# 16-899 Assignment 2

Chi-Chian Wu
chichiaw@andrew.cmu.edu
October 16, 2020

## 1 Part I - System

### 1.1

$$
\begin{aligned}
A &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \\
x_{k+1} &= Ax_k + Bu_k \\
x_{k+1} - r_{k+1} &= A(x_k - r_k) + Bu_k - Ar_k + Ar_k - r_{k+1} \\
x_{k+2} - r_{k+2} &= A(x_{k+1} - r_{k+1}) + Bu_{k+1} - Ar_{k+1} + Ar_{k+1} - r_{k+2} \\
&= A^2(x_k - r_k) + ABu_k + A^2 r_k + Bu_{k+1} - r_{k+1}
\end{aligned}
$$

$$
X_k = \begin{bmatrix} x_k - r_k \\ x_{k+1} - r_{k+1} \\ \vdots \\ x_{k+N} - r_{k+N} \end{bmatrix}, \quad U_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{bmatrix}
$$

$$
X_k = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} (x_k - r_k) + \begin{bmatrix} 0 & 0 & 0 & ...\ 0 \\ B & 0 & 0 & ...\ 0 \\ AB & B & 0 & ...\ 0 \\ \vdots & \vdots & \vdots & ...\ 0 \\ A^{N-1}B & A^{N-2}B & ... & ...\ B \end{bmatrix} U_k - \begin{bmatrix} 0 \\ r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+N} \end{bmatrix} + \begin{bmatrix} 0 \\ Ar_k \\ Ar_k \\ A^2 r_k \\ \vdots \\ A^N r_k \end{bmatrix}
$$

$$
X_k = \bar{f} + \bar{B}U_k - \bar{r} + \bar{C}
$$

$$
\begin{aligned}
J_{k:k+N} &= \frac{1}{2} X_k^T diag(Q, Q, ...Q, S) X_k + \frac{1}{2} U_k^T diag(R, R, ..., R) U_k \\
&= \frac{1}{2} X_k^T \bar{Q} X_k + \frac{1}{2} U_k^T \bar{R} U_k \\
&= \frac{1}{2} (\bar{f} + \bar{B}U_k - \bar{r} + \bar{C})^T \bar{Q} (\bar{f} + \bar{B}U_k - \bar{r} + \bar{C}) + \frac{1}{2} U_k^T \bar{R} U_k \\
&= \frac{1}{2} U_k^T [\bar{B}^T \bar{Q} \bar{B} + \bar{R}] U_k + [\bar{f}^T \bar{Q} \bar{B} - \bar{r}^T \bar{Q} \bar{B} + \bar{C}^T \bar{Q} \bar{B}] U_k + (\frac{1}{2} \bar{f}^T \bar{Q} \bar{f} + \text{...terms not a function of U})
\end{aligned}
$$

$$
\begin{bmatrix} \bar{L_{u1}} \\ \bar{L_{u2}} \\ \bar{L_{x1}} \\ \bar{L_{x2}} \end{bmatrix} Uk \leq \begin{bmatrix} \bar{u} \\ \bar{u} \\ \bar{x} - \bar{L_{x_1}} \bar{f} + \bar{L_{x_1}} \bar{r} - \bar{L_{x_1}} \bar{C} \\ \bar{x} - \bar{L_{x_2}} \bar{f} + \bar{L_{x_2}} \bar{r} - \bar{L_{x_2}} \bar{C} \end{bmatrix}, \quad \bar{L_{u1}} = I, \quad \bar{L_{u2}} = -I, \quad \bar{L_{x1}} = I, \quad \bar{L_{x2}} = -I
$$

### 1.2

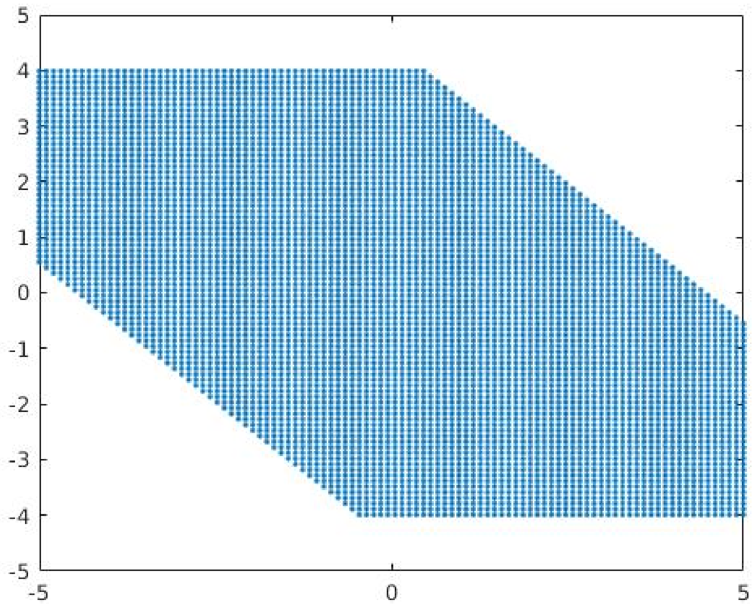



Figure 1: Feasible state

### 1.3

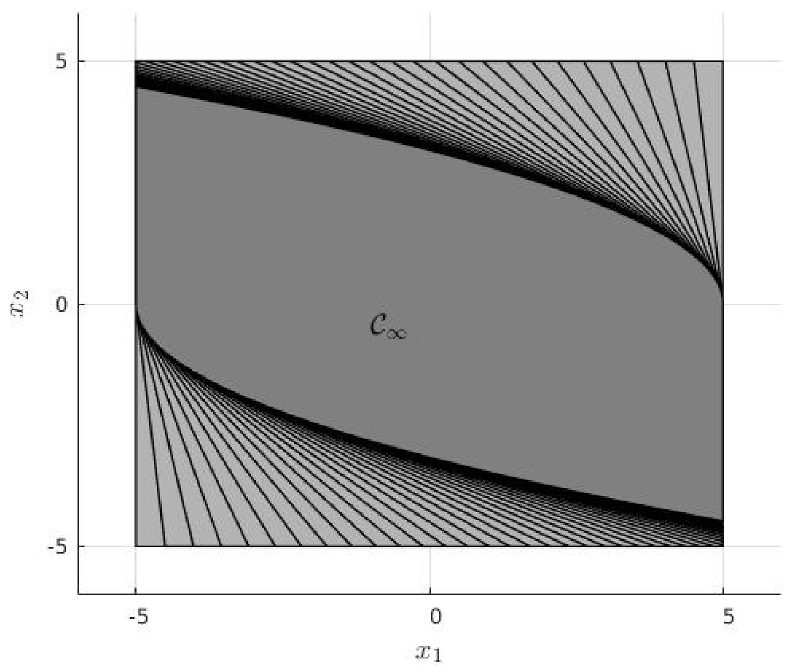
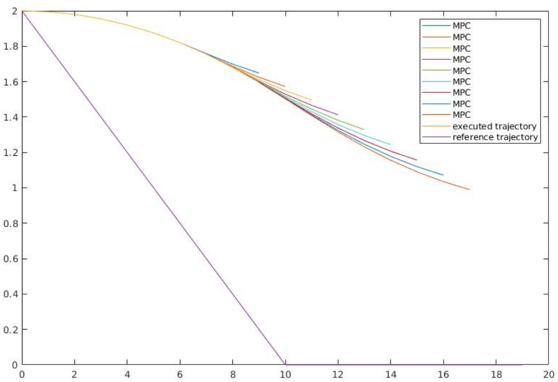
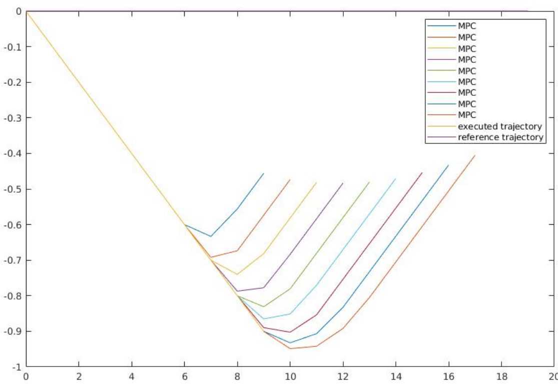



Figure 2: Feasible state

### 1.4




(a) state1

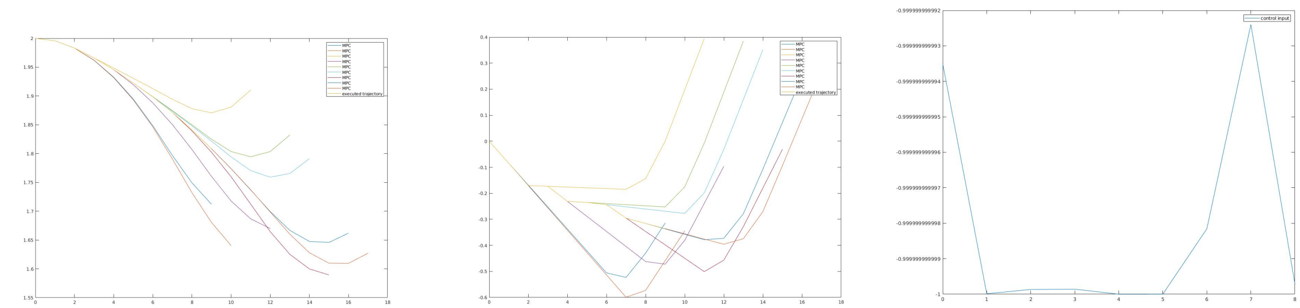



(b) state2

Figure 3: MPC discrete-time system

# 2 Part II - Iterative Learning Control

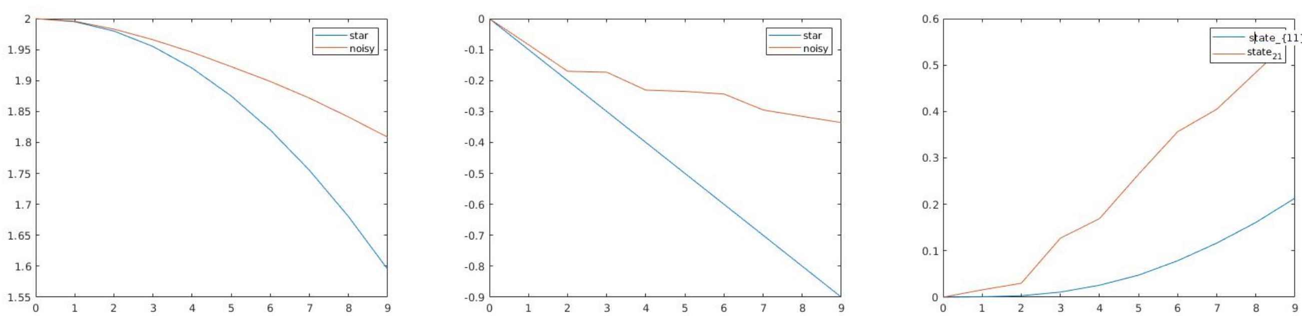
## 2.1



(a) state1 (b) state2 (c) control input

Figure 4: MPC with control noises

## 2.2




(a) state1 (b) state2 (c) errors

Figure 5: Error between state* and noisy state

### 2.3

In this problem, we are using quadratic programming to obtain the control sequence. Therefore, it is very hard to write down the transfer function of the controller. The closed-loop solution for L is not applicable at this scenario. The approach I used was that at each step I tried to find a L that would minimize a cost function $J = x^*_{k+1} - (Ax_k + B(u_k^{MPC} + LE_k))$. This optimization utilized the MATLAB function fminsearch to find the L sequence.

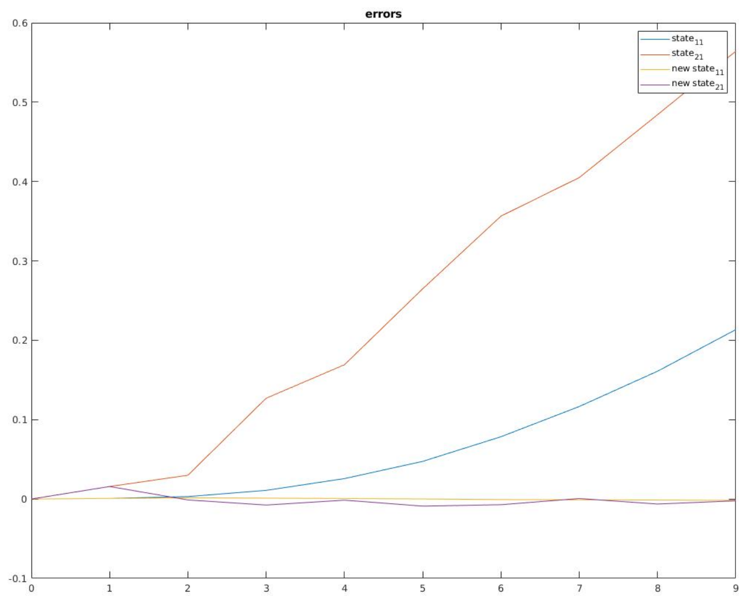
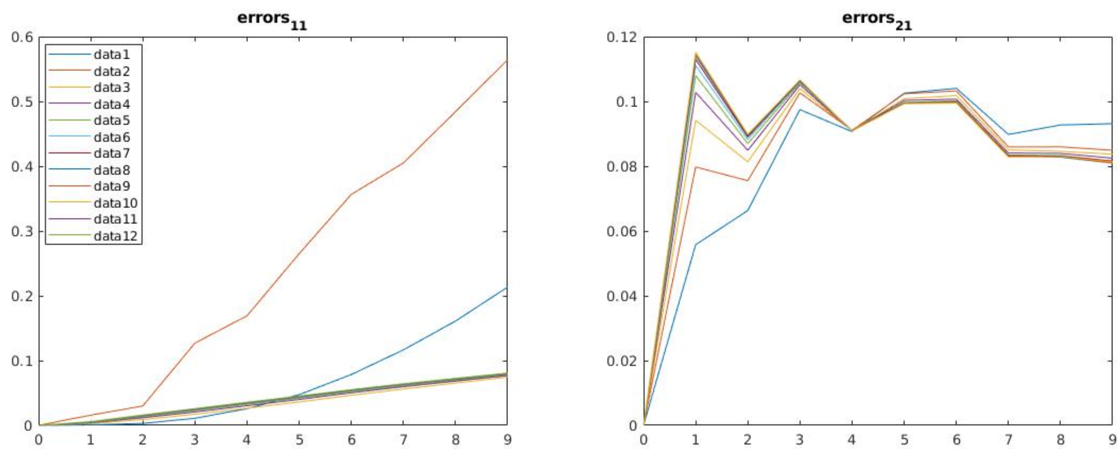



Figure 6: Iterative learning control first iteration

### 2.4

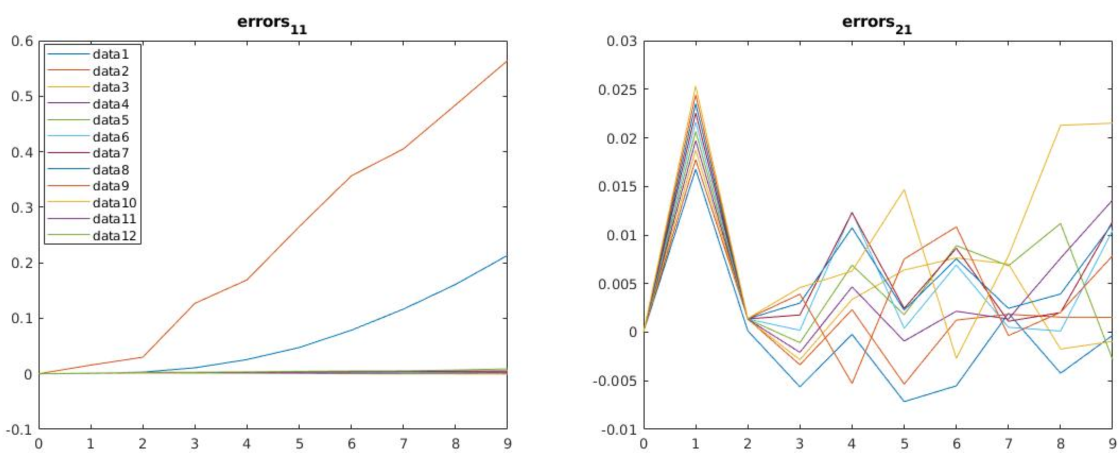
In this section, two different Q value is used. As one can see, two different behaviors are shown in figure7 and figure8. In figure8 the errors in state1 converge to 0 after several iterations, but the errors in state2 does not(even though it is very trivial). One reason for this is that when Q is set to 0.99, we take into account all the noises in the past.




(a) state1 error

(b) state2 error

Figure 7: Q = 0.6




(a) state1 error

(b) state2 error

Figure 8: Q = 0.99

# 3 Part III - Linear Quadratic Gaussian

## 3.1

Although here the cost function is defined respect to a reference, which is not the standard form of LQR, the reference becomes 0 as time goes to infinity. Therefore, the reference terms are dropped. Steady state Riccati equation:

$$P_\infty = Q + A^T PA - A^T PB[R + B^T PB]^{-1} B^T PA$$

## 3.2

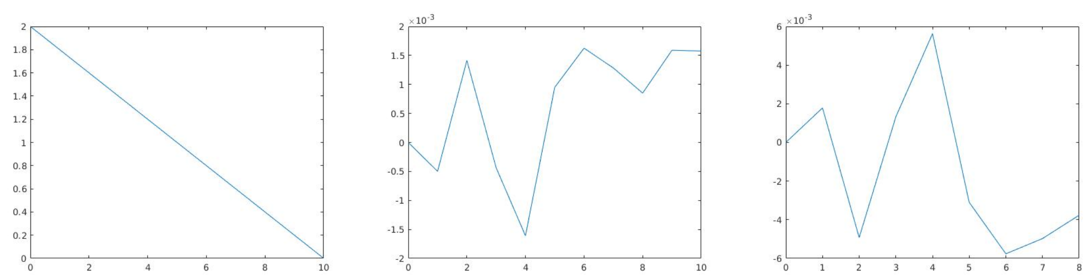
$$P = \begin{bmatrix} 13.9998 & 3.2631 \\ 3.2631 & 4.5875 \end{bmatrix}$$

## 3.3

$$\text{control gain} = -(R + B^T PB)^{-1} B^T PA = \begin{bmatrix} -2.6511 & -3.4431 \end{bmatrix}$$
$$\therefore u_k = g(x_k) = \begin{bmatrix} -2.6511 & -3.4431 \end{bmatrix} x_k$$

## 3.4

Although it is a stochastic system, the noise is relative subtle.



(a) estimated state1 (b) estimated state2 (c) control trajectory

Figure 9: LQR on stochastic system

### 3.5

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad B = B^w = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

dynamic update

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_{k-1} \\ M_k &= AZ_{k-1}A^T + B^w W_{k-1} B^{wT} \end{aligned}$$

measurement update

$$Z_k = M_k - M_k C^T (C M_k C^T + V_k)^{-1} C M_k$$

### 3.6

steady state Ricatti equation:

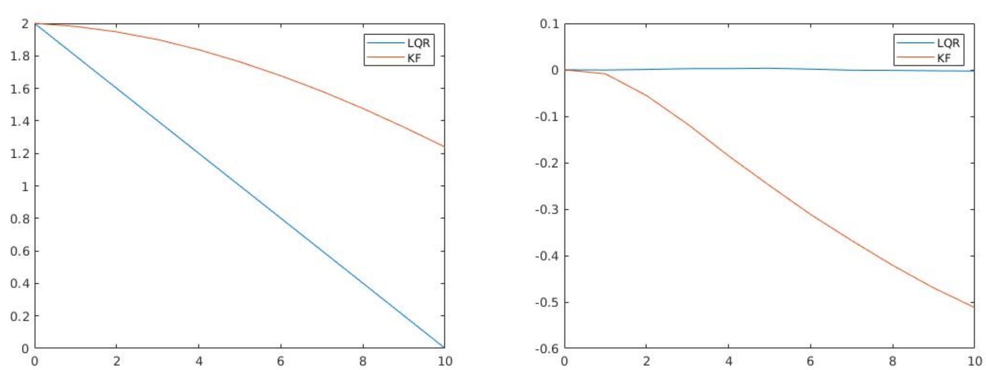
$$\begin{aligned} M_s &= AM_sA^T + B^W (B^w)^T - AM_sC^T(CM_sC^T + V)^{-1}CM_sA^T \\ &= 1.0e-05 \begin{bmatrix} 0.1091 & 0.1306 \\ 0.0629 & 0.3530 \end{bmatrix} \end{aligned}$$

KF gain:

$$\begin{aligned} F_s &= M_sC^T(CM_sC^T + V)^{-1} \\ &= \begin{bmatrix} 0.0934 & 0.0875 \\ 0.0422 & 0.2568 \end{bmatrix} \end{aligned}$$

### 3.7

The two trajectories are quite different, but they are in similar trends. In a kalman filter case, the actual states x are not measured directly, but indirectly through observing y. The noise among the state measurements and control inputs would accumulate and results in increasing errors.




(a) estimated state1 (b) estimated state2

Figure 10: KF

# 4 Part IV - Parameter Adaptation

## 4.1

$$\begin{bmatrix} x_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} A(\theta)x_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} B(\theta)u_k \\ 0 \end{bmatrix} + \begin{bmatrix} B(\theta) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_k \\ w_k^\theta \end{bmatrix}$$
$$y = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + V_k$$

## 4.2

In EKF, the system dynamics need to be linearized. Here in 4.2 I used first order Taylor expansion to linearize the original system.
Dynamic update

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}$$
$$M_k = AZ_{k-1}A^T + B^w W_{k-1} B^{wT}$$

Measurement update

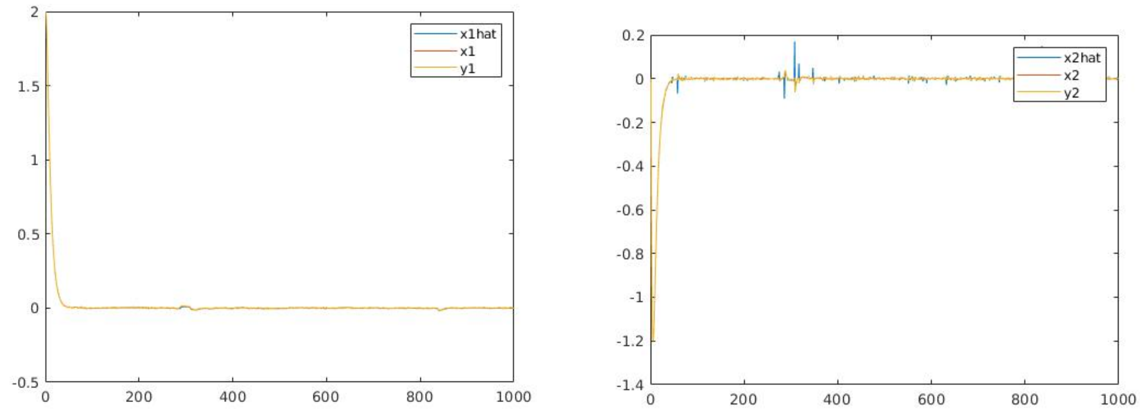
$$Z_k = M_k - M_k C^T (C M_k C^T + V_k)^{-1} C M_k$$

Taylor expansion:

$$\bar{x_{k+1}} = \begin{bmatrix} \bar{x_{k1}} + a_k x_{k1} + b_k u_k \\ \bar{x_{k2}} + c_k u_k \\ a_k \\ b_k \\ c_k \end{bmatrix} = f(\bar{x_k}, u_k), \quad \bar{x_k} = \begin{bmatrix} x_{k1} \\ x_{k2} \\ a_k \\ b_k \\ c_k \end{bmatrix}$$

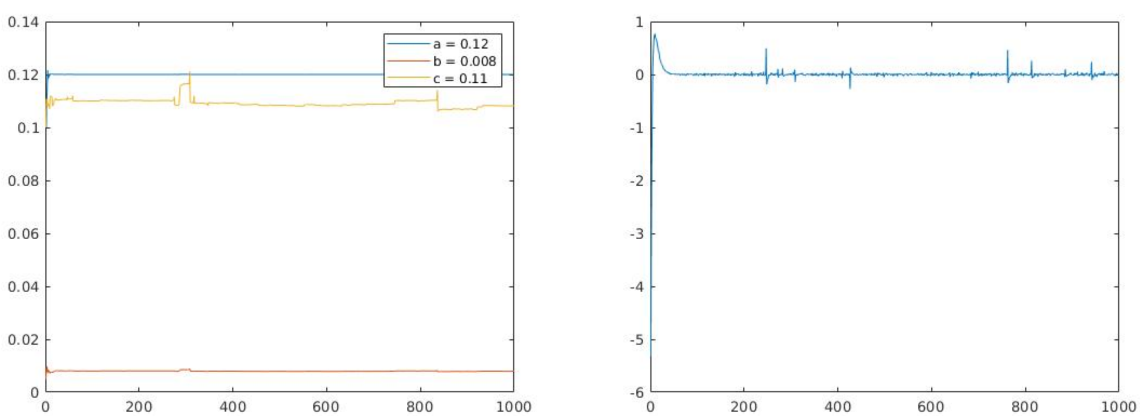
$$f_k(\bar{x_k}, u_k) \sim f_k(\hat{x_{k|k}}, u_k) + \frac{\partial}{\partial \bar{x}} f_k(\hat{x_{k|k}}, u_k)(\bar{x_k} - \hat{x_{k-1|k-1}})$$

$$\frac{\partial}{\partial \bar{x}} f_k(\hat{x_{k|k}}, u_k) = \begin{bmatrix} 1 & a_k & x_{k2} & u_k & 0 \\ 0 & 1 & 0 & 0 & u_k \\ 0 & 0 & 1 & u_k & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \bar{A_k}$$




(a) estimated state1 (b) estimated state2

Figure 11: EKF states




(a) estimated a, b, c (b) control input

Figure 12: EKF estimations

## 4.3

In UKF, instead of linearizing the original dynamics, it linearizes the distribution after the nonlinear transform.
Dynamics Update

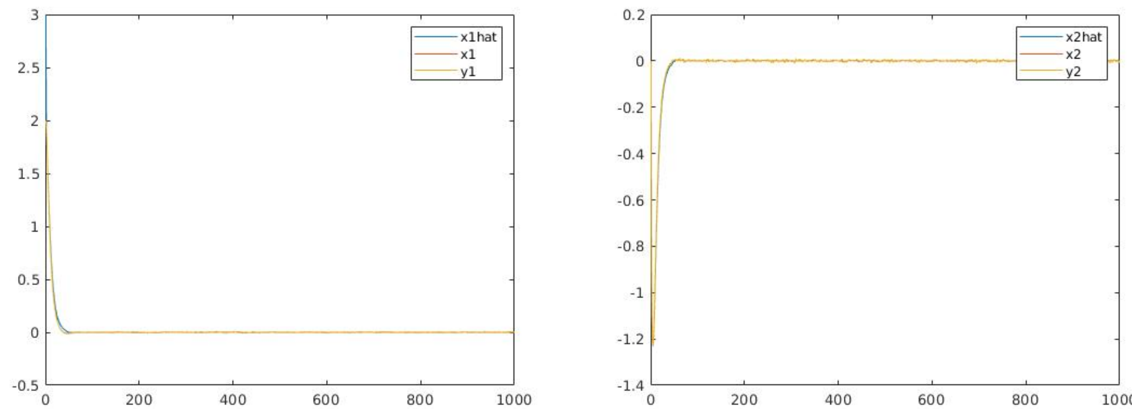
$$X_{k-1} = Sample(x_{k-\hat{1}|k-1}, Z_{k-1}) \qquad X_k = f_{k-1}(X_{k-1}, u_k)$$
$$x_{k|\hat{k}-1} = mean(X_k) \qquad M_k = Var(X_k) + B^w W (B^w)^T$$

Measurement update

$$\bar{X_k} = Sample(x_{k|\hat{k}-1}, M_k) \qquad Y_k = h_k(\bar{X_k})$$
$$\Sigma_k^{x,y} = Cov(\bar{X_k}, Y_k) \qquad \Sigma_k^y = Var(Y_k) + V_k y_{k|\hat{k}-1} = mean(Y_k)$$

Kalman Gain:

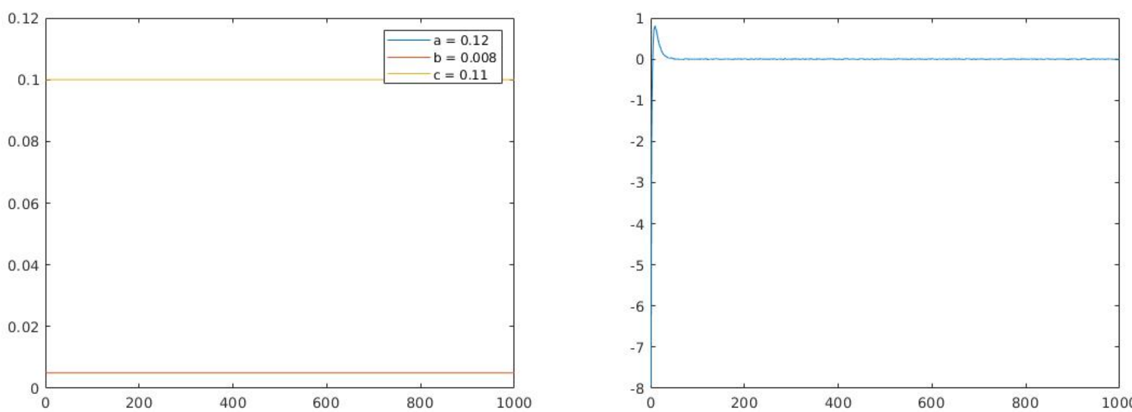
$$x_{\hat{k|k}} = x_{k|\hat{k}-1} + \Sigma_k^{x,y}\Sigma_k^{y-1}(y_k - y_{k|\hat{k}-1})$$
$$Z_k = M_k - \Sigma_k^{x,y}\Sigma_k^{y-1}\Sigma_k^{x,yT}$$




(a) estimated state1

(b) estimated state2

Figure 13: UKF states




(a) estimated a, b, c

(b) control input

Figure 14: UKF estimations