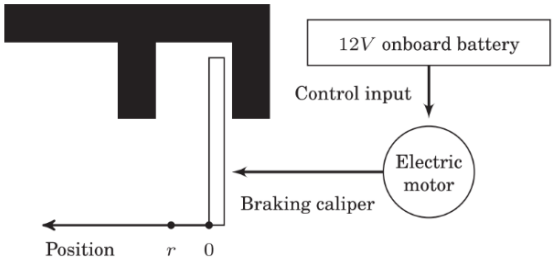
Embedded Control Systems 5LIJ0
Project 2: control design for embedded visual control
February 2019

The purpose of this project is to design a controller for a dynamical system to meet a predefined set of requirements, considering its implementation constraints on a multi-processor platform.

1. Control System



System Description: In the above figure, a simplified model of the Electro-Mechanical Braking (EMB) system is shown. When the EMB is active, the braking caliper should reach a reference position r, which is at the braking disc, within the desired settling time. This is the *position* mode. The requirement for the settling time ensures the reactiveness of the system. After that, a certain force is applied in the *force* mode. The electric motor mobilizing the braking caliper is powered by the onboard battery, which has a voltage of 12V (For more details see "**Chang, Wanli, et al. "OS-Aware Automotive Controller Design Using Non-Uniform Sampling." ACM Transactions on Cyber-Physical Systems 2.4 (2018): 26.**").

In this project, we consider the position mode of the EMB system, which is of interest in several scenarios, such as braking, disc wiping, and pre-crash preparations. The state space matrices of the dynamic system under consideration is given by:

$$A = \begin{bmatrix} -520 & -220 & 0 & 0 & 0 \\ 220 & -500 & -999994 & 0 & 2*10^8 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 66667 & -0.1667 & -1.3333*10^7 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1000 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

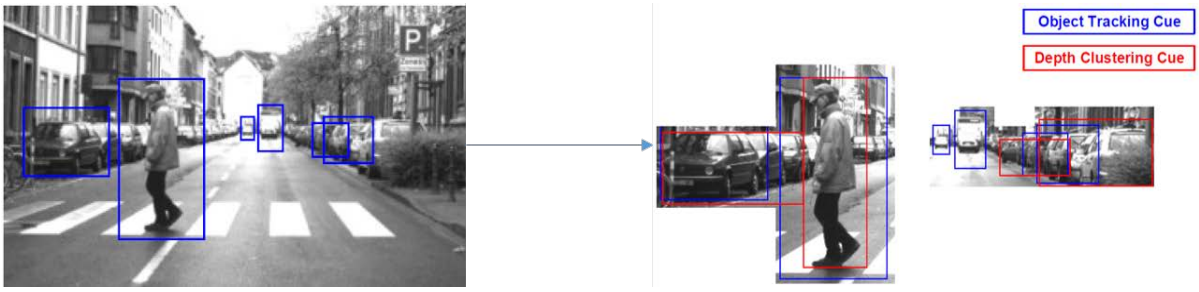
$$C = [0 \quad 0 \quad 0 \quad 0 \quad 1], D = 0$$

It should be noted that there are five system states (state vector $x(t)$): motor current, motor angular velocity, motor angular position, caliper velocity, and caliper position. The control input $u(t)$ is the applied voltage on the motor. Initial value of all states is equal to zero (i.e. $x(0) = 0$). The control system under

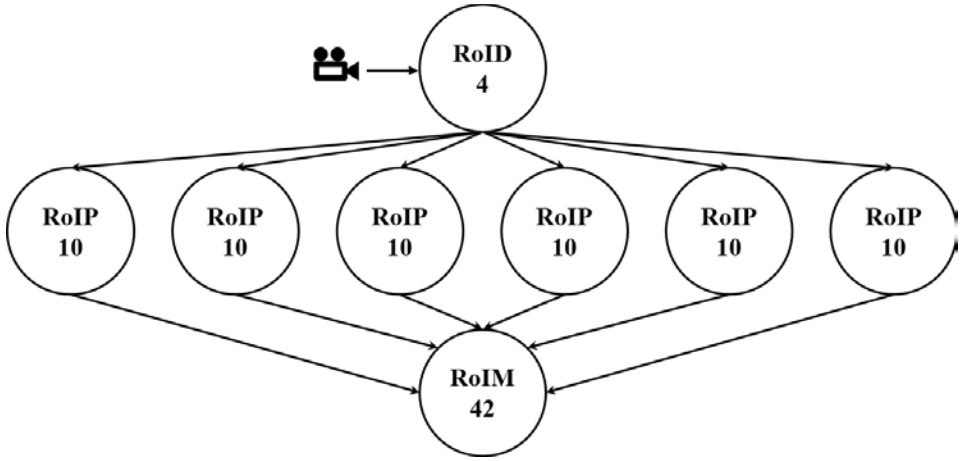
consideration aims to design input $u(t)$ such that the caliper position (output $y(t)$) follows a given constant reference $r = 0.002$. This is called a regulation problem.

2. Embedded Implementation

A control application is composed of three application tasks – τ_s (*sensing*), τ_c (*computation*), and τ_a (*actuation*). The implementation of a control application is done by the execution of these three application tasks. The τ_s task is an image processing algorithm. The algorithm detects the regions-of-interest (RoID), processes each region (RoIP) and merges the data for each region (RoIM). An example is given below: i) the regions-of-interest detection (RoID) for the image (on the left) yields the blue boxes; ii) the blue boxes are then processed (RoIP); and iii) the processed data for the regions are merged (RoIM) to obtain the object tracking and the depth clustering cue.



A task graph description of the above steps is given below. You should use this description for the remainder of the project. The execution times are represented in ms. This task graph implies that the RoIP task can be executed in parallel depending on the availability of processing cores.



Having more number of processing cores mean that you could choose to parallelize or pipeline the execution of sensing tasks and have a sequential or pipelined controller implementation.

The two important parameters defined for the controller design are:

- **Sampling period h**: time between the start times of two consecutive τ_s tasks
- **Sensor to actuation delay τ**: time between start of a τ_s task and the finish time of the corresponding τ_a task

3. Embedded Platform

The platform to consider for implementation is a multi-processor platform. The platform runs on time-triggered static-order scheduling scheme. The camera image frames arrive at a frame rate of **30 fps**.

The timing details of the tasks are given below:

Task	WCET
τ_s	$106ms$
τ_c	$1.5\ ms$
τ_a	$0.5\ ms$

4. Problem Definition

The problem is to design a discrete-time controller $u[k]$ such that all the following constraints are satisfied:

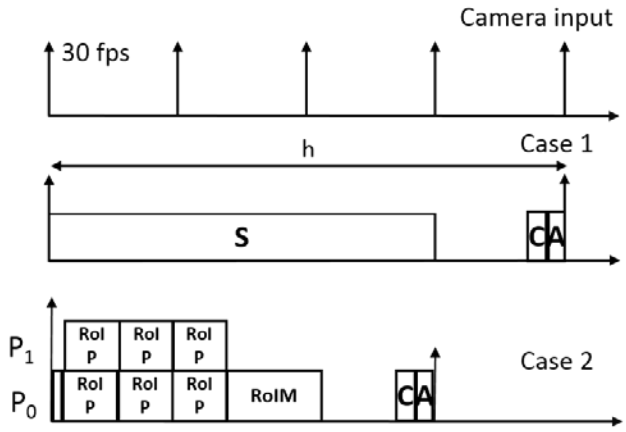
- $y[k] - r \rightarrow 0$ as $k \rightarrow \infty$
- The time needed for $y[k]$ to be within 2% of $r = 0.002$ is called settling time which should be as short as possible.

The **design objective** is to achieve a settling time **as short as possible**.

5. Analysis

i. Design controllers for sequential execution of control tasks with and without parallelization of sensing tasks for the following two cases (Case I and Case II). Sensing tasks can be parallelized if we know the internal structure (e.g. task graph) of the algorithm.

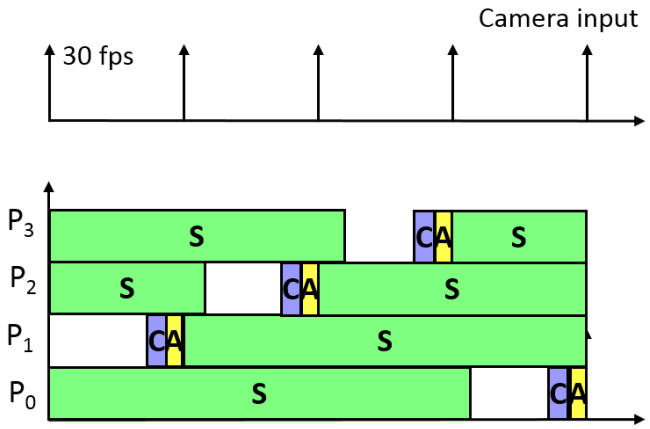
(**5 points**)



- **Case I**: One core is allocated to control tasks $\tau_s, \tau_c\ and\ \tau_a$ (see above figure: Case 1).
- **Case II**: Four cores are available for control tasks $\tau_s, \tau_c\ and\ \tau_a$. E.g., with two cores, the above figure (Case 2) shows timing resulting in the WCRT of $\tau_s = 76\ ms$.
- Compare and comment on the settling time achieved in Case I and Case II.

ii. Design controllers for pipelined execution. In this case, assume that sensing tasks **cannot** be parallelized. This situation occurs when the internal structure (task graph) of the sensing algorithm is not available to the control/embedded systems designer, i.e. the sensing algorithm is a black box.

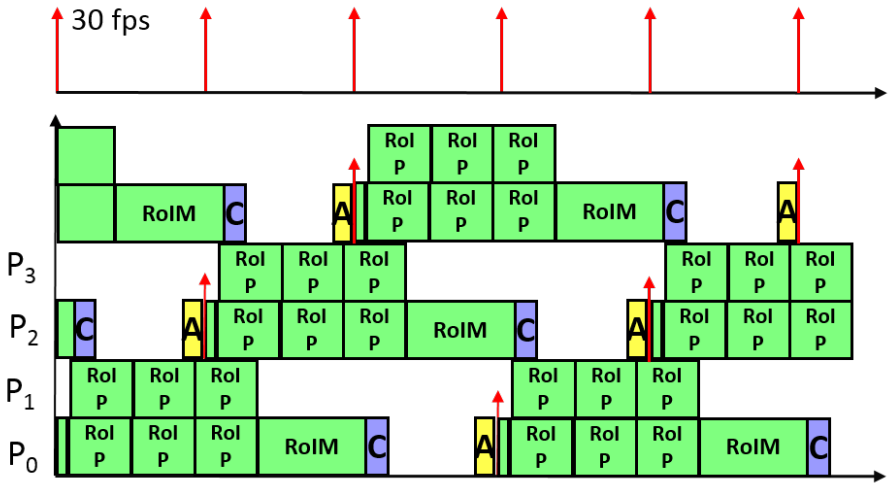
(**8 points**)



- **Case III**: Four cores are available for control tasks τ_s, τ_c *and* τ_a (see the above figure).
- **Case IV**: Evaluate the trade-offs between number of cores and control performance. Assume that you could have any number of cores.

iii. Design a controller for pipelined execution. Assume that sensing tasks **can** be parallelized. This means that the designer knows the internal structure of the sensing algorithm and he could choose to have a hybrid implementation. Hybrid here refers to the combination of parallel and pipelined execution as shown in the figure below. The figure gives the example of hybrid implementation with 4 cores.

(**2 points**)



- **Case V**: Six cores are available for control tasks τ_s, τ_c *and* τ_a.

6. Questions to be answered in this project and deliverables

1. Report **Group_no_project2.pdf** answering the following questions with justifications and explanations.
   - **Cases I-V**: design, results and analysis.
2. Corresponding MATLAB scripts (for each case) to validate the above answers, i.e. **group_no_Case_i.m** (five *.m files, one for each case).
3. Upload the above documents as a zip file, **Group_no_project2.zip.**