, and the measurement noise covariance is $R_{vv} = 4$.
We now study the filtering problem using the Kalman Filter and different initial conditions. For a first approximation, let the process noise covariance be $R_{ww} = 0$.

KF: Basic Linear Example

Simulated System without Process Noise

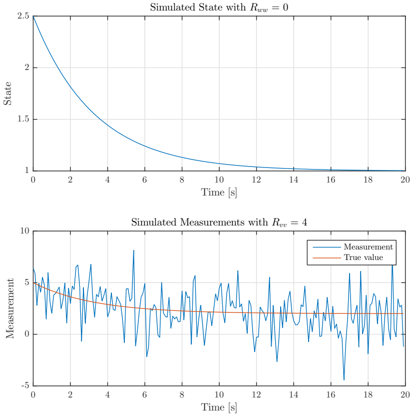

Figure 3: Simulated linear system without process noise.

The system was simulated in MATLAB for 20s. The results of the true state evolution and the measurements are shown in Fig. 3. The *true* measurement (mean) is also shown in red.

KF: Basic Linear Example

Filtering Results for different initial conditions

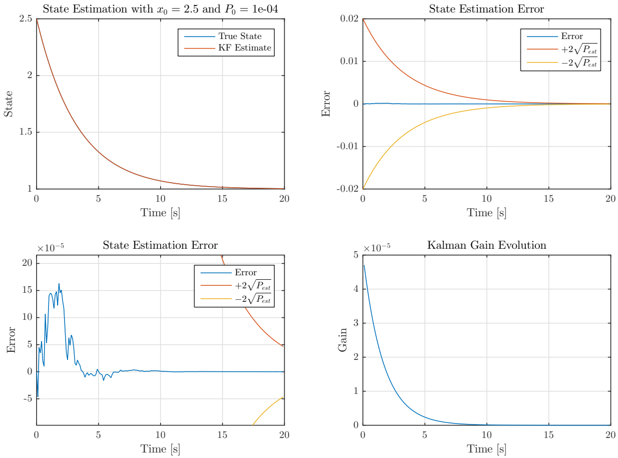

Figure 4: Filtering results with $\hat{x}(0) = 2.5$ and $\hat{P}(0) = 1e-4$.

KF: Basic Linear Example

Filtering Results for different initial conditions

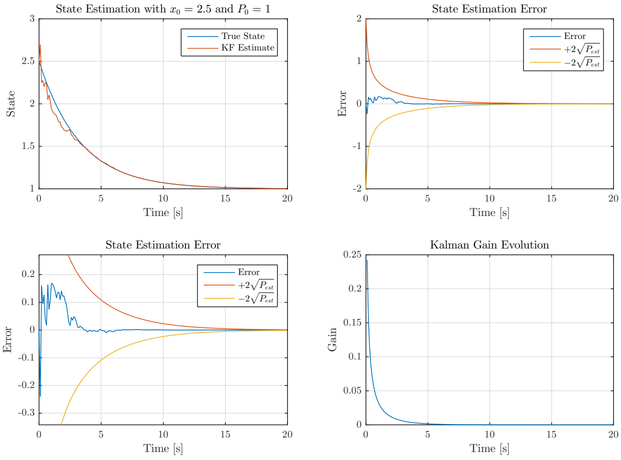

Figure 5: Filtering results with $\hat{x}(0) = 2.5$ and $\hat{P}(0) = 1$.

KF: Basic Linear Example

Filtering Results for different initial conditions

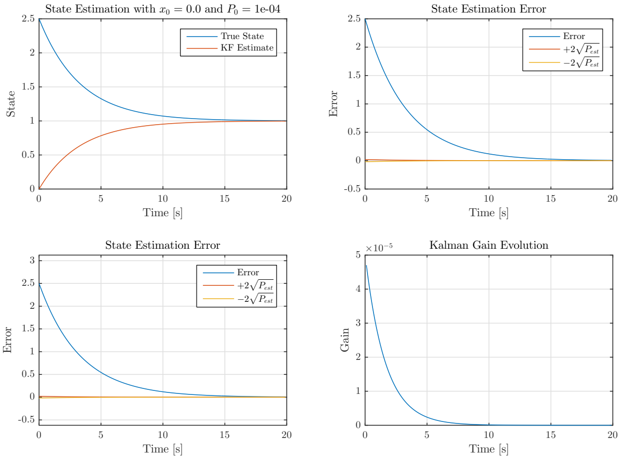

Figure 6: Filtering results with $\hat{x}(0) = 0$ and $\hat{P}(0) = 1e-4$.

KF: Basic Linear Example

Filtering Results for different initial conditions

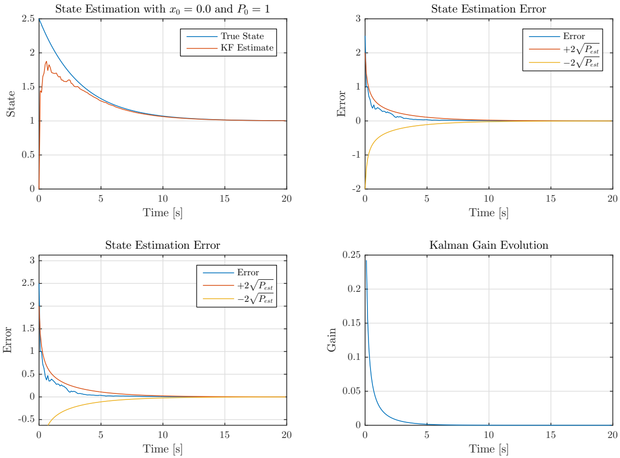

Figure 7: Filtering results with $\hat{x}(0) = 0$ and $\hat{P}(0) = 1$.

KF: Basic Linear Example

Simulated System with Process Noise

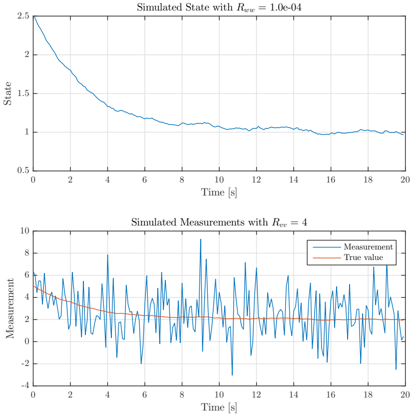

Figure 8: Simulated linear system with process noise.

Now $R_{ww} = 1e-4$, i.e. there is some process noise. The system again was simulated in MATLAB for 20s. The results of the true state evolution and the measurements are shown in Fig. 8. The *true* measurement (mean) is also shown in red.

KF: Basic Linear Example

Filtering Results for different initial conditions

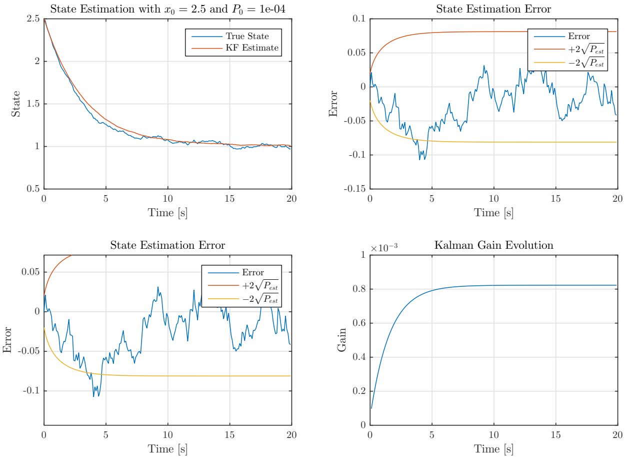

Figure 9: Filtering results with $\hat{x}(0) = 2.5$ and $\hat{P}(0) = 1e - 4$.

KF: Basic Linear Example

Filtering Results for different initial conditions

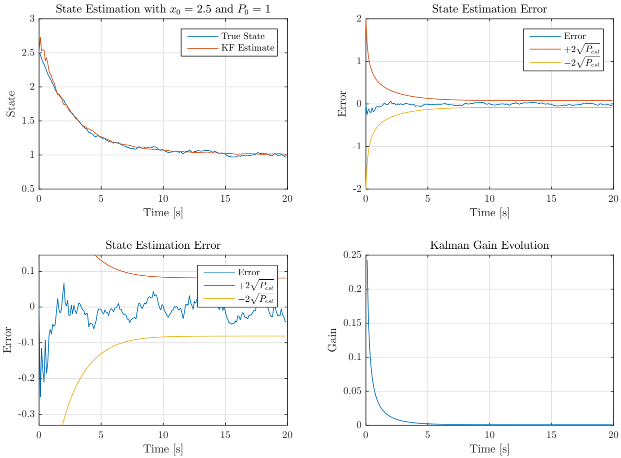

Figure 10: Filtering results with $\hat{x}(0) = 2.5$ and $\hat{P}(0) = 1$.

KF: Basic Linear Example

Filtering Results for different initial conditions

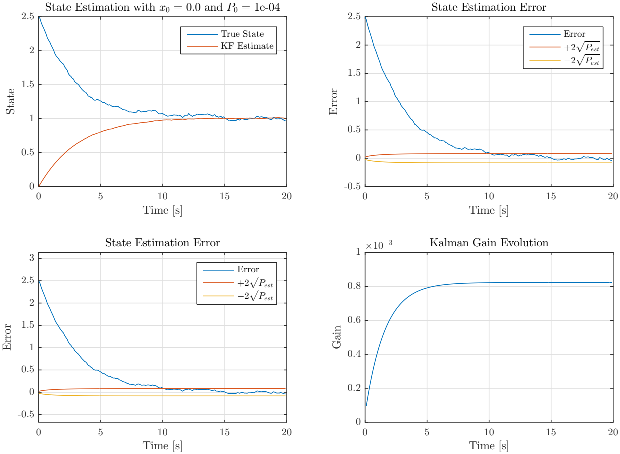

Figure 11: Filtering results with $\hat{x}(0) = 0$ and $\hat{P}(0) = 1e - 4$.

KF: Basic Linear Example

Filtering Results for different initial conditions

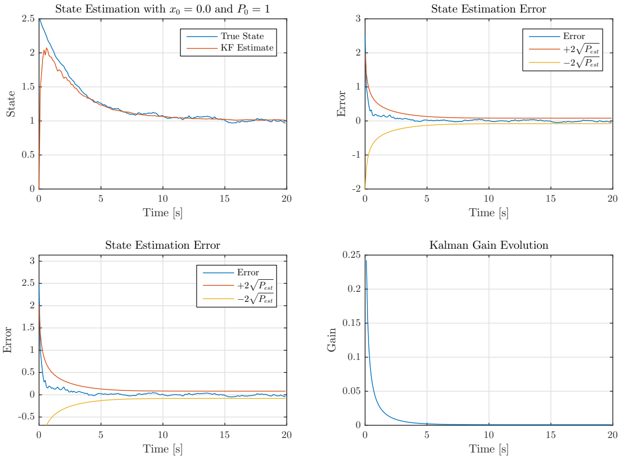

Figure 12: Filtering results with $\hat{x}(0) = 0$ and $\hat{P}(0) = 1$.

KF: Basic Linear Example

Observations

- For different initial conditions, the KF converges to the true state with steady-state covariance and Kalman Gain values.
- Accuracy in $\hat{x}(0)$ should be reflected in the $\hat{P}(0)$ value for better convergence rate.
  - Accurate $\hat{x}(0)$ needs low $\hat{P}(0)$ value to avoid losing track at the beginning due to measurement noise.
  - Inaccurate $\hat{x}(0)$ needs high $\hat{P}(0)$ value, otherwise the correction during the update process will be slow.
- The steady-state value of $\hat{P}$, and so the estimation precision, is highly dependent on the process noise, which sets a lower bound. More process noise becomes the system more difficult to predict.
- Once the steady-state is achieved, the estimation error is well described by the $\hat{P}$ value, whether there is process noise or not.

Outline

EKF: Basic Nonlinear Example

State-Space Model

Suppose we have a nonlinear system whose discrete-time ($\Delta T = 0.01$s) model is given by:

$$x(t) = (1 - 0.05\Delta T)x(t-1) + 0.04\Delta T x^2(t-1) + w(t-1) \quad (7)$$

$$y(t) = x^2(t) + x^3(t) + v(t) \quad (8)$$

Where $w \sim \mathcal{N}(0, R_{ww})$, $v \sim \mathcal{N}(0, R_{vv})$ are AWGN. The initial condition is $x(0) = 2$ and the noise covariances are $R_{vv} = 0.09$ and $R_{ww} = 0$. We now study the filtering problem using the Extended Kalman Filter and the Bootstrap Particle Filter.

EKF: Basic Nonlinear Example

Simulated System

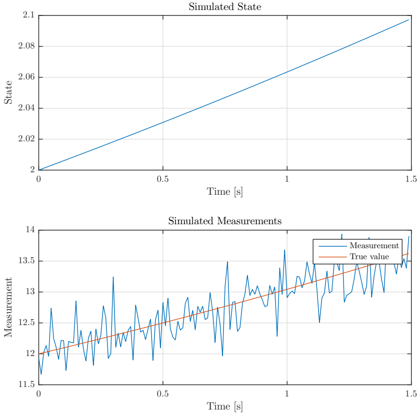

Figure 13: Simulated nonlinear system.

The system was simulated in MATLAB for 1.5s. The results of the true state evolution and the measurements are shown in Fig. 13. The *true* measurement (mean) is also shown in red.

EKF: Basic Nonlinear Example

Filtering with EKF

We address the filtering problem with the EKF. For doing so, we need the Jacobians of the process and measurement models. They are given by:

$$A[x(t-1)] = 1 - 0.05\Delta T + 0.08\Delta T x(t-1) \tag{9}$$

$$C[x(t)] = 2x(t) + 3x^2(t) \tag{10}$$

With these jacobians, let's see how the performance is for a given initial condition.

EKF: Basic Nonlinear Example

Filtering Results for the EKF

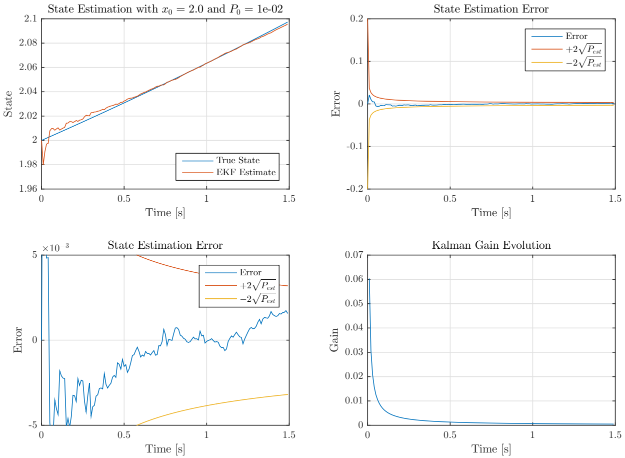

Figure 14: Filtering results with $\hat{x}(0) = 2$ and $\hat{P}(0) = 1e - 2$.

EKF: Basic Nonlinear Example

Filtering Results for the EKF

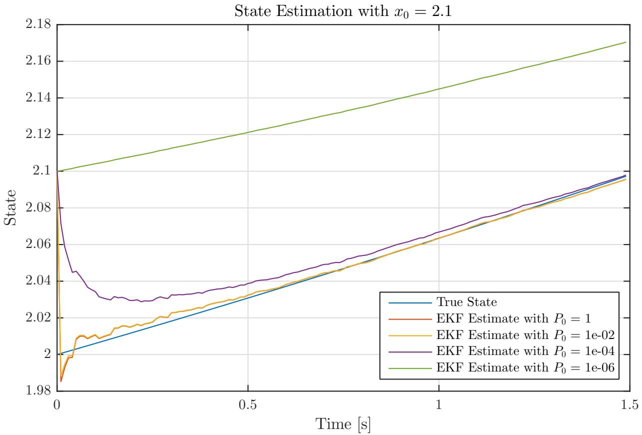

Figure 15: Filtering results with $\hat{x}(0) = 2.1$ and different values for $\hat{P}(0)$.

EKF: Basic Nonlinear Example
Observations

- For this basic example, the EKF followed well the state, although more noisy than in the KF example. The covariance was capable of describing correctly the confidence in the estimate.
- As expected, when giving an incorrect initial condition, it is important not to be overconfident, i.e. we should maintain a relatively high value of P_0. Otherwise, the state may not even converges.

Outline

EKF and BPF: Highly Nonlinear Example

State-Space Model

Now, suppose we have a well-known nonlinear system whose discrete-time ($\Delta T = 1$s) model is given by:

$$x(t) = \frac{1}{2}x(t-1) + \frac{25x(t-1)}{1+x^2(t-1)} + 8\cos(1.2(t-1)) + w(t-1) \quad (11)$$

$$y(t) = \frac{1}{20}x^2 + v(t) \quad (12)$$

Where $w \sim \mathcal{N}(0, R_{ww})$, $v \sim \mathcal{N}(0, R_{vv})$ are AWGN. The initial condition is $x(0) \sim \mathcal{N}(0.1,\ 5)$ and the noise covariances are $R_{vv} = 1$ and $R_{ww} = 10$.

This problem has become a *benchmark* for many filtering algorithms. It is highly nonlinear and nonstationary. We now study the filtering problem using the Extended Kalman Filter and the Bootstrap Particle Filter.

EKF and BPF: Highly Nonlinear Example

Simulated System

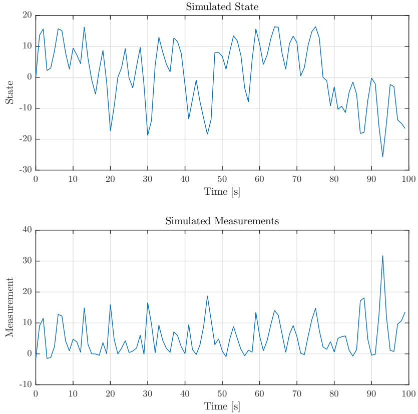

Figure 16: Simulated highly nonlinear system.

The system was simulated in MATLAB for 100s. The results of the true state evolution and the measurements are shown in Fig. 16. The *true* measurement (mean) is also shown in red.

EKF and BPF: Highly Nonlinear Example

Filtering with EKF

First, we address the filtering problem with the EKF. For doing so, we need the Jacobians of the process and measurement models. They are given by:

$$A[x(t-1)] = \frac{1}{2} + \frac{25(1 - x^2(t-1))}{(1 + x^2(t-1))} \tag{13}$$

$$C[x(t)] = \frac{1}{10}x(t) \tag{14}$$

With these jacobians, let's see how the performance is for a given initial condition.

EKF and BPF: Highly Nonlinear Example

Filtering Results for the EKF

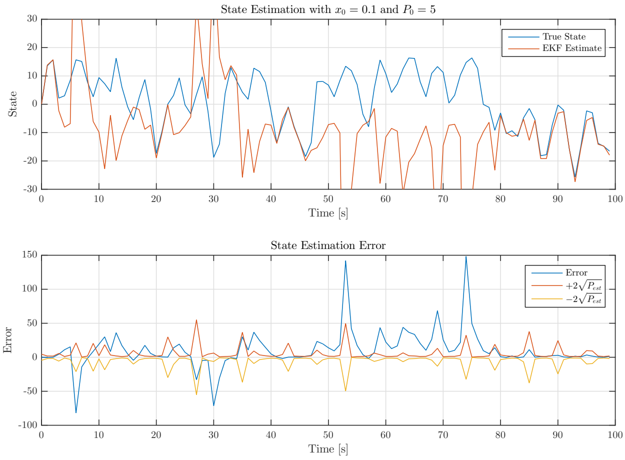

Figure 17: Filtering results for the EKF.

EKF and BPF: Highly Nonlinear Example

Filtering with BPF

Now, we address the filtering problem with the BPF. For doing so, we consider 100 particles initialized with $x(0) \sim \mathcal{N}(0.1,\ 5)$. The state estimate is obtained using the conditional mean of the posterior (MMSE Estimator).

EKF and BPF: Highly Nonlinear Example

Filtering Results for the BPF

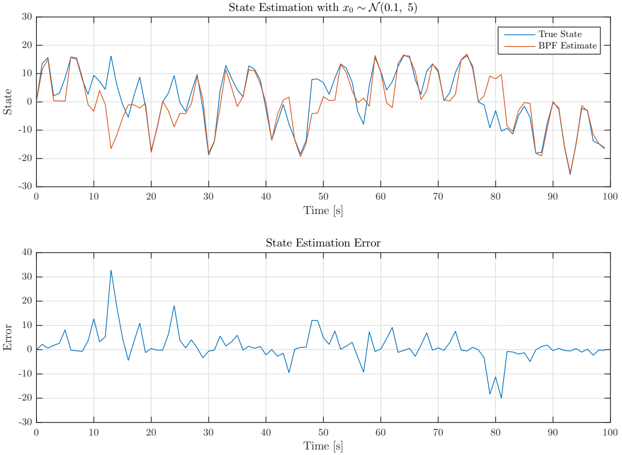

Figure 18: Filtering results for the BPF.

EKF and BPF: Highly Nonlinear Example

Filtering Results Comparison

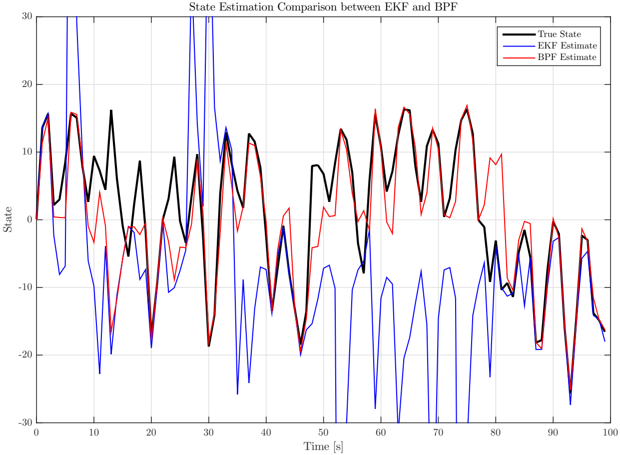

Figure 19: Filtering results comparison between EFK and BPF.

EKF and BPF: Highly Nonlinear Example

Observations

- Due to randomness in the simulation process, it is better to assess performance with several instances. Running 30 simulations, the RMSE obtained for both the EKF and BPF are:

Table 1: RMSE for the two algorithms.

RMSE	Mean	STD
EKF	52.5511	24.6586
BPF	4.7859	1.0199

- As the system becomes more complex, the EKF becomes insufficient for solving the state, because of its Gaussian assumptions and model linearization. In this case, the BPF outperformed the EKF.
- The $\hat{P}$ value of the EKF is inaccurate in this problem. There are several times in which the error is highly underestimate, causing overconfidence when making a decision.

Outline

General Considerations

In this section we will replicate the SOC estimation problem solved in Pola, 2015. For doing so, we extracted the data in the paper for Battery #1, a 18650 cell (3.7 volts, 2.4 ampere-hours). We use $\Delta T = 1$s.

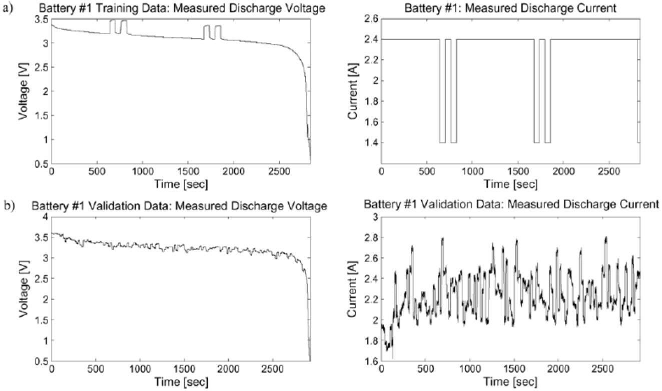

Figure 20: Real data from the paper of Pola, 2015.

General Considerations

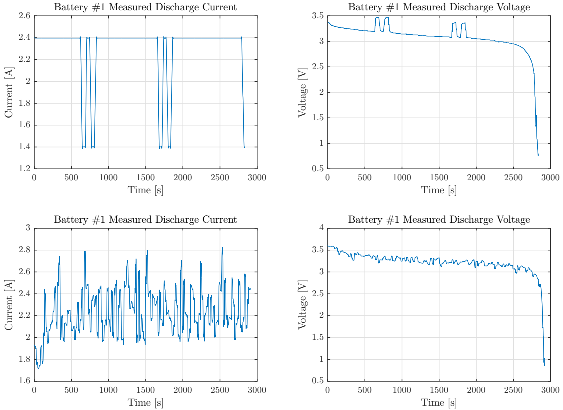

Figure 21: Extracted data from the paper of Pola, 2015.

General Considerations

For the first experiment, we simulate a model with known parameters given by the ones found for Battery #2, using as an input the validation current data extracted previously.
For the second experiment, we use the training data to estimate the parameters and then perform estimation using the validation data.

Table 2: Parameters of the batteries found in the paper of Pola, 2015.

Battery	α	β	γ	v_0	v_L	E_{crit}	Z_p
1	0.08	16	19.65	4.12	3.987	20127	0.30
2	0.15	12	6.61	4.00	3.813	19865	0.20

Outline

Simulated Data and Known Model Parameters

Simulated Battery

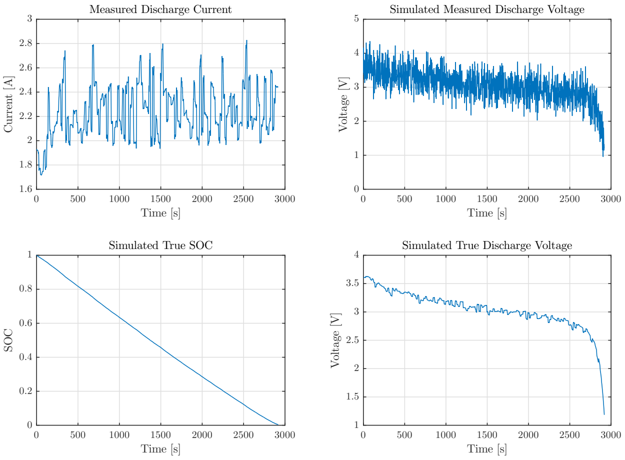

Figure 22: Simulated battery with known parameters.

Simulated Data and Known Model Parameters
Filtering with EKF

First, we address the filtering problem with the EKF. For doing so, we need the Jacobians of the process and measurement models. They are given by:

$$A[x(t-1)] = \begin{pmatrix} 1 & 0 \\ \frac{-i^2(t-1)\Delta T}{E_{crit}} & \frac{i(t-1)\Delta T}{E_{crit}}\lambda(t-1) \end{pmatrix} \tag{15}$$

$$C[x(t)] = \begin{pmatrix} -i(t) & \lambda(t) \end{pmatrix} \tag{16}$$

$$\lambda(t) = \gamma(v_0 - v_L)e^{\gamma(x_2(t)-1)} + \alpha v_L + \frac{\beta(1-\alpha)v_L}{2\sqrt{x_2(t)}}e^{-\beta\sqrt{x_2(t)}} \tag{17}$$

We also set the initial condition for the SOC to be $x_2(0) = 0.8$ even when we know it is 1. Also, we set

$$P_0 = \begin{pmatrix} 1e-4 & 0 \\ 0 & 1e-1 \end{pmatrix}$$

Simulated Data and Known Model Parameters

Filtering Results for the EKF

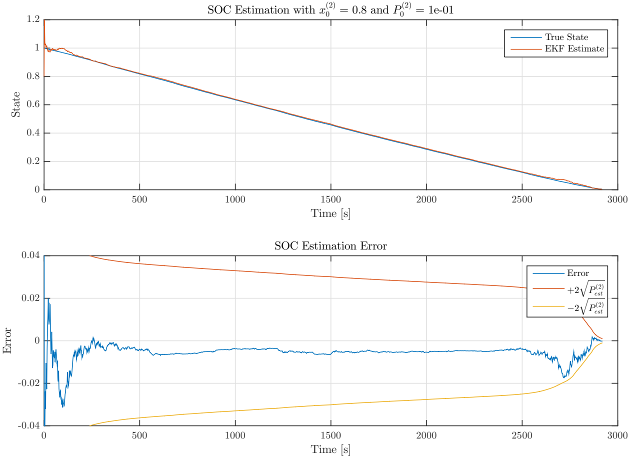

Figure 23: Filtering results for the EKF.

Simulated Data and Known Model Parameters

Filtering with BPF

Now, we address the filtering problem with the BPF. For doing so, we consider 100 particles initialized with $x(0) \sim \mathcal{U}(0.8, 0.9)$, even when we know it is 1. The state estimate is obtained using the conditional mean of the posterior (MMSE Estimator). The process noise, specifically the one associated with the SOC, and the measurement were slightly augmented from $1e-8$ and $1e-4$ to $1e-5$ and $1e-2$ respectively.

Simulated Data and Known Model Parameters

Filtering Results for the BPF

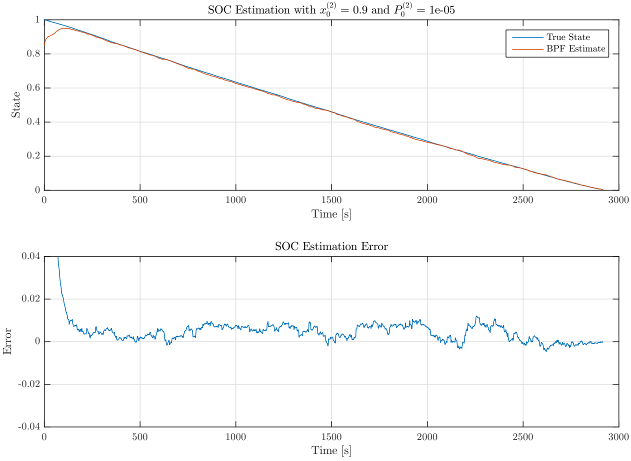

Figure 24: Filtering results for the BPF.

Outline

Experimental Data and Unknown Model Parameters
Training Data

Now, we estimate the parameters of the battery through the methodology described in the paper of Pola, 2015, using the training data set.

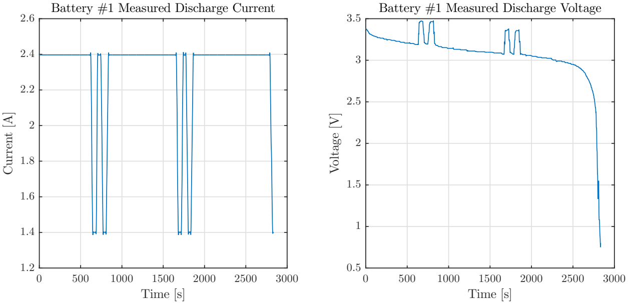

Figure 25: Training data set.

Experimental Data and Unknown Model Parameters

Training Data

Given the low voltage value achieved, E_{crit} is estimated just by the total energy delivered $E_{crit} = \sum v(k)i(k)\Delta t$. Using the amplitude of the pulses, Z_p es estimated as $Z_p = |\Delta V/\Delta I|$.

$$Z_p = \text{mean}\left\{\left|\frac{3.196 - 3.464}{2.399 - 1.401}\right|, \left|\frac{3.077 - 3.356}{2.395 - 1.404}\right|\right\}$$

Using the fact that, at the beginning of this data set, $SOC = 1$, we can calculate an OCV vs SOC curve, separated by its zones, by using

$$v_{OCV}(t) = v(t) + i(t) \cdot Z_p$$

$$SOC(t) = SOC(t-1) - \frac{1}{E_{crit}} v(t)i(t)\Delta T$$

Experimental Data and Unknown Model Parameters

Training Data

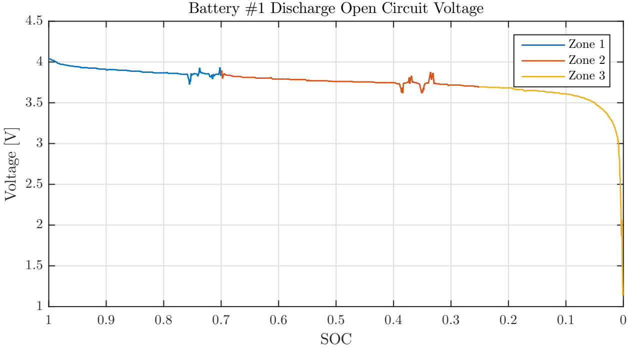

Figure 26: OCV vs SOC curve obtained in the training data.

Experimental Data and Unknown Model Parameters

Training Data

We can perform a linear fit in the zone 2 to obtain v_L and α according to

$$v_{OCV}^{(z1)}(SOC) \approx v_L + \alpha v_L (SOC - 1)$$

We can estimate v_0 as $v_{OCV}(0)$. The remaining parameters, β and γ, are estimated minimizing the squared error in the zone 1 and 3 respectively.

Experimental Data and Unknown Model Parameters

Training Data

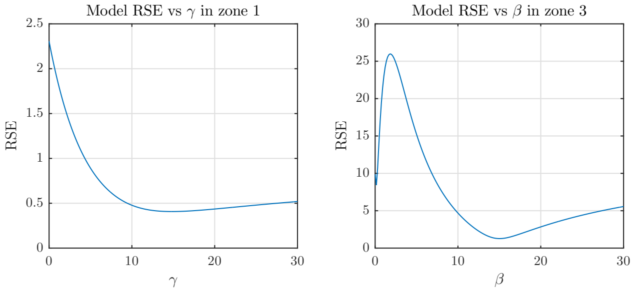

Figure 27: Root Squared Error as a function of γ and β in each zone.

Table 3: Parameters of the battery # 1 found here in contrast with the ones found in the paper of Pola, 2015.

Battery	α	β	γ	v_0	v_L	E_{crit}	Z_p
1 (Here)	0.067	15.026	14.825	4.044	3.901	20118.627	0.275
1 (Pola)	0.08	16	19.65	4.12	3.987	20127	0.30

Experimental Data and Unknown Model Parameters

Training Data

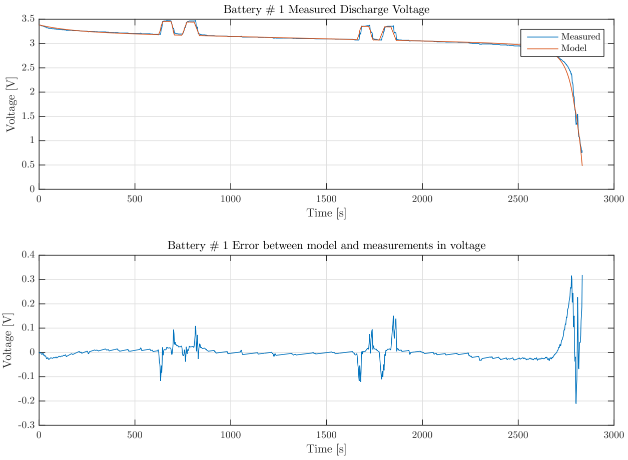

Figure 28: Training data set and its found model.

Experimental Data and Unknown Model Parameters
Validation Data

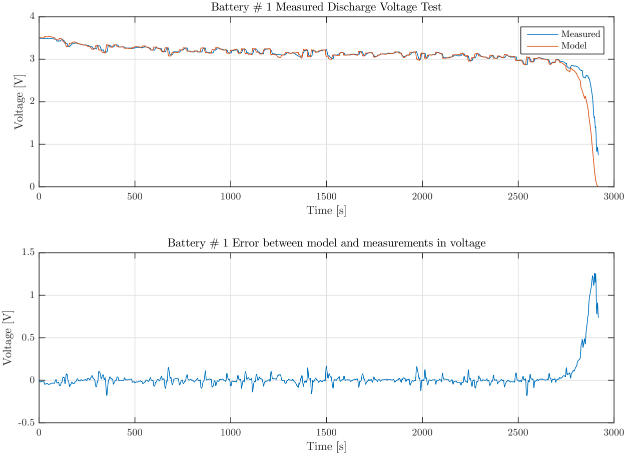

Figure 29: Validation data set and the found model.

Experimental Data and Unknown Model Parameters

Filtering with EKF

First, we address the filtering problem with the EKF. For doing so, we need the Jacobians of the process and measurement models. They are given by the same equations as before. We also set the initial condition for the SOC to be $x_2(0) = 0.8$ even when we know it is 1. Again, we set

$$P_0 = \begin{pmatrix} 1e-4 & 0 \\ 0 & 1e-1 \end{pmatrix}$$

Also, we set the process noise to be $w_1, w_2 \sim \mathcal{N}(0, 1e-8)$ and $\eta \sim \mathcal{N}(0, 1e-2)$.

Experimental Data and Unknown Model Parameters

Filtering Results for the EKF

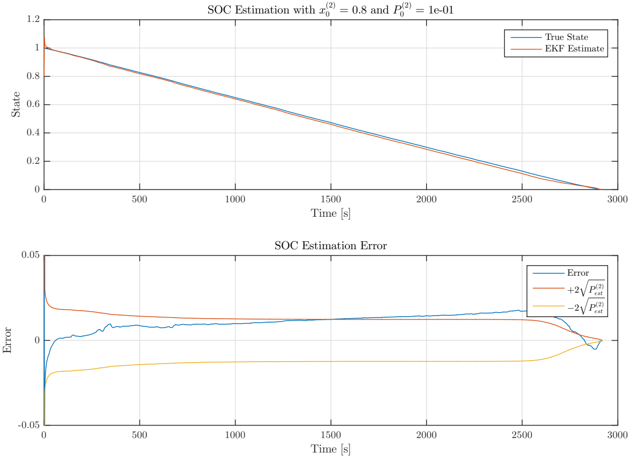

Figure 30: Filtering results for the EKF.

Experimental Data and Unknown Model Parameters

Filtering with BPF

Now, we address the filtering problem with the BPF. For doing so, we consider 100 particles initialized with $x(0) \sim \mathcal{U}(0.8, 0.9)$, even when we know it is 1. The state estimate is obtained using the conditional mean of the posterior (MMSE Estimator). We also augmented the SOC process noise to $1e-5$ as before.

Experimental Data and Unknown Model Parameters

Filtering Results for the BPF

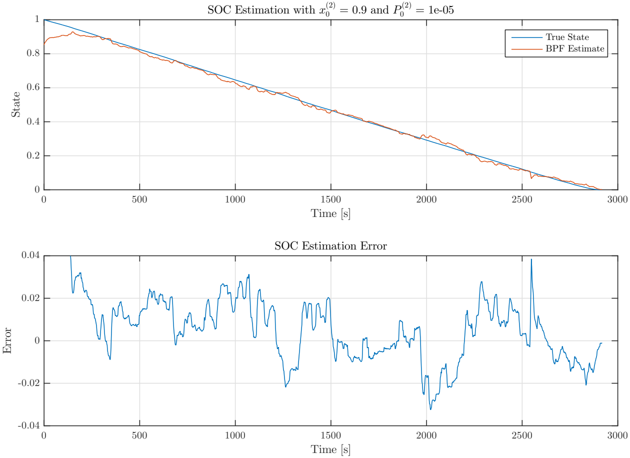

Figure 31: Filtering results for the BPF.

Experimental Data and Unknown Model Parameters

Filtering Results for the BPF

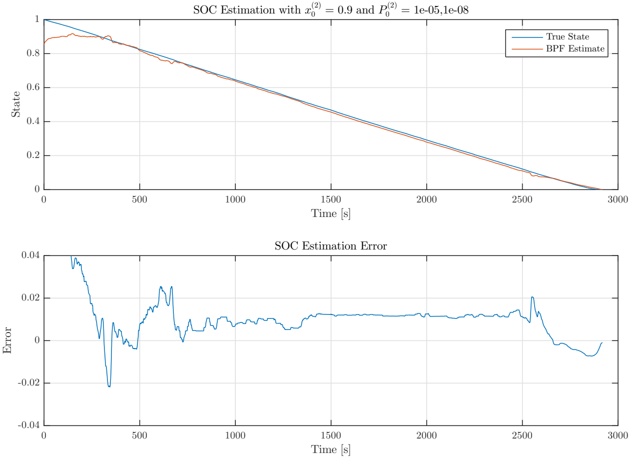

Figure 32: Filtering results for the BPF with a changed in the SOC process noise at 350s from $1e-5$ to $1e-8$.

Summary

- The EKF demonstrated to be a good choice when the hypothesis are hold. Its implementation is simple and can quickly correct bad initial conditions.
- Facing more complex problems, the EKF fails not only in the state estimation but also in the confidence interval.
- In these cases, it is better to try with a Particle Filter, which has almost no hypothesis. It is necessary just to be capable of sampling the importance distribution, and making point evaluation of the distributions of the system.

Bibliography I

J. Candy
Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods
John Wiley & Sons Inc., 2009

D. Pola *et al.*
Particle-Filtering-Based Discharge With a Statistical Characterization of Use Profiles
IEEE Transactions on Reliability, 64(2):710–720, 2015

D. Acuña and M. Orchard
Particle-filtering-based failure prognosis via sigma-points: Application to Lithium-Ion battery State-of-Charge monitoring
Mechanical Systems and Signal Processing, 85:827–848, 2017.

M. Arulampalam *et al.*
A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking
IEEE Transactions on Signal Processing, 50(2):174–188, 2002.

Bibliography II

A. Tulsyan *et al.*
Particle filtering without tears: A primer for beginners
Computers and Chemical Engineering, 95:130–145, 2016.