How do timing misalignments affect real-time control systems?

Yde Sinnema
Lund University, Department of Automatic Control
Supervisors: Martina Maggio, Anders Rantzer

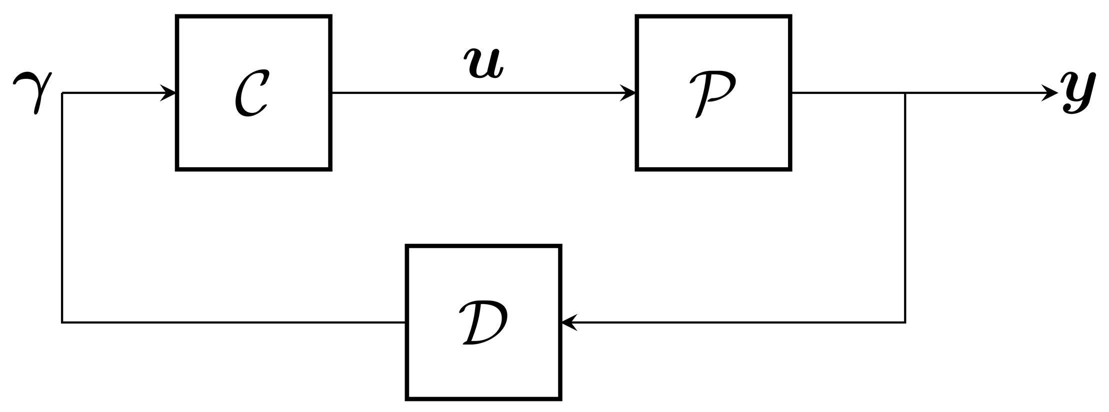
LUND UNIVERSITY

Summary

In **real-time control systems** such as autonomous vehicles, a quick reaction to sensor measurements is desired. However, there is always a processing time, which can be unknown or non-constant. In systems with multiple sensor channels, different time-varying delays result in **timing misalignments**. We present a model for such systems and use Markov Jump Linear Systems theory to assess the impact on the **stability** and quantify the **performance** in the presence of misalignments.

Model: system and delays

A plant $\mathcal{P}$ is regulated by a controller $\mathcal{C}$. Both systems are linear and time-invariant.



The plant output $\boldsymbol{y}$ is sampled periodically. The controller calculates its control signal $\boldsymbol{u}$ using the measurement $\boldsymbol{\gamma}$.
$\boldsymbol{\gamma}$ is a *non-uniformly* delayed version of $\boldsymbol{y}$.

$$\begin{bmatrix} \gamma_1(k) \\ \vdots \\ \gamma_n(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_1(k-\boldsymbol{\delta}_1(k)) \\ \vdots \\ \boldsymbol{y}_n(k-\boldsymbol{\delta}_n(k)) \end{bmatrix}$$

We use a probabilistic delay model to describe $\boldsymbol{\delta}(k)$. The dynamical system $\mathcal{D}$ integrates the time-varying delays into the closed-loop system.

Theory: Markov Jump Linear Systems

A Markov Jump Linear System is a linear system parameterised by a discrete state. A Markov chain governs the discrete state (here $\boldsymbol{\delta}$). At time step k, the continuous state $\tilde{\boldsymbol{x}}$ evolves according to

$$\tilde{\boldsymbol{x}}(k+1) = \Phi_{\boldsymbol{\delta}(k)}\tilde{\boldsymbol{x}}(k).$$

The state transition matrix $\Phi_{\boldsymbol{\delta}(k)}$ is made up of the matrices that constitute $\mathcal{P}$, $\mathcal{C}$, and $\mathcal{D}$. Only the latter depends on $\boldsymbol{\delta}$.

Notation: $\mathcal{S}$ contains all possible values of $\boldsymbol{\delta}(k)$. The Markov chain has a probability transition matrix Π.

- **Stability**: the system is *mean-square stable* if and only if

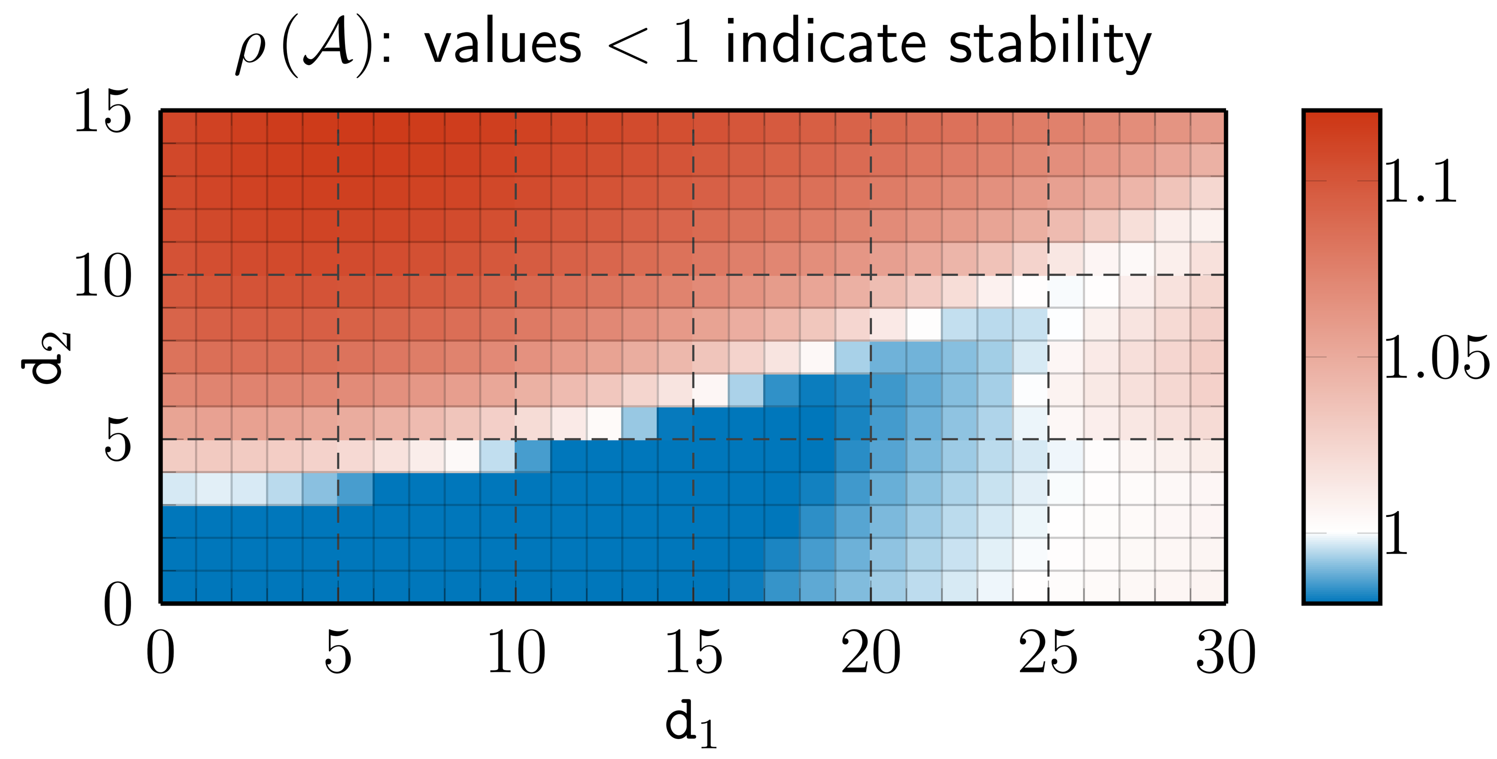
$$\rho(\mathcal{A}) = \max\{|\mathrm{eig}(\mathcal{A})|\} < 1,$$

$$\mathcal{A} = (\Pi^T \otimes I_{n_{\tilde{x}}^2}) \cdot \mathrm{blkdiag}\left\{\Phi_{\boldsymbol{\delta}(k)}^T \otimes \Phi_{\boldsymbol{\delta}(k)}\right\}_{\boldsymbol{\delta}(k)\in\mathcal{S}}.$$

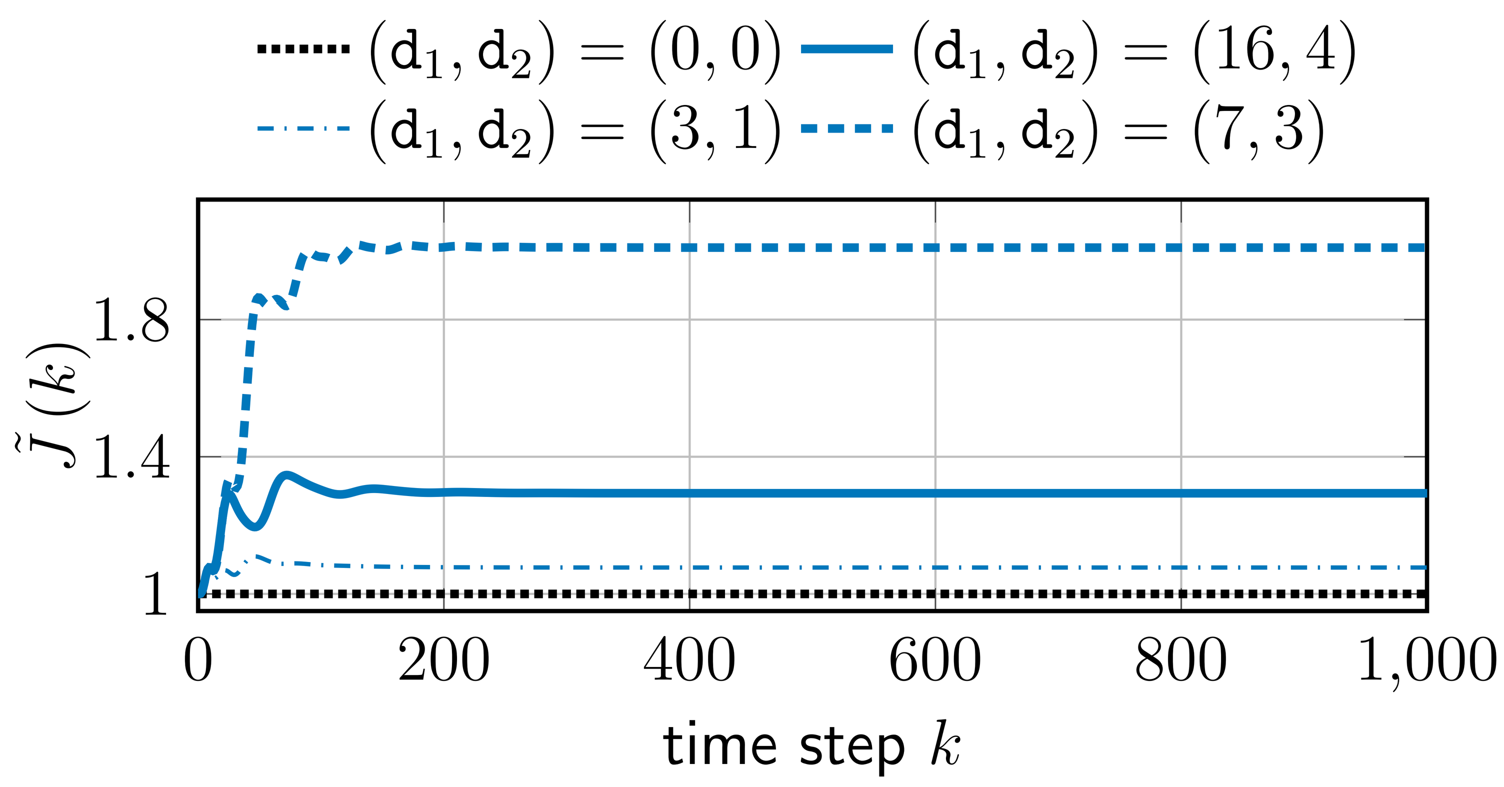
- **Performance**: a cost $J(k)$ is calculated based on the evolution of the covariance of $\tilde{\boldsymbol{x}}$.

Case study: Adaptive Cruise Control

$\mathcal{P}$ is a car with two sensor channels: position $\boldsymbol{y}_1$ and velocity $\boldsymbol{y}_2$. The control objective is to keep a safe distance to a leading vehicle.
In this example, both sensor channels have a constant delay: $\boldsymbol{\delta}_1(k) = \mathtt{d}_1$ and $\boldsymbol{\delta}_2(k) = \mathtt{d}_2$.



Stability: plot of the spectral radius $\rho(\mathcal{A})$ for a range of different $\mathtt{d}_1$ and $\mathtt{d}_2$.
The effect of delays in both sensor channels is asymmetric. The spectral radius does not change monotonically with the delays.



Performance: evolution of $J(k)$ for selected delay values. We show the normalised cumulative cost relative to the reference cost J_0 in the delayless case:

$$\tilde{J}(k) = \frac{\sum_{i=0}^{k} J(i)}{\sum_{i=0}^{k} J_0(i)}.$$

A larger delay doesn't always imply a worse performance.

WASP
WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM