Embedded Control Systems 5LIJ0
Project 1: Control/computation co-design (40 Points)
February 2023

The purpose of this assignment is to design and implement controllers for two dynamical systems to meet a predefined set of requirements, considering the implementation constraints on an embedded platform.

1. Dynamic system and model

These are two dynamic systems considered in this project.

1.1 Dual rotary system

The first system is a dual rotary fourth-order single input multiple output (SIMO) motion system as depicted in Figure 1.a. The input to the system is the input current to motor driver and the output is the angular position of the two masses.

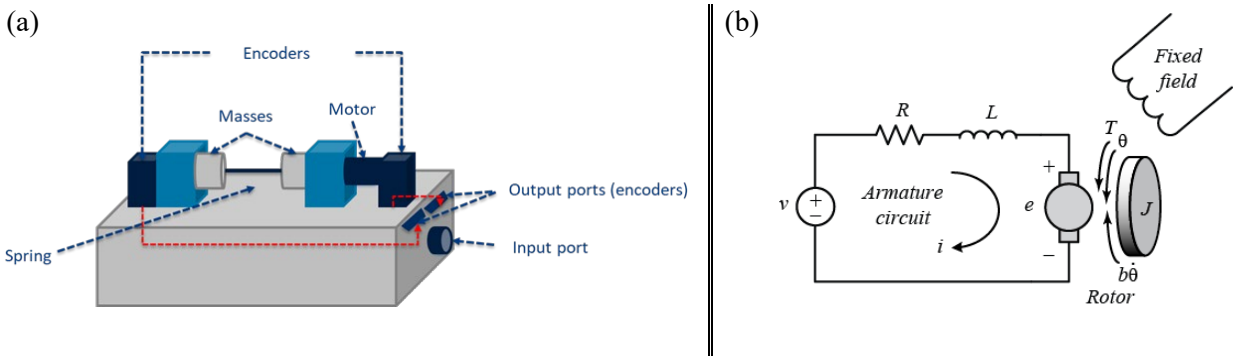
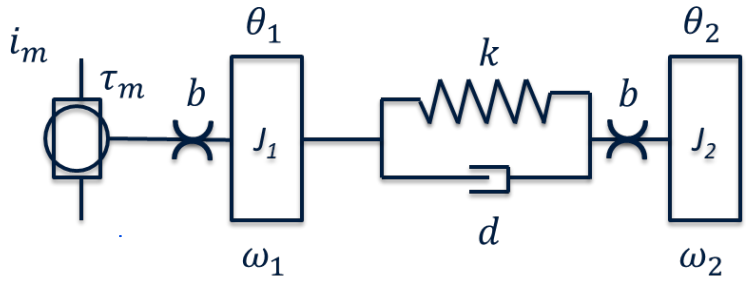


Figure 1- The schematics of the two dynamic systems

To model this mechanical system, the model can first be simplified to spring-damper model as shown in Figure 2.



θ_1	Position mass 1
θ_2	Position mass 2
i_m	Motor current
$\tau_m = K_m i_m$	Motor torque
J_1	Inertia mass 1
J_2	Inertia mass 2
b	Friction coefficient
k	Torsional spring
d	Torsional damping

Figure 2- Spring-damper equivalent model of the motion system

Considering this model, the system dynamics can be derived as:

$$J_1\ddot{\theta}_1 = K_m i_m - k(\theta_1 - \theta_2) - d(\dot{\theta}_1 - \dot{\theta}_2) - b(\dot{\theta}_1 - \dot{\theta}_2)$$
$$J_2\ddot{\theta}_2 = -k(\theta_2 - \theta_1) - d(\dot{\theta}_2 - \dot{\theta}_1) - b(\dot{\theta}_2 - \dot{\theta}_1)$$

The values for system parameters are presented in Table 1.

Table 1- System parameters of the dual rotary system

K_m [Nm/A]	J_1 [Kgm^2]	J_2 [Kgm^2]	b [Nms/rad]	k [Nm/rad]	d [Nms/rad]
$4.4\text{x}10^{-2}$	$3.75\text{x}10^{-6}$	$3.75\text{x}10^{-6}$	$1\text{x}10^{-5}$	0.2656	$3.125\text{x}10^{-5}$

1.2 DC motor speed control system

The second dynamic system is the speed control of a DC motor. A DC motor directly provides rotary motion and, coupled with wheels or drums and cables, it can provide translational motion. The electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in Figure 1.b. For this example, we will assume that the input of the system is the voltage source (V) applied to the motor's armature, while the output is the rotational speed of the shaft $\dot{\theta}$ .The rotor and shaft are assumed to be rigid. We further assume a viscous friction model, that is, the friction torque is proportional to shaft angular velocity. We can derive the following governing equations based on Newton's 2nd law and Kirchhoff's voltage law,

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

where i is the armature current and the values of the system parameters are presented in Table 2.

Table 2- System parameters of the DC motor speed control system

K [Nm/A]	J [Kgm^2]	b [Nms/rad]	R [Ohm]	L [H]
0.01	0.01	0.1	1	0.5

2. Embedded Platform and implementation

2.1 Platform properties

The embedded platform considered in this assignment for implementation is CompSOC. It is a composable and predictable platform that runs under a time division multiplexing (TDM) policy with preemption [1]. The platform runs the tasks periodically using a TDM table as Figure 3. The table is divided into *3 slots*. Each slot is composed of two parts – a slot for context switch overhead of fixed length ω=*2000 clock cycles* and an partition slot of length Ψ_i where i=*1,2,3.* An application can be allocated to one or multiple partition slots in a TDM table. However, a partition slot can only be assigned to one application. The following figure illustrates various parameters of the platform scheduling policy. The platform runs at the frequency of 40 MHz.

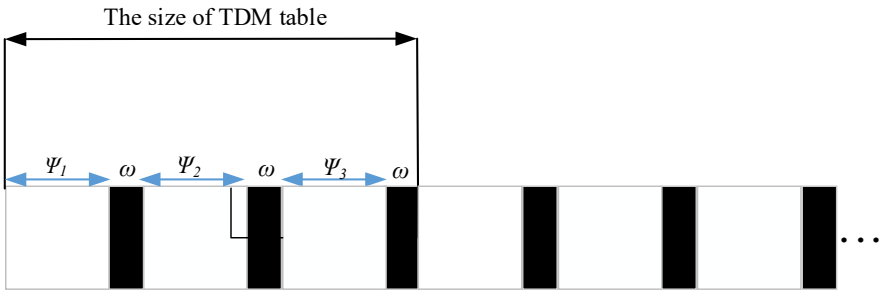


Figure 3- Timing and scheduling on CompSOC

2.2 Embedded implementation

A control application is composed of three application tasks – T_s (*sensing*), T_c (*computation*), and T_a (*actuation*). The implementation of a control application is done by the periodic sequential execution of these three application tasks. The execution time of the above tasks are to be measured as a project requirement.

The two important parameters defined for the controller design are:

- Sampling period h: time between start times of two consecution T_s tasks.
- Sensor to actuation delay τ: time between start of a T_s task and the finish time of a T_a task within a sampling period h.

The execution time of the control application can be measured through processor-in-the-loop (PIL) simulations. You can find more details over this on the implementation tutorial that is provided to you.

3. Assignment Tasks

3.1 Controller Design (20 points)

The first step in the assignment is to derive the state-pace model of the system. For the dual rotary system consider the states vector as x= [θ_1, θ_2, ω_1, ω_2], where $\omega_1=\dot{\theta}_1$ and $\omega_2=\dot{\theta}_2$. For the DC motor speed system consider the states vector as x= [$\dot{\theta}$, i].

Since our purpose is to design a discrete-time controller, we need to discretize the systems with a sampling period (h) which is derived from execution time and TDM scheduling of the platform. Doing so, the sampling instants are defined as t=k*h where k=0,1,2,… . The equivalent discretized states are defined as ξ [k]=x(kh) and the discrete-time state-space is defined as:

$$\xi[k+1] = \varphi\,\xi[k] + \Gamma\,u[k],$$
$$y[k] = C_d\,\xi[k].$$

Where:

$$\varphi = e^{Ah},$$
$$\Gamma = \int_0^h e^{As} B\, ds.$$

Now, the problem is to design a discrete-time controller u[k], given r[k] is the reference signal such that all the following constraints are satisfied assuming all four states are measurable and can be used as feedback signals:

- y[k] – r[k] → 0 as k→ ∞, where $y = x_1$ and r[k]=1. This is true for both of the dynamical systems.
- The time needed for y[k] to reach within 2% of r and stay is called settling time which should be shorter than 70 ms for the dual rotary system and 2 sec for the DC motor speed control system.
- The control input u[k] should be in range of (-1 <u[k] < 1) for the dual rotary system and (-12 <u[k] < 12) for the DC motor speed control system;
- The states (x_3<50 and x_4 <50) for the dual rotary system;

And the controller is to be implemented on the embedded platform described in Section 2 where:

- The controllers of each dynamic system are separate applications and both control applications are implemented on a shared processor tile of the platform (which is called controller tile).
- It is assumed that there is a shared sensor unit that is able to provide the sensor values for both of the dynamic systems. This sensor unit is a part of dual rotary controller application.
- A summary of the expected implementation is provided in Figure 4 and discussed in Section 3.2.

The expected results for control design have these steps:

a. **Design 1**: For the dual rotary system, consider a sampling period in range of h_1<3 ms and design the controller. For the DC motor speed control consider $h_2=2*h_1$.
b. **Design 2**: Repeat Design 1 and assume that the delay is not measurable and not known to the designer. Therefore the designer assumes τ=0 in control design (but not in the simulation). Plot the system response and input signal and then, compare it to the result of Design 1.
c. **Design 3**: For the dual rotary system, consider a sampling period in range of h_1>3 ms and design the controller. For the DC motor speed consider $h_2=2*h_1$.
d. **Design 4**: Repeat Design 3 and assume that the delay is not measurable and not known to the designer. Therefore the designer assumes τ =0 in control design (but not in the simulation). Plot the system response and input signal and then, compare it to the result of Design 3.

A comprehensive report of the results should be able to answer these questions:

a. Present the derived state space of the system.
b. What are the execution times for different tasks on the platform?
c. For each design:
   - What are the values of Ψ_i (i=1,2,3) of the platform? Consider them as design parameters.
   - What is the allocation of control applications to the partition slots?
   - What are the values of h and τ?
   - What are the controller gains used in u[k] for both control systems?
   - What is the settling time achieved for both control systems?

3.2 HIL Implementation (20 points)

After designing the controllers and verifying them with simulation, you need to also verify them through hardware-in-the-loop (HIL) simulation on the embedded platform. For this simulation, the model of the two motion systems are implemented on separate processor tiles. The DC motor dynamic system is implemented on tile 1 and the dual rotary system is implemented on tile 2. You are expected to implement your controllers as two separate applications on tile 0. If we assume S_1, C_1, and A_1 as sensing, computation and actuation of

the DC motor system respectively and S_2, C_2, and A_2 as sensing, computation and actuation of the Dual rotary system, the implementation depicted in Figure 4 is expected.

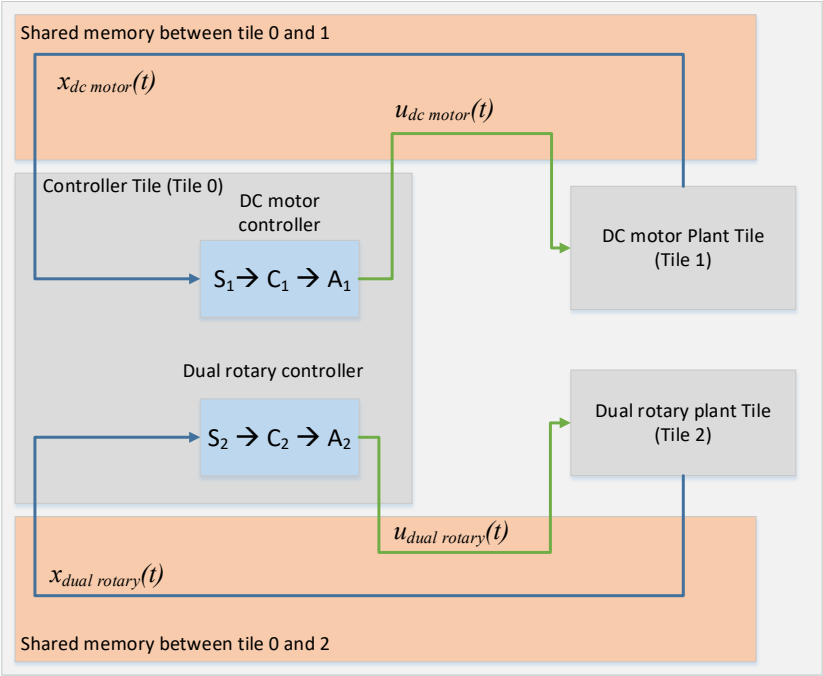


Figure 4 – Expected implementation of the controllers on the embedded platform

Each group will be granted access to a PYNQ FPGA board which is the host of this architecture. Through the provided Simulink models, you can modify the controller parameters as well as scheduling parameters for controller processor tile. You will perform PIL simulations using the provided model to extract the required execution times.

After modifying the controller parameters and measuring the execution times, you should perform HIL simulation. Necessary Simulink models are provided to help you do this. Next you need to compare the result of HIL with the result of the MATLAB simulation. You only need to simulate and compare the results for **Design 1** and **Design 3** in Section 3.1.

What you should deliver for this part is the resulting comparison graphs.

3.3 Provided materials

1. **MIL_PIL_Simulation_2022.slx**:
   This file is a Simulink that is provided to you to help you visualizing the output of both dynamic systems to your designed controllers. In order to use this file you need to initialize its variables using two provided MATLAB function: assignment1_2022_Simulink_init_DCmotor and assignment1_2022_Simulink_init_Dualrotary.
2. **assignment1_2022_Simulink_init_DCmotor.p**:
   This file is a MATLAB function which initialize variables for the DC motor system in the Simulink file: Assignment1_2022_Simulink.slx. Please put this function at the very last line of your code for control design, run your code, and then execute the Simulink. Here is the function description:

   assignment1_2022_Simulink_init_DCmotor (D,H,K,F);

This function takes 4 input arguments. D (1x1) is the sensor-to-actuator delay, H (1x1) is the sampling period, K(1xN where N is 2 or 3) is the feedback gain and F(1x1) is the feedforward gain. Calling this function will define all the necessary variables for simulation. If you call this function correctly, you will see the response "Variables initialized for simulink" in MALTAB command window. You can also see that some variables are added to your workspace. This function can also be used for the Designs 2 & 4 in Section 3.1, where you do not consider the delay in your control design. To activate this feature you need to give a 1x2 input as feedback gain (K). In this case you will see this response in MATLAB command window:
Warning: Feedback gain K length is 2. "Control design without considering delay" is activated.

3. **assignment1_2022_Simulink_init_Dualrotary.p**:
   This file is a MATLAB function which initialize variables for the Dual rotary system in the Simulink file: Assignment1_2022_Simulink.slx. Please put this function at the very last line of your code for control design, run your code, and then execute the Simulink. Here is the function description:

   assignment1_2022_Simulink_init_Dualrotary (D,H,K,F);
   This function takes 4 input arguments. D (1x1) is the sensor-to-actuator delay, H (1x1) is the sampling period, K(1xN where N is 4 or 5) is the feedback gain and F(1x1) is the feedforward gain. Calling this function will define all the necessary variables for simulation. If you call this function correctly, you will see the response "Variables initialized for Simulink" in MALTAB command window. You can also see that some variables are added to your workspace. This function can also be used for the Designs 2 & 4 in Section 3.1, where you do not consider the delay in your control design. To activate this feature you need to give a 1x4 input as feedback gain (K). In this case you will see this response in MATLAB command window:
   Warning: Feedback gain K length is 4. "Control design without considering delay" is activated.

4. **ECS_2022 Virtual Machine:**
   A virtual machine for performing PIL and HIL simulations is provided. The virtual machine contains a MATLAB installation and necessary simulation files to perform simulations. The Virtual machine can be accessed through the link below (The link only works in TU/e network):

   https://virtualmachine.compsoc.eu/

   It is highly advised to study the provided tutorial. The provided PDF *Tutorial 1 The virtual machine* will help you through the installation of the virtual machine.

5. **PYNQ FPGA board**:
   Each group is granted access to a PYNQ FPGA board, which makes you able to implement your controller, extracting execution times using PIL, and perform HIL simulation. In order to work with your board you need to do some steps:
   a. Install ECS virtual machine and connect to the board.
   b. It is highly advised to study and implement the provided tutorial example. The provided PDF "Tutorial_guide" will help you through the implementation.

6. PIL and HIL Simulink models:

The necessary files to perform the PIL and HIL simulations on the platform are provided inside the virtual machine. After setting up the virtual machine and the PYNQ board, you can use *Tutorial 3 PIL and HIL simulations* to learn how to perform PIL and HIL simulations.

4. Deliverables

1. Report **Groupnumber_assignment1.pdf** answering the following questions with justifications and explanations.

   a. Answer the questions asked at the end of section 3.1.

   b. Compare the results of MATLAB simulation with the results of HIL simulation.

   c. Conclusion and discussion about the provided results.

2. Corresponding MATLAB script to validate the above answers. Name of the script should be **Group-number_assignment1.m**.
3. Modified Simulink models for PIL and HIL simulations (MIL_PIL_Simulation_2022.slx, DC_Motor_HIL.slx, and Dual_rotary_HIL.slx).

Please provide all the deliverables inside a zip file with the name: **5LIJ0-A1-groupnumber-version.**

References:

[1] K. Goossens *et. al*., "Noc-based multiprocessor architecture for mixed time-criticality applications," Handbook of hardware/Software Codesign, pp. 491–530, 2017.