Designing Distributed Applications: Easing the Burden of Real-Time Systems Engineering with Control Methods

Paolo Pazzaglia,
Corporate Research, Robert Bosch GmbH

RT-Cloud 2025 08.07.25

Outline

1. Reliable Distributed Systems
2. The Research Problem
3. Implementing Control Applications in RDSs
   – Be Adaptive
   – Be Robust
4. Conclusion

1

Reliable Distributed Systems

Distributed Systems
The basic vision…

Distributed Systems
enable

functionality on **off-the-shelf distributed infrastructure** (HW&SW) across all industries

BOSCH

Distributed Systems

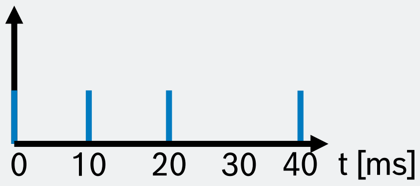
Impact of distributed systems (aka cloud) in different industries

Industry	Classic	Cloud	Key enabler
Music	 COMPACT disc DIGITAL AUDIO	 deezer Music	*Bandwidth*
Film	DVD	  NETFLIX tv	*Bandwidth*
Navigation	 	 Google Maps 280	*Wireless Coverage*
Gaming	 XBOX	  STADIA NVIDIA GEFORCE NOW	*Latency*

Cloud capabilities reduce the value of things significantly

Reliable Distributed Systems

… and the consequences



Ensure
proper timing

26262
21448

Ensure
proper operation
in case of failure

Prevent
unauthorized
influence

Timing unpredictability

Connection losses

Reliable Distributed Systems
enable

real-time,
safety-critical, and
secure

functionality on **off-the-shelf distributed infrastructure** (HW&SW) across all industries

Cloud SW stacks

RDS

Component failures

…

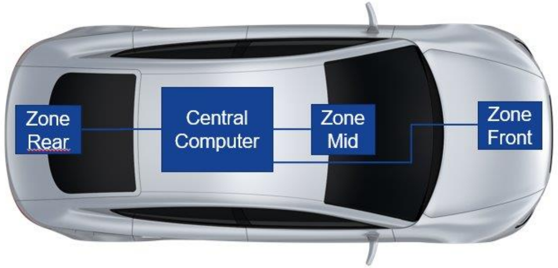
ISO26262: Functional Safety; ISO/PAS21448: SOTIF (Safety of the Intended Function)

BOSCH

Why RDS is Interesting for Bosch
A market shift

- Bosch is world-leading in things centered around **real-time**, **safety**, and **embedded security**
- Current applications run in embedded HW, sold as monolithic parts

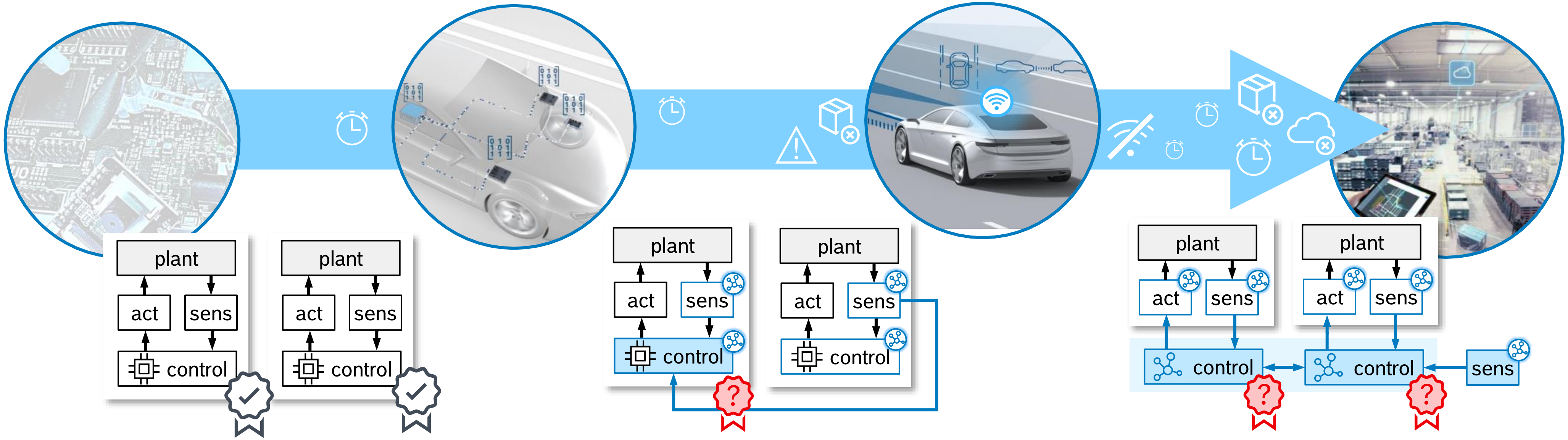
- RDSs are disruptive threat to market
  - **Bosch wants to be part of the change**



- From selling boxes (HW + SW) to selling functionalities
  - Also safety-critical! E.g.: braking, steering…

Demand & Business Cases	HW/SW Decoupling (e.g. SDV) Flexible Manufacturing Infrastructure-Based AD
Technology	Low Latency Wireless Communication Performance Isolation
Funding	Large Government Programs e.g., IPCEI-CIS in Europe

Which Kind of RDS?
Different shades of distribution



- The platform cannot hide all dynamic effects to time-critical applications
- The distribution (wired, wireless, cloud) introduces **delays and connection losses** that must be treated in application design for stability and functional performance

How to design functions with guaranteed performance in distributed systems?

BOSCH

2

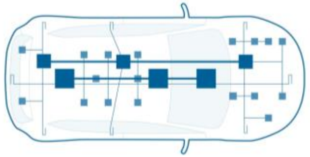
The Research Problem

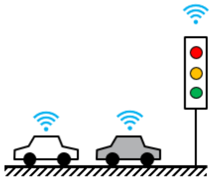
Developing Reliable Distributed Applications
Beyond the hard real-time model

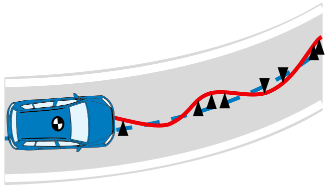
- Building truly reliable distributed systems requires **cooperation between real-time and control engineers**
  - But the two fields still pursue different objectives

- **Few control systems are truly hard real-time**
  - ...and for those you have **already dedicated** embedded **solutions**
    - ... and will not realistically be moved to distributed platforms!

- In many industrial-relevant use cases, **missing (some) deadlines and experiencing (limited, sporadic) delays and/or jitter is not a big issue**
  - Existing control methods targeting noise and uncertain parameters can be adapted to **timing uncertainties**!

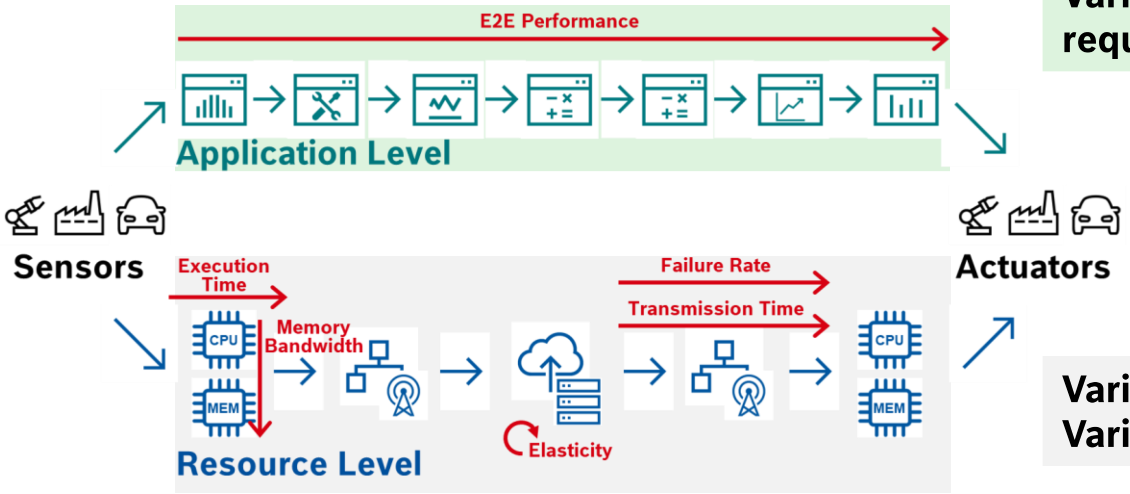






BOSCH

Developing Reliable Distributed Applications
Design Implications



Variable performance requirements

Variable computational resources
Variable communication delays

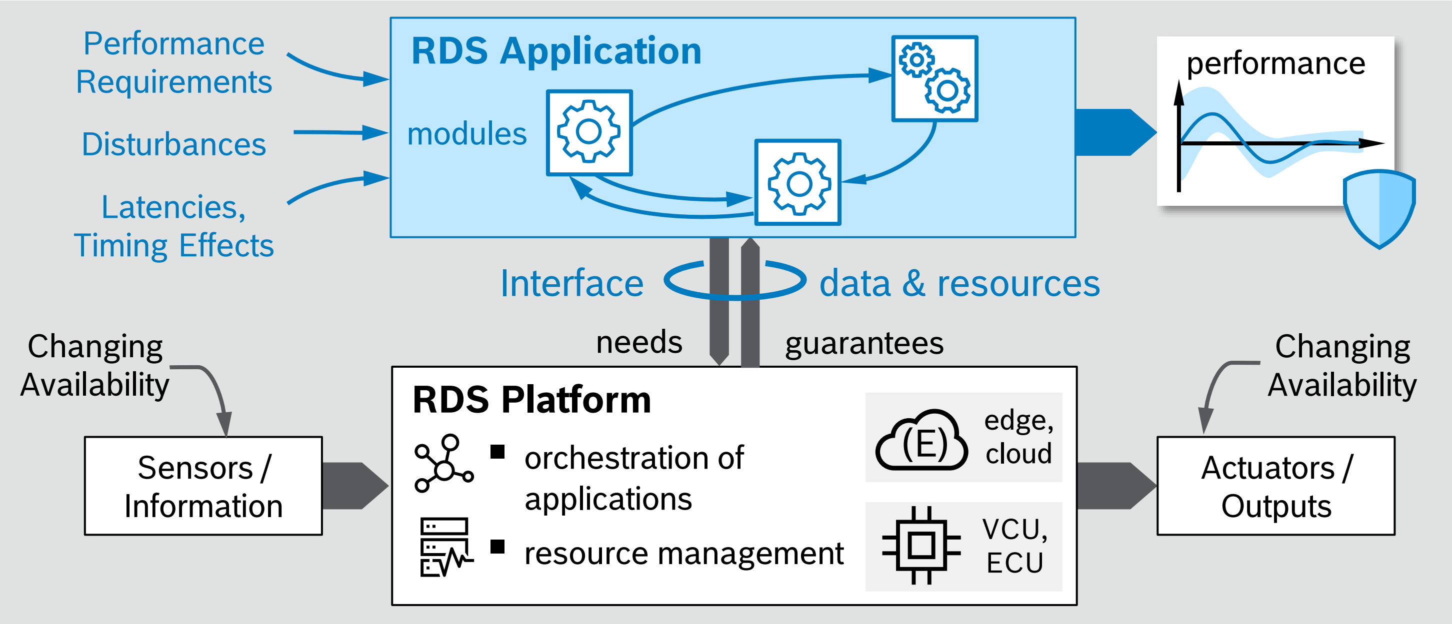
The **barrier** between functional and timing aspects **must be removed**

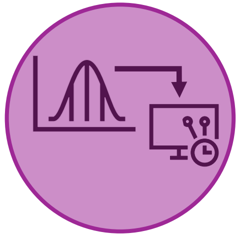
- **Need for** exchanging data between control application and real-time platform/orchestrator
  - ✓ Information flow in both directions
  - ✓ Delay/resource usage and control performance/requirements

Large design space if real-time and control methods are combined

Developing Reliable Distributed Applications

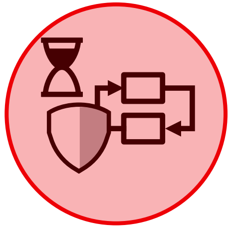
The components in operation





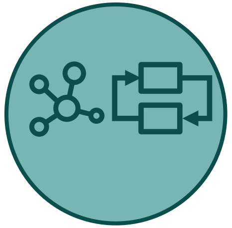
Increase Credibility

Uncertainty Quantification for timing effects integrated in system simulation



Increase Robustness

Timing-aware control design techniques with **probabilistic** methods



Increase Flexibility

Methodology for **timing-adaptive** distributed control design and platform interaction

BOSCH

You are not alone

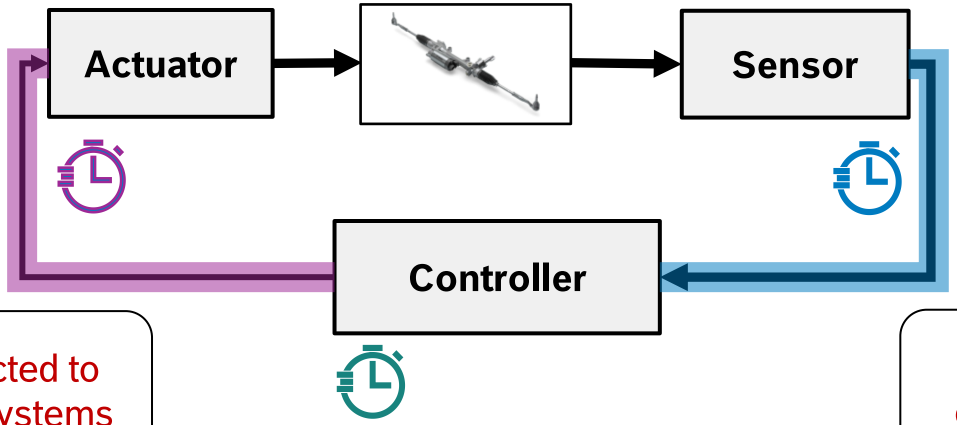
- No need to aim too high with real-time methods alone
  - Some amount of delays and some variability is inevitable
- Control applications can work surprisingly well **also with non-ideal timing conditions**

3

Implementing Control Applications in RDSs

Classical vs Modern Control

Objectives of ‘classical’ control engineering: Electric Drive



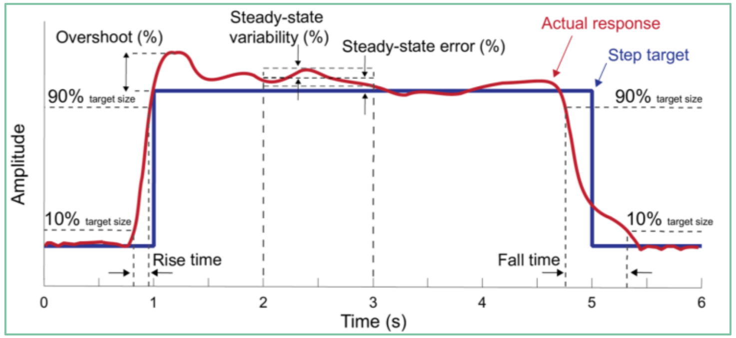
- Abstract timing models

Restricted to SISO systems

Neglects environment

Performance

- Steady-state variability
- Steady-state error

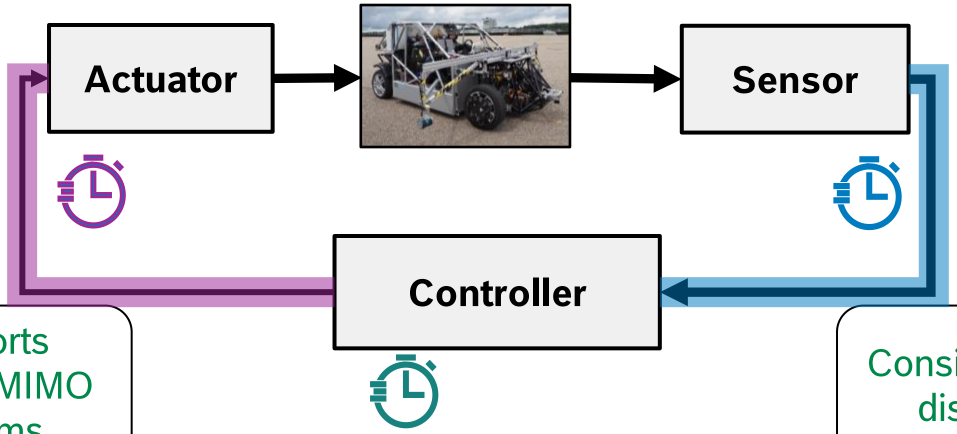


Safety

- Stability
- Rise- and Fall time
- Overshoot

Classical vs Modern Control

Objectives of ‘modern’ control engineering: Motion Control



- Abstract timing models

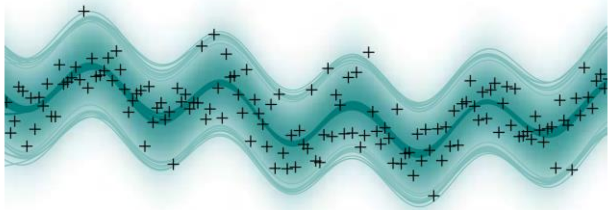
Supports general MIMO systems

Considers arbitrary disturbances

Performance

- Optimize general objective $J(\pi)$
- Consider uncertainties

$$\min_{\pi} \mathbb{E}_z[J(\pi, z)] \quad \text{s.t.} \quad \Pr(c(\pi, z) \leq p)$$



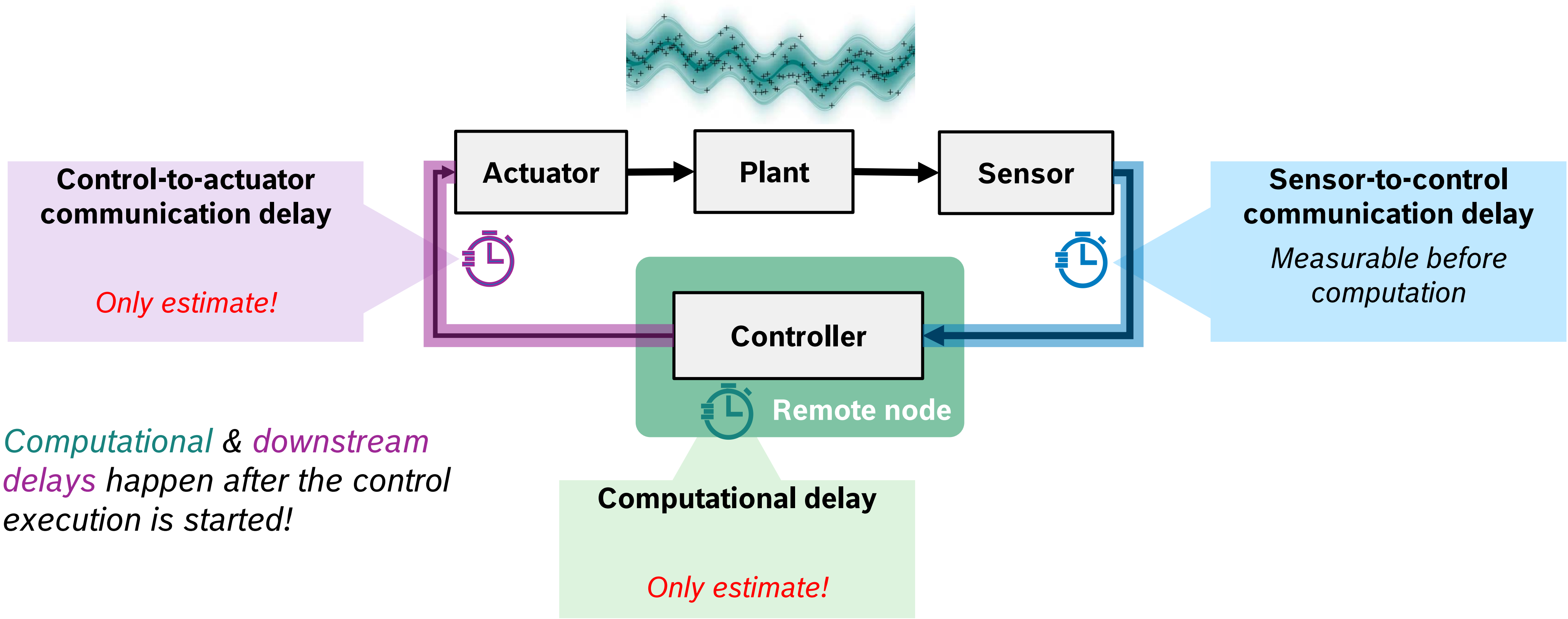
z: *stochastic uncertainty (state | plant | environment)*
π: *controller 'policy'*

Safety

- General chance constraints $c(\pi)$
- Consider uncertainties

[1] Hewing, Lukas, et al. "Learning-based model predictive control: Toward safe learning in control." *Annual Review of Control, Robotics, and Autonomous Systems* 3.1 (2020): 269-296.

Shifting to RDSs
Focus on timing uncertainties



Computational & downstream delays happen after the control execution is started!

Access to delay information can be used to design better controllers

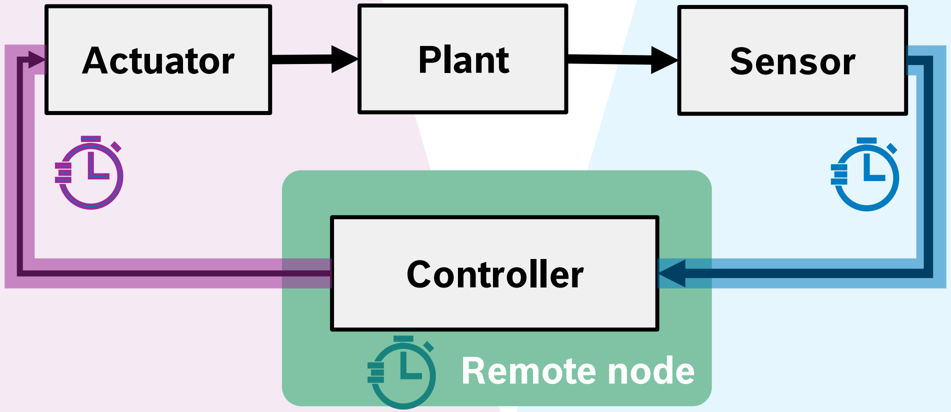
Possible Solutions
Modern control methods can handle RT uncertainties

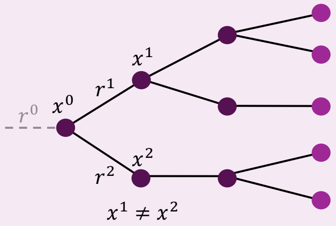
Strategy 1: “Be adaptive”

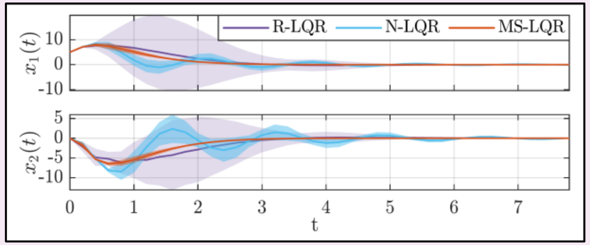
Adapt controller **at runtime** based on timing

Multi-mode control, each mode optimized for different delays

Move part of the decision away from controller



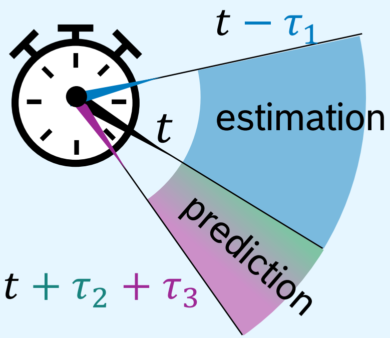


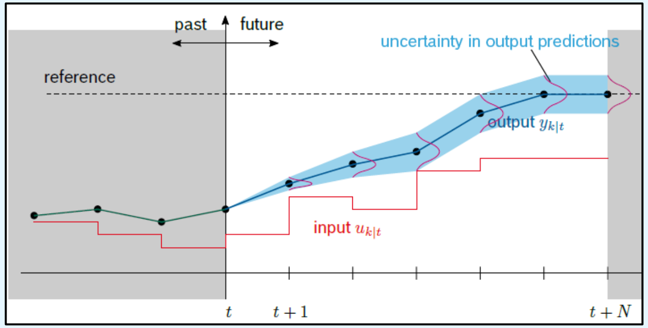


Strategy 2: “Be robust”

Design controllers that can **withstand uncertain and variable delays**

Incorporate knowledge of timing at design phase, deal with **partial** knowledge at runtime





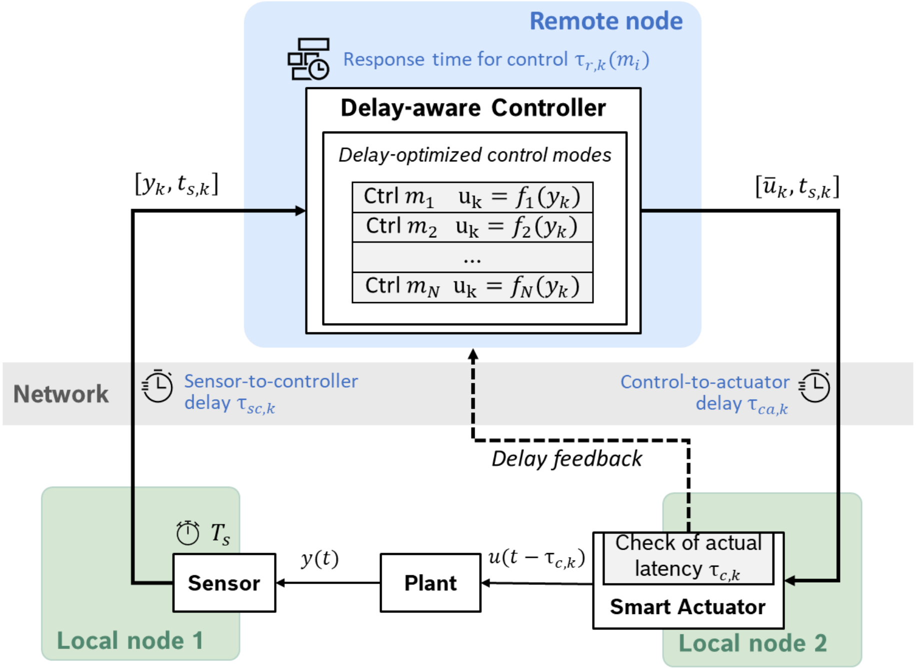
BOSCH

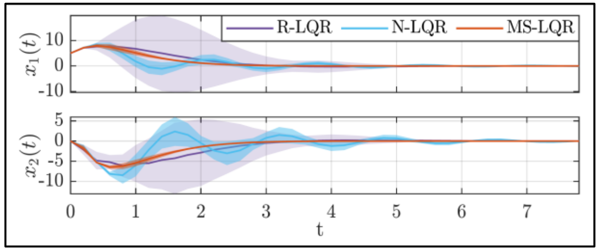
Be adaptive

Be Adaptive with Multiple Modes
Delay-aware control design

- **Delay-aware control:** features multiple modes that are run all together at remote node
- each optimized for a specific interval of sensor-to-actuator latency.

- **Smart actuator** measures the delay of received control input:
    1. chooses the best control input among the provided ones according to the current delay
    2. reacts in case of packet losses with compensation strategy
    3. provides an updated estimate of delays, to enable dynamic reconfiguration



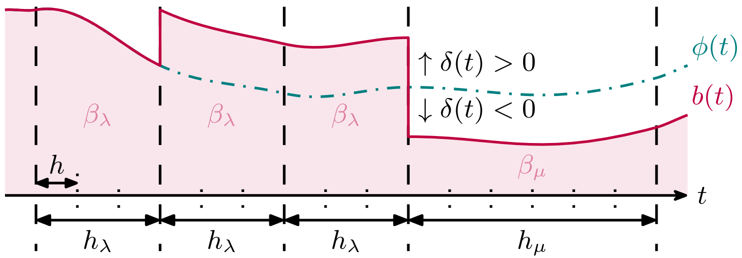


[1] Pazzaglia, P., Mark, C., Pourmohseni, B., Smirnov, F., Schmidt, K., & Beermann, L. (2024). "**Brief Industry Paper: Delay-Aware Control in Networked Systems Using Smart Actuators**." In 2024 IEEE 30th Real-Time and Embedded Technology and Applications Symposium (RTAS)

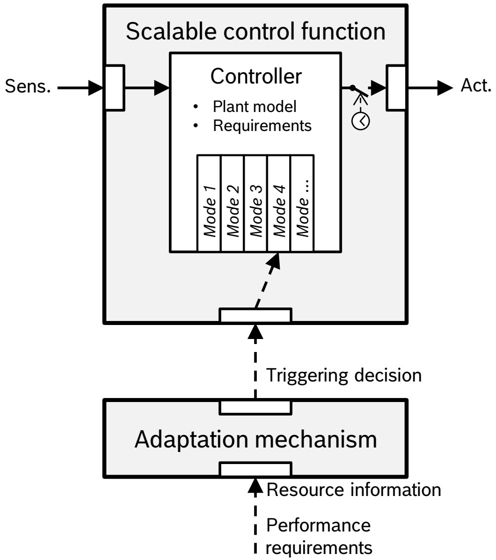
Be Adaptive and Resource-Aware

Adapting the controller based on available resources

- Multiple control applications on a shared platform with **limited computational resources** and **variable performance requirements**
- Resources and performance are traded to optimize the capabilities of the distributed system



- Using a **flexible resource model** for the computational resource availability



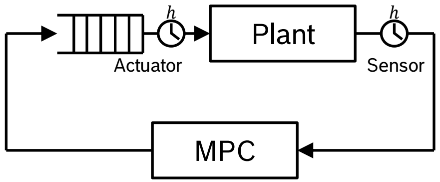
Proposed solution

- A **resource-aware design** that integrates the resource information into the design of the control functionality.
- A **scalable control functionality** capable of adapting to the current resource and state-space conditions

BOSCH

Resource-Aware Control System Design
Example Usecase

- Model Predictive Control (MPC) executing with variable periods
- Using buffered (past) projected control values while executing



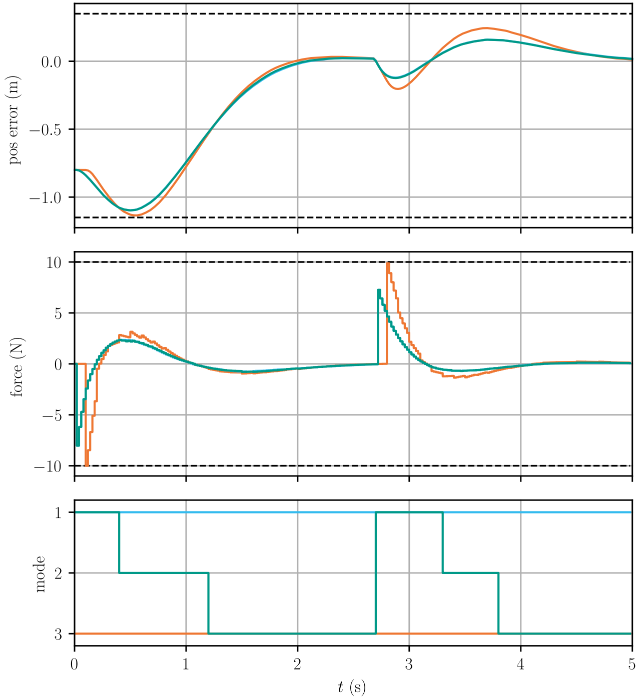
$$
\begin{aligned}
\min_{u_k,\, x_k} \quad & V_f(x_{N|k}) + \sum_{i=0}^{N-1} \|x_{i|k}\|_Q + \|u_{i|k}\|_R \\
\text{s.t.} \quad & x_{i+1|k} = Ax_{i|k} + Bu_{i|k}\,, && i = 0, \dots, N-1 && \text{System dynamics} \\
& x_{i|k} \in X\,, && i = 0, \dots, N-1 && \text{State constraints} \\
& u_{i|k} \in U\,, && i = 0, \dots, N-1 && \text{Input constraints} \\
& x_{0|k} \in x_k && && \text{Initial state constraint} \\
& \boxed{u_{i|k} = u^*_{\sigma_\lambda+i|k-\sigma_\lambda}\,, \quad i = 0, \dots, \sigma_\mu - 1 \quad \text{Initial input constraints}} \\
& x_{N|k} \in X_f && && \text{Terminal state constraint}
\end{aligned}
$$

Stabilization of cart-pole system

Reject impulse disturbance at $t \approx 0.26$ s

Multi-mode MPC with 3 modes:

- Mode 1: period = $1h$ (high accuracy)
- Mode 2: period = $5h$ (mid accuracy)
- Mode 3: period = $10h$ (low accuracy)



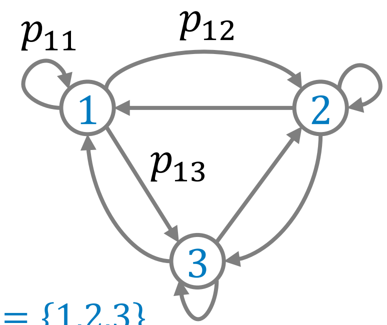
	control perf. (MSE)	resource util.
Fixed mode 3	0.828	**20%**
Fixed mode 1	**0.653**	100%
Multi-mode	**0.656**	**43.8%**

[2] Domenighini M., Pazzaglia P., Mark C., Schmidt K., Beermann L. & Papadopoulos A. V. (2025). "**Control Period Adaptation for Resource-Constrained MPC Applications**". *23rd European Control Conference (ECC), 2025.*

Be robust

Be Robust and Deal with Uncertainties
Accounting for unknown and time-varying delays

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k$$
$$\theta_k \in \Theta, \quad |\Theta| < \infty$$



$\Theta = \{1,2,3\}$

Trans. prob. can be:

- *Known*
- *Unknown*
- *Time-varying (within bounds)*

$$P_k = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}_k$$

- Control applications may **behave differently under different delays**
  - Switching system
- Often also not full knowledge of delays is available (e.g., varying network reliability) → probabilistic characterization with ambiguity

System Model

- *Time-inhomogeneous* discrete-time Markov jump linear system (MJLS)
- Mixed transition probabilities

Control Objective

- Design a feedback controller $K = \{K_i\}_{i \in \Theta}$ with $u_k = K_{\theta_k} x_k$
- Ensure mean-square stability $\lim_{k \to \infty} \mathbb{E}[x(k)\, x(k)^\top] = 0 \quad \forall x_0, \theta_0$

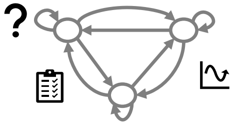
Challenge: Robustness under uncertainty in transition probabilities

BOSCH

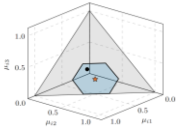
Structure-Exploiting Distributionally Robust Control
In a nutshell

Approach

Time-inhomogeneous MJLS model



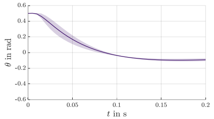
Collect data samples to set up ambiguity sets



Solve set of Linear Matrix Inequalities

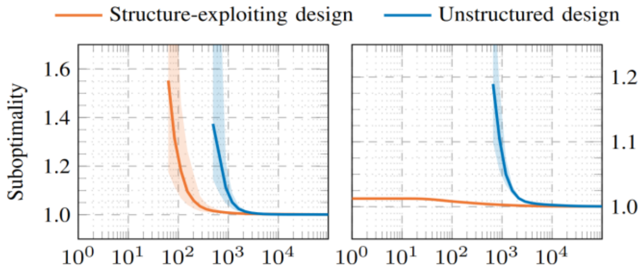
$$\begin{bmatrix} E_i + p_i^{\mathrm{UV}} \tilde{S}_{ij} - S_i^{\Sigma} & E_i Q^{1/2} & Y_i^{\top} R^{1/2} & \sqrt{p_i^{\mathrm{UV}}} \Phi_i'^{\top} & \bar{\Phi}_i^{\mathrm{K}} \\ \star & I & 0 & 0 & 0 \\ \star & \star & I & 0 & 0 \\ \star & \star & \star & E_j & 0 \\ \star & \star & \star & \star & \bar{E}_i^{\mathrm{K}} \end{bmatrix} \succeq 0$$

Distributionally robust controller



Key results

- State feedback controller for mean-square stabilization of MJLS
- **Probabilistic stability guarantee** for mixed transition probabilities
- **Performance** and **feasibility** improved through usage of prior knowledge



(a) Constant K (b) Switching K

[3] Gallant, M., Mark, C., Pazzaglia, P., von Keler, J., Beermann, L., Schmidt, K., & Maggio, M. (2024). "**Structure-Exploiting Distributionally Robust Control of Non-Homogeneous Markov Jump Linear Systems**". *IEEE Control Systems Letters & 2025 American Control Conference*

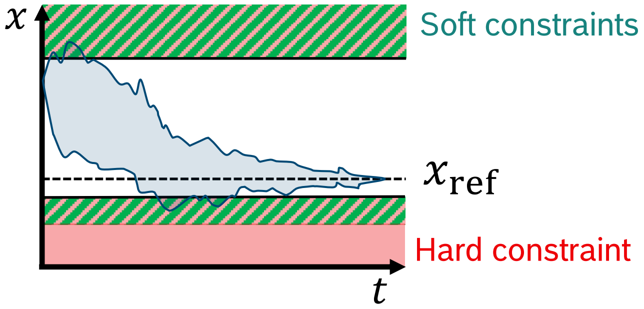
BOSCH

Be Robust and Consider Constraints

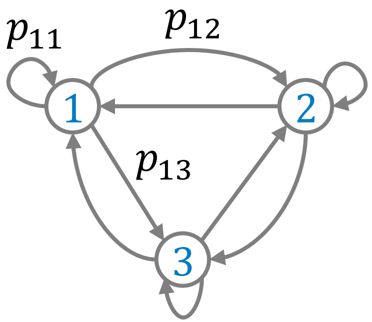
Integrating delays and constraints with scenario tree

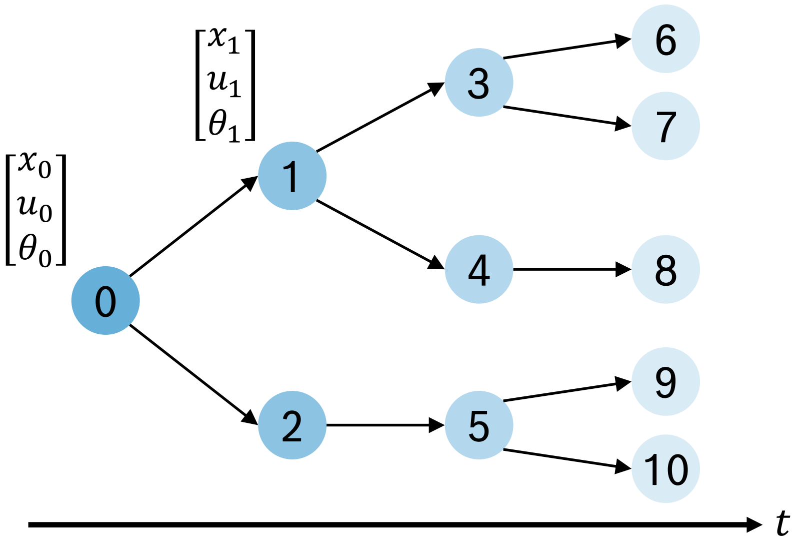
Adding constraints

- Hard constraints: must always be satisfied (e.g., safety limits)
- Soft constraints: can be violated at a cost (e.g., performance trade-offs)



Switching system MJLS





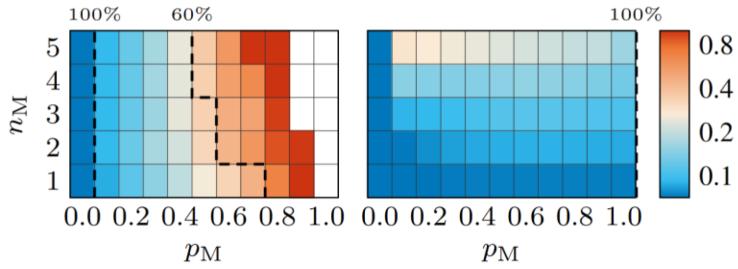
Predictive Behaviour

- Using MPC formulation to anticipate future system states casted over a scenario tree
- Each branch represents a possible evolution of the system with potential deadline misses
- The system is controlled by optimizing over this tree at each time step

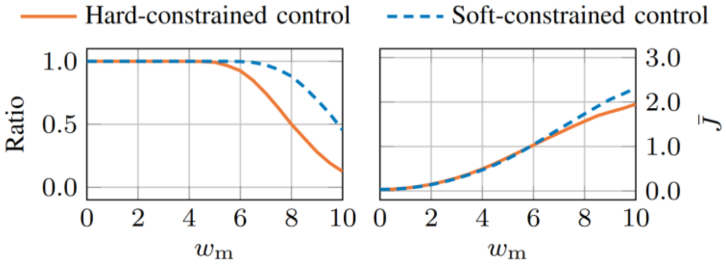
Soft-Constrained Stochastic MPC
Application and results

Results

- Mixed **hard and soft constrained** scenario-tree MPC for optimal predictive control of MJLS
- Optimal control problem is recursively feasible and the origin **mean-square stable**
- Showed improved **performance** and **feasibility**



(a) Nominal control ⏱ 9.8 ms
(b) Stochastic control ⏱ 11.3 ms



(a) Feasibility
(b) Average cost

$$
\begin{aligned}
\min_{x,u,\epsilon}\quad & J_N^x(x_{\cdot|k}, u_{\cdot|k}) + J_N^\epsilon(\epsilon_{\cdot|k}) \\
\text{s.t.}\quad & x_{0|k} = x(k) \\
& x_{\mathtt{succ}(\ell,i)|k} = A_\ell\, x_{\ell|k} + B_\ell\, u_{\ell|k} \\
& H u_{\ell|k} \le h \\
& \boxed{G^{\mathrm{H}} x_{\ell|k} \le g^{\mathrm{H}}} \quad \text{Hard constraints} \\
& \boxed{G^{\mathrm{S}} x_{\ell|k} \le g^{\mathrm{S}} + \epsilon_{\ell|k} + \epsilon_k^{\mathrm{f}}} \quad \text{Soft constraints} \\
& x_{\ell|k} \in \mathcal{X}_\ell^{\mathrm{f}} \\
& c\,\|x_{\ell|k}\|_{P_\ell} \le g^{\mathrm{S}} + \epsilon_k^{\mathrm{f}} \\
& \epsilon_{\ell|k} \ge 0 \\
& \epsilon_k^{\mathrm{f}} \ge 0
\end{aligned}
$$

[4] Gallant, M., Mark, C., Pazzaglia, P., von Keler, J., Beermann, L., Schmidt, K., & Maggio, M. (2025). "**Soft-Constrained Stochastic MPC of Markov Jump Linear Systems: Application to Real-Time Control with Deadline Overruns**". *IEEE Control Systems Letters*.

5

Conclusions

Conclusions

- A **revolution** is ongoing for safety-critical systems
  - "RDS will be for CPS what Cloud has been for IT"
- Large spectrum of applications for RDSs that work **without hard deadlines**
- **Modern control methods** can help against **inevitable timing uncertainties**
  - Increasing credibility, robustness, flexibility
  - Require some level of **knowledge** of delays and resource usage
- We are not at the end, we need much more.
- An **interface** between platform and applications, to exchange runtime constraints and requirements
- Huge opportunity to **build more reliable systems with better performance**

Conclusions

- **Real-time and control communities cannot solve all RDS problems alone**
  - Seeking the pass between the two mountains
- Finding a solution knowing the models, assumptions and capabilities **of the methods of the other community**

To be successful we need to work together, real-time and control communities

Thank you!

paolo.pazzaglia@de.bosch.com

A big thanks also to all colleagues, managers and PhD students that are helping making all this possible

REFERENCES

[1] Pazzaglia, P., Mark, C., Pourmohseni, B., Smirnov, F., Schmidt, K., & Beermann, L. (2024). "**Brief Industry Paper: Delay-Aware Control in Networked Systems Using Smart Actuators**." In 2024 IEEE 30th Real-Time and Embedded Technology and Applications Symposium (RTAS)

[2] Domenighini M., Pazzaglia P., Mark C., Schmidt K., Beermann L. & Papadopoulos A. V. (2025). "**Control Period Adaptation for Resource-Constrained MPC Applications**". *23rd European Control Conference (ECC), 2025*.

[3] Gallant, M., Mark, C., Pazzaglia, P., von Keler, J., Beermann, L., Schmidt, K., & Maggio, M. (2024). "**Structure-Exploiting Distributionally Robust Control of Non-Homogeneous Markov Jump Linear Systems**". *IEEE Control Systems Letters & 2025 American Control Conference (ACC)*

[4] Gallant, M., Mark, C., Pazzaglia, P., von Keler, J., Beermann, L., Schmidt, K., & Maggio, M. (2025). "**Soft-Constrained Stochastic MPC of Markov Jump Linear Systems: Application to Real-Time Control with Deadline Overruns**". *IEEE Control Systems Letters*.