Control Performance-Aware Task Mapping and Schedule Synthesis for Distributed Controllers on Multiprocessor Platforms

Samarjit Chakraborty*

joint work with

Harald Voit*, Reinhard Schneider*, Dip Goswami*, Anuradha Annaswamy[+]

*TU Munich, Germany, [+]Massachusetts Institute of Technology, USA

Problem Description

- Gap between high-level control models and their actual implementations
- Platform architectures consist of multiple processors connected by one or more communication buses
- Multiple control applications share a platform and need to be mapped and scheduled appropriately
- Implementation platform has an impact on control performance

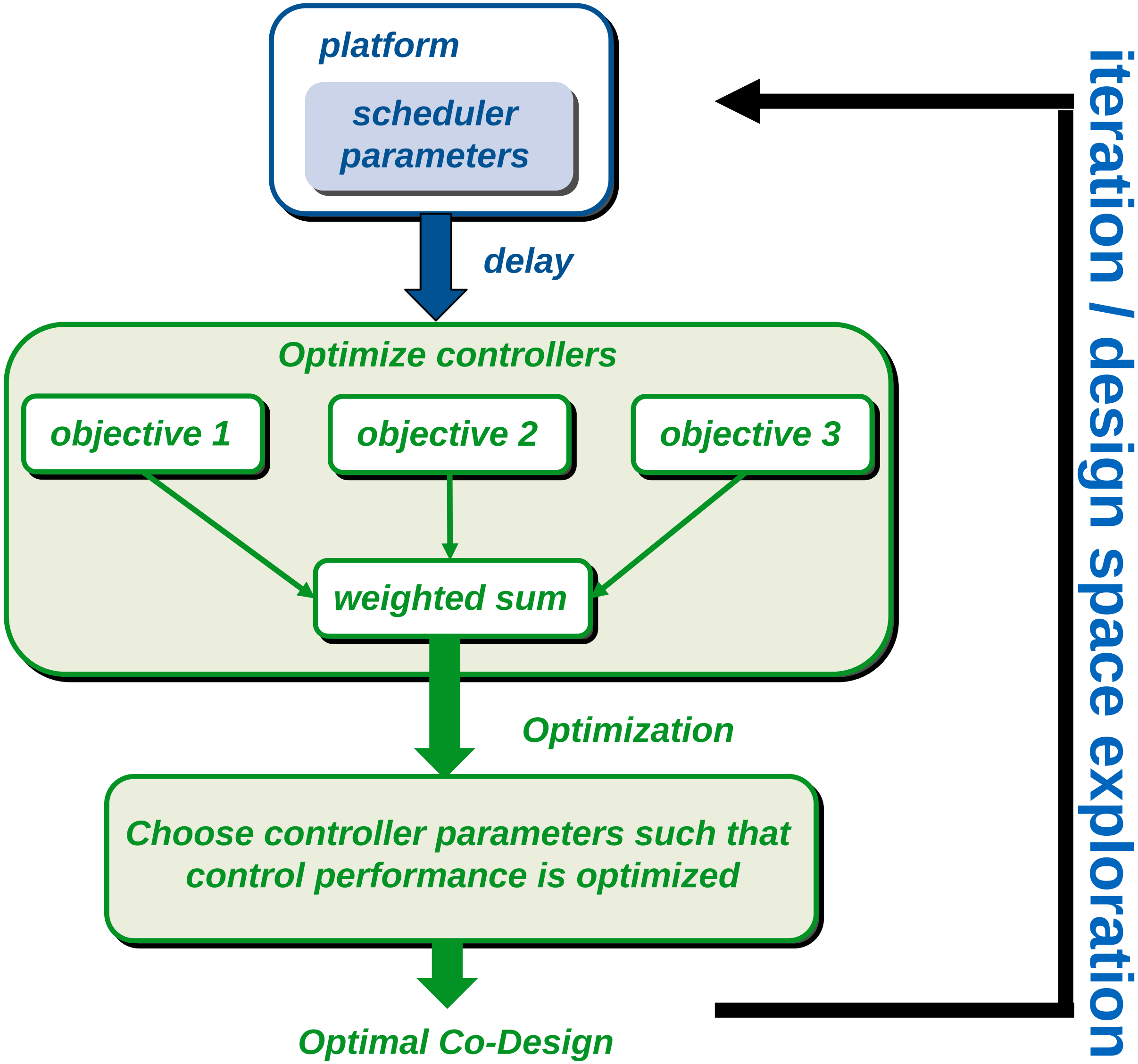
How to quantify or account for the semantic gap between the control models and their implementations?

Controller-aware platform design / Controller-Platform Co-design

Setup

- Multiple feedback controllers being implemented on a platform consisting of multiple processing units (PUs)
- PUs communicate over a shared bus according to a hierarchical scheduling policy
- Several performance metrics reflecting system properties
- Closed form formulation of delays as a function of scheduler parameters
- Delay values are used to estimate control performance
- Identification of optimal scheduling parameters with respect to control performance

High-Level Design Flow

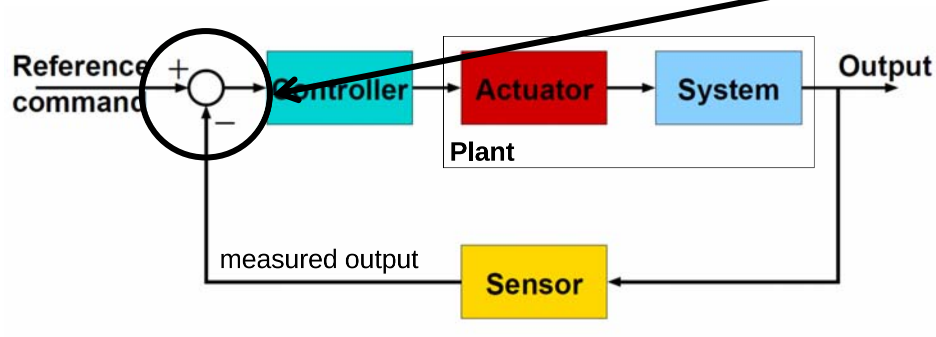


Challenges

- Improving one performance metric might deteriorate others
- Improving the performance of one controller might adversely affect the performance of the others
- For the controllers we study, control performance improves monotonically with decreasing delay
- However, the rate of improvement is not constant
- Hence, this becomes a challenging optimization problem
- Each choice of platform parameters is associated with a controller optimization ➔ two optimization problems coupled together

Feedback Control Systems

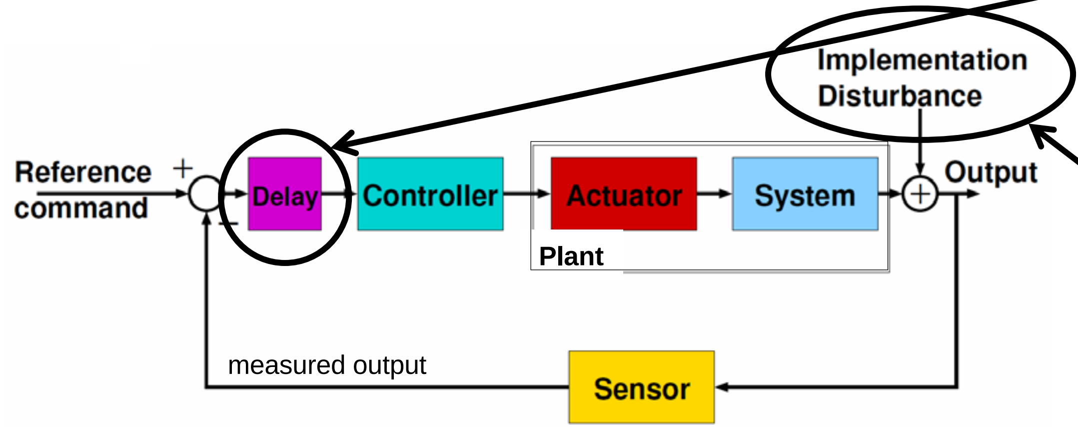
- Model of feedback control system:



Error is the difference between reference command and output

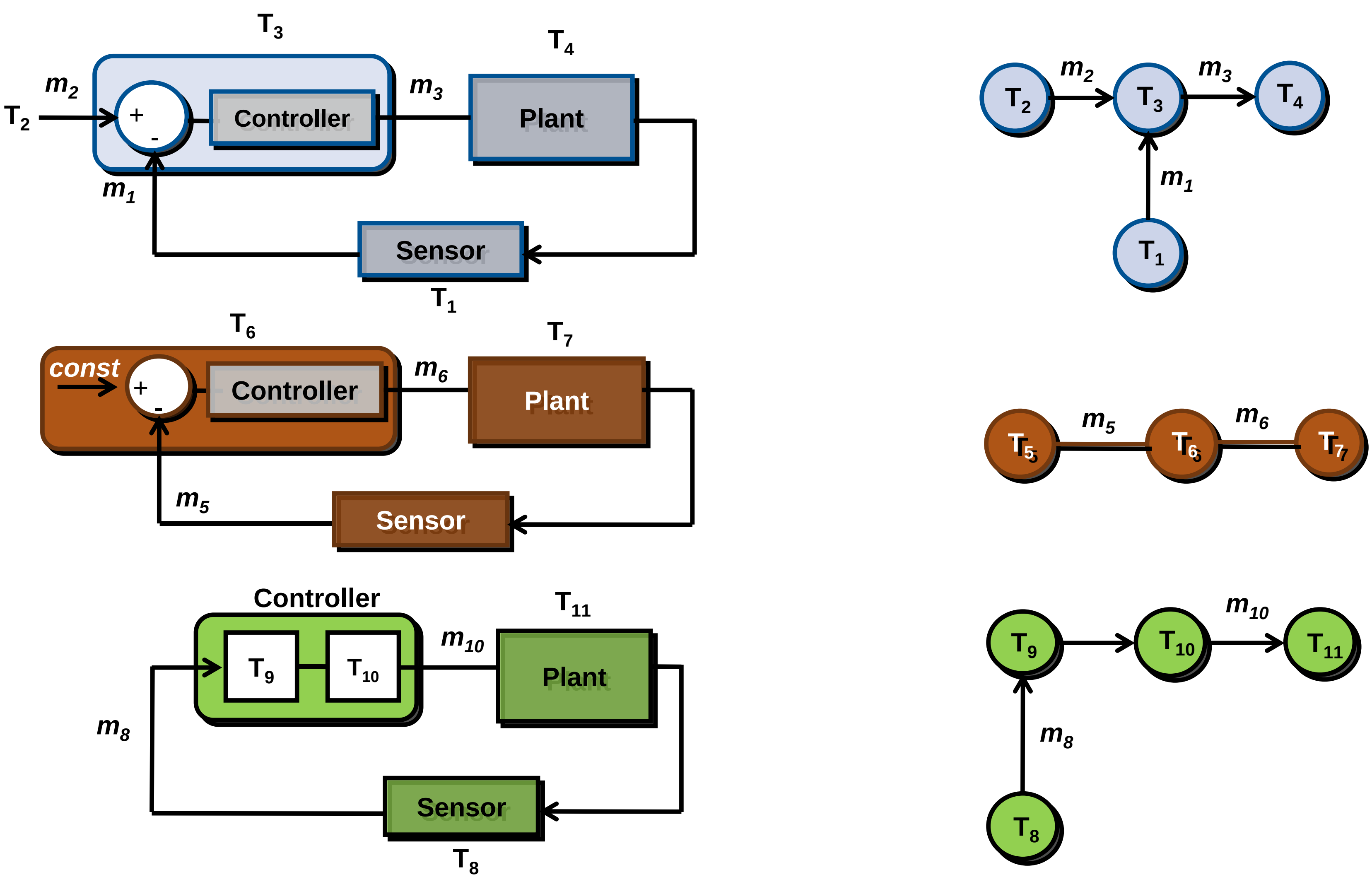
- Model of feedback control system with time delay:

Total delay due to communication and computation



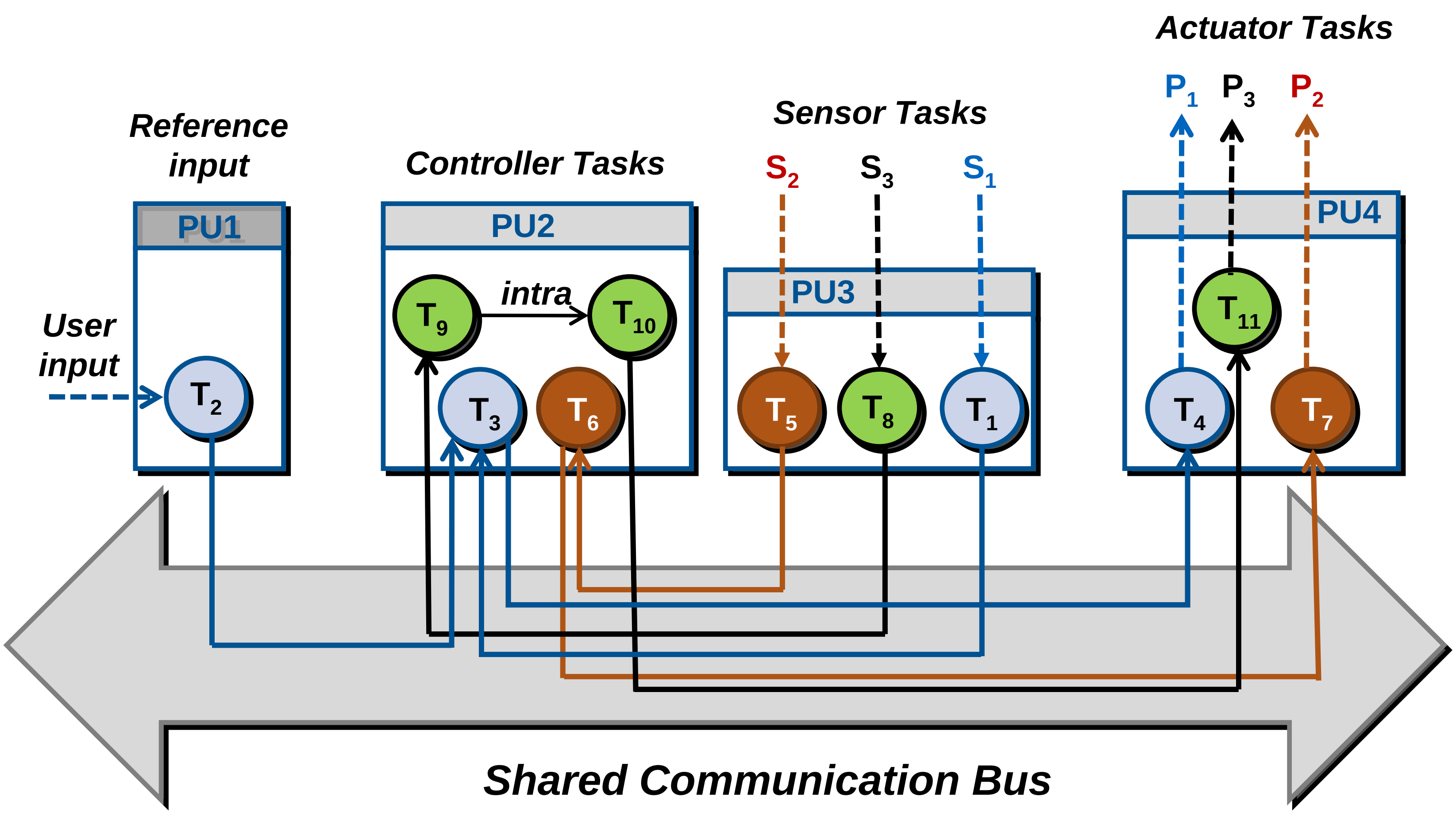
Due to sampling, discretization and quantization errors

Task Partitioning of Control Applications

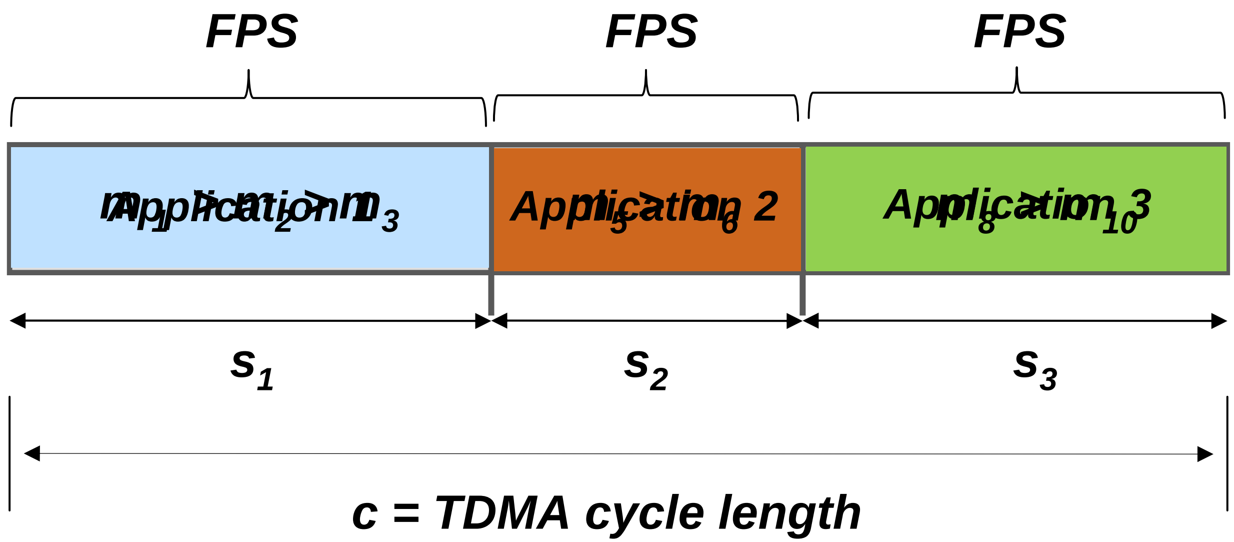


Implementation on Distriuted Architectures

- Tasks are mapped on shared processing and communication resources
- Tasks are scheduled on PUs and on the communication bus



Hierarchical Scheduling Policy



- Top-level scheduler: Time Division Multiple Access (TDMA)
- Every control application is assigned one slot
- Messages in each slot follow a fixed priority scheduler (FPS)

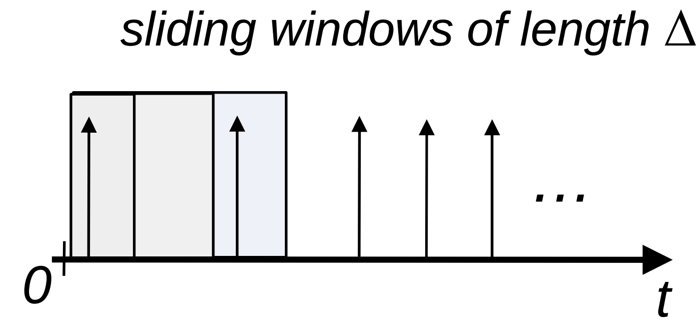
Analysis Using Real-Time Calculus

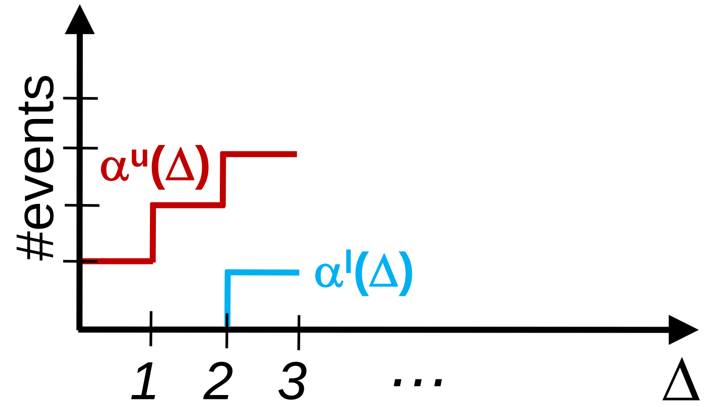
Event stream R[s,t):
Number of events that arrive in the time intervall [s,t)

count based abstraction

Arrival function $\alpha(\Delta)$***:***
Min. and max. number of events that arrive in *any* time interval of length Δ

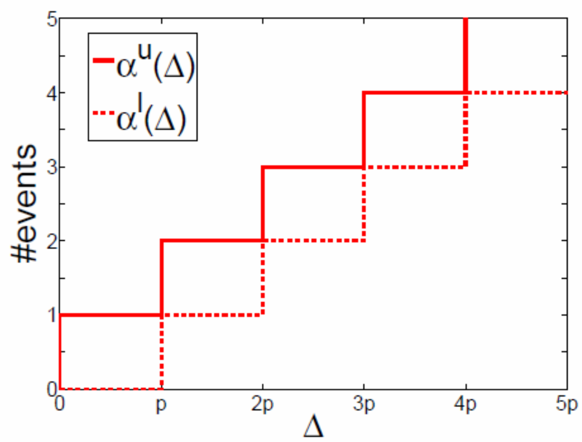
similarly



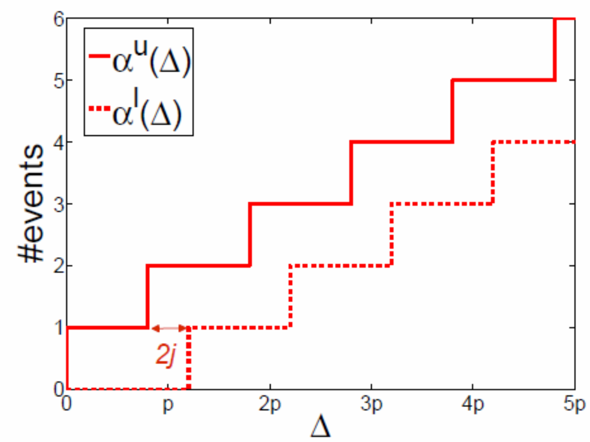


Service function $\beta(\Delta)$***:***
Models processing capacity of processor and bus resources

Examples



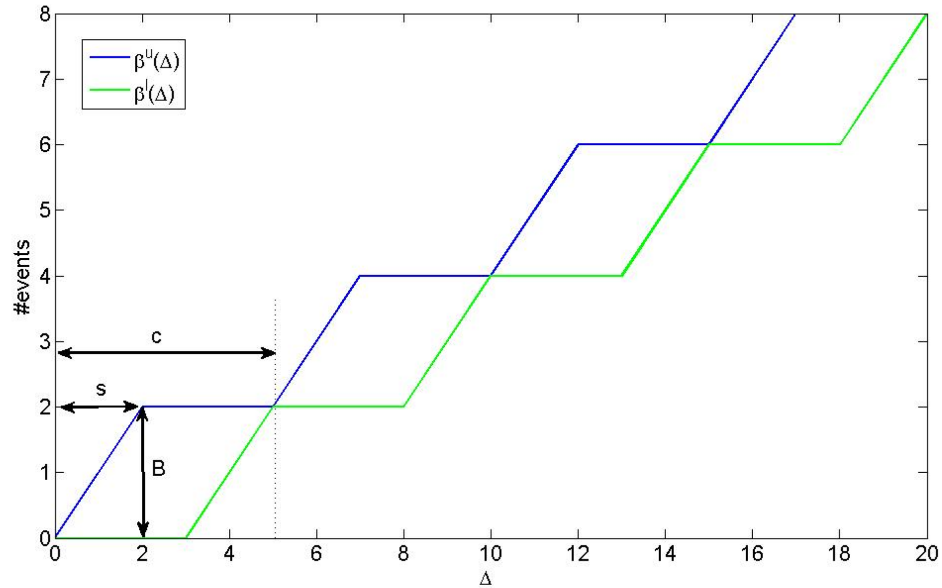
periodic



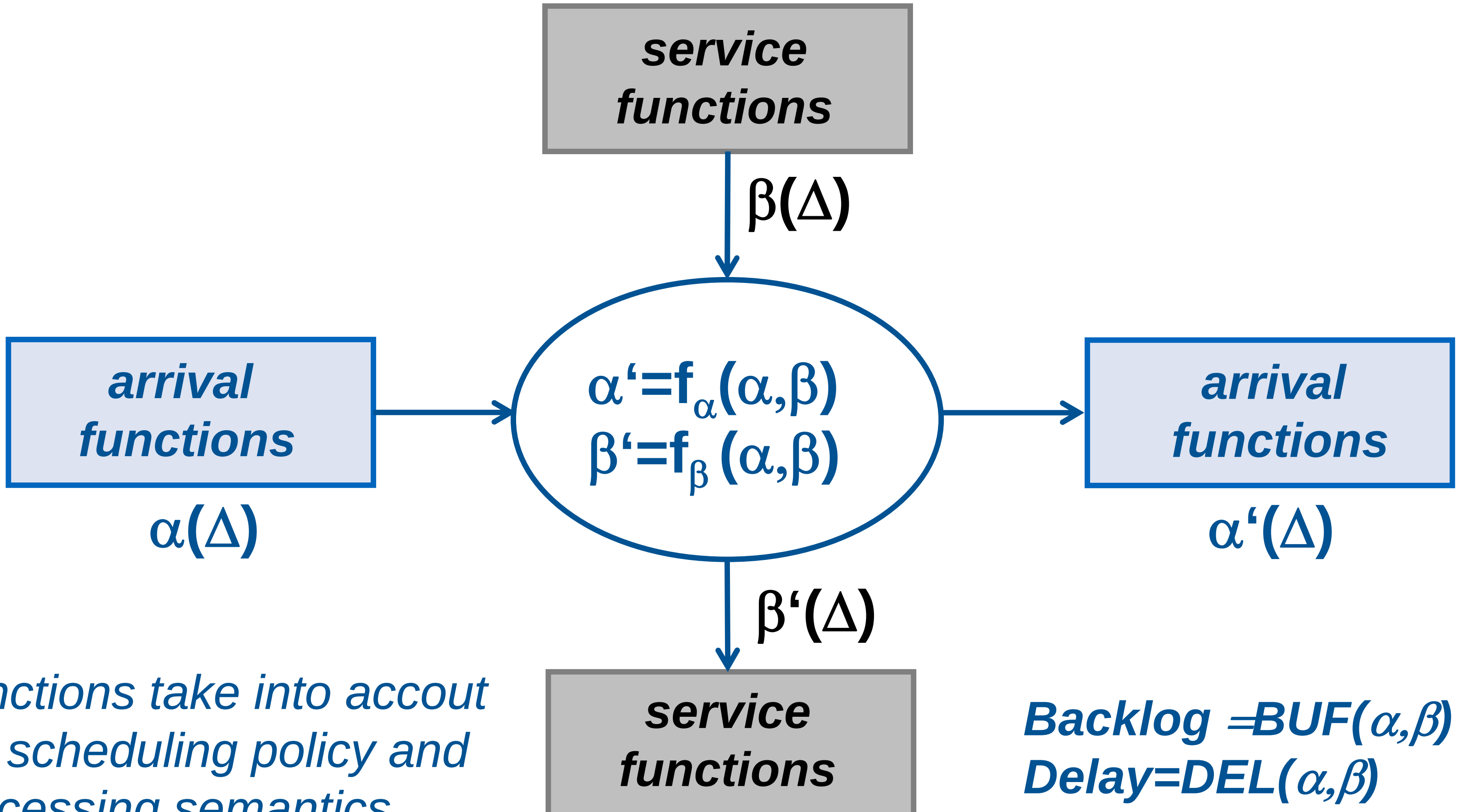
periodic with jitter

TDMA resource:

- c: TDMA round
- s: slot size
- B: bandwith



Real-Time Calculus (RTC)

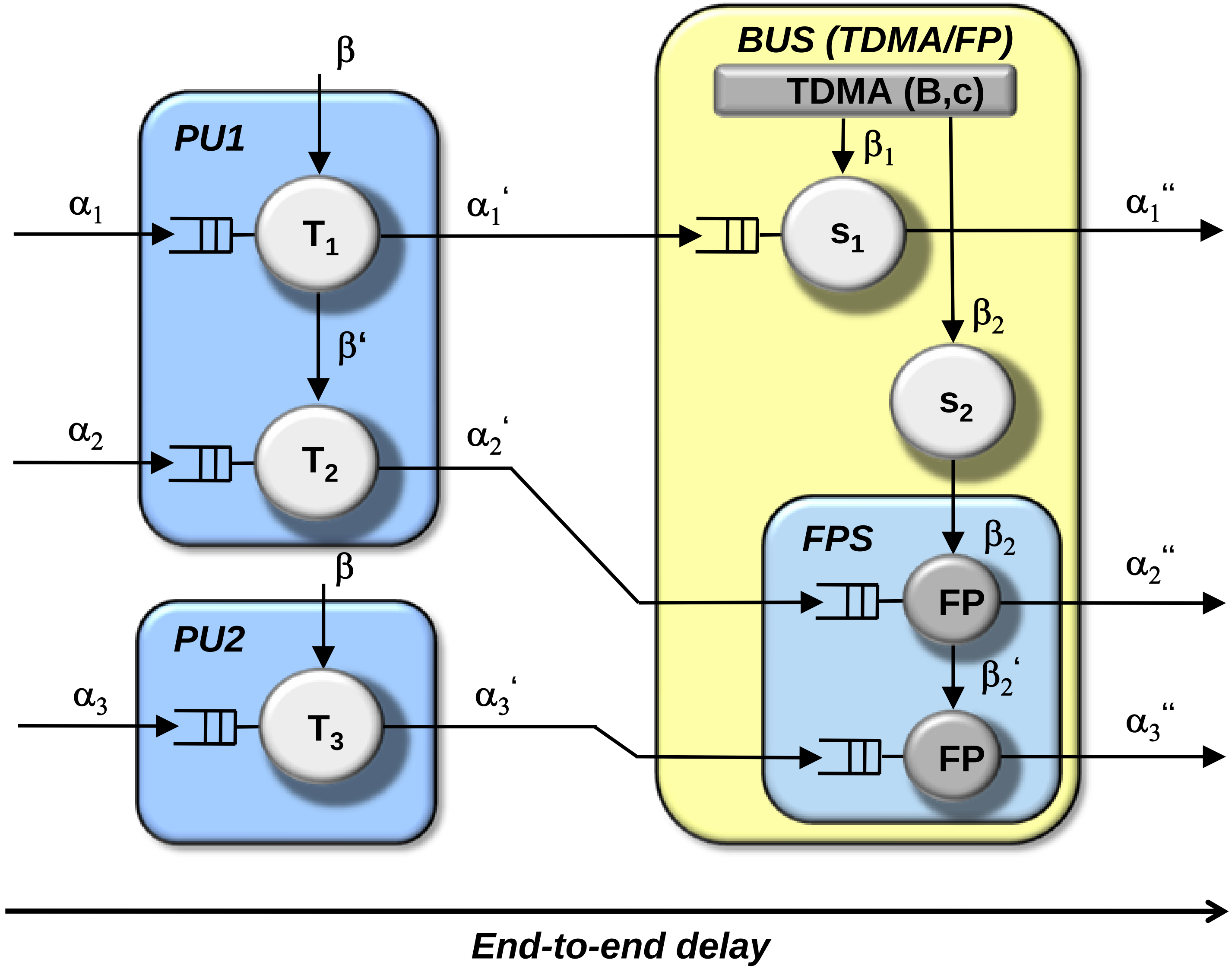


Functions take into accout the scheduling policy and processing semantics

Backlog =BUF(α,β)
Delay=DEL(α,β)

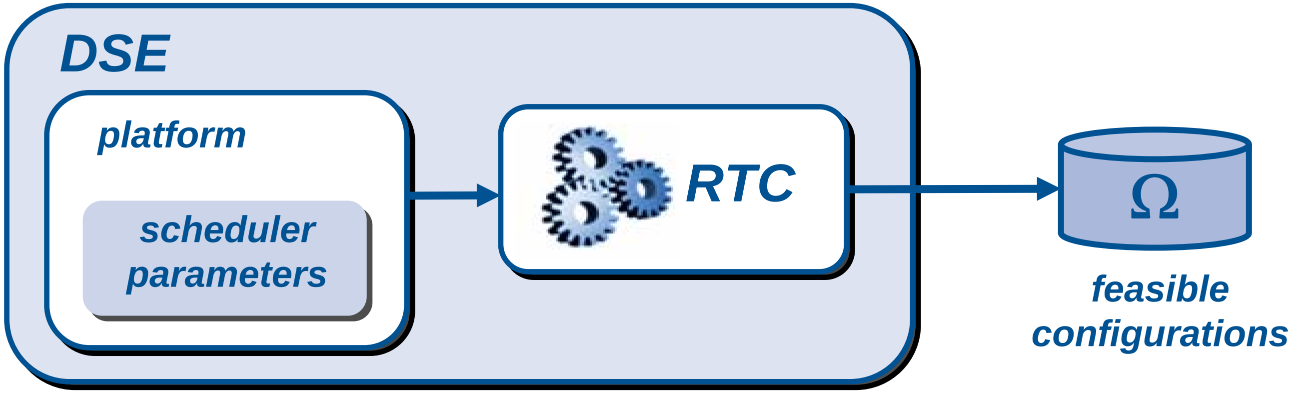
S. Chakraborty, S. Künzli, and L. Thiele. A general framework for analysing system properties in platform-based embedded system designs, In DATE, 2003

Compositional Timing Analysis



Co-Design

- End-to-end delay computation from sensor to actuator by RTC
- For each delay value, a controller needs to be optimized
- Design Space Exploration (DSE) on platform configurations



Control performance depends on end-to-end delay / scheduler parameters

System Model

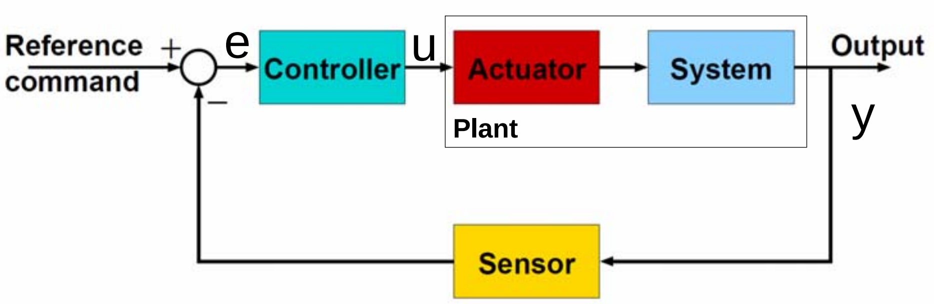
- A **plant** is a physical device, like a brake in a car, or a chemical process
- A plant is usually described by a system of linear ordinary differential equations that are transformed into Laplace-space, e.g.:

$$\ddot{y}(t) + \dot{y}(t) = K_P u(t)$$

$$Y(s)s^2 + Y(s)s = K_P U(s)$$

$$\frac{Y}{U} = \frac{K_P}{s^2+s}$$

s: Laplace variable
K_P: Plant constant



$$P(s) = \frac{K_P}{s^2 + s}$$

System Model

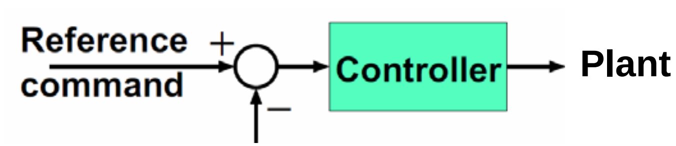
- A **controller** is a device which computes the input to the plant based on an error between measurements of the output of the plant and the reference command

- Here, we consider a Proportional-Derivative (PD) controller:

  – Proportional gain: The error is multiplied with the constants *K and b*

  – Derivative term: Adjusts the output based on the rate of change of the input. The error is multiplied by $K \times s$

  – PD controller: sum of proportional gain and derivative term



$$C(s) = K(s+b)e^{-\tau s}$$

- The term $e^{-\tau s}$ is due to the delay due to communication and computation and is the Laplace-transform of ($t - \tau$)

System Model

- Consider the system (negative unit feedback):

Plant: $$P(s) = \frac{K_P}{s^2 + s}$$

Controller: $$C(s) = K(s+b)e^{-\tau s}$$

- The **transfer function** of the closed loop system is given by:

$$G(s) = \frac{K_P K(s+b)e^{-\tau s}}{s^2 + s + K(s+b)e^{-\tau s}}$$

Control Performance

- **Stability** is (nearly) always a priority for control engineers
- Delay margin L_m denotes the amount of delay a system can tolerate before it gets unstable (if $L_m < \tau$, *the system is unstable*)
- We define the cost $\mathcal{P}_0$ for stability:

$$\mathcal{P}_0 = \frac{1}{L_m}$$

- The delay margin is determined analytically:

$$L_m(K, b, K_P) = \frac{\varphi_m(K, b, K_P)}{\omega_c(K, b, K_P)}$$

Control Performance

- The transient behaviour of a system describes the behaviour until the system reaches its steady state (if the system is stable)
- The **peak overshoot** is the maximum amplitude of the system output
- The amplitude of the peak overshoot and the question if it is feasible to have an overshoot are important design considerations
- Peak Overshoot $\mathcal{P}_1$ for transient performance is determined by:

$$\mathcal{P}_1(e) = \begin{cases} \max(g(t)) \text{ for } t > t_0 & \text{if } t_0 \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

- We <u>determined the peak overshoot numerically</u> as there are infinitely many maxima

Control Performance

- Steady state is the value of the system after transient phase
- If the controller is designed well, the output follows the reference signal, i.e., the steady state error is zero
- If the output is different from the desired one, the steady state error is not zero
- Squared integral error $\mathcal{P}_2$ denotes the **steady state performance**:

$$\mathcal{P}_2 = \int_0^{\infty} e(t)^2 dt$$

- Steady state performance is <u>determined analytically</u> using Padé <u>approximation</u>

Optimal Co-design

- Total weighted cost for a given controller j:

$$\bar{\mathcal{P}}_i^j(\theta_j) = \sum_{k=0}^{2} \lambda_k \mathcal{P}_{k,i}^j(\theta_j)$$

- Total cost over all controllers:

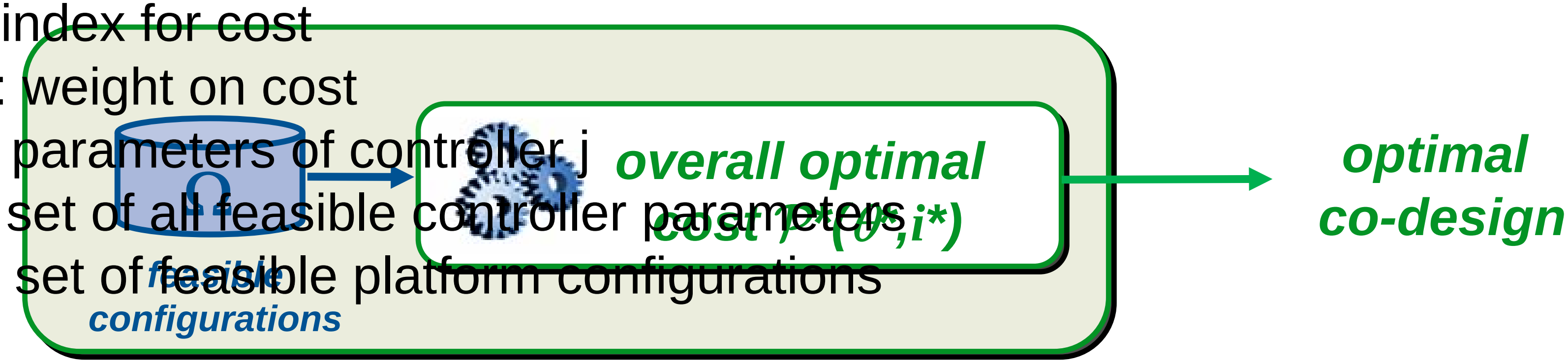
$$\tilde{\mathcal{P}}(\theta_j) = \sum_{j=1}^{m} \bar{\mathcal{P}}_i^j(\theta)$$

- Overall optimal cost:

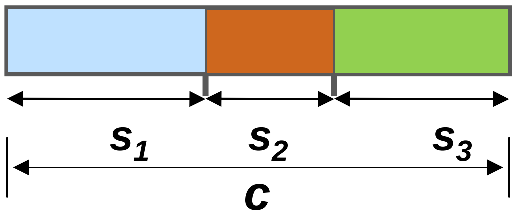
$$\mathcal{P}^*(\theta^*, i^*) = \min_{\substack{i=1,\ldots,|\Omega| \\ \theta\in\Gamma}} (\tilde{\mathcal{P}}_i(\theta))$$

i: platform configuration
j: controller
k: index for cost
λ_k: weight on cost
θ_j: parameters of controller j
Γ: set of all feasible controller parameters
Ω: set of feasible platform configurations

Co-Design objective: *Select platform configuration with optimal scheduler parameters in order to achieve maximum control performance*



Experimental Results (1/2)



overall optimal cost for different performance weights λ_k

scheduler parameters:
c = TDMA cycle length
s_1 = slot size for slot 1

feasible equally weighted configurations

performance metrics P_k

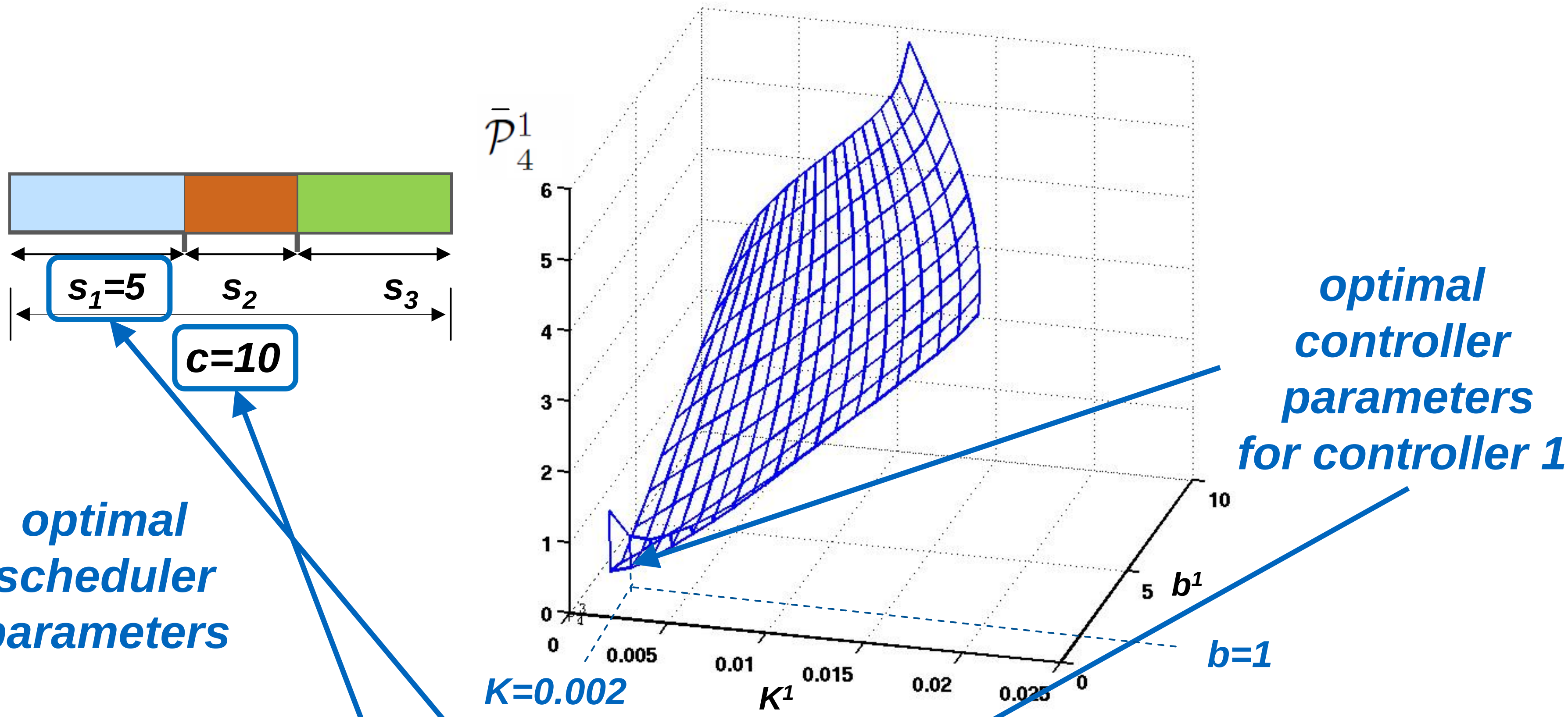
config.	c [ms]	s_1 [ms]	$\lambda_k = \frac{1}{3}$, $k = 0, 1, 2$	$\lambda_0 = \frac{1}{21}$, $\lambda_k = \frac{10}{21}$, $k = 1, 2$
1	10	3.5	1.5579	**0.5662**
2	10	4.0	1.5540	0.5671
3	10	4.5	1.5534	0.5675
4	10	5.0	**1.5527**	0.5664
5	15	5.0	1.5775	0.5913
6	15	5.5	1.5766	0.5908
⋮	⋮	⋮	⋮	⋮

heavy weight on transient- and steady state performance

optimal design

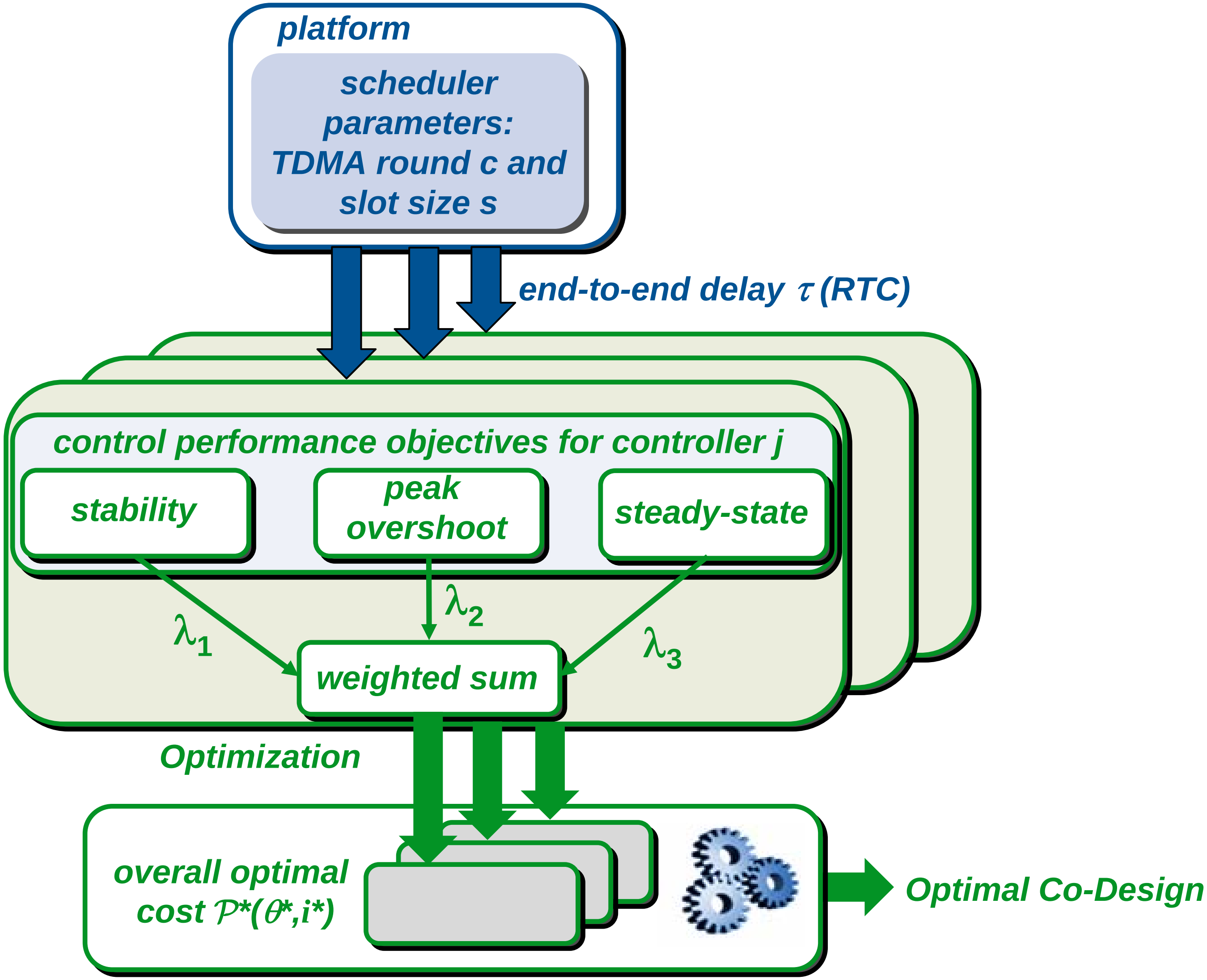
Experimental Results (2/2)

Example: total weighted cost for controller 1 with λ_k =1/3 in configuration 4



config.	$c[ms]$	$s_1[ms]$	K^1	b^1	K^2	b^2	K^3	b^3
1	10	3.5	0.0032	1	0.0034	1	0.0030	1
4	10	5.0	0.002	1	0.002	1	0.002	1

High-Level Design Flow



Concluding Remarks

- *Challenge 1*: Closed-form formulations of optimal controllers
- *Challenge 2*: Analytically compute average (instead of worst-case) delay values
- *Opportunities*
  - Automotive architectures and control software
  - Smart / zero-net energy buildings
  - Sacrifice control performance to save energy

Questions?