Multivariable Control (ME-422) - Exercise session 14 SOLUTIONS

Prof. G. Ferrari Trecate

1. Consider the nonlinear system

$$\begin{aligned} x_{k+1} &= \alpha x_k + \cos(x_k) + w_k & w_k &\sim WGN(0,1) \\ y_k &= x_k^2 + v_k & v_k &\sim WGN(0,0.5) \end{aligned}$$

Determine the Extended Kalman Filter (EKF).

Solution: The relevant matrices are

$$\begin{aligned} \hat{A}_k &= \left.\frac{\partial\left(\alpha x_k + \cos(x_k)\right)}{\partial x_k}\right|_{x_k=\hat{x}_{k-1|k-1}} = \alpha - \sin(\hat{x}_{k-1|k-1}) \\ \hat{C}_k &= \left.\frac{\partial\left(x_k^2\right)}{\partial x_k}\right|_{x_k=\hat{x}_{k|k-1}} = 2\hat{x}_{k|k-1}. \end{aligned}$$

Using the formulae seen in the lecture for EKF:

- Filtering step:

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + \bar{L}_{k|k}\left(y_k - \hat{x}_{k|k-1}^2\right) \\ \bar{L}_{k|k} &= \bar{\Sigma}_{k|k-1}\left(2\hat{x}_{k|k-1}\right)\left(\left(2\hat{x}_{k|k-1}\right)^2\bar{\Sigma}_{k|k-1} + 0.5\right)^{-1} \\ \bar{\Sigma}_{k|k} &= \bar{\Sigma}_{k|k-1} - \bar{\Sigma}_{k|k-1}^2\left(2\hat{x}_{k|k-1}\right)^2\left(\left(2\hat{x}_{k|k-1}\right)^2\bar{\Sigma}_{k|k-1} + 0.5\right)^{-1} \end{aligned}$$

- Prediction step:

$$\begin{aligned} \hat{x}_{k+1|k} &= \alpha\hat{x}_{k|k} + \cos\left(\hat{x}_{k|k}\right) \\ \bar{\Sigma}_{k+1|k} &= \left(\alpha - \sin\left(\hat{x}_{k-1|k-1}\right)\right)^2\bar{\Sigma}_{k|k} + 1 \end{aligned}$$

2. Consider the system

$$x_{t+1} = Ax_t + Bu_t\,, \tag{1}$$

with

$$A = \begin{bmatrix} 0.5 & -2 \\ 6 & -6 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where $x_0 \sim \mathcal{N}(0, \Sigma_0)$ is distributed according to a Gaussian distribution with covariance matrix $\Sigma_0 = I$.

Since the system is open-loop unstable, your colleague selects a stabilizing control law

$$u_t = -K_0 x_t\,,$$

that sets the eigenvalues of $(A - BK_0)$ equal to $(0, 0.5)$. Letting

$$J(K) = \mathbb{E}_{x_0}\left[\sum_{t=0}^{\infty} x_t^\intercal x_t + u_t^\intercal u_t\right], \tag{2}$$

denote the average IH cost achieved by a control law $u_t = -Kx_t$, determine the *Performance Improvement* (PI)

$$\text{PI} = 100\left(1 - \frac{J(K^\star)}{J(K_0)}\right)\%\,, \tag{3}$$

that can be achieved by replacing K_0 with the optimal S-LQR controller $K^\star$.

Hint: You can use the MATLAB functions `dlyap` and `idare` appropriately.

Solution: To compute the controller K_0 that sets eigenvalues to $(0, 0.5)$, we let

$$K_0 = \begin{bmatrix} x & y \end{bmatrix}\,,$$

so that

$$\det(\lambda I - A + BK_0) = \lambda^2 + \lambda(5 + y) + 6 - 2x - y\,.$$

To impose $\lambda = 0$ is a root of the characteristic polynomial above, we set

$$2x + y = 6\,.$$

Then, to impose that $\lambda = 0.5$ is also a root, we set

$$y + 5 = -0.5 \to y = -5.5\,,$$

which implies $x = 5.75$. Hence, we have $K_0 = \begin{bmatrix} 5.75 & -5.5 \end{bmatrix}$.

To compute the average IH cost achieved by K_0, we need to solve the Lyapunov equation

$$P_{K_0} = I + K_0^\mathsf{T} K_0 + (A - BK_0)^\mathsf{T} P_{K_0}(A - BK_0)\,.$$

By using the MATLAB command

```
dlyap((A-B*K0)', eye(2)+K0'*K0);
```

we obtain that $P_{K_0} = \begin{bmatrix} 61 & -85.5 \\ -85.5 & 139 \end{bmatrix}$, and hence

$$J(K_0) = \text{Trace}(P_{K_0}) = 200\,.$$

Last, we compute the minimum cost achievable with the optimal S-LQR controller $K^\star$. We have that

$$J(K^\star) = \text{Trace}(P^\star)\,,$$

where

$$P^\star = A^\mathsf{T} P^\star A - A^\mathsf{T} P^\star B(B^\mathsf{T} P^\star B + I)^{-1} B^\mathsf{T} P^\star A + I\,.$$

In MATLAB, $P^\star$ is computed by running

```
idare(A,B,eye(2),eye(1),zeros(2,1),eye(2));
```

We obtain that $J(K^\star) = 81.8442$. Hence, we conclude that the Performance Improvement obtained by switching to an LQR controller is

$$PI = 100\left(1 - \frac{81.8442}{200}\right)\% = 59\%.$$

Quite a large improvement! Your colleague should really go for the LQR controller.

3. In this exercise, you will implement the Gradient Descent (GD) algorithm to solve S-LQR. You are given a system (1) with

$$A = 0.1\begin{bmatrix} -1 & 1 & -2 \\ -3 & 3.2 & 3.5 \\ -5 & 6 & -7 \end{bmatrix}\,, \quad B = I\,,$$

where $x_0 \sim \mathcal{N}(0, \Sigma_0)$ and $\Sigma_0 = I$. The average IH cost is defined as (2) as per Exercise 2.

Your task is to implement the GD algorithm on MATLAB to compute the optimal LQR controller $K^\star$.

Task 1 Complete the passages in the pseudocode implementation of GD

```
K0 = ???    (explain how to select an initial controller)


eta = ???    (explain, in words, how you would select a stepsize eta)


K = K0
while( ???                          )  (Insert a condition for stopping the algorithm)
    P_K = ???       (How to compute P_K?)


    Sigma_K = ???   (How to compute Sigma_K?)


    DeltaJ = ???    (How to compute the gradient DeltaJ)


    K = ???         (How to compute an updated value of K?)


end
```

Task 2 Complete the file "Ex14_3.m" to implement the GD algorithm. Verify that the GD converges to the optimal LQR controller.

Solution Task 1 The pseudocode is given by

```
K0 = ???
ANSWER: K0 such that (A-B*K0) is Hurwitz. Since A is already Hurwitz, K0 = 0 works.

eta = ???
ANSWER: The stepsize eta should be small enough such that,
for all K, J(K-eta*DeltaK) <= J(K).In practice, we reduce
eta until we observe convergence.

K = K0
while(  ANSWER: ||DeltaJ||<=0.00001 )
    P_K =         ANSWER: dlyap((A-B*K)',Q+K'*R*K)
    Sigma_K =     ANSWER: dlyap(A-B*K,Sigma_0);
    DeltaJ =      ANSWER: 2*((R+B'*P*B)*K-B'*P*A)*Sigma_K;
    K =           ANSWER: K-eta * DeltaJ;
end
```

Solution Task 2

```
%% SYSTEM DEFINITION
A=-0.1*[1 -1 2;3 -3.2 -3.5;5 -6 7];
B=eye(3);
n = size(A,1);
m = size(B,2);
Sigma_0 = eye(n);
```

```
disp('We consider the S-LQR problem with A, B, Sigma_0 given by')
A
B
Sigma_0
disp('and cost matrices Q and R given by')
%Weight matrices for the cost
Q = eye(n)
R = eye(m)

fprintf('\n\n')
disp('We start with a  controller K0 ')
K0 = zeros(3,3)
disp('that is stabilizing and has a cost')
cost_K0 = trace(dlyap((A-B*K0)',Q+K0'*R*K0)*Sigma_0)


eta=0.0005; %step-size
norm_gradient = 9999; %stopping criterion
count=0;
fprintf('***********+\n**********\n Start GD\n **********\n********\n')

Sigma=zeros(n,n);
K = K0;
while(norm_gradient>0.000001)
    count = count+1;
    P = dlyap((A-B*K)',Q+K'*R*K);
    Sigma = dlyap(A-B*K,Sigma_0);
    DeltaJ = 2*((R+B'*P*B)*K-B'*P*A)*Sigma;
    norm_gradient = norm(DeltaJ);
    if(mod(count,100)==0)
        cost_K = trace(P*Sigma_0);
        fprintf('Iteration: %d, Cost: %f\n\n\n', count, cost_K)
    end
    K = K-eta * DeltaJ;
end

%% LQR Controller comparison
P = idare(A,B,Q,R,zeros(n,m),eye(n));
K_LQR = inv(B'*P*B+R)*B'*P*A
cost_LQR = trace(P * Sigma_0)
disp('The controller found with gradient descent is')
K
disp('The LQR controller is')
K_LQR
```

4. In this exercise you will use projected GD to compute locally optimal distributed controllers. You are given a system (1) with

$$A = 0.2 \begin{bmatrix} -1 & 1 & -2 \\ -3 & 3.2 & 3.5 \\ -5 & 6 & -7 \end{bmatrix}, \quad B = I, \tag{4}$$

where $x_0 \sim \mathcal{N}(0, \Sigma_0)$ and $\Sigma_0 = I$. The average IH cost is defined as (2) as per Exercise 2 and Exercise 3. Notice that the definition of A has changed with respect to Exercise 3.

Task 1 You work for the company *ControlX* who is using a distributed controller

$$K_0 = \begin{bmatrix} -1.1189 & 0 & 0 \\ -0.6894 & 0 & -1.9114 \\ 0 & 1.7018 & 0 \end{bmatrix},$$

for their plant whose state-space model is given by (4). Letting $S = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, find a locally optimal controller $\hat{K} \in \text{Sparse}(S)$ using PGD starting from K_0. What PI can you achieve (see definition of PI in (3))?

Task 2 Your locally optimal controller from Task 1 is quite successful and improves the company's profits by $\sim 86\%$. Congratulations, you got a promotion!

A few months later your main competitor on the market, *iControl*, deploys a novel distributed controller $K_{\texttt{iControl}} \in \text{Sparse}(S)$ that significantly improves over your solution. How is it possible?

Can you find a controller in the sparsity subspace $\text{Sparse}(S)$ that improves over $\hat{K}$? How much does it improve?

Solution Task 1 We apply projected GD to improve over K_0 until convergence. We modify the centralized GD algorithm as follows

```
S = [1 0 0;1 0 1;0 1 0];
eta=0.000005;
while(norm_gradient>0.000001)
    count = count+1;
    P = dlyap((A-B*K)',Q+K'*R*K);
    Sigma = dlyap(A-B*K,Sigma_0);
    DeltaJ = 2*((R+B'*P*B)*K-B'*P*A)*Sigma;

    DeltaJ = DeltaJ .* S;

    norm_gradient = norm(DeltaJ);
    if(mod(count,100)==0)
        cost_K = trace(P*Sigma_0);
        fprintf('Iteration: %d, Cost: %f\n\n\n', count, cost_K)
    end
    K = K-eta * DeltaJ;
end
```

We converge to a controller

$$\hat{K} = \begin{bmatrix} -0.7597 & 0 & 0 \\ -0.8956 & 0 & -0.2621 \\ 0 & 2.7016 & 0 \end{bmatrix}.$$

We verify that $J(K_0) = 465.5270$ and $J(\hat{K}) = 63.077438$, yielding a PI of

$$PI = 100\left(1 - \frac{63.077438}{465.5270}\right)\% = 86.45\%\,.$$

You deserve a promotion!

Solution Task 2 Yes, it is possible that *iControl* comes up with a better distributed controller. Indeed, $\hat{K}$ may only be a local minimum of $J(\cdot)$ when imposing sparsity constraints.

In order to find a better controller than $\hat{K}$, we should find a new initial controller K_0 that lies in a different region closer to a better local minimum. For instance, by starting from

$$K_0' = \begin{bmatrix} -0.0900 & 0 & 0 \\ -1.2633 & 0 & 3.3683 \\ 0 & 0.7425 & 0 \end{bmatrix},$$

we converge to

$$\hat{K}' = \begin{bmatrix} -0.5843 & 0 & 0 \\ -0.0446 & 0 & 1.5751 \\ 0 & -0.2419 & 0 \end{bmatrix},$$

that yields a cost of $J(\hat{K}) = 43.0588$.

The controller K_0' was found by brute force by running

```
done=0;
while(done==0)
    done=1;
    K=500*(rand-rand)*(rand(m,n)-rand(m,n));
    K = K.*S
    eigs=eig(A-B*K);
    for(i=1:size(eigs,1))
        eigs(i)=norm(eigs(i));
        if(eigs(i)>0.90)
            done=0;
        end
    end
end
```

which looks for a stabilizing controller within the given sparsity structure. Gradient descent then allows to converge to the closest stationary point.