Design and Implemenation of Image Based Control Systems

- Dip Goswami (TU/e)

In collaboration with Sajid Mohamed (TU/e), Robinson Medina (TNO), Sayandip De (TU/e), Mojtaba Haghi (TU/e), Raghu Rajappa (Sioux Logena) , Vishak Nathan (Sioux Logena), Koen Rutten (Sioux CCM), Vigneswar Madras (Sioux CCM), Chinmay Kamate (TU/e), Kees Goossens (TU/e) and Twan Basten (TU/e)

Electronic Systems group, Electrical Engineering

What is Image Based Control?

- Regulate behavior of a system
- System: Dynamical systems
  - $\dot{x} = Ax + Bu; \mathrm{y} = \mathrm{Cx}$

- Camera is used as a sensor
- Sensing operation
  - Image Signal Processing (ISP)
  - Percenption (PR)

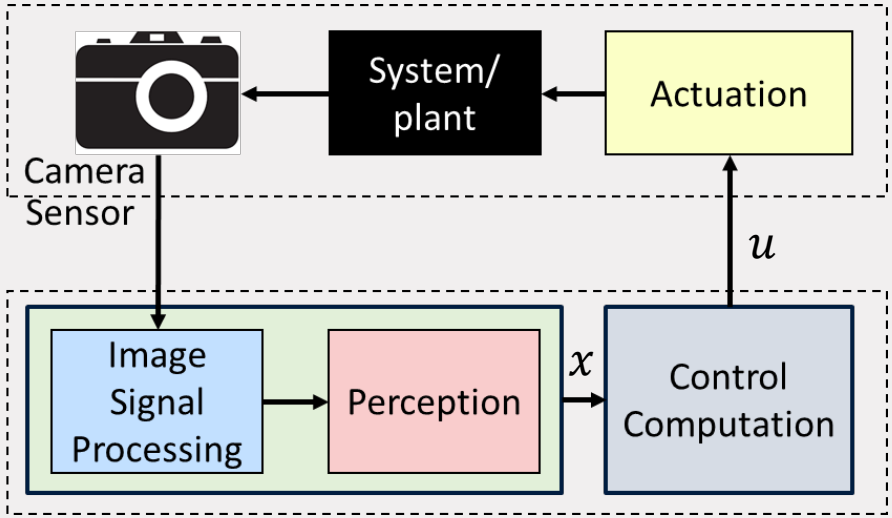
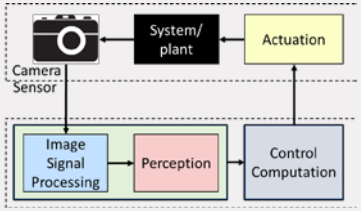


Image Based Sensing



- ISP pipeline converts the raw image to a compressed image
- ISP performs a series of signal processing
- PR performs a set of application specific processing

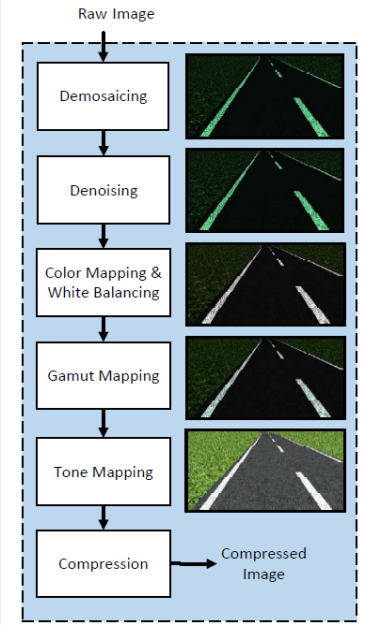
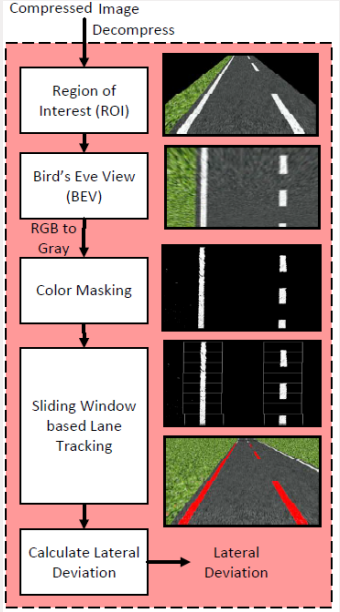
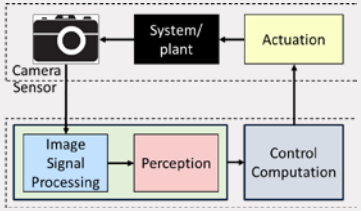


Image Signal Processing (ISP)

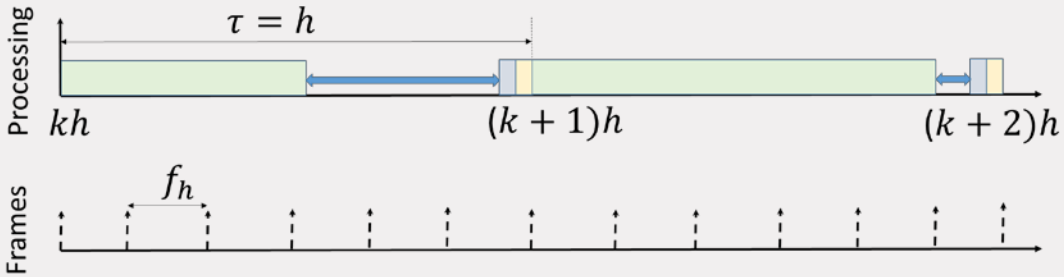


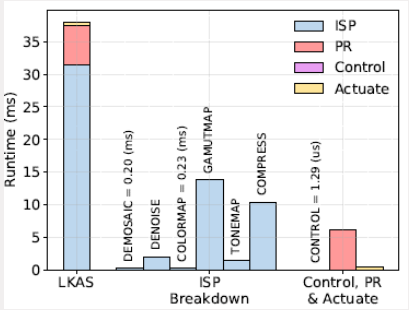
Perception (PR)

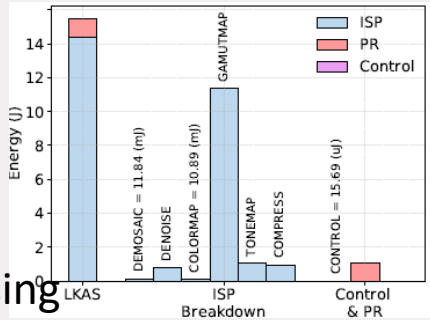
Image Based Sensing



- ISP takes 82% of runtime
- ISP takes 93% of energy







f_h = period of camera frame arrival;
h = sampling period or start of two successive sensor processing
τ = sensing-to-actuator delay

*Lane Keeping Assist Sys (LKAS)
8-core Intel i9; 256 KB L2+16MBL3+64GB RAM

Some known examples… autonomous cars

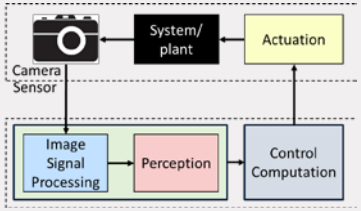
Courtesy: Tesla.com

Courtesy: Daimler.com

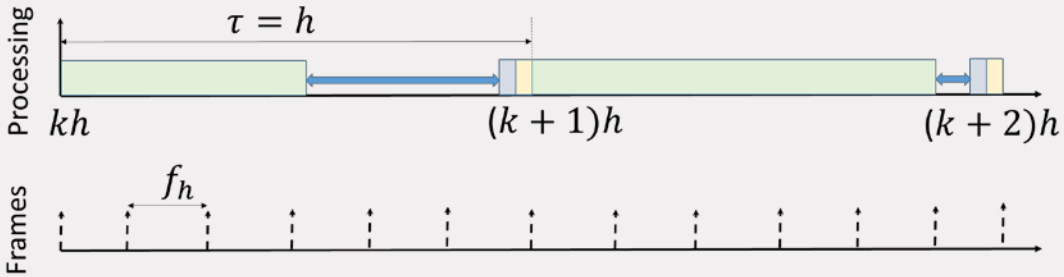
Why Image (and Camera) in the feedback loop?

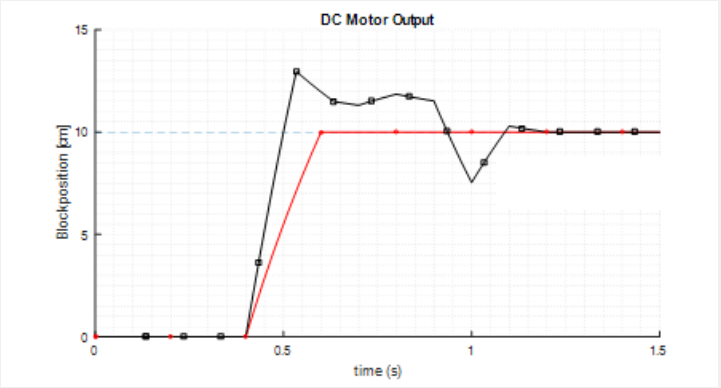
- Richer information which is hard to by one sensor;
- Low-cost of processing units and advancement of CMOS-based imaging technology;
- No alternative sensor in some scenarios;

What is the challenge?



- ISP takes 82% of runtime
- ISP takes 93% of energy

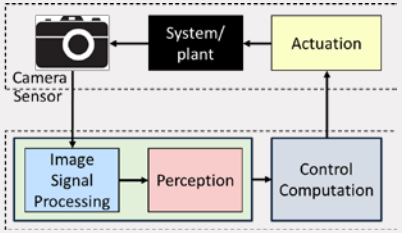




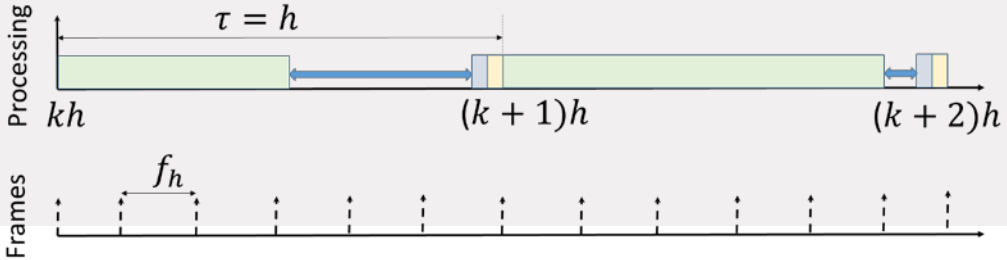
Too slow response!

f_h = period of camera frame arrival;
h = sampling period or start of two successive sensor processing
τ = sensing-to-actuator delay

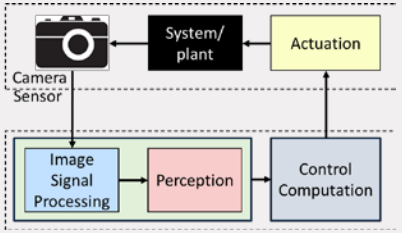
How do we deal with it?

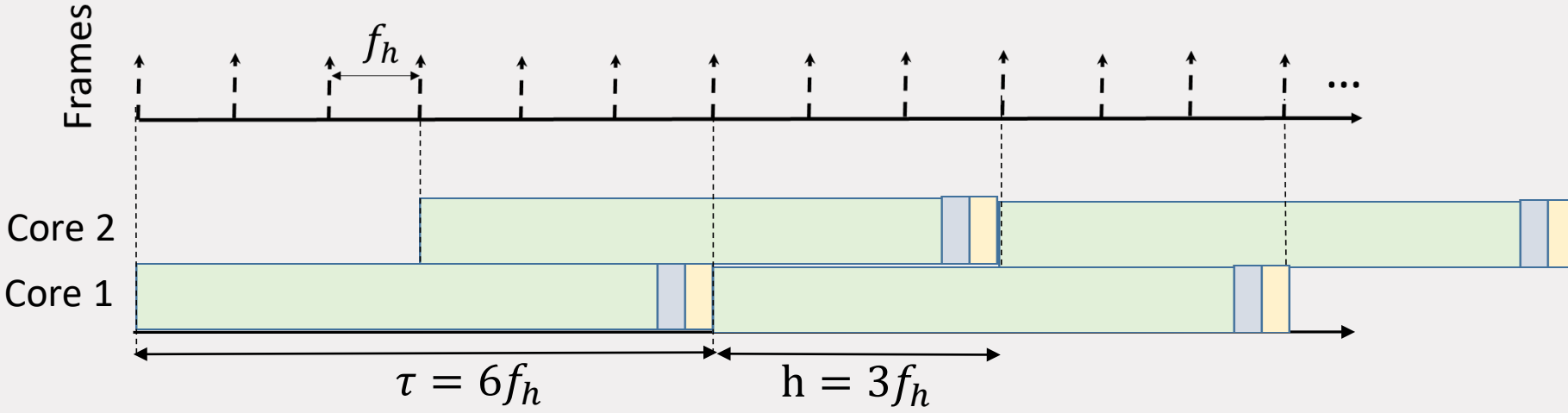


- Pipelining sensing task → shorten sampling period h and unaltered long delay τ
- Parallelizing sensing task → shorten both sampling period h and delay τ; limited by degree of parallelism;
- Approximation of the sensing block → shorten both sampling period h and delay τ;



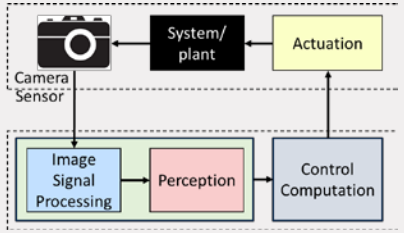
Soln 1: Pipelined control

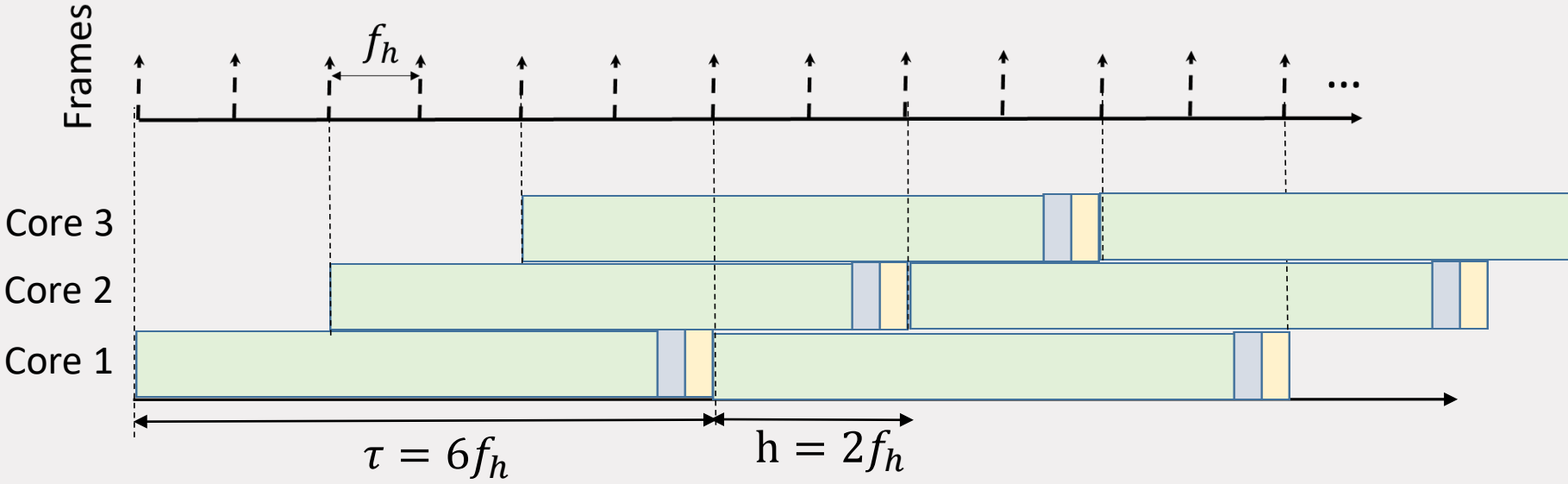




- Sampling period $\mathrm{h} = \frac{\tau}{2} = 3f_h$; Delay = $\tau = 6f_h$

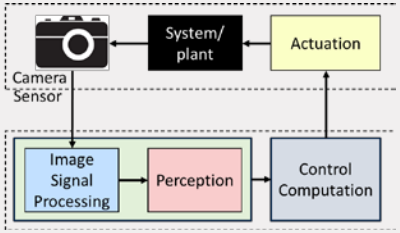
Pipelined control



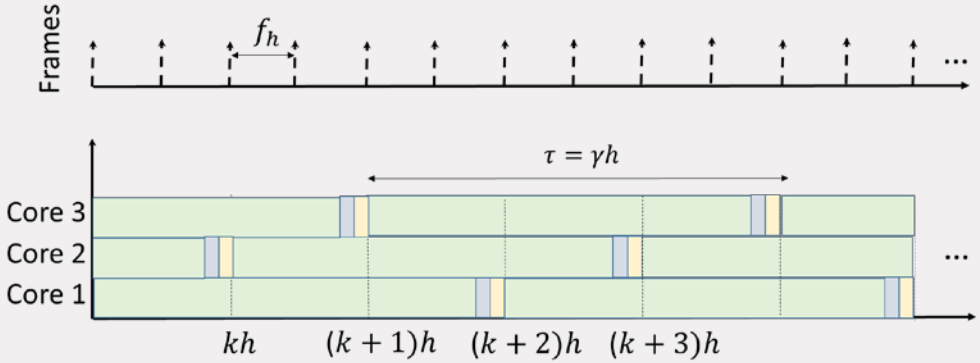


- Sampling period $\mathrm{h} = \frac{\tau}{3} = 2f_h$; Delay = $\tau = 6f_h$

Pipelined control



- Sampling period $\mathrm{h} = \frac{\tau}{\gamma}$
- γ= number of available control computing cores



- Plant in discrete-time: $x(kh + h) = \phi x(kh) + \Gamma\, u(kh - \gamma h)$

Pipelined control: model

Discretized model: $x(kh+h) = \phi x(kh) + \Gamma\, u(kh-\gamma h)$

Define augmented states:

$$z(kh) = \begin{bmatrix} x(kh) \\ u(kh-\gamma h) \\ u(kh-\gamma h+h) \\ u(kh-\gamma h+2h) \\ \dots \\ u(kh-h) \end{bmatrix}$$

$$z(kh+h) = [\dots]z(kh) + [\dots]u(kh)$$

Pipelined control: model

Discretized model: $x(kh+h) = \phi x(kh) + \Gamma\, u(kh-\gamma h)$

Define augmented states:

$$z(kh) = \begin{bmatrix} x(kh) \\ u(kh-\gamma h) \\ u(kh-\gamma h+h) \\ u(kh-\gamma h+2h) \\ \dots \\ u(kh-h) \end{bmatrix}$$

$$z(kh+h) = \begin{bmatrix} x(kh+h) \\ u(kh-\gamma h+h) \\ u(kh-\gamma h+2h) \\ u(kh-\gamma h+3h) \\ \dots \\ u(kh) \end{bmatrix} = \begin{bmatrix} \phi x(kh) + \Gamma\, u(kh-\gamma h) \\ u(kh-\gamma h+h) \\ u(kh-\gamma h+2h) \\ u(kh-\gamma h+3h) \\ \dots \\ u(kh) \end{bmatrix}$$

Pipelined control: model

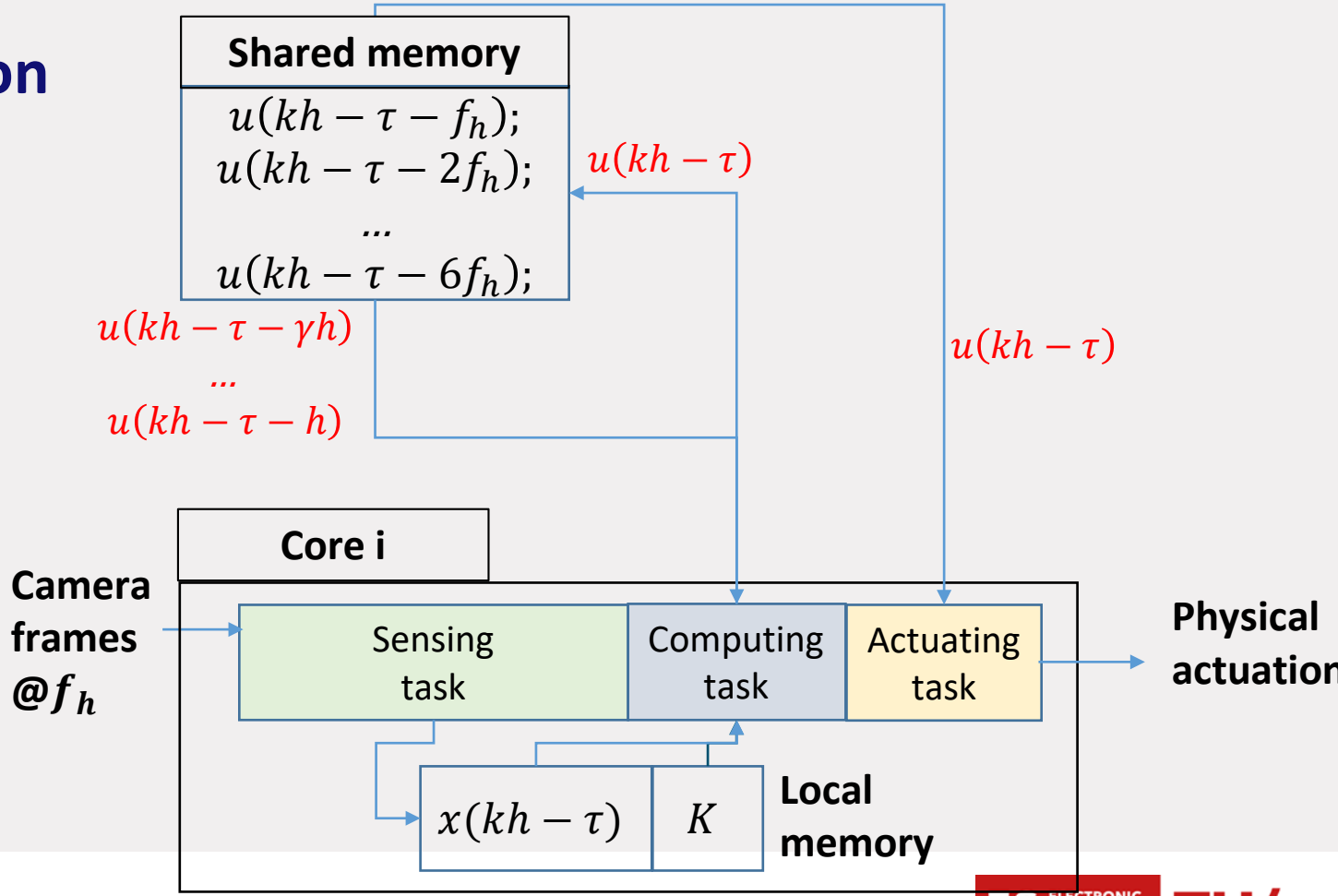
Discretized model: $\boldsymbol{x(kh+h) = \phi x(kh) + \Gamma\, u(kh-\gamma h)}$

$$z(kh+h) = \begin{bmatrix} x(kh+h) \\ u(kh-\gamma h+h) \\ u(kh-\gamma h+2h) \\ u(kh-\gamma h+3h) \\ \dots \\ u(kh) \end{bmatrix} = \begin{bmatrix} \phi x(kh) + \Gamma\, u(kh-\gamma h) \\ u(kh-\gamma h+h) \\ u(kh-\gamma h+2h) \\ u(kh-\gamma h+3h) \\ \dots \\ u(kh) \end{bmatrix}$$

$$z(kh+h) = \begin{bmatrix} \phi & \Gamma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} z(kh) + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 1 \end{bmatrix} u(kh)$$

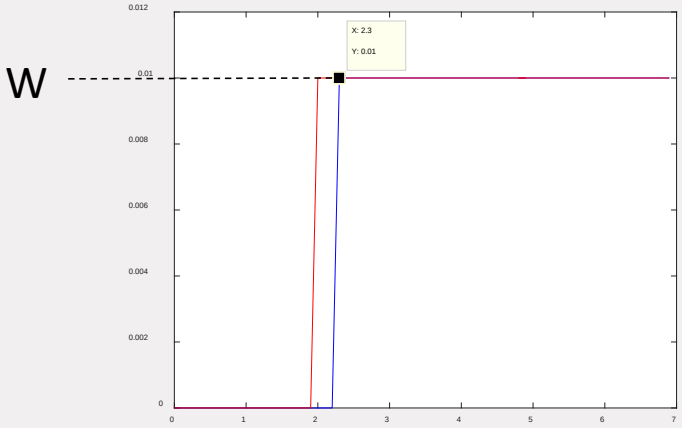
Controller: $\boldsymbol{u(kh) = K\, z(kh)}$

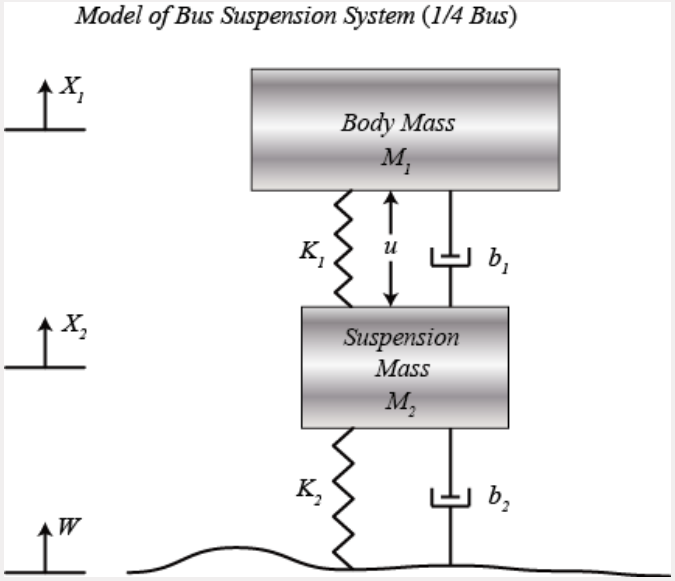
Implementation architecture



Pipelined control: Predictive suspension system

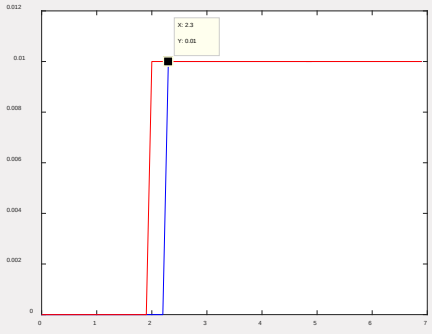
- Sequential implementation
- Settling time = 400 ms → too slow



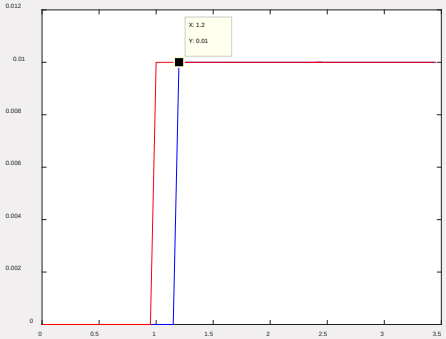


Camera is used to sense W

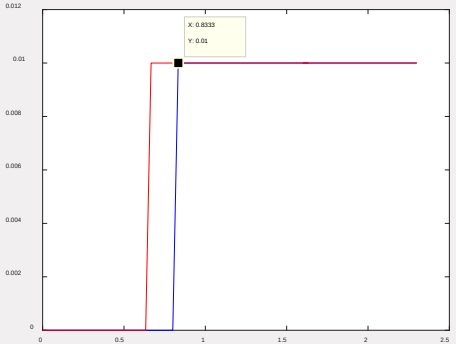
Performance vs cores



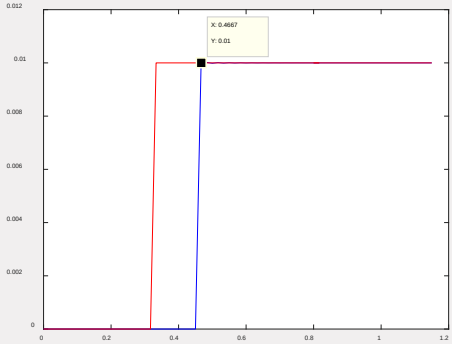
$\gamma = 1$
Settling time = 400 ms



$\gamma = 2$
Settling time = 250 ms

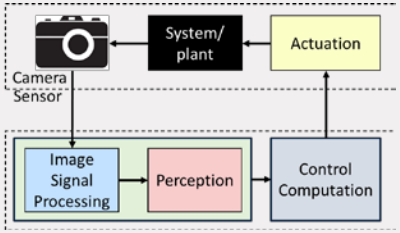


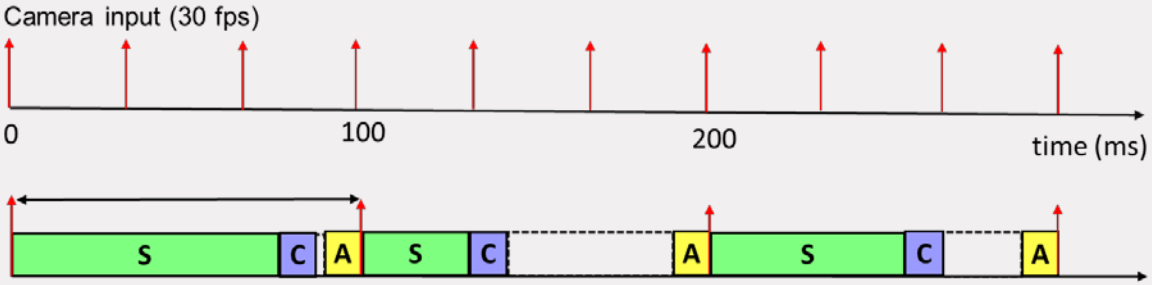
$\gamma = 3$
Settling time = 200 ms



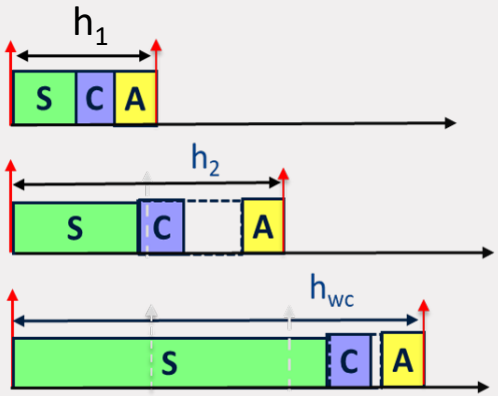
$\gamma = 6$
Settling time = 150 ms

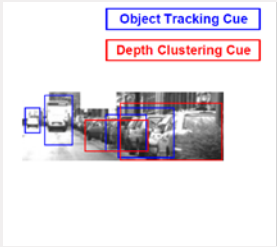
Soln 2: Parallelized sensing operation

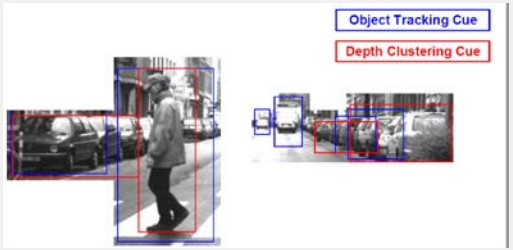




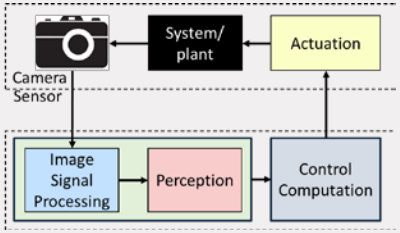
- WC design is resource inefficient;
- Workload scenarios: wide variation;
- Average-case execution time is much lower than the WC execution time



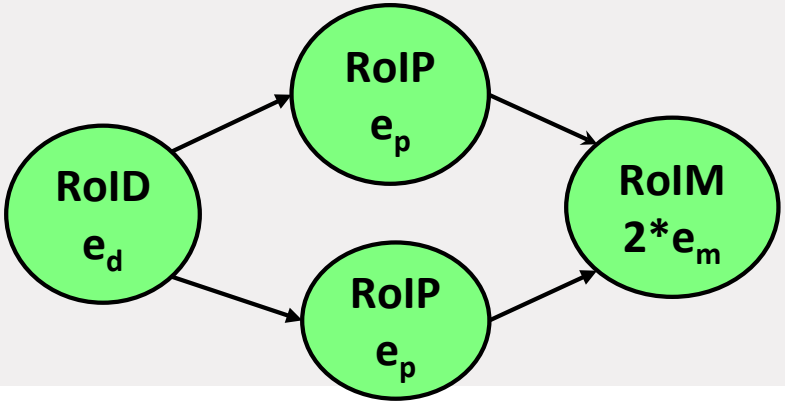


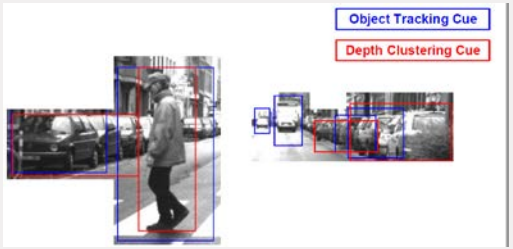


Parallel sensor processing

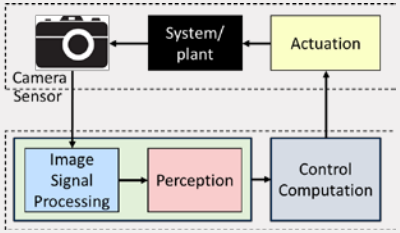


- Identify parameters in input data that relates to workload
- For our example, workload is quantified by #RoI (=w)
- #RoI for workload, w = 2

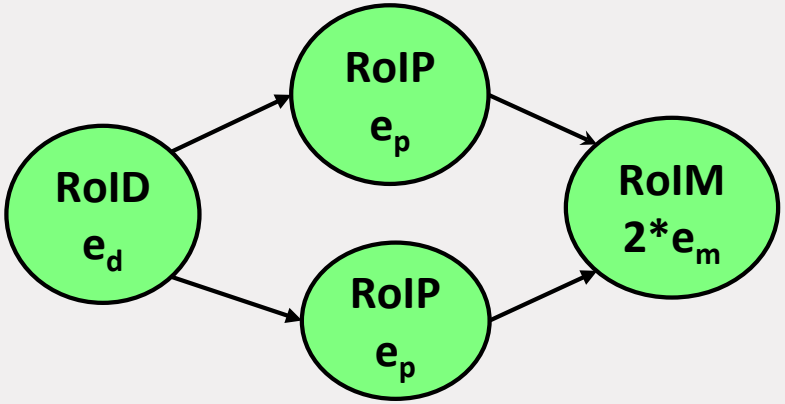


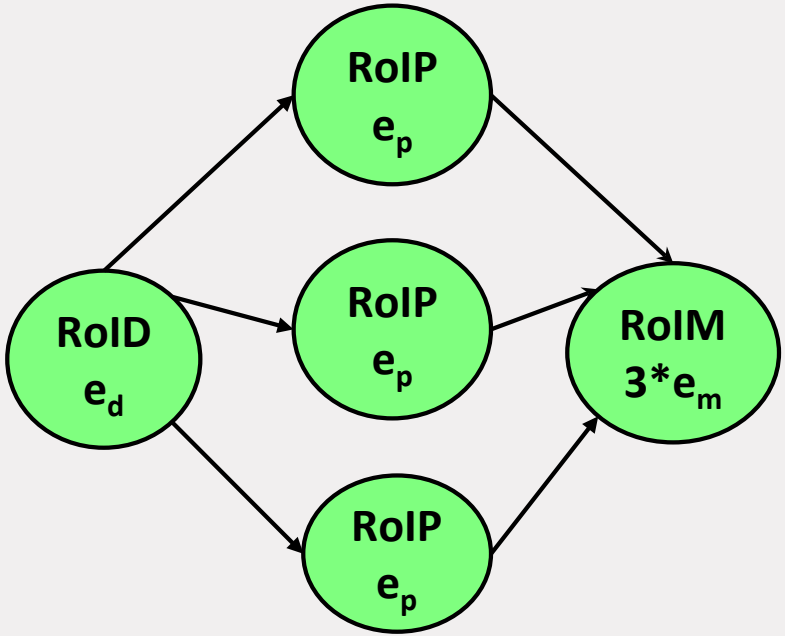


Workload variation in sensing

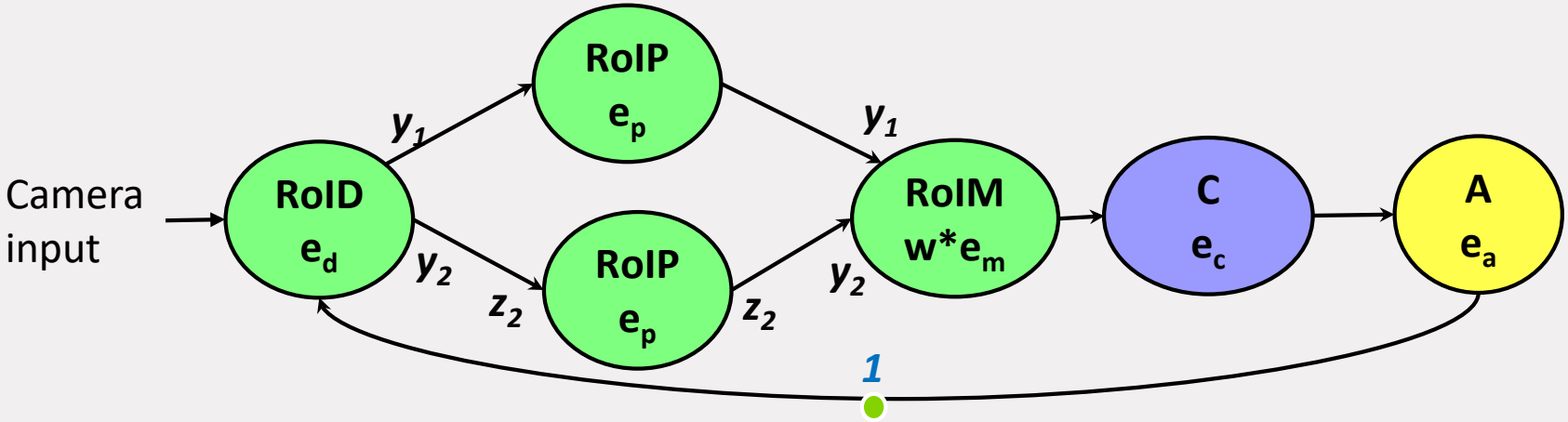


- #RoI for workload, w = 2
- #RoI for workload, w = 3





Application model: dataflow – 2-core case



$$y_1 = \{1, \ldots, \lceil w/2 \rceil\}$$

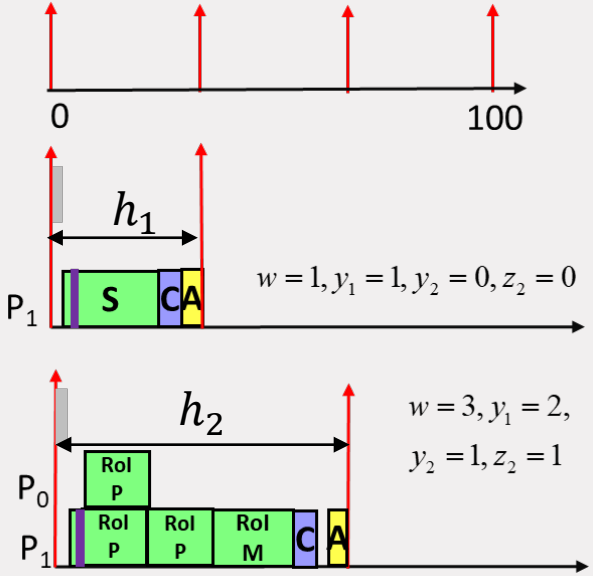
$$y_2 = \{0, 1, \ldots, \lfloor w/2 \rfloor\}$$

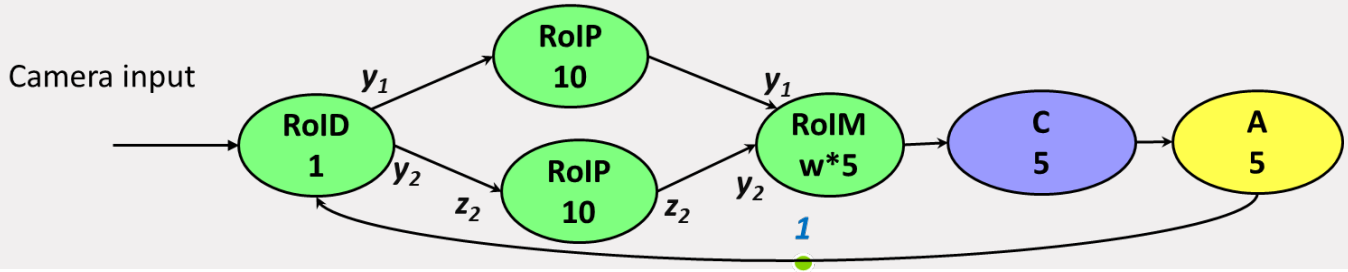
$$y_1 + y_2 = w$$

$$z_2 = \begin{cases} 0, & \text{if } y_2 = 0 \\ 1, & \text{otherwise} \end{cases}$$

Application mapping

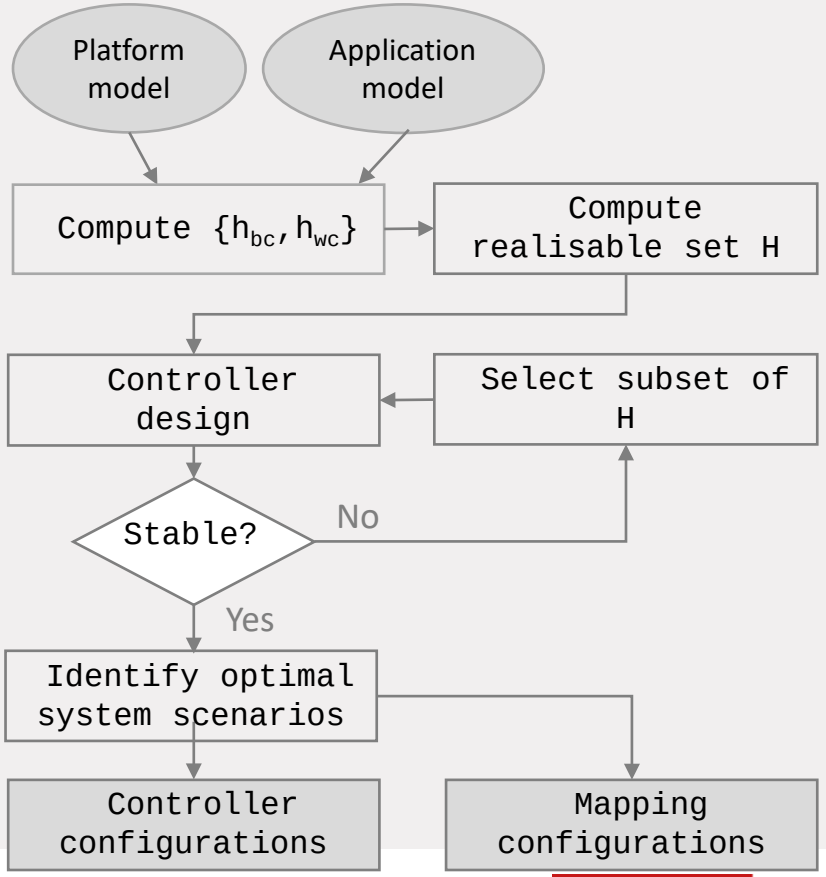
- Mapping depends on available cores;
- Different mapping gives different mapping period and hence, difference control performance;



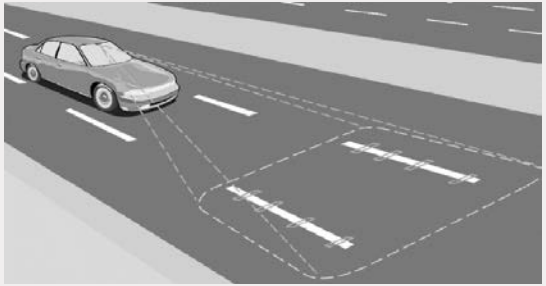


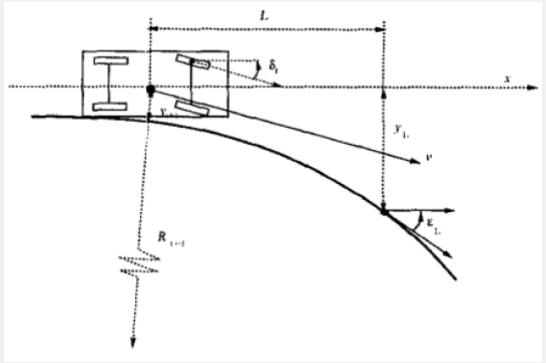
IBC design: parallelization

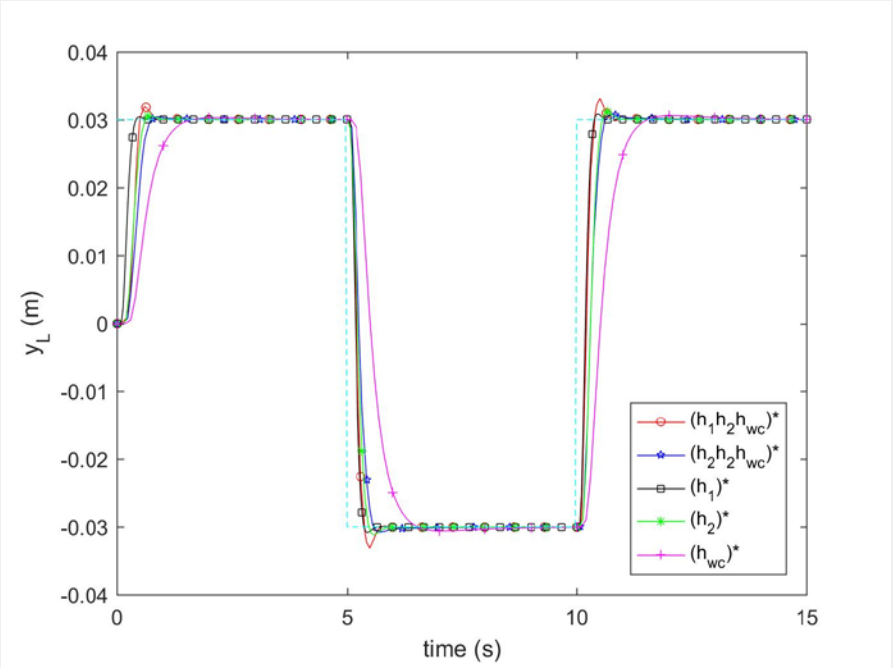
- Compute the best-case and worst-case sampling periods
- The realizable set H is frame-rate depended discretized set
- System = controller + mapping



IBC design: parallelization

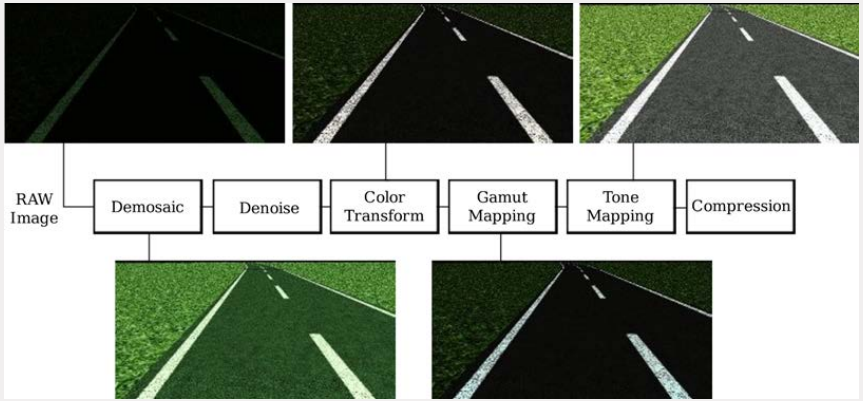






Soln 3: Approximation-in-the-loop

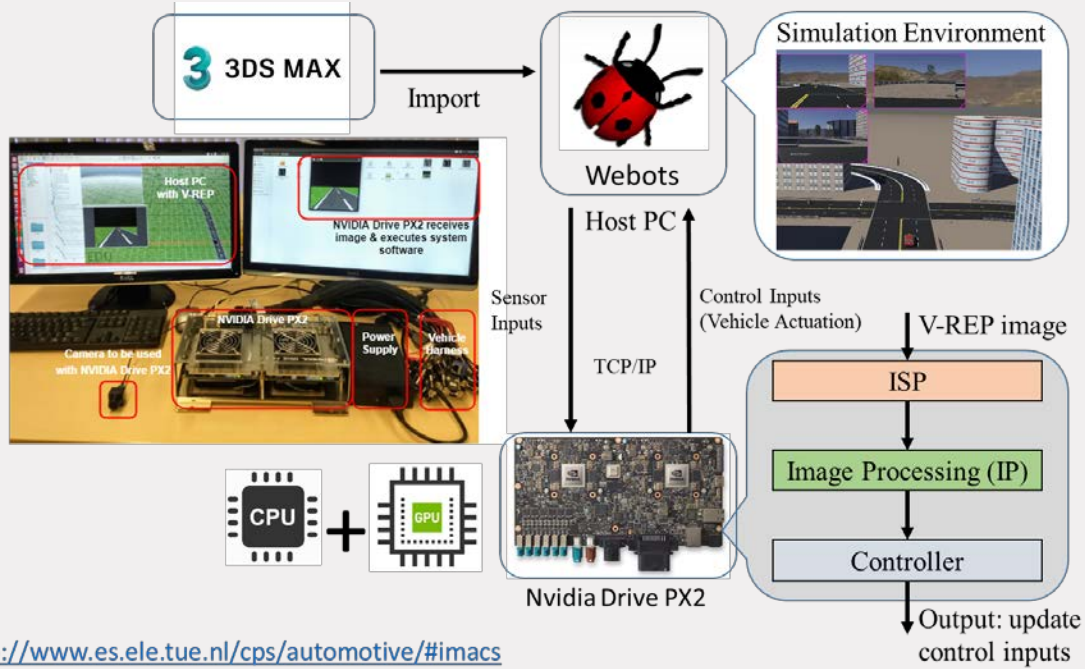
- ISP Approximation: tuning knobs



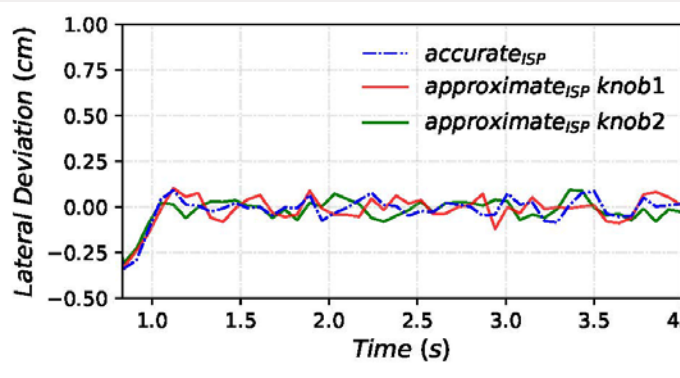
Version	ISP Stages	Explanation
v0	None	No change to data
v1	Rto, Rg, Rtr, Ftr, Fto	Skip gamut mapping
v2	Rto, Rg, Rtr, Ftr, Fg	Skip tone mapping
v3	Rto, Rg, Rtr, Fg, Fto	Skip color transform
v4	Rto, Rg, Rtr, Ftr	Only do color transform
v5	Rto, Rg, Rtr, Fg	Only do gamut mapping
v6	Rto, Rg, Rtr, Fto	Only do tone mapping
v7	Rto, Rg, Rtr	Reverse to demosaic

	v0	v1	v2	v3	v4	v5	v6	v7
Delay (ms)	68.3	24.3	57.6	69.6	20.1	55.2	23	19.5
Sampling Period (ms)	70	25	60	70	25	60	25	20

Performance evaluation: HiL setup



http://www.es.ele.tue.nl/cps/automotive/#imacs



Conclusions

- Image based control – getting popular due to low cost of processing and imaging technology;
- Main bottleneck is image signal processing which takes 80-90% of the total runtime and energy in a typical control loop;
- ISP needs low latency and high throughput (a harder constraint due to periodicity requirement in sampling period);
- Pipelining, parallelizing and approximating – three explored directions have potential;