Filtering and Identification (SC42025)
Home exercise 2

Exercise 1

a) From the state space model

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad w_k \sim \mathcal{N}(0, Q)$$
$$y_k = Cx_k + v_k, \quad v_k \sim \mathcal{N}(0, R)$$

According Bayes theorem, the conditional distribution $p(x_k | x_k, y_{1:k-1})$ is written as

$$p(x_k | x_k, y_{1:k-1}) = \frac{p(x_k | x_{k-1}, y_{1:k-1}) p(x_{k-1} | y_{1:k-1})}{p(x_k | x_{k-1})}$$

$$p(x_{k-1} | y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$$

$$p(x_k | x_{k-1}) = \mathcal{N}(x_k; Ax_{k-1} + Bu_{k-1}, Q)$$

$$p(x_k | x_{k-1}, y_{1:k-1}) = p(x_k | x_{k-1})$$

$$p(x_k | x_k, y_{1:k-1}) = \frac{p(x_k | x_{k-1}, y_{1:k-1}) p(x_{k-1} | y_{1:k-1})}{p(x_k | x_{k-1})} = \frac{p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1})}{p(x_k | x_{k-1})}$$

$$p(x_k | x_k, y_{1:k-1}) = p(x_{k-1} | y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$$

According the system $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, w_{k-1} \sim \mathcal{N}(0, Q)$

From theorom 1 (in video2_handout.pdf), random variables x_k, x_{k-1} have the Gaussian probability distributions, $p(x_k | y_{1:k-1}) = \mathcal{N}(x_k : A\hat{x}_{k-1|k-1} + Bu_{k-1}, AP_{k-1|k-1}A^T + Q)$ and $p(x_{k-1} | y_{1:k-1}) = \mathcal{N}(x_{k-1} : \hat{x}_{k-1|k-1}, P_{k-1|k-1})$, the joint distribution of x_k, x_{k-1} is still Gaussian probability distributions and written as

$$p(x_k, x_{k-1} | y_{1:k-1}) = \mathcal{N}\left(\begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix}; \begin{pmatrix} A\hat{x}_{k-1|k-1} + Bu_{k-1} \\ \hat{x}_{k-1|k-1} \end{pmatrix}, \begin{pmatrix} AP_{k-1|k-1}A^T + Q & AP_{k-1|k-1} \\ P_{k-1|k-1}A^T & P_{k-1|k-1} \end{pmatrix} \right)$$

The predictive distribution $p(x_k | y_{1:k-1})$ is written as

$$p(x_k | y_{1:k-1}) = \int p(x_k | y_{1:k-1}, x_{k-1}) dx_{k-1} = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1}$$

According to the Lemma 1, given the distribution $p(x_{k-1} | y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$ and the model $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$ with A, B deterministic and $w_{k-1} \sim \mathcal{N}(0, Q)$, it holds that

$$p(x_k | y_{1:k-1}) = \mathcal{N}(x_k; A\hat{x}_{k-1|k-1} + Bu_{k-1}, AP_{k-1|k-1}A^T + Q)$$

b) The control input has the distribution that $u_k \sim \mathcal{N}(\mu_k, \Sigma)$ and u_k v_k, w_k uncorrelated

From the state space model

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad w_k \sim \mathcal{N}(0, Q)$$
$$y_k = Cx_k + v_k, \quad v_k \sim \mathcal{N}(0, R)$$

According Bayes theorem, the conditional distribution $p(x_k|x_k, y_{1:k-1})$ is written as

$$p(x_k|x_k, y_{1:k-1}) = \frac{p(x_k|x_{k-1}, y_{1:k-1})\,p(x_{k-1}|y_{1:k-1})}{p(x_k|x_{k-1})} \quad \text{where } p(x_{k-1}|y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$$

$p(x_k|x_{k-1}) = \mathcal{N}(x_k; Ax_{k-1}+Bu_{k-1}, Q)$, $p(x_k|x_{k-1}, y_{1:k-1}) = p(x_k|x_{k-1})$

$$p(x_k|x_k, y_{1:k-1}) = \frac{p(x_k|x_{k-1}, y_{1:k-1})\,p(x_{k-1}|y_{1:k-1})}{p(x_k|x_{k-1})} = \frac{p(x_k|x_{k-1})\,p(x_{k-1}|y_{1:k-1})}{p(x_k|x_{k-1})}$$

$p(x_k|x_k, y_{1:k-1}) = p(x_{k-1}|y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$

According the system $x_k = Ax_{k-1}+Bu_{k-1}+w_{k-1}$, $w_{k-1} \sim \mathcal{N}(0, Q)$, $u_{k-1} \sim \mathcal{N}(\mu_k, \Sigma)$

$E[x_k] = E[Ax_{k-1}+Bu_{k-1}+w_{k-1}] = AE[x_{k-1}]+BE[u_{k-1}]+E[w_{k-1}] \Rightarrow \hat{x}_k = A\hat{x}_{k-1}+B\hat{u}_{k-1} = A\hat{x}_{k-1}+B\mu_k$

$$E\left[\begin{pmatrix} x_k \\ x_{k-1}\end{pmatrix}\right] = \begin{pmatrix} A\hat{x}_{k-1|k-1}+B\mu_{k-1} \\ \hat{x}_{k-1|k-1}\end{pmatrix}$$ The covariance matrix can be written as

$$E\left[\begin{pmatrix} x_k-(A\hat{x}_{k-1|k-1}+B\mu_{k-1}) \\ x_{k-1}-\hat{x}_{k-1|k-1}\end{pmatrix}\begin{pmatrix}(x_k-(A\hat{x}_{k-1|k-1}+B\mu_{k-1}))^T & (x_{k-1}-\hat{x}_{k-1|k-1})^T\end{pmatrix}\right]$$

$E[(x_{k-1}-\hat{x}_{k-1|k-1})(x_{k-1}-\hat{x}_{k-1|k-1})^T] = P_{k-1|k-1}$

$E[(x_k-(A\hat{x}_{k-1|k-1}+B\mu_{k-1}))(x_k-(A\hat{x}_{k-1|k-1}+B\mu_{k-1}))^T] = E[(Ax_{k-1}+Bu_{k-1}+w_{k-1}-A\hat{x}_{k-1|k-1}-B\mu_{k-1})\cdot(Ax_{k-1}+Bu_{k-1}+w_{k-1}-A\hat{x}_{k-1|k-1}-B\mu_{k-1})^T] = E[[A(x_{k-1}-\hat{x}_{k-1|k-1})+B(u_{k-1}-\mu_{k-1})+w_{k-1}]\cdot[A(x_{k-1}-\hat{x}_{k-1|k-1})+B\cdot(u_{k-1}-\mu_{k-1})+w_{k-1}]^T] = AE[(x_{k-1}-\hat{x}_{k-1|k-1})(x_{k-1}-\hat{x}_{k-1|k-1})^T]A^T + BE[(u_{k-1}-\mu_{k-1})(u_{k-1}-\mu_{k-1})^T]B^T + E[w_{k-1}w_{k-1}^T] =$

$AP_{k-1|k-1}A^T+B\Sigma B^T+Q$ (note that w_k and u_k are uncorrelated, assume that $E[(x_k-\hat{x}_{k|k-1})[v_k^T\ w_k^T\ u_k^T]^T]=0$)

$E[(x_{k-1}-\hat{x}_{k-1|k-1})(x_k-(A\hat{x}_{k-1|k-1}+B\mu_{k-1}))^T] = E[(x_{k-1}-\hat{x}_{k-1|k-1})(Ax_{k-1}+Bu_{k-1}+w_{k-1}-A\hat{x}_{k-1|k-1}-B\mu_{k-1})^T]$
$= E[(x_{k-1}-\hat{x}_{k-1|k-1})[A(x_{k-1}-\hat{x}_{k-1|k-1})+B(u_{k-1}-\mu_{k-1})+w_{k-1}]^T] = E[(x_{k-1}-\hat{x}_{k-1|k-1})(x_{k-1}-\hat{x}_{k-1|k-1})^T]A^T +$
$E[(x_{k-1}-\hat{x}_{k-1|k-1})\cdot(-\mu_{k-1}^T B^T)] = P_{k-1|k-1}A^T + E[(x_{k-1}-\hat{x}_{k-1|k-1})]\cdot(-\mu_{k-1}^T B^T) = P_{k-1|k-1}A^T$

In the simalar way $E[(x_k-(A\hat{x}_{k-1|k-1}+B\mu_{k-1}))(x_{k-1}-\hat{x}_{k-1|k-1})^T] = AP_{k-1|k-1}$

From theorom1 (in video2_handout.pdf), random variables x_k, x_{k-1} have the Gaussian probability distributions, $p(x_k|y_{1:k-1}) = \mathcal{N}(x_k: A\hat{x}_{k-1|k-1}+B\mu_k, AP_{k-1|k-1}A^T+B\Sigma B^T+Q)$ and $p(x_{k-1}|y_{1:k-1}) = \mathcal{N}(x_{k-1}: \hat{x}_{k-1|k-1}, P_{k-1|k-1})$, the joint distribution of x_k, x_{k-1} is still Gaussian probability distributions and written as

$$p(x_k, x_{k-1}|y_{1:k-1}) = \mathcal{N}\left(\begin{pmatrix} x_k \\ x_{k-1}\end{pmatrix}; \begin{pmatrix} A\hat{x}_{k-1|k-1}+B\mu_k \\ \hat{x}_{k-1|k-1}\end{pmatrix}, \begin{pmatrix} AP_{k-1|k-1}A^T+B\Sigma B^T+Q & AP_{k-1|k-1} \\ P_{k-1|k-1}A^T & P_{k-1|k-1}\end{pmatrix}\right)$$

The predictive distribution $p(x_k|y_{1:k-1})$ is written as

$$p(x_k|y_{1:k-1}) = \int p(x_k|y_{1:k-1}, x_{k-1})\,dx_{k-1} = \int p(x_k|x_{k-1})\,p(x_{k-1}|y_{1:k-1})\,dx_{k-1}$$

According to the Lemma 1, given the distribution $p(x_{k-1}|y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$ and the model $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$ with A, B deterministic and $w_{k-1} \sim \mathcal{N}(0, Q)$, it holds that

$$p(x_k|y_{1:k-1}) = \mathcal{N}(x_k; A\hat{x}_{k-1|k-1} + B\mu_k, AP_{k-1|k-1}A^T + B\Sigma B^T + Q)$$

In question b), we take control input u_k as random variable with distribution $u_k \sim \mathcal{N}(\mu_k, \Sigma)$ the time update becomes $\hat{x}_k = A\hat{x}_{k-1} + B\hat{u}_{k-1} = A\hat{x}_{k-1} + B\mu_k$ and $P_{k+1|k} = AP_{k-1|k-1}A^T + B\Sigma B^T + Q)$

c) The problem with the noisy input can be recast into a similar form in (1) with deterministic input

The noisy input can be decomposed as a deterministic term and stochastic term

$$u_k = \mu_k + g_k, \quad g_k \sim \mathcal{N}(0, \Sigma) \quad ①$$

The state space model is written as

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad w_k \sim \mathcal{N}(0, Q) \quad ②$$

Combining ① and ②, we obtain $x_{k+1} = Ax_k + B(\mu_k + g_k) + w_k = Ax_k + B\mu_k + Bg_k + w_k$

we assume $w_k' = Bg_k + w_k \sim \mathcal{N}(0, Q')$ where $Q' = B\Sigma B^T + Q$, $u_k' = \mu_k$

Then the state space model is rewritten as

$$x_{k+1} = Ax_k + Bu_k' + w_k', \quad w_k' \sim \mathcal{N}(0, Q')$$

$$y_k = Cx_k + v_k, \quad v_k \sim \mathcal{N}(0, R)$$

where $u_k' = \mu_k$ is the state space deterministic term, and also new input of system

Exercise 2

a) $e_k = x_k - \hat{x}_{k|k-1} = (Ax_{k-1}+w_{k-1}) - A\hat{x}_{k-1|k-2} - L_{k-1}(y_{k-1}-\hat{y}_{k-1})$

$y_{k-1} = Cx_{k-1}+v_{k-1}$ $\quad$ $\hat{y}_{k-1} = C\hat{x}_{k-1|k-2}$

$e_k = (Ax_{k-1}+w_{k-1}) - A\hat{x}_{k-1|k-2} - L_{k-1}(Cx_{k-1}+v_{k-1} - C\hat{x}_{k-1|k-2})$

$e_k = (A-L_{k-1}C)x_{k-1} - (A-L_{k-1}C)\hat{x}_{k-1|k-2} + w_{k-1} - L_{k-1}v_{k-1} = (A-L_{k-1}C)(x_{k-1}-\hat{x}_{k-1|k-2}) + w_{k-1} - L_{k-1}v_{k-1}$

b) The distribution of estimation error at time k $e_{k|k-1} \sim N(0, P_{k|k-1})$

The covariance matrix is written as,

$E[(e_k - \tilde{e}_{k|k-1})(e_k-\tilde{e}_{k|k-1})^T] = E[[(A-L_{k-1}C)(x_{k-1}-\hat{x}_{k-1|k-2})+w_{k-1}-L_{k-1}v_{k-1}][(A-L_{k-1}C)(x_{k-1}-\hat{x}_{k-1|k-2})+w_{k-1}-L_{k-1}v_{k-1}]^T]$

$= (A-L_{k-1}C)E[(x_{k-1}-\hat{x}_{k-1|k-2})(x_{k-1}-\hat{x}_{k-1|k-2})^T](A-L_{k-1}C)^T + E[w_{k-1}w_{k-1}^T] + E[L_{k-1}v_{k-1}v_{k-1}^T L_{k-1}^T] + E[(A-L_{k-1}C)\cdot$
$(x_{k-1}-\hat{x}_{k-1|k-2})\cdot w_{k-1}^T] + E[(A-L_{k-1}C)(x_{k-1}-\hat{x}_{k-1|k-2})\cdot(-L_{k-1}v_{k-1})^T] + E[w_{k-1}(x_{k-1}-\hat{x}_{k-1|k-2})^T(A-L_{k-1}C)^T] +$
$E[-L_{k-1}v_{k-1}\cdot(x_{k-1}-\hat{x}_{k-1|k-2})^T(A-L_{k-1}C)^T] + E[w_{k-1}v_{k-1}^T L_{k-1}^T] + E[-L_{k-1}v_{k-1}\cdot w_{k-1}^T]$

As $E\left[\begin{pmatrix} w_k\\ v_k\end{pmatrix}(w_k^T\ v_k^T)\right] = E\begin{bmatrix} w_kw_k^T & w_kv_k^T\\ v_kw_k^T & v_kv_k^T\end{bmatrix} = \begin{pmatrix} Q & S\\ S^T & R\end{pmatrix}\Delta(k-j) > 0$

$E[(x_k-\hat{x}_{k|k-1})v_k^T]=0$ and $E[(x_k-\hat{x}_{k|k-1})w_k^T]=0$ $\quad E[(x_{k-1}-\hat{x}_{k-1|k-2})(x_{k-1}-\hat{x}_{k-1|k-2})^T] = P_{k-1|k-2}$

$E[(e_k-\tilde{e}_{k|k-1})(e_k-\tilde{e}_{k|k-1})^T] = (A-L_{k-1}C)E[(x_{k-1}-\hat{x}_{k-1|k-2})(x_{k-1}-\hat{x}_{k-1|k-2})^T](A-L_{k-1}C)^T + E[w_{k-1}w_{k-1}^T] + E[L_{k-1}v_{k-1}v_{k-1}^T L_{k-1}^T]$
$E[w_{k-1}v_{k-1}^T L_{k-1}^T] - E[L_{k-1}v_{k-1}\cdot w_{k-1}^T] = (A-L_{k-1}C)P_{k-1|k-2}(A-L_{k-1}C)^T + Q + L_{k-1}RL_{k-1}^T - L_{k-1}S^T - SL_{k-1}^T$

$P_{k|k-1} = (A-L_{k-1}C)P_{k-1|k-2}(A-L_{k-1}C)^T - SL_{k-1}^T - L_{k-1}S^T + Q + L_{k-1}RL_{k-1}^T$

Then we can conclude that

$P_{k+1|k} = (A-L_kC)P_{k|k-1}(A-L_kC)^T - SL_k^T - L_kS^T + Q + L_kRL_k^T$

c) The completion of squares Lemma states that the solution of the optimazition problem

$\min_{L_k} P_{k+1|k} = \min_{L_k}(A-L_kC)P_{k|k-1}(A-L_kC)^T - SL_k^T - L_kS^T + Q + L_kRL_k^T$, as Q is deterministic matrix, the optimazition can be written as

$\min_{L_k}(A-L_kC)P_{k|k-1}(A-L_kC)^T - SL_k^T - L_kS^T + L_kRL_k^T$ $\quad$ Assume that $A-L_kC = X$, $F = P_{k|k-1}$,

$\min_{L_k}(A-L_kC)P_{k|k-1}(A-L_kC)^T - SL_k^T - L_kS^T + L_kRL_k^T$

$= \min_{L_k}(A-L_kC)P_{k|k-1}(A^T - C^TL_k^T) - SL_k^T - L_kS^T + L_kRL_k^T$

$= \min_{L_k} AP_{k|k-1}A^T - \underline{L_k}CP_{k|k-1}A^T - AP_{k|k-1}C^T\underline{L_k^T} + \underline{L_k}CP_{k|k-1}C^T\underline{L_k^T} - S\underline{L_k^T} - \underline{L_k}S^T + \underline{L_k}R\underline{L_k^T}$

As A is deterministic, $P_{k|k-1}$ is known, we can get rid of term $AP_{k|k-1}A^T$, the optimization problem be written as

$$\min_{L_k} L_k(CP_{k|k-1}C^T+R)L_k^T+(-AP_{k|k-1}C^T-S)L_k^T-L_k(CP_{k|k-1}A^T+S^T)$$

We assume $L_k=X$, $F=CP_{k|k-1}C^T+R$, $G=CP_{k|k-1}A^T+S^T$

the optimization problem can be written as $\min_X XFX^T-XG-G^TX^T$ the optimization problem arrive at minimal when $X=G^TF^{-1}$, $L_k=(CP_{k|k-1}A^T+S^T)^T(CP_{k|k-1}C^T+R)^{-1}=(AP_{k|k-1}^TC^T+S)(CP_{k|k-1}C^T+R)^{-1}$ the as $P_{k|k-1}$ is symmetric matrix $P_{k|k-1}=P_{k|k-1}^T$, $L_k=(S+AP_{k|k-1}C^T)(CP_{k|k-1}C^T+R)^{-1}$, Luenberger observer has minimum error covariance $P_{k+1|k}$

d) If the covariance is stationary, we assume that $\lim_{k\to\infty} P_{k|k-1}=P_\infty>0$ the gain is given by $\tilde{K}_k$, $L_k=(S+AP_\infty C^T)(CP_\infty C^T+R)^{-1}$, when $k\to\infty$, the equation

$$P_{k+1|k}=(A-L_kC)P_\infty(A-L_kC)^T-SL_k^T-L_kS^T+Q+L_kRL_k^T \quad \text{becomes}$$

$$P_\infty=(A-L_kC)P_\infty(A-L_kC)^T-SL_k^T-L_kS^T+Q+L_kRL_k^T,\ L_k=(S+AP_{k|k-1}C^T)(CP_{k|k-1}C^T+R)^{-1}$$

$$P_\infty=AP_\infty A^T+L_k(CP_\infty C^T+R)L_k^T+(-AP_\infty C^T-S)L_k^T-L_k(CP_\infty A^T+S^T)+Q$$

substitute L_k with $(S+AP_\infty C^T)(CP_\infty C^T+R)^{-1}$

$$P_\infty=AP_\infty A^T+(S+AP_\infty C^T)[(CP_\infty C^T+R)^{-1}]^T(S+AP_\infty C^T)^T+(-AP_\infty C^T-S)[(CP_\infty C^T+R)^{-1}]^T(S+AP_\infty C^T)^T$$
$$-(S+AP_\infty C^T)(CP_\infty C^T+R)^{-1}(CP_\infty A^T+S^T)+Q$$

$$\Rightarrow P_\infty=AP_\infty A^T+Q+(S+AP_\infty C^T)\left[[(CP_\infty C^T+R)^{-1}]^T-[(CP_\infty C^T+R)^{-1}]^T-(CP_\infty C^T+R)^{-1}\right](S+AP_\infty C^T)^T$$

$$\Rightarrow P_\infty=AP_\infty A^T+Q-(S+AP_\infty C^T)(CP_\infty C^T+R)^{-1}(S+AP_\infty C^T)^T \leftarrow \text{discrete Algebraic Riccati Equation (DARE)}$$

when the pair (A,C) is observable, and the pair $(A,Q^{\frac{1}{2}})$ is controllable, $A-\tilde{K}C$ is asymptotically stable, $A-\tilde{K}C$ is positive definite, then DARE has unique solution such that $P>0$

e) The constant Kalman Gain cannot be used as L_k for all values of $k\geq 0$, because P_∞ can be achieved when $k\to\infty$, or some certain number K_t. when $k\geq k_t$, $P=P_\infty$. Thus for some k which makes $P_{k|k-1}\neq P_\infty$, we cannot use $\tilde{K}$ as $\tilde{K}_\infty$. The difference between time-varying gain $\tilde{K}_k$ and constant gain $\tilde{K}_\infty$ is we can use $\tilde{K}_k$ for all time steps, for $\tilde{K}_\infty$ we can only use it when $P_{k|k-1}=P_\infty$. When we use $\tilde{K}_k$, the observaver changes over time, for different time steps having different covariance, when we use $\tilde{K}_\infty$, the observaver has fixed distribution, having the same covariance matrix

SC42025 Filtering & Identification

MATLAB EXERCISE HW2

We want to localize a vehicle with GPS data. However the data is noisy and we cannot accurately estimate the state of the vehicle relying merely on the GPS signal. In order to improve the estimate, we wish to combine the data with vehicle acceleration measurements from an accelerometer mounted on the vehicle.

$$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{p}^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$$

Where $\boldsymbol{p} = \begin{bmatrix}p_x & p_y\end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{v} = \begin{bmatrix}v_x & v_y\end{bmatrix}^{\mathrm{T}}$ are the vector position and velocity in the ground plane respectively. The equations of motion are

$$\boldsymbol{p}_k = \boldsymbol{p}_{k-1} + \boldsymbol{v}_{k-1}\Delta t + \frac{1}{2}\boldsymbol{a}_{k-1}\Delta t^2$$

$$\boldsymbol{v}_k = \boldsymbol{v}_{k-1} + \boldsymbol{a}_{k-1}\Delta t$$

Where $\boldsymbol{a}_k$ is the acceleration vector at time step k. It is determined using the measured acceleration $\widetilde{\boldsymbol{a}}$ from the accelerometer as

$$\mathbf{a}_k = \widetilde{\mathbf{a}}_k + \epsilon_k \qquad \epsilon_k \sim \mathcal{N}(0, \sigma_e^2 I_{2\times 2})$$

The accelerometer compensates for the acceleration due to gravity, and therefore only the vehicle accelerations are recorded.

At each time step we also have available measured GPS state information as

$$\mathbf{z}_k = \begin{bmatrix}\mathbf{p}_k \\ \mathbf{v}_k\end{bmatrix} + \eta_k \qquad \eta_k \sim \mathcal{N}(0, R)$$

It is given that $\sigma_e = 0.3\ \mathrm{ms}^{-2}$ and $R = diag(3^2, 3^2, 0.03^2, 0.03^2)$. The timestep $\Delta t = 0.1\mathrm{s}$, and the total time-period of the trial is 10s. The vehicle is at the origin with zero velocity at the beginning of the trial. Also important to note here is that all the measurements are already in the global coordinate system, so no coordinate transformations are required.

Using the given equations, model a state space equation of the form

$$\begin{aligned}\mathbf{x}_{k+1} &= A\mathbf{x}_k + B\widetilde{\mathbf{a}}_k + w_k \qquad w_k \sim \mathcal{N}(0, Q)\\ \mathbf{z}_k &= C\mathbf{x}_k + \eta_k \qquad \eta_k \sim \mathcal{N}(0, R)\end{aligned}$$

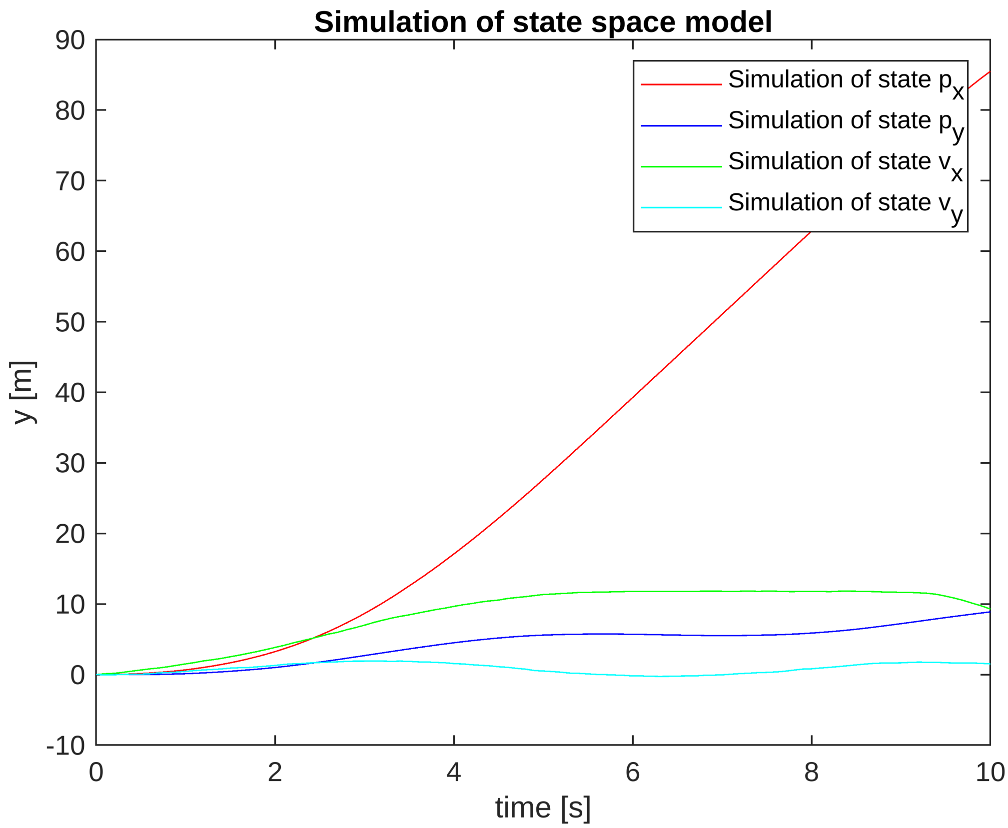
Note that in the above state space model, the measured accelerations are used as an input to the system, while the GPS measurements are considered as the outputs.

What are the matrices $A, B, C \text{ and } Q$?

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix},\ C = I_{4\times 4},\ Q = \sigma_e^2 \begin{bmatrix} \frac{1}{4}\Delta t^4 & 0 & \frac{1}{2}\Delta t^3 & 0 \\ 0 & \frac{1}{4}\Delta t^4 & 0 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & 0 & \Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^3 & 0 & \Delta t^2 \end{bmatrix}$$

Import the data from the file *data_hw2.mat* file. It contains the noisy accelerometer readings ($\tilde{a}_k$), and the GPS state measurements(z_k). Using the acceleration measurements simulate the state space model.

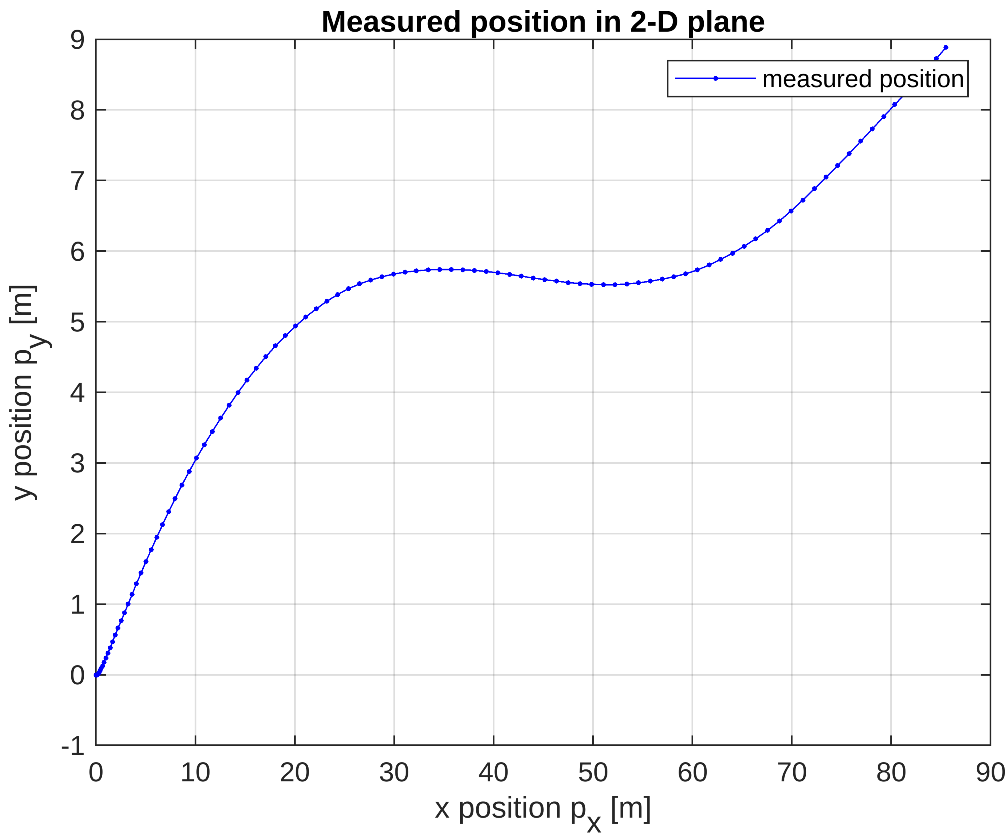
```
clear;
dt = 0.1;                                   % timestep
t = 0:0.1:10;                               % simulation time
T = length(t);                              % total simulation time
x0 = zeros(4,1);                            % initial state
load("data_hw2.mat")                             % load data
R=[9 0 0 0;0 9 0 0;0 0 0.0009 0; 0 0 0 0.0009;];
A=[1 0 dt 0;0 1 0 dt; 0 0 1 0; 0 0 0 1];
B=[0.5*dt^2 0; 0 0.5*dt^2; dt 0; 0 dt];
C=eye(4,4);
D=0;
Q=0.3^2.*[dt^4/4 0 dt^3/2 0; 0 dt^4/4 0 dt^3/2;dt^3/2 0 dt^2 0;0 dt^3/2 0 dt^2];
epsilon=[0.09 0;0 0.09];
u=a_meas;
sys = ss(A,B,C,D,dt);
y1=lsim(sys,u,t,x0);
figure
plot(t,y1(:,1),'r-');
hold on
plot(t,y1(:,2),'b-');
hold on
plot(t,y1(:,3),'g-');
hold on
plot(t,y1(:,4),'c-');
xlabel('time [s]');ylabel('y [m]');title('Simulation of state space model');
legend('Simulation of state p_x','Simulation of state p_y','Simulation of state v_x','S
```



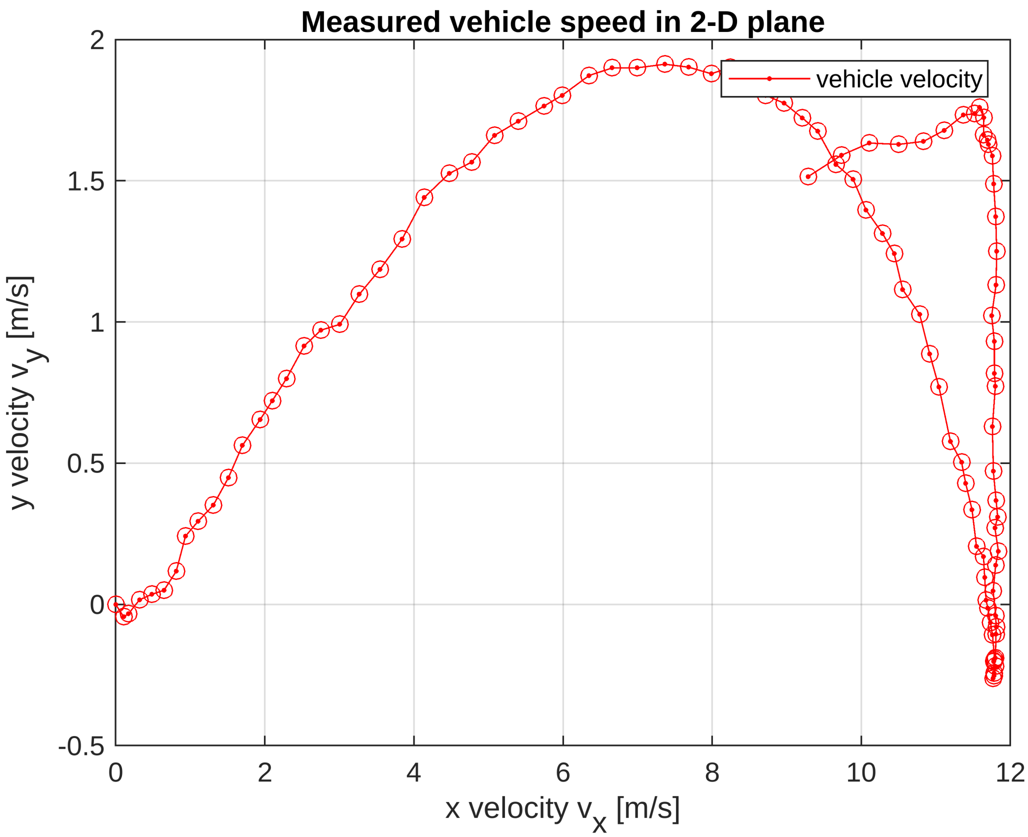
Hint: You can use the function lsim() for simulating the state space model.

From the GPS data, plot the measured position (in the 2D plane) and the vehicle speed.

```
figure
plot(y1(:,1),y1(:,2),'b.-',y1(:,1),y1(:,2),'b.-');grid on;
xlabel('x position p_x [m]');ylabel('y position p_y [m]');
hold on
title('Measured position in 2-D plane');
legend('measured position');
```




```
figure
plot(y1(:,3),y1(:,4),'r.-',y1(:,3),y1(:,4),'ro');grid on;
xlabel('x velocity v_x [m/s]');ylabel('y velocity v_y [m/s]');
title('Measured vehicle speed in 2-D plane');
legend('vehicle velocity');
```



Construct an asymptotic Luenberger observer such that the poles of the observer model are placed at *diag(0.95, 0.95, 0.9, 0.9)* i.e. construct a system of the form

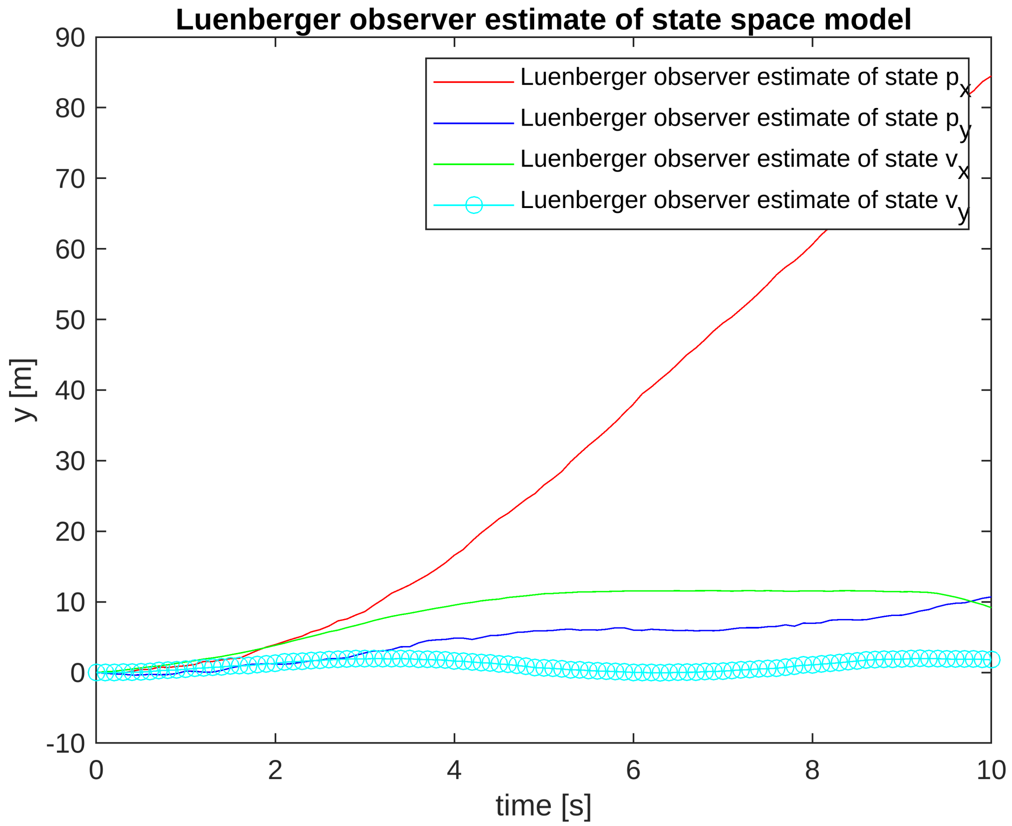
$$\hat{\boldsymbol{x}}_{k+1} = (A - KC)\hat{\boldsymbol{x}}_k + B\tilde{\boldsymbol{a}}_k + K z_{\text{meas},\,k}$$

$$\hat{z}_k = C\hat{\boldsymbol{x}}_k$$

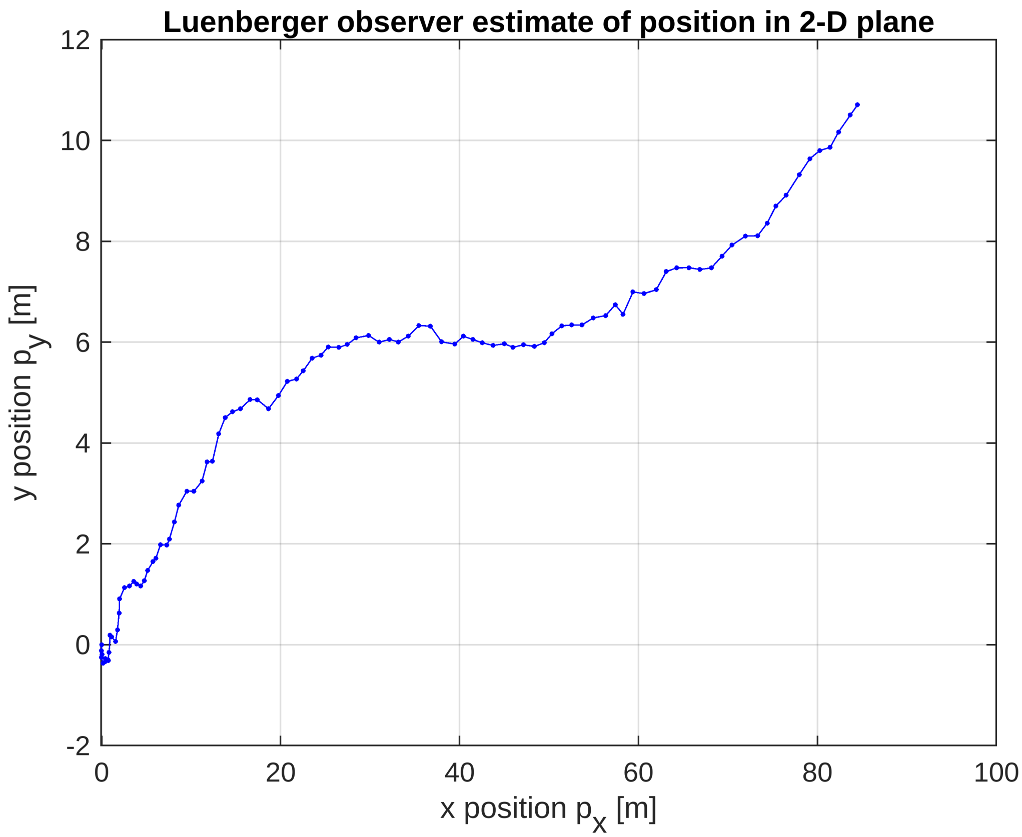
Simulate the system using the acceleration measurements($\tilde{a}_k$) and the GPS readings (z_{meas}).

```
% Design an observer to estimate state of vehicle using both acceleration
% and GPS measurements
p=[0.95 0.95 0.9 0.9];
K = place(A',C',p)';
u_new=B*a_meas+K*z_meas';
A_new=A-K*C;
B_new=eye(4,4);
sys = ss(A_new,B_new,C,D,dt);
y2=lsim(sys,u_new,t,x0);
figure
plot(t,y2(:,1),'r-');
hold on
plot(t,y2(:,2),'b-');
hold on
plot(t,y2(:,3),'g-');
hold on
```

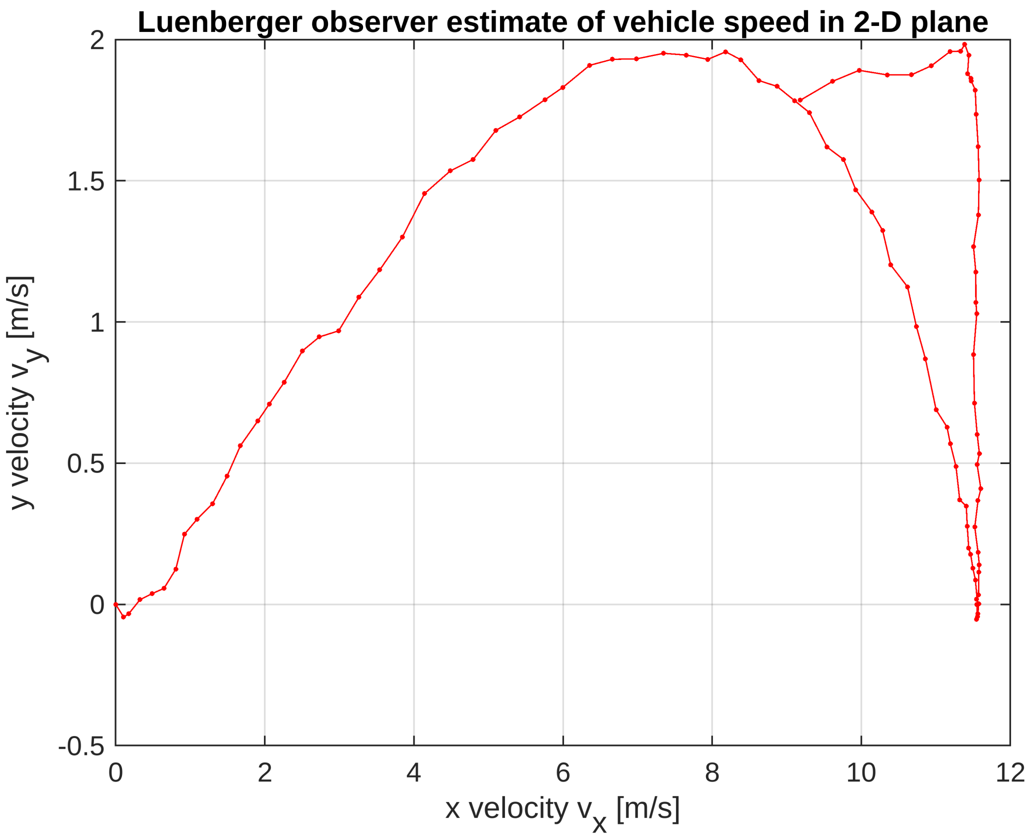
```
plot(t,y2(:,4),'co-');
xlabel('time [s]');ylabel('y [m]');title('Luenberger observer estimate of state space m
legend('Luenberger observer estimate of state p_x','Luenberger observer estimate of sta
```




```
figure
plot(y2(:,1),y2(:,2),'b-',y2(:,1),y2(:,2),'b.');grid on;
hold on
xlabel('x position p_x [m]');ylabel('y position p_y [m]');
title('Luenberger observer estimate of position in 2-D plane');
```




```
figure
plot(y2(:,3),y2(:,4),'r-',y2(:,3),y2(:,4),'r.');grid on;
hold on
xlabel('x velocity v_x [m/s]');ylabel('y velocity v_y [m/s]');
title('Luenberger observer estimate of vehicle speed in 2-D plane');
```



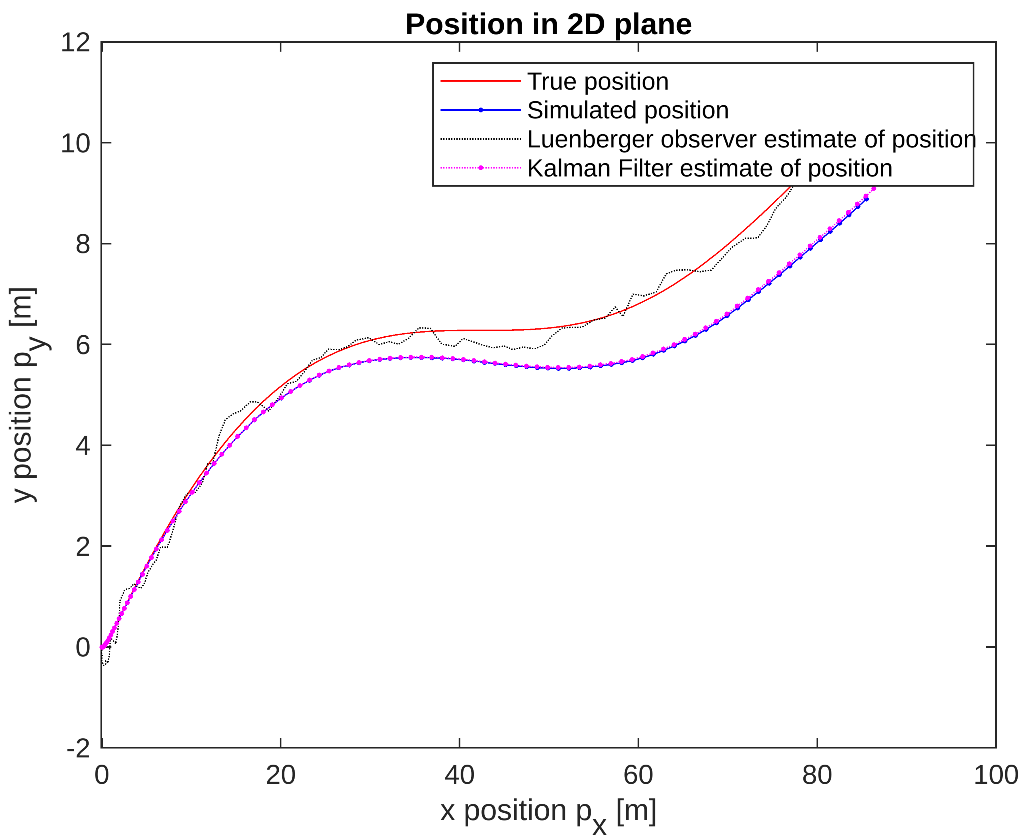
Hint: You can use the command K = place(A',C',p)' to find K for the given poles.

Construct a Kalman Filter to estimate the state of the given state space model, using the same data as done in the previous section. No use of MATLAB toolboxes is allowed, you need to implement the Kalman Filter by yourself. Use the matrix Q you found in the first question, and the matrix R as given.

```
%%Kalman filter algorithm
xmean=x0;%initial mean
xcor=0;%initial covariance
x_estimate=zeros(4,102);
x_estimate(:,1)=x0;
for i=1:length(t)
%measurement update
K_update=xcor*C'*inv(C*xcor*C'+R)*C*xcor;
xmean_update=xmean+K_update*(z_meas(i,:)'-C*xmean);
xcor_update=xcor-xcor*C'*inv(C*xcor*C'+R)*C*xcor;
%time update
xmean=A*xmean_update+B*u(:,i);
x_estimate(:,i+1)=xmean;
xcor=A*xcor_update*A'+Q;
end
```

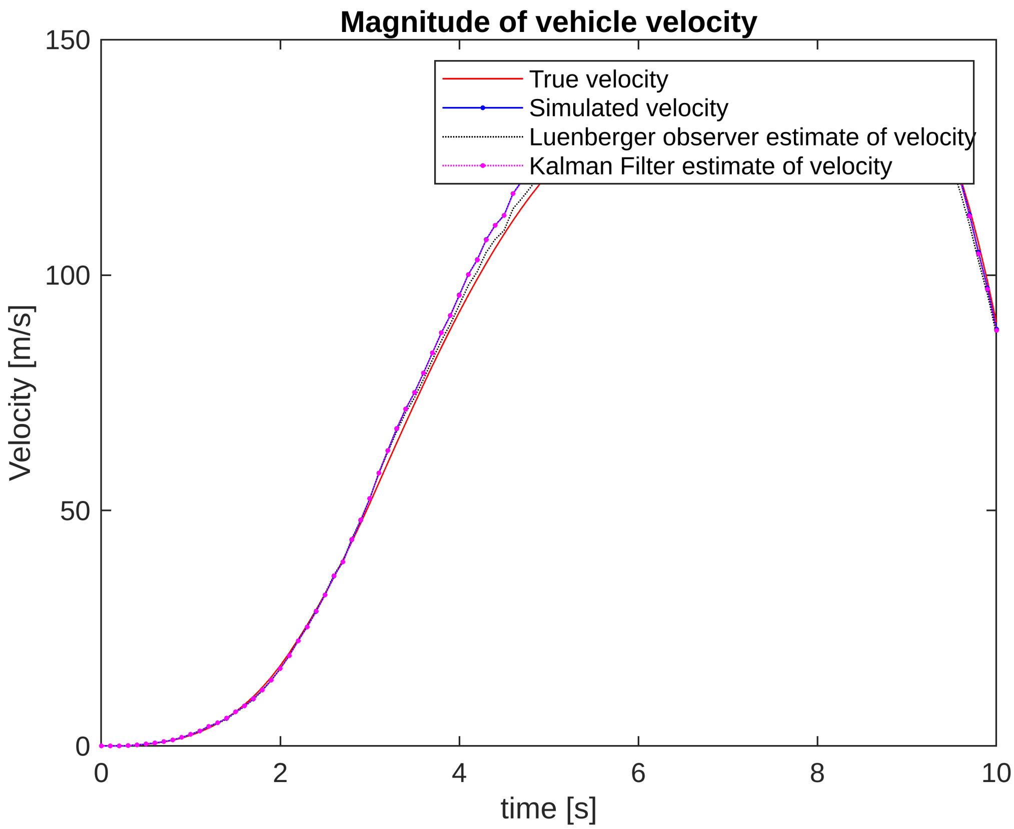
Load the file *ground_truth_hw2.mat.* This file contains the actual states of the vehicle. Plot the true position in the 2D plane, along with the simulated state, Luenberger observer estimate, and Kalman Filter estimate in a graph. In another figure, plot the magnitude of the true vehicle velocity, with the simulated velocity, Luenberger observer estimate, and Kalman Filter estimate. Make certain to provide clear titles, axis labels and legends.

```
load("ground_truth_hw2.mat") % load true state data
%postion
figure
plot(z_true(:,1),z_true(:,2),'r-');
hold on
plot(y1(:,1),y1(:,2),'b.-');
hold on
plot(y2(:,1),y2(:,2),'k:');
hold on
plot(x_estimate(1,:),x_estimate(2,:),'m:.');
xlabel('x position p_x [m]');ylabel('y position p_y [m]');title('Position in 2D plane')
legend('True position','Simulated position','Luenberger observer estimate of position',
```



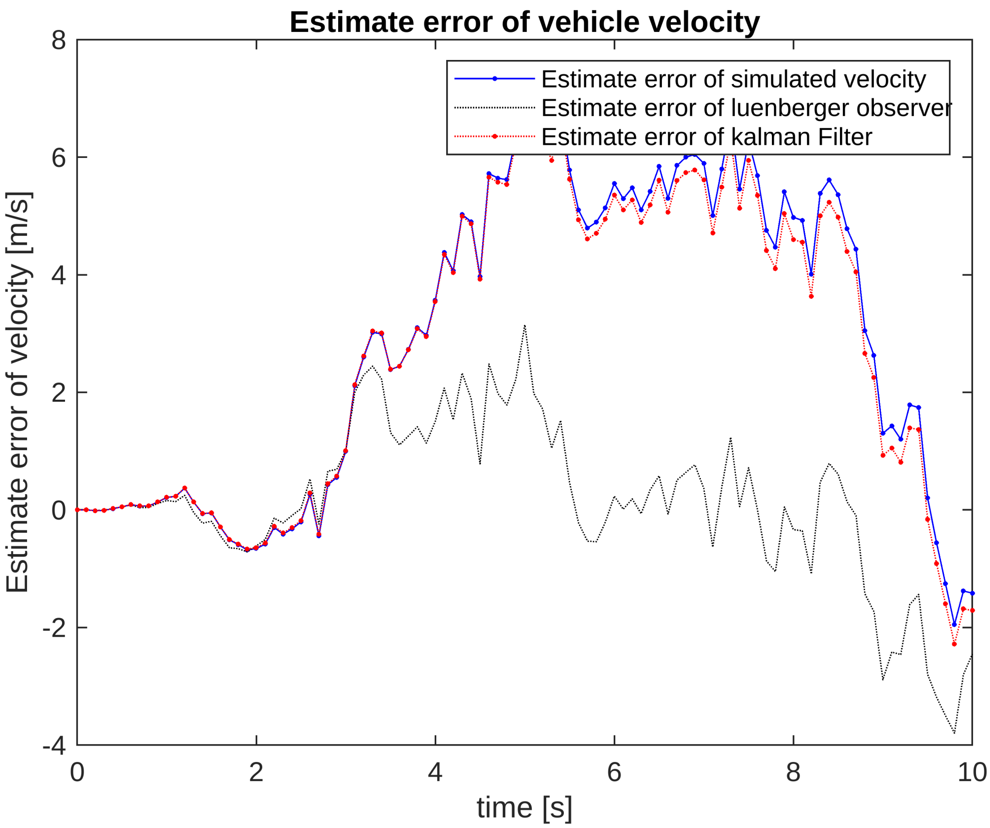

```
%magnitude of vehicle velocity
v_true=z_true(:,3).*z_true(:,3)+z_true(:,4).*z_true(:,4);
v_simulation=y1(:,3).*y1(:,3)+y1(:,4).*y1(:,4);
v_luenberger=y2(:,3).*y2(:,3)+y2(:,4).*y2(:,4);
v_kalman=x_estimate(3,1:101).*x_estimate(3,1:101)+x_estimate(4,1:101).*x_estimate(4,1:1
figure
plot(t,v_true,'r-');
```

```
hold on
plot(t,v_simulation,'b.-');
hold on
plot(t,v_luenberger,'k:');
hold on
plot(t,v_kalman,'m:.');
xlabel('time [s]');ylabel('Velocity [m/s]');title('Magnitude of vehicle velocity');
legend('True velocity','Simulated velocity','Luenberger observer estimate of velocity',
```



Finally calculate and plot the difference between the magnitude of estimated velocity and the magnitude of the true velocity for the simulation, the Luenberger observer, and the Kalman Filter.

```
%estimate error of vehicle velocity
v_true=z_true(:,3).*z_true(:,3)+z_true(:,4).*z_true(:,4);
v_simulation_error=v_simulation-v_true;
v_luenberger_error=v_luenberger-v_true;
v_kalman_error=v_kalman'-v_true;
figure
plot(t,v_simulation_error,'b.-');
hold on
plot(t,v_luenberger_error,'k:');
hold on
plot(t,v_kalman_error,'r:.');
xlabel('time [s]');ylabel('Estimate error of velocity [m/s]');title('Estimate error of
legend('Estimate error of simulated velocity','Estimate error of luenberger observer','
```


Estimate error of vehicle velocity
Estimate error of simulated velocity
Estimate error of luenberger observer
Estimate error of kalman Filter
Estimate error of velocity [m/s]
time [s]
8
6
4
2
0
-2
-4
0
2
4
6
8
10