Assignment 2

Data Intensive Control Systems

Yiting Li (1567713)

April 3, 2022

Course:	Embedded Control Systems
Professor:	Prof. Dip Goswami
Date:	April 3, 2022

1 Time Analysis & Implementation and Result for Each Cases

1.1 Case 1

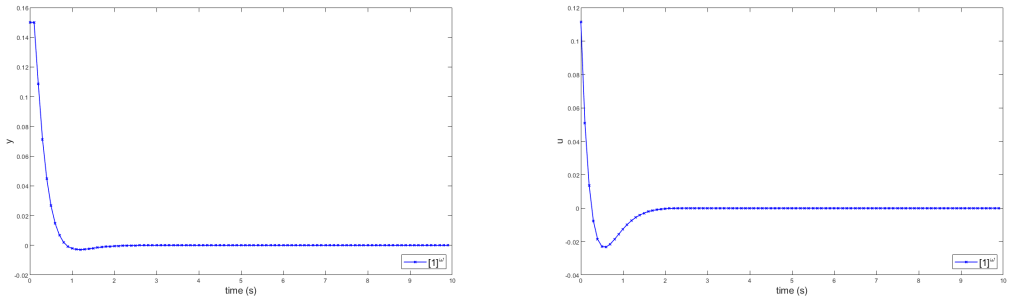
Firstly, time analysis is performed to get the sampling time and time delay, $h = 0.1s$ and $\tau = 0.097s$in this case.

$$\begin{aligned} \tau &= t_{isp1} + \frac{ROI * t_{isp2}}{parallel} + t_{isp3} + t_{RoID} + \frac{ROI * t_{RoIP}}{parallel} + t_{RoIM} + t_C + t_A \\ h &= fh * \lceil (\frac{\tau}{fh * n_{pipeline}}) \rceil, \quad fh = 0.1(s) \end{aligned} \tag{1}$$

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ -0.9843 & 0.1755 & -0.0107 & 0.0134 & 0 & 0 \\ 0.0188 & 0.0083 & -0.6979 & 0.7159 & 0 & 0 \\ -0.0206 & -0.1061 & -0.7127 & -0.6931 & 0 & 0 \\ 0.1742 & 0.9787 & -0.0694 & -0.0836 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$
$$K = \begin{bmatrix} 0.0083 & -0.0872 & -0.9460 & -0.1177 & -0.5438 \end{bmatrix} \tag{2}$$

Plot the system input and output(under a positive deviation, bias is 0.15), the MiL simulation result shows, in the figure 1, that the system settling time is around 0.8.



(a) Input u Performance (b) Output y Performance

Figure 1: Case 1 MIL Controller Performance

In the SiL implementation, we enable one pipeline and connect no memory block in the simulation, shown as in the figure2.

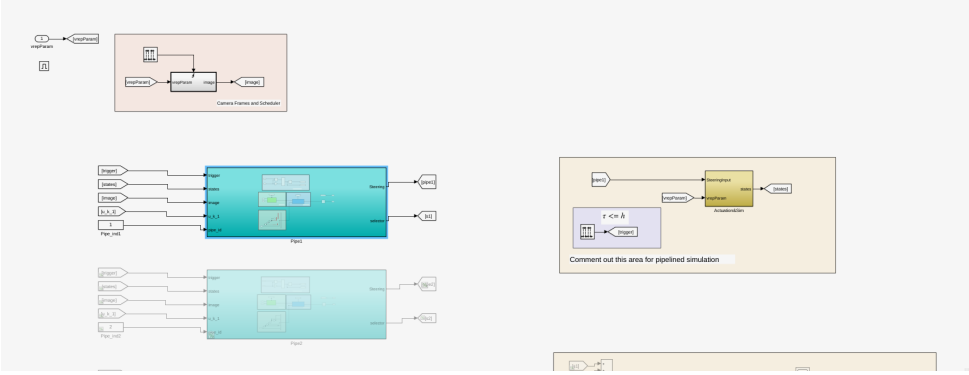


Figure 2: Case 1 SiL Implementation

Using the above implementation, we obtained the system output from IMACS. The deviation output shows an oscillation leading to a larger settling time(around 3s), as in the figure3. Assisted by the Vrep sense, we can observe the car trying to turn its head towards the straight road. If the controller isn't good enough, the car will turn very dramatically.

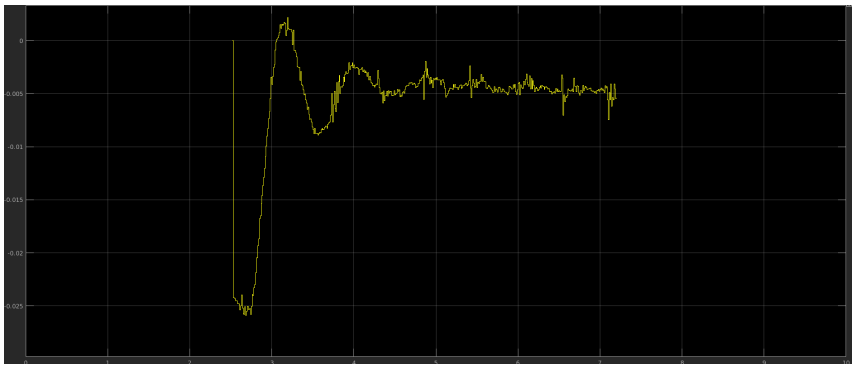


Figure 3: Case 1 SiL Output

summary

Although the controller design performs quite well in the MiL, its performance will degrade in the SiL. This difference may enlighten us that even though we had a good design in simulation, in reality, we would assign extra resources in case of some uncertainty.

1.2 Case 2

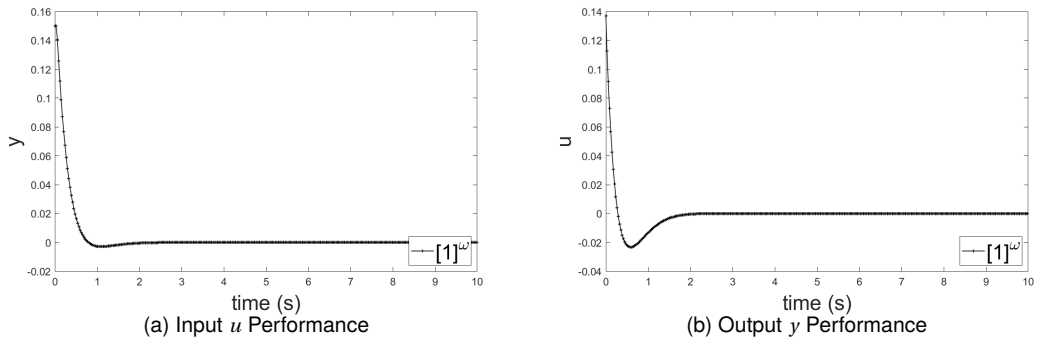
From the time analysis, we get $h = 0.03s$ and $\tau = 0.028s$ in this case.

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ -0.1857 & 0.0364 & -0.7371 & 0.6487 & 0 & 0 \\ 0.9414 & -0.1816 & -0.2814 & -0.0401 & 0 & 0 \\ -0.2130 & -0.0155 & -0.6142 & -0.7597 & 0 & 0 \\ 0.1843 & 0.9826 & -0.0150 & -0.0194 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$K = \begin{bmatrix} 0.3322 & -0.2882 & -0.9563 & -0.0402 & -0.1770 \end{bmatrix}$$

Plot the system input and output(under a positive deviation, bias is 0.15), the MiL simulation result shows that the system settling time is around 0.7.



(a) Input u Performance (b) Output y Performance

Figure 4: Case 2 MIL Controller Performance

We get the result from the SiL, in the figure5. In the system output, we can observe a smaller oscillation and also a smaller settling time comparing to the Case 1, which justify the necessity of extra resources.

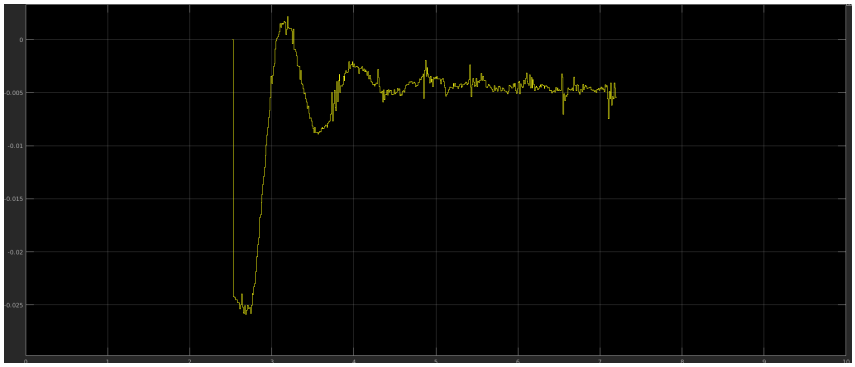


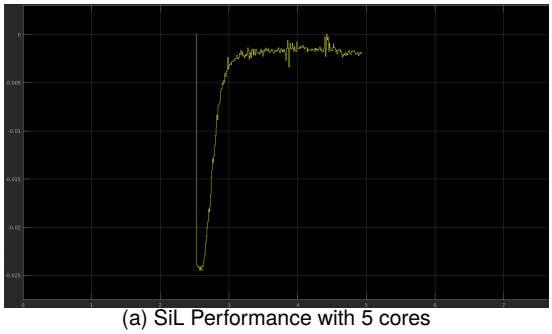
Figure 5: Case 2 SiL Output

We continue with different parallelization, the MiL is firstly conducted, results are shown in the table 1. We obverse that under some configurations, the MiL results are the same, which means that we don't need to keep increasing the number of parallelization until reaching the hardware limitations.

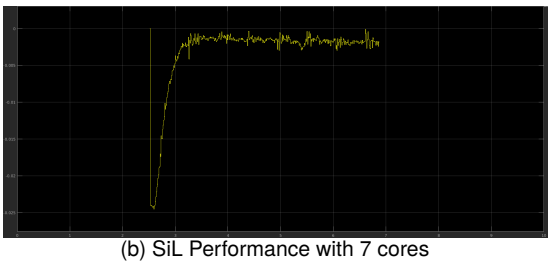
Table 1: MiL Result of Different Parallelization

parallelization	sampling period	delay	settling time(MiL)
2	0.060	0.056	0.72
3	0.050	0.043	0.71
4	0.040	0.036	0.70
5	0.040	0.032	0.70
6	0.030	0.030	0.69
7	0.030	0.028	0.69
8	0.030	0.026	0.69

We compare the SiL output between 8 cores, 7 cores and 5 cores in the figure 6, and find no big difference. Increasing the number of the core from 7 to 8, or from 5 to 8 has no significant impact on the system.



(a) SiL Performance with 5 cores



(b) SiL Performance with 7 cores

Figure 6: Case 2 MIL SiL Performance in Different Cores

Summary

Increasing the parallelization could improve the system, it can reduce the oscillation of the system output and reduce the settling time. But the relation between system performance and the amount of parallelization is not linear, in reality, we may not choose the parallelization as too large, because larger parallelization would have large power consumption while having a small upgrade to system performance

1.3 Case 3

In the pipelined version controller design, we first extract the sampling period and time delay by SDF time analysis.

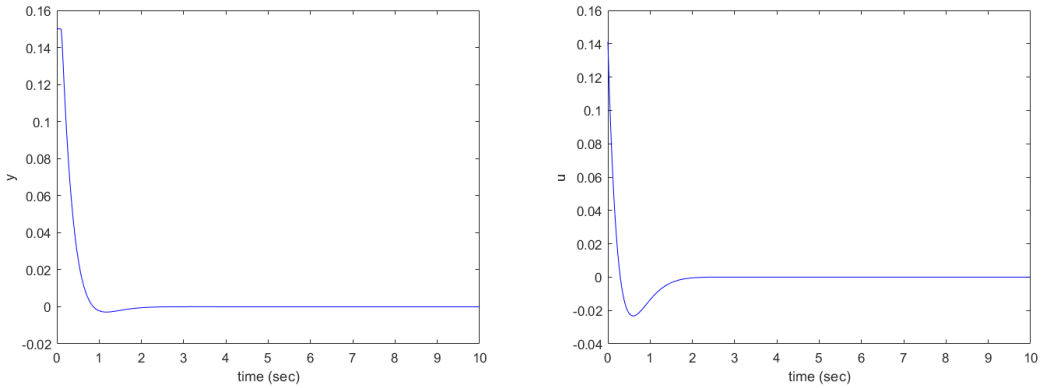
The sampling period is 0.02s, and the sensor-to-actuator delay is 0.097.

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -0.0692 & 0.0140 & -0.7654 & 0.6396 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8970 & -0.1790 & -0.3055 & -0.2646 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3934 & .0618 & -0.5663 & -0.7216 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1893 & 0.9818 & -0.0092 & -0.0120 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$K = \begin{bmatrix} -0.2891 & -0.4753 & -1.0149 & -0.0356 & -0.1370 & -0.1326 & -0.1283 & -0.1239 & -0.1192 \end{bmatrix} \tag{4}$$

Plot the system input and output(under a positive deviation, bias is 0.15), the MiL simulation result shows that the system settling time is around 0.76.



(a) Input u Performance (b) Output y Performance

Figure 7: Case 3 MIL Controller Performance

In the SiL implementation, we enable eight pipelines and connect four memory block in the simulation.

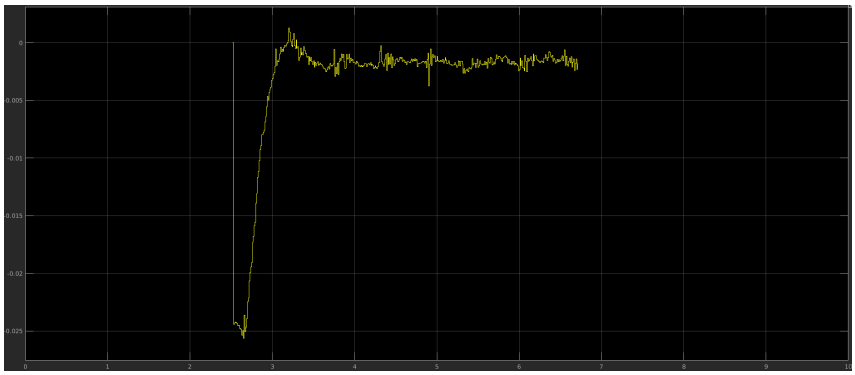


Figure 8: Case 3 SiL Output

Summary

Adding more pipelines can also reduce the setting time, comparing to the one core sequential design. However, a overshoot is observed.

1.4 Case 4

We inspect designs with a smaller amount of pipelines to check whether the overshoot can be avoided. By the time analysis, designs with pipeline numbers from 5 lines to 8 lines have no difference from each other, in the table 2.

Table 2: MiL Result of Different Pipelines

pipeline	sampling period	delay	settling time(MiL)
2	0.050	0.097	0.76
3	0.040	0.097	0.78
4	0.030	0.097	0.78
5	0.020	0.097	0.76
6	0.020	0.097	0.76
7	0.020	0.097	0.76
8	0.020	0.097	0.76

5(or larger) pipelines

We first check the difference between the controller with 5-pipelines and the controller with 8-pipelines.

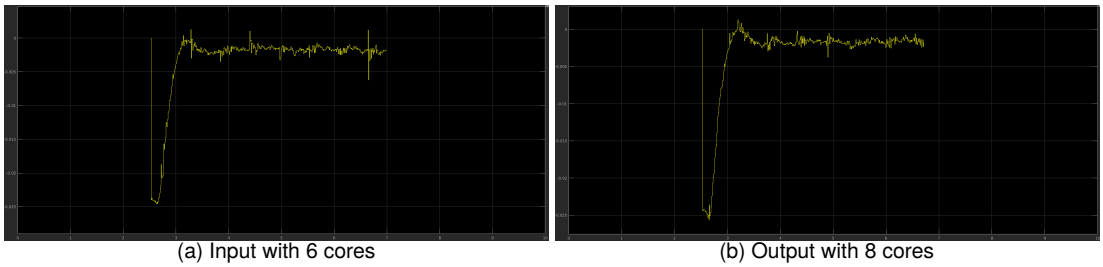
By time analysis, the sampling time of 5-pipelines(or larger amount of pipelines) version is $h = 0.02$.

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -0.0692 & 0.0140 & -0.7654 & 0.6396 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8970 & -0.1790 & -0.3055 & -0.2646 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3934 & .0618 & -0.5663 & -0.7216 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1893 & 0.9818 & -0.0092 & -0.0120 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$K = \begin{bmatrix} -0.2891 & -0.4753 & -1.0149 & -0.0356 & -0.1370 & -0.1326 & -0.1283 & -0.1239 & -0.1192 \end{bmatrix} \tag{5}$$

In the SiL implementation, we enable five or larger amount of pipelines and connect four memory blocks in the simulation to all situations.



(a) Input with 6 cores (b) Output with 8 cores

Figure 9: Case 4 SIL Controller Performance(h = 0.02)

From the SiL result, compared to the 8-cores case, the overshoot from the 6-cores case is less obvious.

4 pipelines

By time analysis, the sampling time of the 4-pipelines version is $h = 0.04$.

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -0.1857 & 0.0364 & -0.7371 & 0.6487 & 0 & 0 & 0 & 0 & 0 \\ 0.9414 & -0.1816 & -0.2814 & 0.0401 & 0 & 0 & 0 & 0 & 0 \\ -0.2130 & 0.0155 & -0.6142 & 0.7597 & 0 & 0 & 0 & 0 & 0 \\ 0.1843 & 0.9825 & -0.0150 & -0.0194 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$K = \begin{bmatrix} -0.2148 & -0.2950 & -1.0938 & -0.0464 & -0.2058 & -0.1963 & -0.1868 & -0.1770 \end{bmatrix}$$

In the SiL implementation, we enable four pipelines and connect three memory blocks in the simulation.

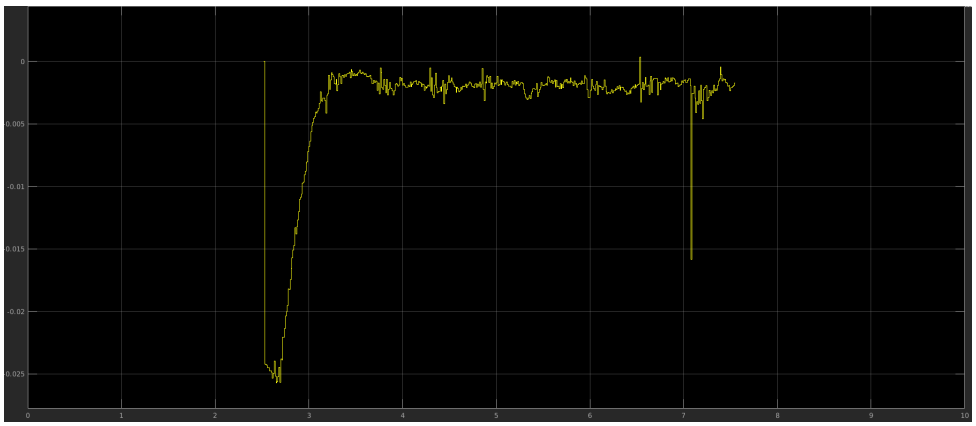


Figure 10: SiL Results of 4 Pipelines Controller

The SiL results show that, compared to larger pipelines number, this controller has a smaller overshoot and the settling time is almost the same as MiL result.

2 pipelines

The sampling time of 2-pipelines version is $h = 0.05$.

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -0.7356 & 0.1369 & -0.4743 & 0.4639 & 0 & 0 & 0 \\ 0.6506 & -0.1180 & -0.5900 & 0.4633 & 0 & 0 & 0 \\ -0.0614 & -0.0357 & -0.6528 & -0.7542 & 0 & 0 & 0 \\ 0.1784 & 0.9829 & -0.0285 & -0.0364 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$K = \begin{bmatrix} -0.1017 & -0.2153 & -1.0479 & -0.0657 & -0.3136 & -0.2886 \end{bmatrix}$$

The Software-in-the-Loop (SiL) Validation implementation is shown in figure 11, where we enable two pipelines and connect one memory block in the simulation.

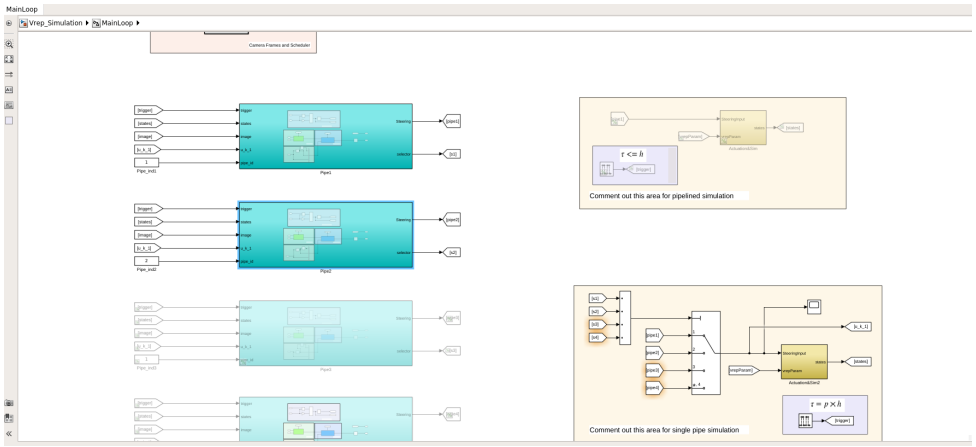


Figure 11: SiL Realization of 2 Pipelines Controller

The SiL output has a slight difference compared to the previous one, which means 2 pipelines may be ample for control tasks.

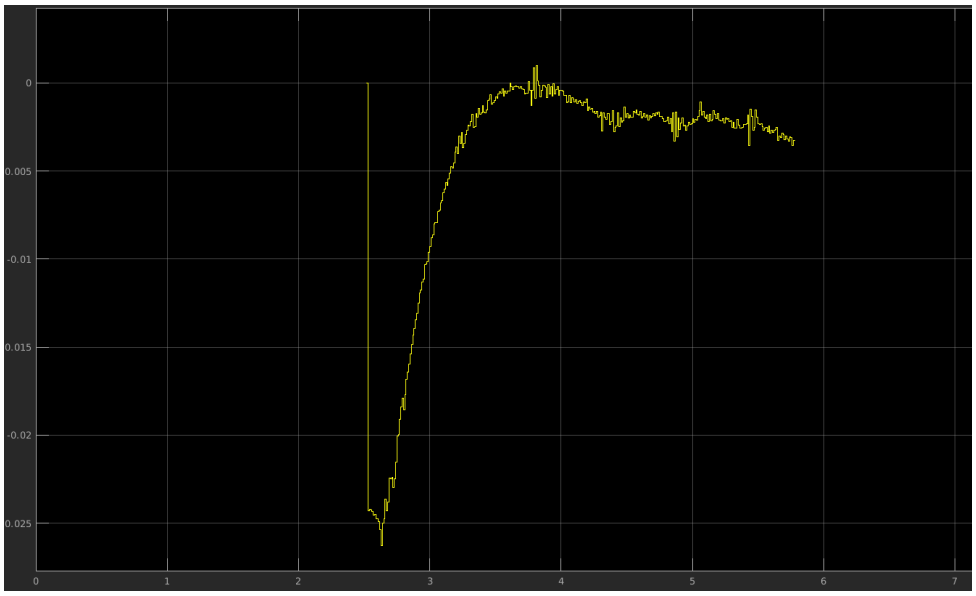


Figure 12: SiL results of 2 Pipelines Controller

Summary

Increasing the number of pipelines could have a large overshoot, therefore, we should avoid adding redundant pipelines.

1.5 Case 5

Design a controller for pipelined execution, and sensing tasks can be parallelized. Since more pipelines may cause overshoot and more parallelization has little help, there are three possible plans for core partition, shown in the table 3.

Table 3: Cores Partition Plans

	pipeline	parallelization	settling time	sampling time
A	4	2	0.72	0.02
B	2	4	0.70	0.02
C	2	2	0.72	0.03

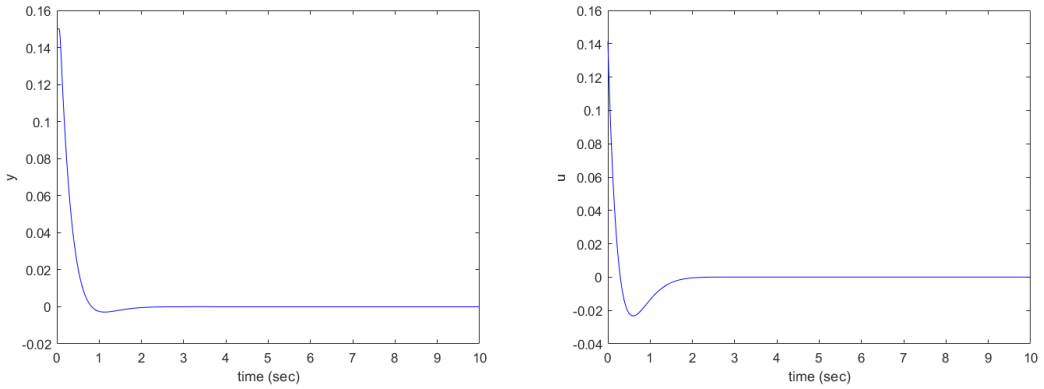
Plan A

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -0.0692 & 0.0140 & -0.7654 & 0.6396 & 0 & 0 & 0 & 0 & 0 \\ 0.8969 & -0.1790 & -0.3055 & -0.2645 & 0 & 0 & 0 & 0 & 0 \\ -0.3934 & 0.0617 & -0.5663 & -0.7215 & 0 & 0 & 0 & 0 & 0 \\ 0.1893 & 0.9817 & -0.0091 & -0.0119 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$K = \begin{bmatrix} -0.3686 & -0.4427 & -0.9250 & -0.0323 & -0.1238 & -0.1191 & -0.1238 & -0.1191 \end{bmatrix}$$

The MiL result is as follow:



(a) Input u Performance (b) Output y Performance

Figure 13: Case 5 MIL Controller Performance of Plan A

In the SiL implementation, we enable four pipelines and connect three memory blocks in the simulation.

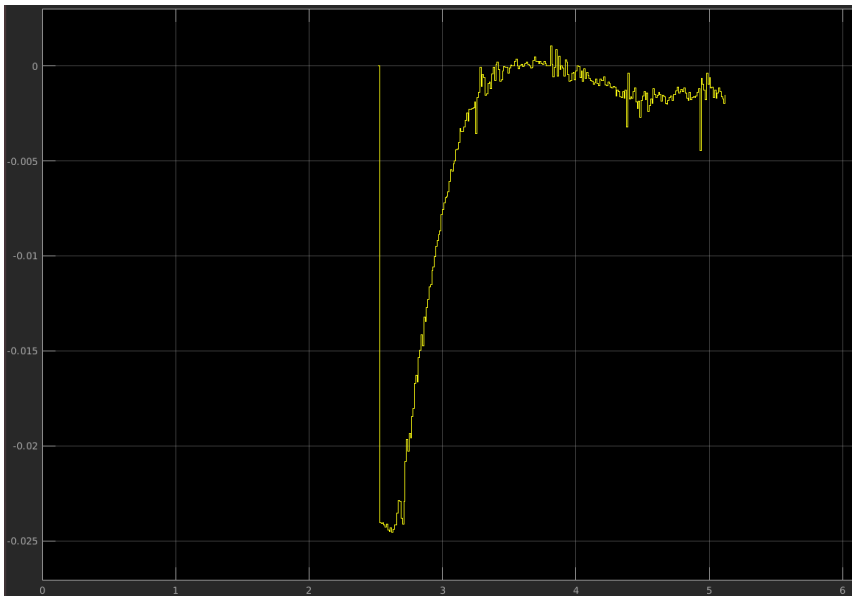


Figure 14: Case 5 SiL Output on Plan A

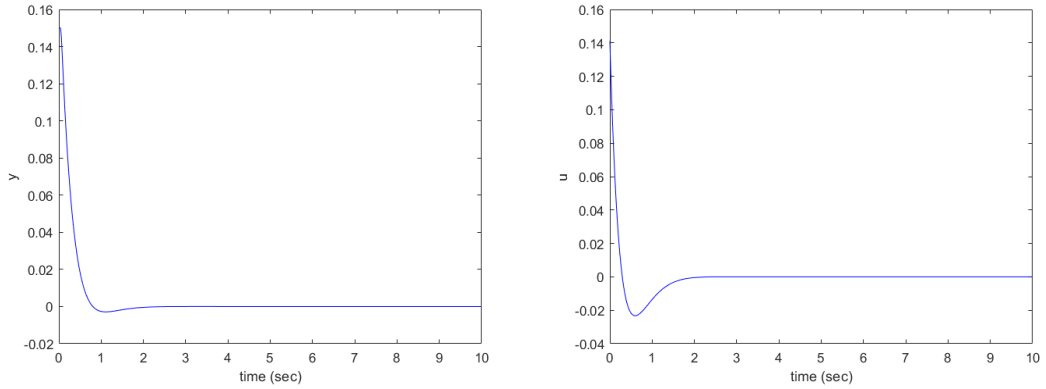
Plan B

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -0.0692 & 0.0140 & -0.7654 & 0.6396 & 0 & 0 & 0 \\ 0.8970 & -0.1791 & -0.3055 & -0.2646 & 0 & 0 & 0 \\ -0.3934 & 0.0617 & -0.5663 & -0.7215 & 0 & 0 & 0 \\ 0.1893 & 0.9817 & -0.0091 & -0.0119 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$K = \begin{bmatrix} -0.3686 & -0.4427 & -0.9250 & -0.0323 & -0.1238 & -0.1191 \end{bmatrix}$$

The MiL result is as follow:



(a) Input u Performance (b) Output y Performance

Figure 15: Case 5 MIL Controller Performance of Plan B

In the SiL implementation, we enable two pipelines and connect one memory block in the simulation.

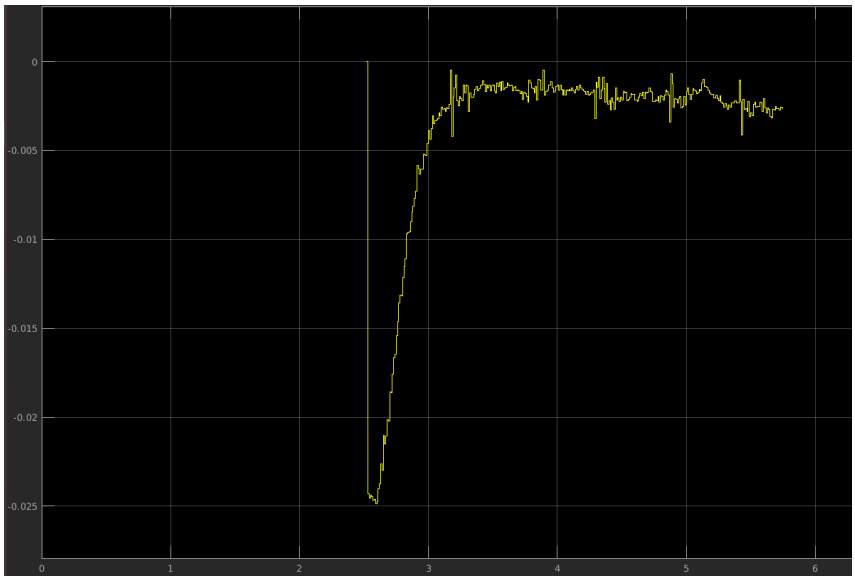


Figure 16: Case 5 SiL Output on Plan B

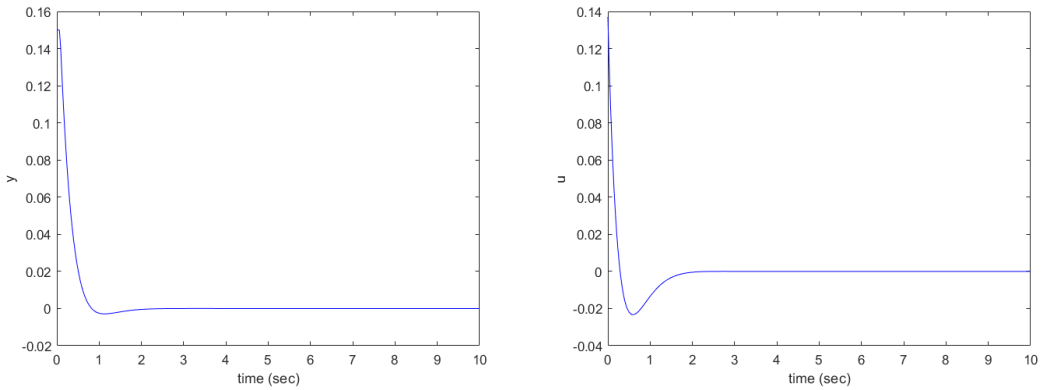
Plan C

Design the controller in the LQR way, we can get the transform matrix T and feedback vector K_T of the controllable part are as follow:

$$K\_T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -0.1857 & 0.0364 & -0.7371 & 0.6487 & 0 & 0 & 0 \\ 0.9413 & -0.1816 & -0.2814 & -0.0400 & 0 & 0 & 0 \\ -0.2129 & 0.0155 & 87347463155 & -0.7597 & 0 & 0 & 0 \\ 0.1842 & 0.9825 & -0.0149 & -0.0194 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$K = \begin{bmatrix} -0.2930 & -0.2904 & -1.0021 & -0.0422 & -0.1868 & -0.1770 \end{bmatrix}$$

The MiL result is as follow:



(a) Input u Performance (b) Output y Performance

Figure 17: Case 5 MIL Controller Performance of Plan C

In the SiL implementation, we enable two pipelines and connect one memory block in the simulation.

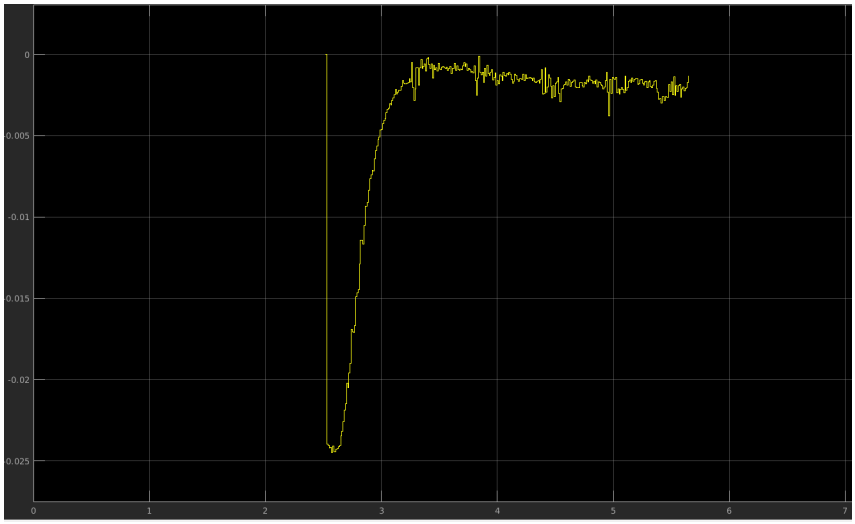


Figure 18: Case 5 SiL Output on Plan C

Summary

These three versions have no big difference from each other(the setting times are almost the same, no obvious overshooting), to avoid complexity, Plan C(2 pipelines and 2 parallelization) is preferable.

2 Comparison & Conclusion

In the Data-intensive control design, even if we get a good sequential design in the MiL simulation, the controller may have an oscillation and lead to a larger settling time. As a result, more resources are expected.

However, there is a trade-off between resources and performance, firstly, too many pipelines may lead to an overshoot, moreover, the power consumption will also grow. We should not increase the number of the core without constraints. Furthermore, Combining parallelism and pipeline is a good option, it will get a smoother output than other designing. For this specific case, using 2 pipelines and assigning two cores on parallelization to design controller is the most preferable plan.