۱۳۰۷

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده مهندسي برق و کامپیوتر

FINAL PROJECT REPORT

LARGE SCALE SYSTEMS

گزارش پروژه نهایی درس سیستم های ابعاد بزرگ

نگارش:

مینو جهان سیر

محمد علی‌خانی

۴۰۱۰۹۴۹۴

استاد درس:

دکتر بیژن معاونی

بهار ۱٤۰۲

بسم الله الرحمن الرحیم

Abstract

Distributed systems have become an essential component of modern traffic control systems due to their ability to handle large amounts of data and provide real-time responses. In this paper, we explore the use of distributed systems in controlling traffic flow, focusing on the challenges and benefits of implementing such a system. After that we optimize a specific traffic plant. In section 2 the MATLAB simulation has been demonstrated.

Table Of Content

Title **Page**

Table of Figures

Page **Title**

Introduction

DSC vs DES

In today's world, the need for efficient and effective control systems has become increasingly important. One such system that has gained popularity in recent years is the Distributed Control System (DCS). A DCS is a computerized control system used to manage and monitor complex industrial processes. It consists of a network of controllers, sensors, and actuators that work together to control various aspects of the process.

While DCS and Decentralized Control Systems (DEC) may seem similar, there are some key differences between the two. In a decentralized system, each controller operates independently and communicates with other controllers as needed. On the other hand, in a distributed system, all controllers are connected through a central network and communicate with each other constantly.

The difference between these two systems can be represented mathematically using the following formula:

DCS = Centralized Control + Distributed Intelligence

DEC = Decentralized Control + Local Intelligence

In large scale control systems, choosing between DCS and DEC can have significant impacts on efficiency and effectiveness. Understanding these differences is crucial for selecting the right system for your specific needs.

Proposed traffic system

Consider the below system:

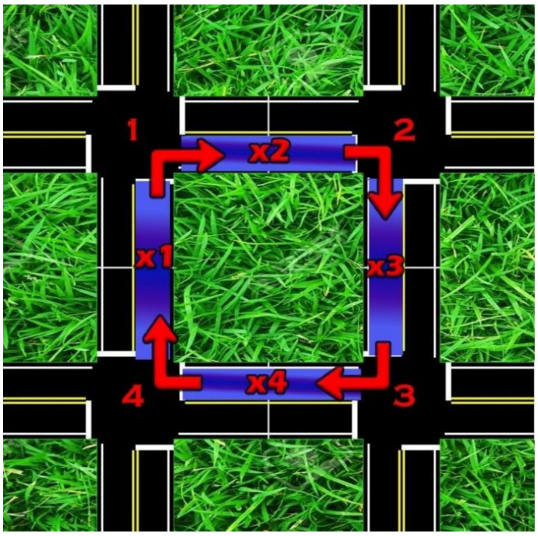

Figure 1 the schematic of the proposed problem

As can be seen in the above figure there are four intersections which are connected successively:

- Intersection #1 is influenced by Intersection #4

- Intersection #2 is influenced by Intersection #1
- Intersection #3 is influenced by Intersection #2
- Intersection #4 is influenced by Intersection #1

The below picture demonstrates the schematic of the distributed control system is chosen for the control method.

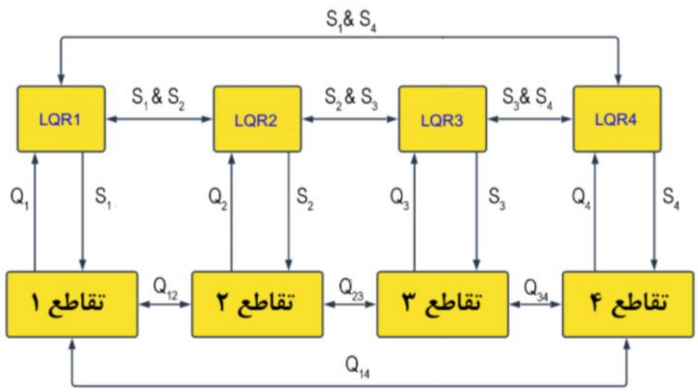

Figure 2 DSC method for this problem

So, each intersection has two cost function; first the one which the cost is influenced by itself and second is influenced by others.

Two control signals have been determined according to our cost functions for each intersection. Our goal is finding the best control signals that minimize the total cost function:

$$J_{total} = J_1 + J_2 + J_3 + J_4$$

The overall algorithm is shown in below:

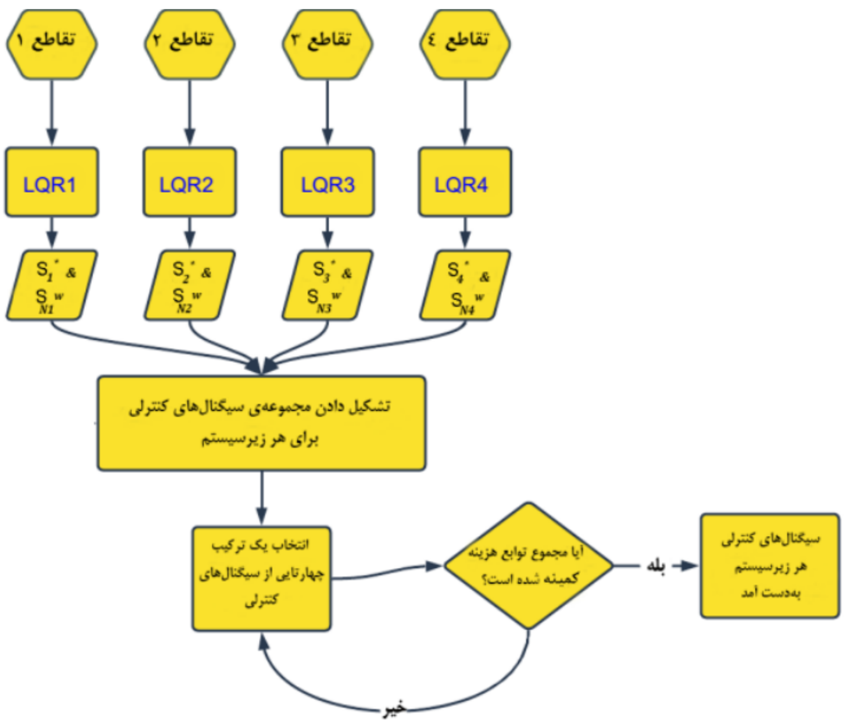

Figure 3 Algorithm of solution for optimum pairs

In this algorithm we should find the best pair of control signals that can minimize the cost function.

The state spaces of intersections are:

$A_1 = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}, Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_1 = 1$

$A_2 = \begin{bmatrix} -5.5 & -3 \\ 2 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, R_2 = 0.5$

$A_3 = \begin{bmatrix} -4 & -1.75 \\ 1 & 0 \end{bmatrix}, B_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, Q_3 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, R_3 = 1$

$A_4 = \begin{bmatrix} -4 & -2 \\ 2 & 0 \end{bmatrix}, B_4 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, Q_4 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, R_4 = 2$

All subsystems have the same initial value of 1.5. The LQRs are designed discrete for 10 timesteps.

Simulation of The Proposed Traffic Plant

Simulation in MATLAB

In this section we simulate the proposed system in MATLAB. After that, we plot the result that contains the most optimal set of controllers:

First, we define state-space matrices, which are A and B and the relevant LQR matrices, which are Q and R:

```
A1 = [-3 -2;1 0];
A2 = [-5.5 -3;2 0];
A3 = [-4 -1.75;1 0];
A4 = [-4 -2;2 0];

B1 = [4;0];
B2 = [2;0];
B3 = [1;0];
B4 = [2;0];

R1 = 1;
R2 = 0.5;
R3 = 1;
R4 = 2;

Q1 = [1 0;0 1];
Q2 = [1 0;0 2];
Q3 = [2 0;0 1];
Q4 = [2 0;0 2];
```

Then, we compute LQR gains of systems. (ks represents the gain which is influenced by itself and kw represents the gain which is influence by other subsystems.)

```
k1s = dlqr(A1,B1,Q1,R1);
k2s = dlqr(A2,B2,Q2,R2);
k3s = dlqr(A3,B3,Q3,R3);
k4s = dlqr(A4,B4,Q4,R4);

k1w = dlqr(A1,B1,Q4,R4);
k2w = dlqr(A2,B2,Q1,R1);
k3w = dlqr(A3,B3,Q2,R2);
k4w = dlqr(A4,B4,Q3,R3);
```

After that the control signals and states are computed by below function for 10 timesteps: (This function is attached to the final project folder and it’s name is controlsignalState.)

```
function [u,x] = controlsignalState(A,B,k,samples)
%CONTROLSIGNALSTATE Summary of this function goes here
%   Detailed explanation goes here
x0 =  [1.5;1.5];
x = zeros(2,10);
x(:,1) = x0;
u = zeros(1,10);
u(1,1) = -k*x(:,1);
for i = 1:samples - 1
    x(:,i+1) = A*x(:,i) + B*(-k)*x(:,i);
    u(i+1) = -k*x(:,i+1);
end
end
```

About code snippet up here, we can say it computes state variables x1 and x2 and the control signal u, in 10 iterations. Inputs to this function are matrices A, B and LQR gains, and the outputs of this function are states and control signal.

We write all pairs of control signal and states which are affected by itself:

```
[u1s,x1s] = controlsignalState(A1,B1,k1s,samples);
[u2s,x2s] = controlsignalState(A2,B2,k2s,samples);
[u3s,x3s] = controlsignalState(A3,B3,k3s,samples);
[u4s,x4s] = controlsignalState(A4,B4,k4s,samples);
```

And all pairs of control signals and states which are affected by previous subsystems(in a cascade sence):

```
[u1w,x1w] = controlsignalState(A1,B1,k1w,samples);
[u2w,x2w] = controlsignalState(A2,B2,k2w,samples);
[u3w,x3w] = controlsignalState(A3,B3,k3w,samples);
[u4w,x4w] = controlsignalState(A4,B4,k4w,samples);
```

Now by these lines of code we store all control signals and all states for each subsystem no matter centralized or distributed:

```
U1 = [u1s;u1w];
U2 = [u2s;u2w];
U3 = [u3s;u3w];
U4 = [u4s;u4w];

X1 = [x1s;x1w];
X2 = [x2s;x2w];
X3 = [x3s;x3w];
X4 = [x4s;x4w];
```

By “allcomb” function, we determine indices which contain all possible sets of control signals and states automatically, and the results are stored in combinations variable which is a 16x4 matrix:

```
combinations = allcomb([1     2],[1     2],[1     2],[1     2]);
```

These new variables U_i are here to aid us compute the cost:

```
U_1 = [];
U_2 = [];
U_3 = [];
U_4 = [];
```

After that, we write two for loops to compute all possible cases of LQR costs.
And about the code, note snippet computes cost for every 16 sets of controllers, and in the end determines which one has the least cost:
We loop over all 16 possible cases, and then for each case we put the control signal and the states into U_i and X_i.
Since the elements of combinations matrix are either 1 or 2 we can put this property into use and associate the controllers with their states and control signals.

```
for k = 1:length(combinations)
   U_1 = U1(combinations(k, 1), :);
   U_2 = U2(combinations(k, 2), :);
   U_3 = U3(combinations(k, 3), :);
   U_4 = U4(combinations(k, 4), :);
```

```
    X_1 = X1(2 * combinations(k, 1) - 1:2 * combinations(k, 1),:);
    X_2 = X2(2 * combinations(k, 2) - 1:2 * combinations(k, 2),:);
    X_3 = X3(2 * combinations(k, 3) - 1:2 * combinations(k, 3),:);
    X_4 = X4(2 * combinations(k, 4) - 1:2 * combinations(k, 4),:);

    for j = 1:length(u1s)
        Sum = (X_1(:,j)'*Q1*X_1(:,j) + U_1(:,j)'*R1*U_1(:,j))+...
              (X_2(:,j)'*Q2*X_2(:,j) + U_2(:,j)'*R2*U_2(:,j))+...
              (X_3(:,j)'*Q3*X_3(:,j) + U_3(:,j)'*R3*U_3(:,j))+...
              (X_4(:,j)'*Q4*X_4(:,j) + U_4(:,j)'*R4*U_4(:,j));
        J = Sum + J;
    end
    JJ(k,1) = J;
end
```

We are going to have a little demonstration of what we said above:
Grant k = 1, corresponding to the first row of the combinations, which give us a vector of [1 1 1 1], for the controllers, remember from definition 1 is for decentralized controller and 2 is for distributed controller.
For X_1, X_2, X_3 and X_4 we will have:

```
    X_1 = X1(2 * 1 – 1 : 2 * 1,:);
    X_2 = X2(2 * 1 – 1 : 2 * 1,:);
    X_3 = X3(2 * 1 – 1 : 2 * 1,:);
    X_4 = X4(2 * 1 – 1 : 2 * 1,:);
```

Then we have:

```
    X_1 = X1(1 : 2 ,:);
    X_2 = X2(1 : 2 ,:);
    X_3 = X3(1 : 2 ,:);
    X_4 = X4(1 : 2 ,:);
```

If you remember from X1 = [x1s;x1w], x1s and x1w are 2*10 matrices, and the first 2 lines are for x1s.
And so is applicable to U_1.

```
    U_1 = U1(combinations(k, 1), :);
    U_2 = U2(combinations(k, 2), :);
    U_3 = U3(combinations(k, 3), :);
    U_4 = U4(combinations(k, 4), :);
```

If k = 1, we have these matrices below:

```
    U_1 = U1(combinations(1, 1), :);
    U_2 = U2(combinations(1, 2), :);
    U_3 = U3(combinations(1, 3), :);
    U_4 = U4(combinations(1, 4), :);
```

And cause all combinations elements on the first row are 1, we get the below results:

```
    U_1 = U1(1 , :);
    U_2 = U2(1 , :);
    U_3 = U3(1 , :);
    U_4 = U4(1 , :);
```

U1, U2, U3 and U4 are 1x10 matrices, so the results for U_1, U_2, U_3 and U_4 will be the first row and all columns of their corresponding Ui matrices.
This was small example, for the other 15 cases we can show that, this holds true.
The below code will compute the cost and store it in variable J:

```
for j = 1:length(u1s)
        Sum = (X_1(:,j)'*Q1*X_1(:,j) + U_1(:,j)'*R1*U_1(:,j))+...
              (X_2(:,j)'*Q2*X_2(:,j) + U_2(:,j)'*R2*U_2(:,j))+...
              (X_3(:,j)'*Q3*X_3(:,j) + U_3(:,j)'*R3*U_3(:,j))+...
              (X_4(:,j)'*Q4*X_4(:,j) + U_4(:,j)'*R4*U_4(:,j));
        J = Sum + J;
end
```

And at the end of the outer for loop we put the cost into a 1x16 matrix named JJ.

```
JJ(k,1) = J;
```

Now all we have to do is determine what combination of controllers has the optimal cost.
By this line we carry out this task:

```
[minimum, index] = min(JJ);
```

What does this line do is `min` function of matlab gets a matrix as input and finds the minimum and returns this minimum with its corresponding index.
After running the code, we get this result, the optimal case will be the centralized controller.

```
minimum =

  208.5996

index =

     1
```

And finally, we plot the results:

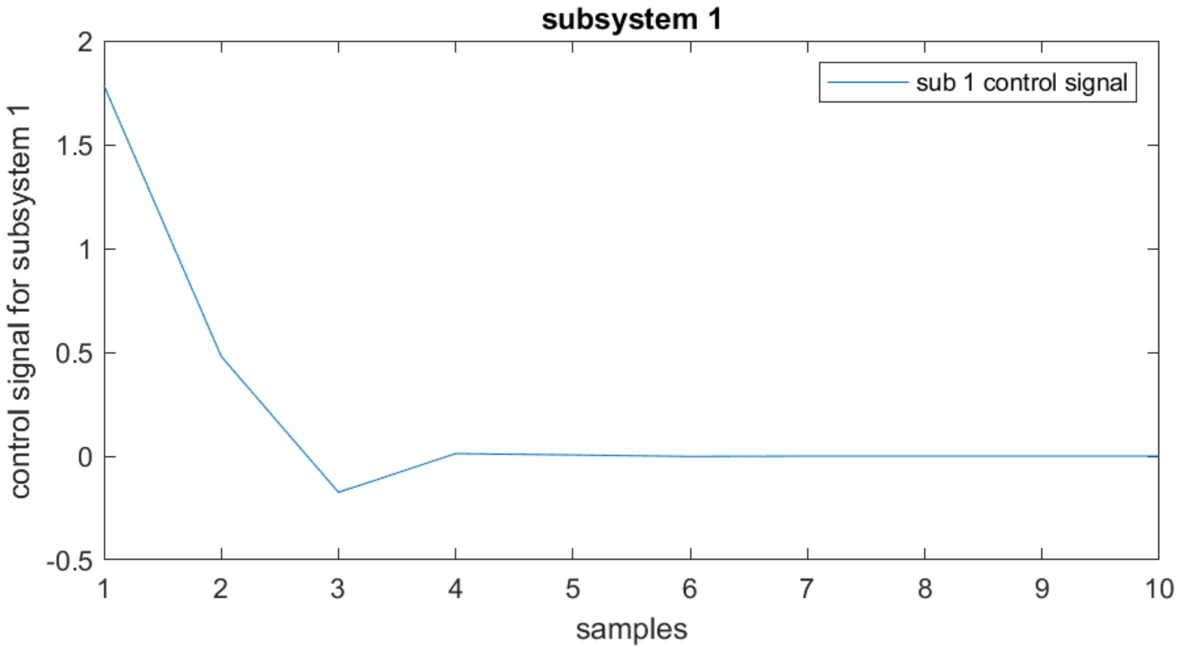

Figure 4 optimum control signal for subsystem #1

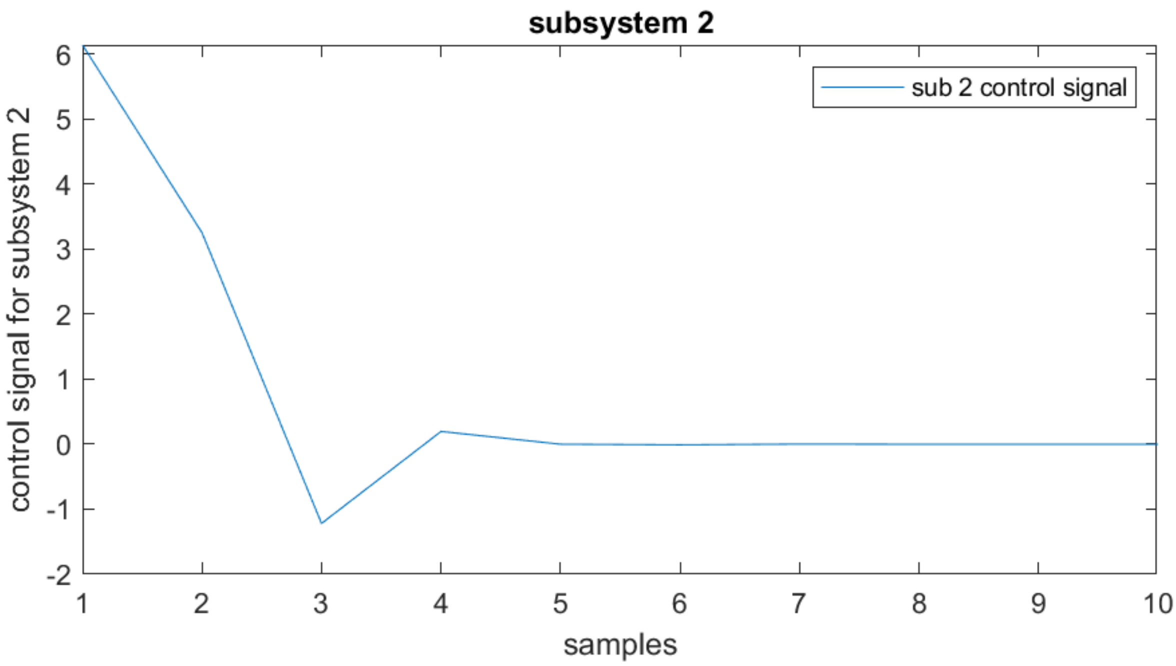

Figure۵ optimum control signal for subsystem #2

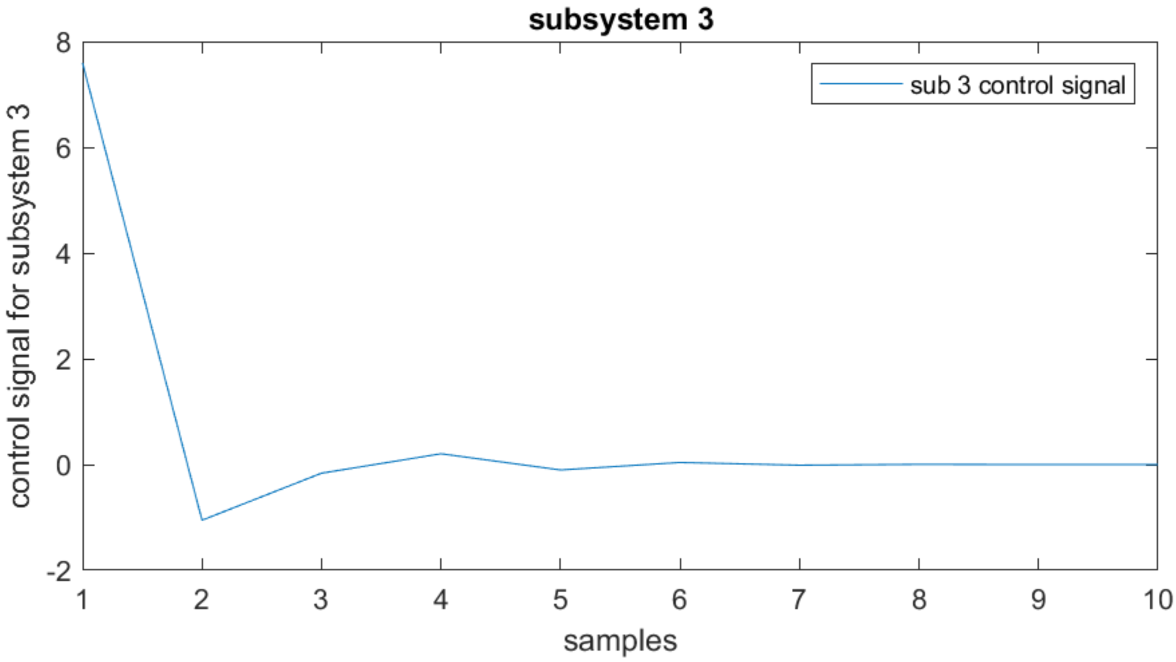

Figure ۶ optimum control signal for system #3

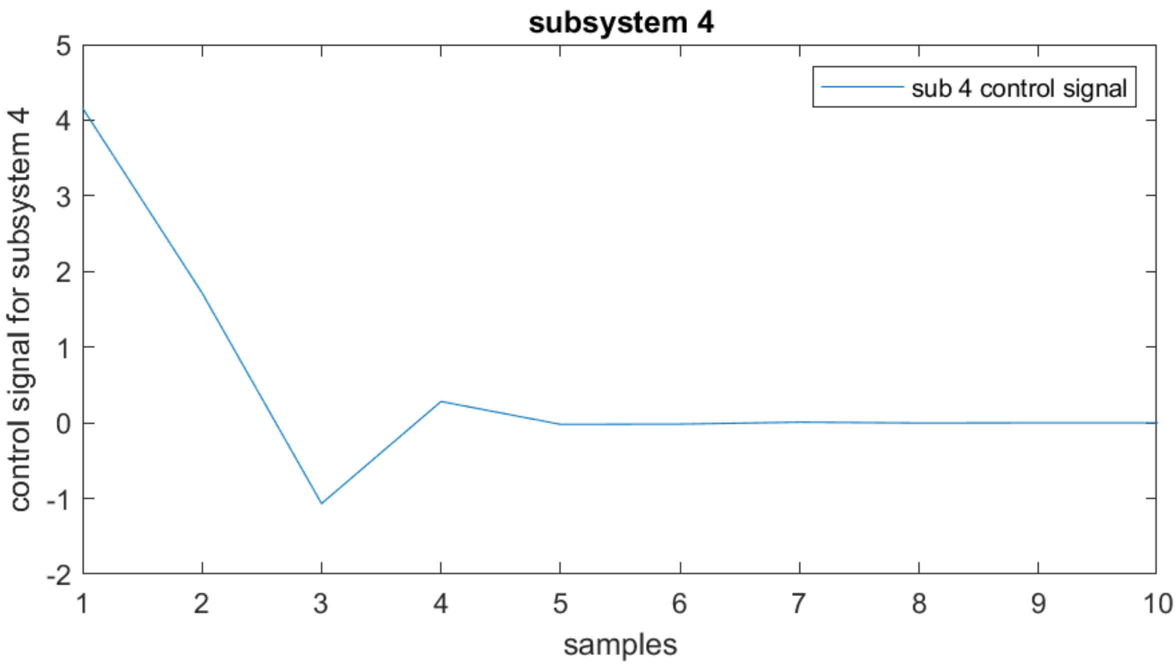

Figure 7 optimum control signal for system #4

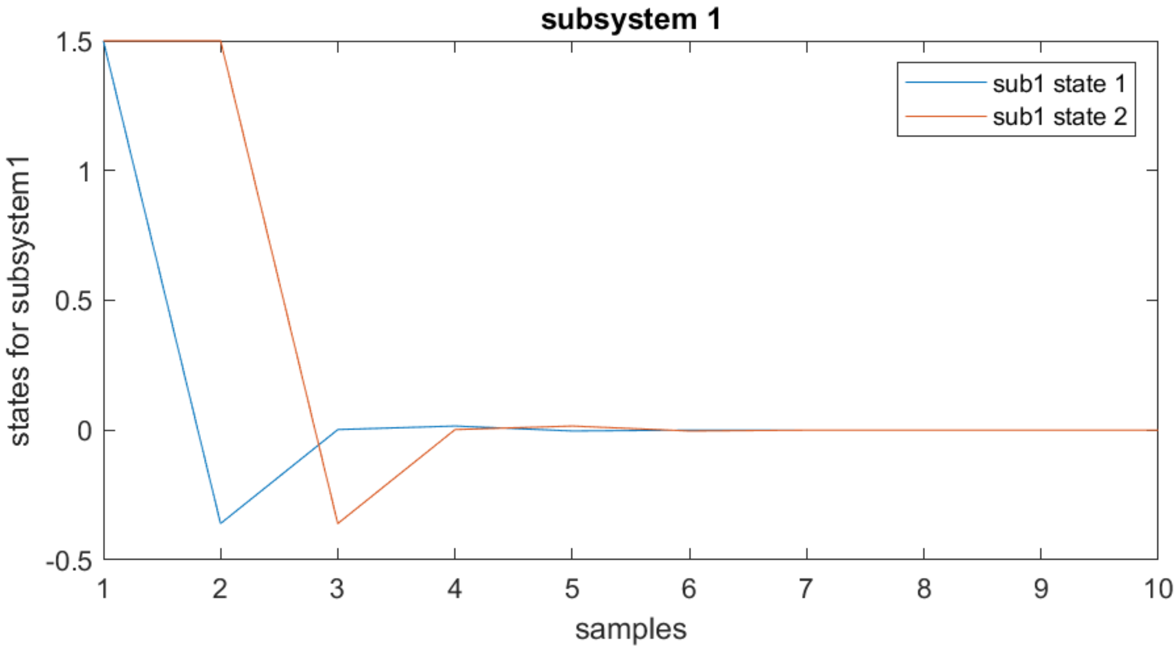

Figure٨ subsystem #1 states

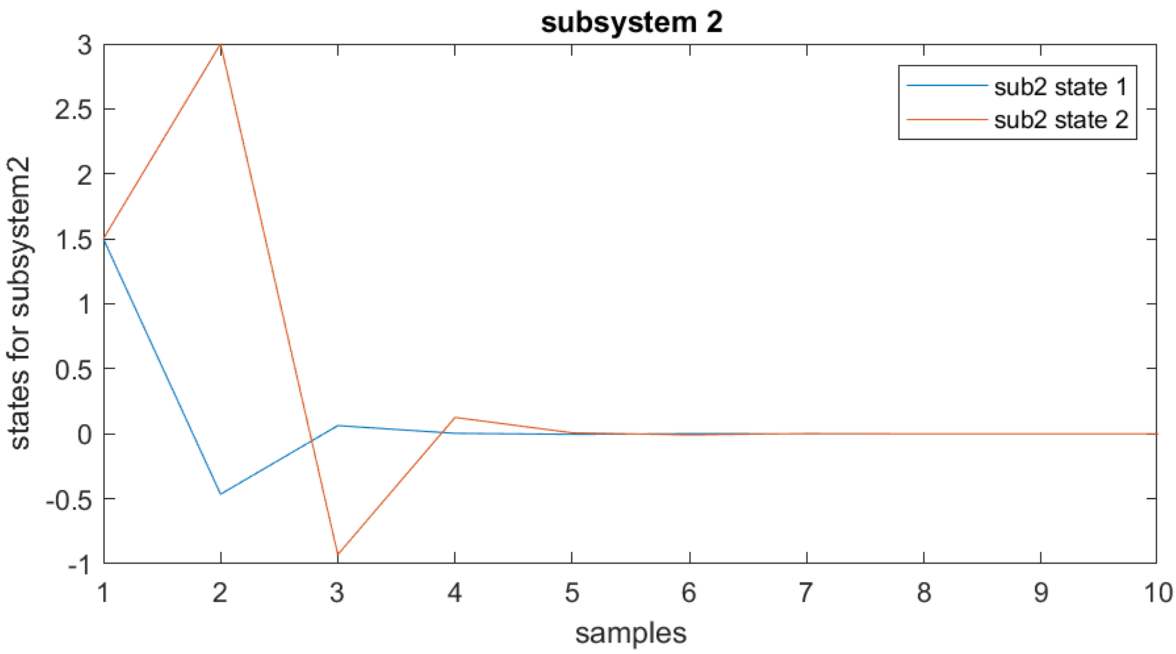

Figure٩ subsystem #2 states

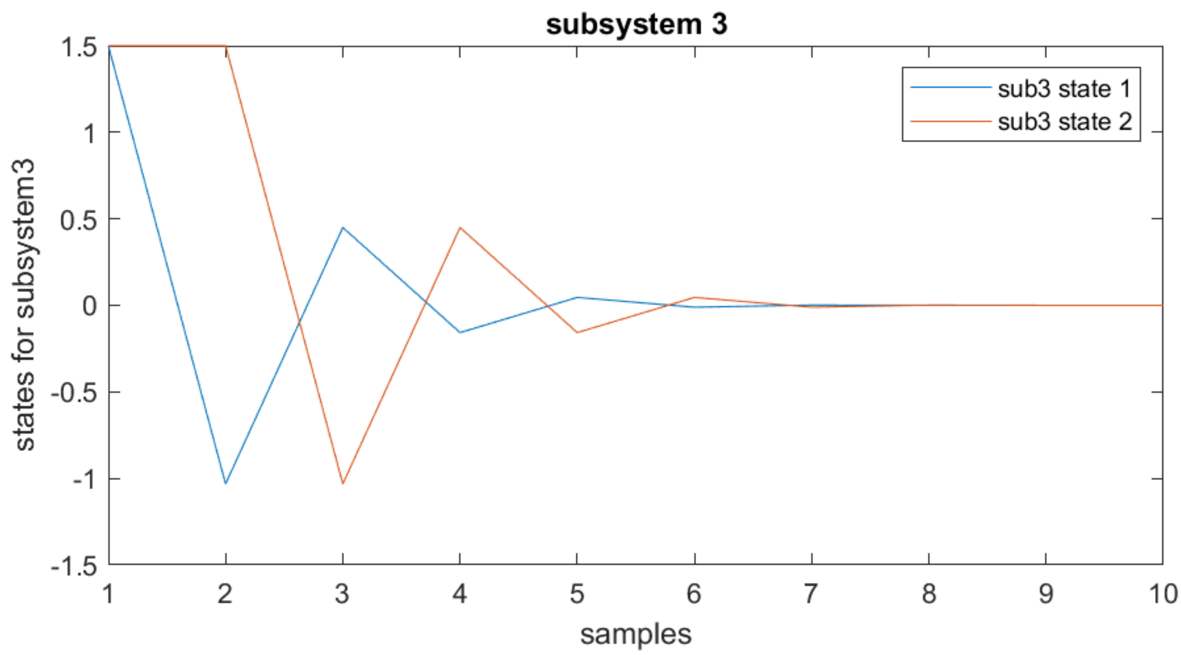

Figure ١٠ subsystem #3 states

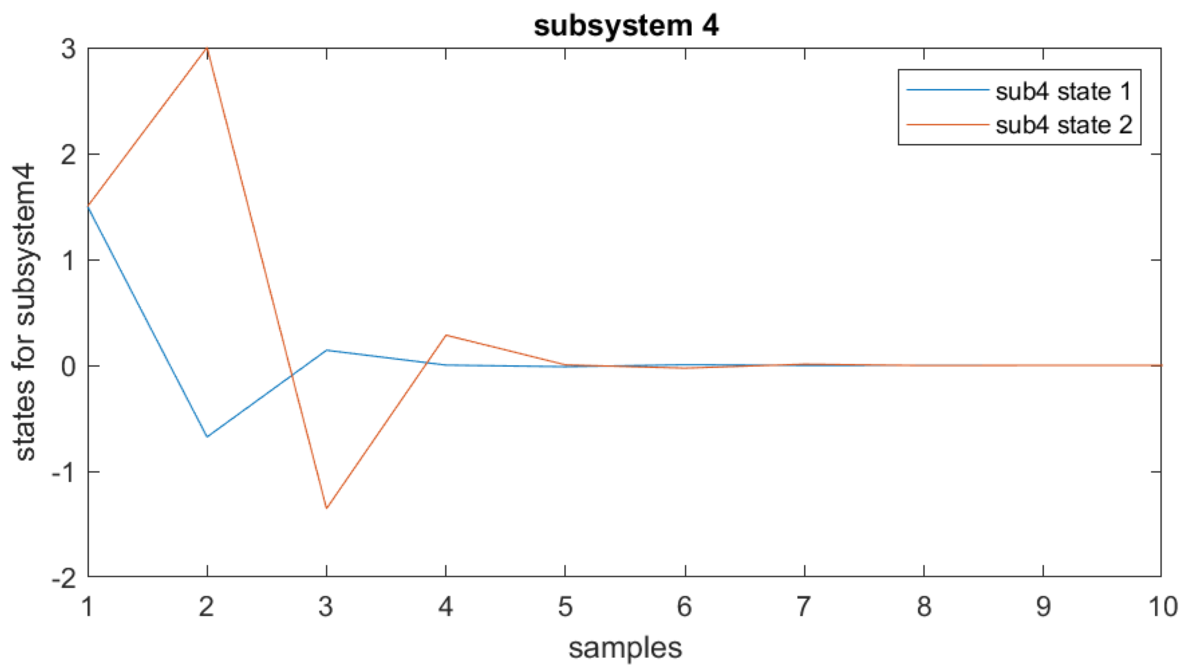

Figure ١١ subsystem #4 states

Conclusion

Conclusion

The cost of a centralized controller versus a decentralized controller can vary depending on several factors, including the specific context, requirements, and implementation details. It's challenging to definitively state which one has the least cost in all situations, as cost considerations can be complex and multifaceted.
That being said, here are some general points to consider:

Centralized Controller:

1. Cost advantages: Centralized controllers can potentially benefit from economies of scale as they consolidate resources and operations. They may require fewer hardware components and infrastructure, leading to lower upfront costs.
2. Maintenance and upgrades: With a centralized controller, maintenance and upgrades typically need to be performed on a single system. This centralized approach may simplify the management and reduce ongoing costs compared to maintaining multiple decentralized controllers.
3. Scalability: Depending on the scale and growth of the system, centralized controllers may be more cost-effective to scale up since the focus is on a single controller rather than multiple decentralized ones.
4. Single point of failure: However, it's important to note that a centralized controller represents a single point of failure. If the centralized controller fails, it can disrupt the entire system, leading to potential downtime and higher costs for recovery.

Decentralized Controller:

1. Redundancy and fault tolerance: Decentralized controllers offer redundancy and fault tolerance since multiple controllers operate independently. If one controller fails, the others can continue to function, minimizing downtime and associated costs.
2. Distribution of resources: In certain scenarios, a decentralized controller architecture may allow for better utilization of resources by distributing control closer to the systems being managed. This can potentially lead to increased efficiency and cost savings in terms of hardware and infrastructure.
3. Initial setup and complexity: Implementing a decentralized controller architecture can require additional setup and configuration compared to a centralized system. This may introduce higher upfront costs, particularly when considering the need for multiple controllers and associated networking components.

Ultimately, the cost-effectiveness of a centralized or decentralized controller depends on the specific requirements, system complexity, scalability needs, fault tolerance considerations, and other factors unique to the application at hand. Evaluating these factors and conducting a thorough cost analysis based on your specific context would be necessary to determine which approach is the least costly in your particular situation.

In this specific case the cost for centralized controller is lowest of all possible combination of the controllers, which we can associate with the specific problem we are trying to solve, which has no model uncertainty could be a reason. The minimum value for J(cost) is 208.5996 and the Index is #1. This means that the optimal set happens when every subsystem is influenced by itself. The thing is, when there is no coupling and inference between subsystems it is better to design isolated controllers. In designing distributed, there are some uncertainties, and the Q Matrix may vary. So,

Conclusion

most control signals are computed by Optimizer functions like “Fmincon”. But in this example, there is no uncertainty; we use LQR directly.

In this method, we have one control system. If our desire is to design a decentralized controller, we have to design four controllers.

References

[1] Dr B. Moaveni Lecture Notes.
[2] Dr A. Khakisedigh Lecture Notes.

References

K. N. Toosi University of Technology

Faculty of Electrical and Computer Engineering

[Large Scale Final Project Report]

By:

[Mohammad Alikhani]

40109494

[Minoo JahanSeir]

40104424

Course Instructor:

Dr. Moaveni

Spring 2023