Embedded Control Systems 5LIJ0
Project 2: Data-intensive Control

Muze Li(1794728),
Idan Grady (1912976)

March 30, 2023

1 Sequential vision-in-the-loop control

1.1 Maximum Execution Time

System	Sensing	Merging	Controller
Maximum Execution in ns	28,189425	245550	43950

Table 1: Maximum execution times for subsystems in the sequential system

1.2 TDM Scheduling

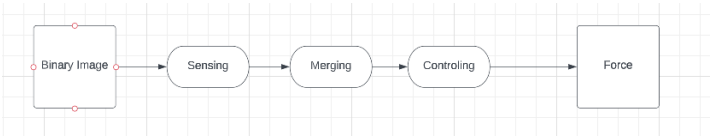

Figure 1: Sequential Data Flow

Tile	P1	P2	P3	P4
Tile 0	System app	Monitoring App(32 Kb)		controller(32 Kb)
Tile 1	System app	Sensing (64 Kb)	Merging (32 Kb)	

Table 2: Schedule Diagram and Memory Allocation

Tile	Partition	Cycles
0	1	1,600,000
	2	10,000
	3	777,000
1	1	1,150,000
	2	1,239,000

Table 3: Cycles Configuration

1.3 Sampling Period

$$\tau = 28.28189425 + 2450 + 43960$$
$$h = \lceil \frac{\tau}{f_h} \rceil f_h = \lceil \frac{28.478925}{10} \rceil f_h = 30ms$$

1.4 Controller Gains

K_p	K_d	K_i
70	20	10

Table 4: Controller Gain Sequential

To derive the settling time achieved for the control system, we consider the time that the system output first comes to the reference value. The settling time is:

$$t = 0.510s \tag{1}$$

As for the behavior we observed, the system does not show much overshoot. It vibrates around the reference value after reaching, we consider this to be due to the difference between sampling time and sampling steps.

1.5 Output

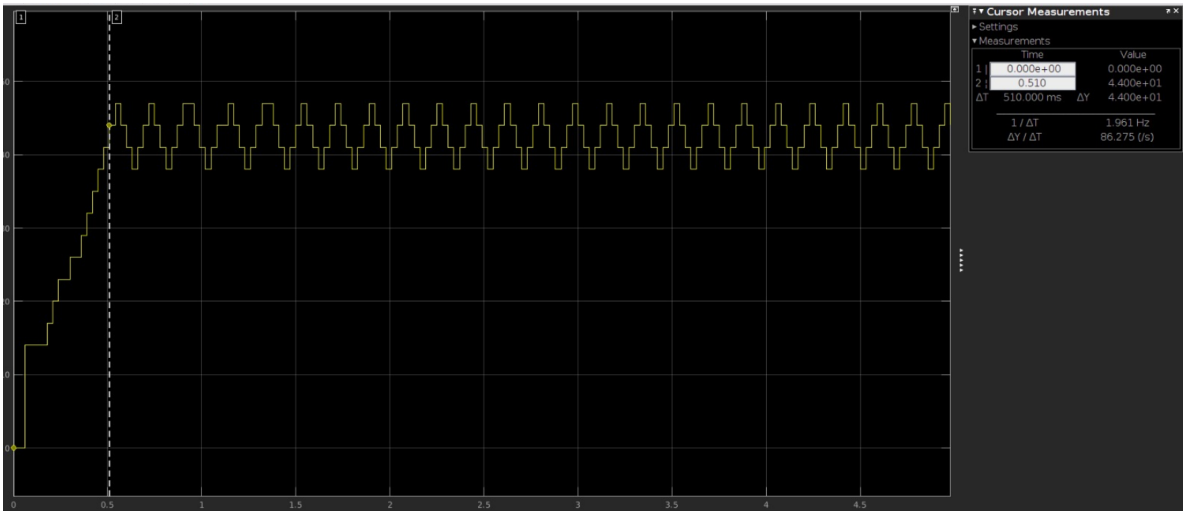

Figure 2: Sequential Plot

As observed, the PID converges after 0.5 seconds and stabilizes in the range of 40 to 48 units.

2 Parallelized vision-in-the-loop control

2.1 Maximum Execution Time

System	Sensing1	Sensing2	Merging	Controller
Maximum Execution in ns	14,294050	17394375	54475	31600

Table 5: Maximum execution times for subsystems in the parallel system

2.2 TDM Scheduling

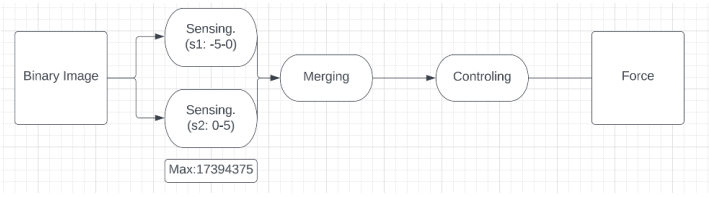

Figure 3: Parallelized Data Flow

Tile		P1	P2	P3
Tile 0	System app	Monitoring App(32 Kb)	Merging (32 Kb)	controller (32 Kb)
Tile 1	System app	Sensing 1 (64 Kb)		
Tile 2	System app	Sensing 2 (64 Kb)		

Table 6: Schedule Diagram and Memory Allocation

Tile	Partition	Cycles
0	1	800,000
	2	10,000
	3	1,577,000
1	1	700,000
	2	1,689,000
2	1	800,000
	2	1,589,000

Table 7: Cycles Configuration

2.3 The Sampling Period

$$\tau = 17.394375 + 2450 + 43960$$
$$h = \lceil \frac{\tau}{f_h} \rceil f_h = \lceil \frac{17.480450}{10} \rceil f_h = 20ms$$

2.4 Controller Gains

K_p	K_d	K_i
15	0.4	53

Table 8: Controller Gain Parallel

To derive the settling time achieved for the control system, we consider the time that the system output first comes to the reference value. The settling time is:

$$t = 0.715s \tag{2}$$

As for the behavior we observed, the parallel system requires a longer settling time than the sequential system. After reaching the reference value, the system has almost no vibration, we consider that it is more stable.

2.5 Output

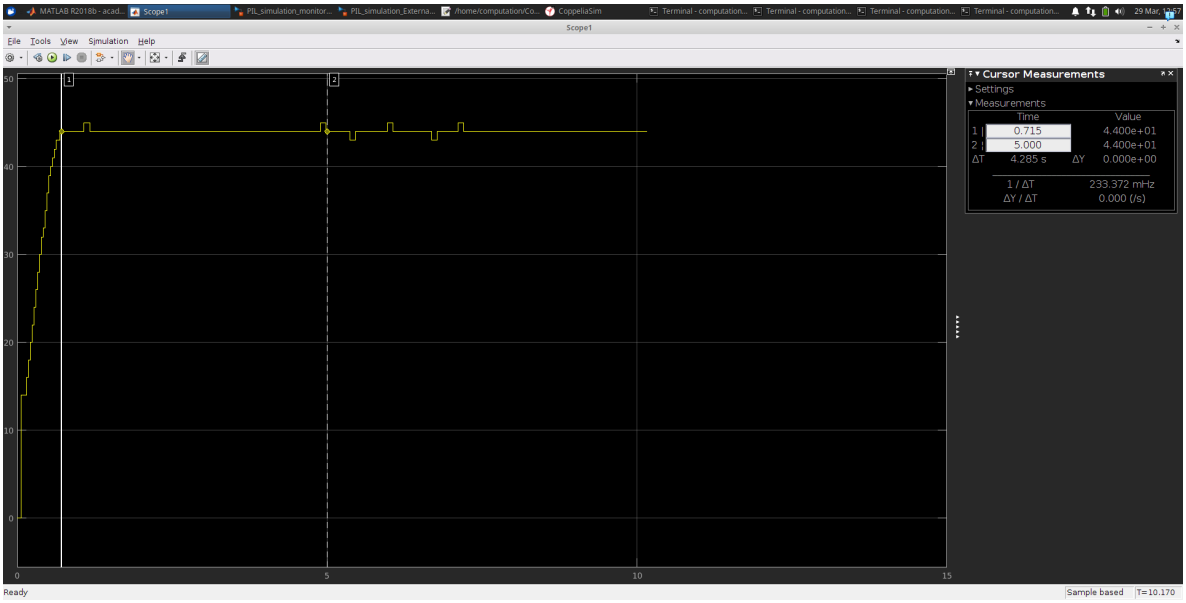

Figure 4: Parallel execution output

As seen in the figure, the PID controller with parallel execution converges faster and shows more stable response compared to the Sequential execution.

3 Pipelined vision-in-the-loop control

3.1 TDM Scheduling

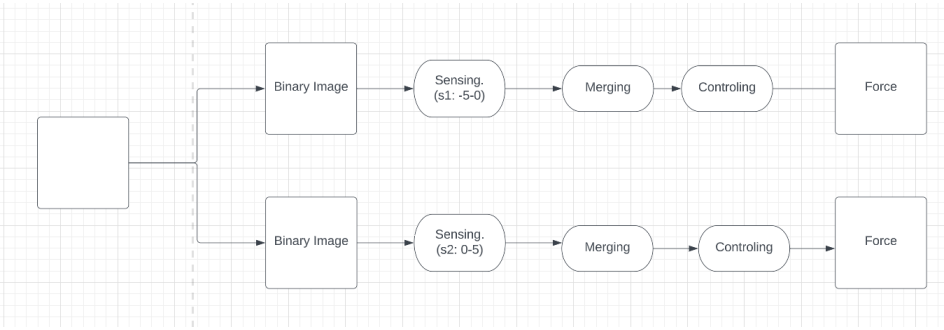

Figure 5: Pipelined Data Flow

Tile	P1	P2	P3
Tile 0	System app	Monitoring App(64 Kb)	
Tile 1	System app	Sensing + Merging + Controller (64 Kb)	
Tile 2	System app	Sensing + Merging + Controller (64 Kb)	

Table 9: Schedule Diagram and Memory Allocation

Tile	Partition	Cycles
0	1	2,000,000
1	1	1,200,000
2	1	1,200,000

Table 10: Cycles Configuration

3.2 Sampling Period

$$\tau = 28.28189425 + 2450 + 43960$$
$$h = \lceil \frac{\tau}{2 * f_h} \rceil f_h = \lceil \frac{28.478925}{2 * 10} \rceil f_h = 20ms$$

3.3 Controller Gains

K_p	K_d	K_i
40	0.8	30

Table 11: Controller Gain Pipelined

To derive the settling time achieved for the control system, we consider the time that the system output first comes to the reference value. The settling time is:

$$t = 0.585s \tag{3}$$

As for the behavior we observed, the pipelined system quickly reached the reference value, but it has large vibration.

3.4 Output

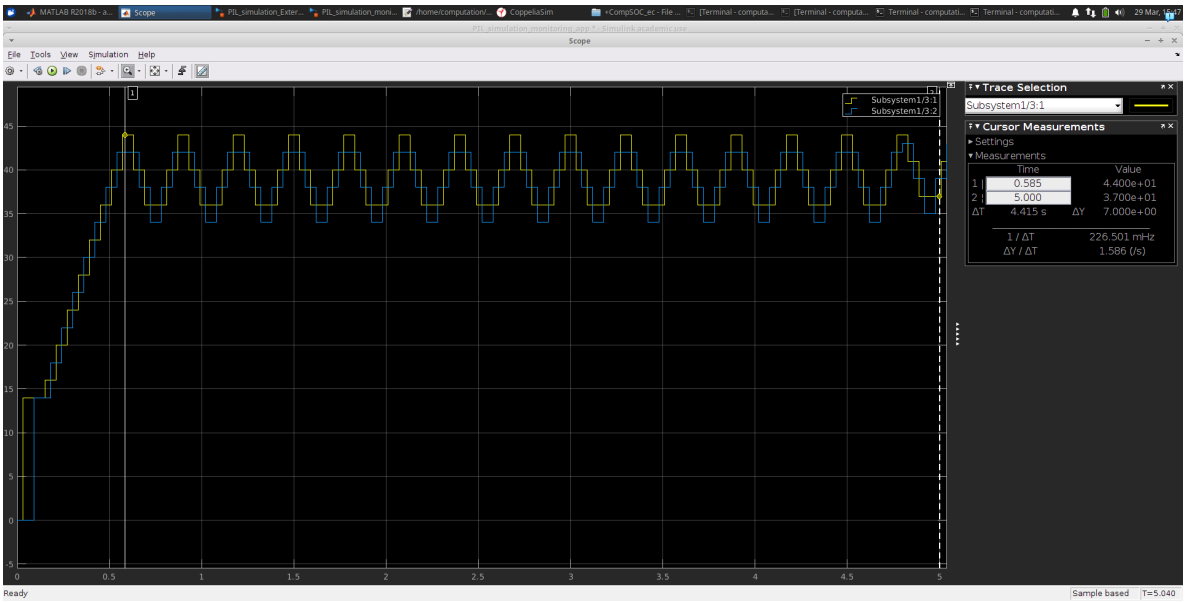

Figure 6: Pipelined Output

As can be seen, the PID with pipelined execution converges slower than the parallel execution. However, in terms of convergence time and output tolerance, it is in a similar range to the sequential.