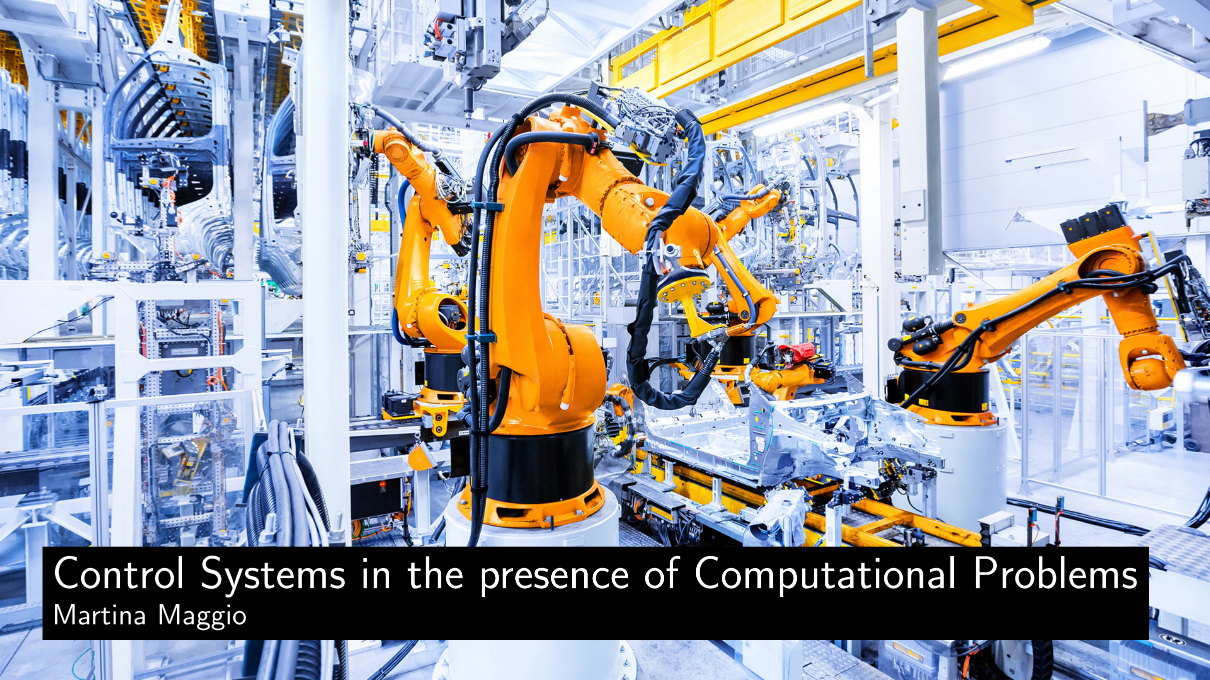


Control Systems in the presence of Computational Problems

Martina Maggio

Control: The Hidden Technology[1]

[1] This is the title of a famous *lectio magistralis* on control given by Karl-Johan Åstrom.

In a nutshell...

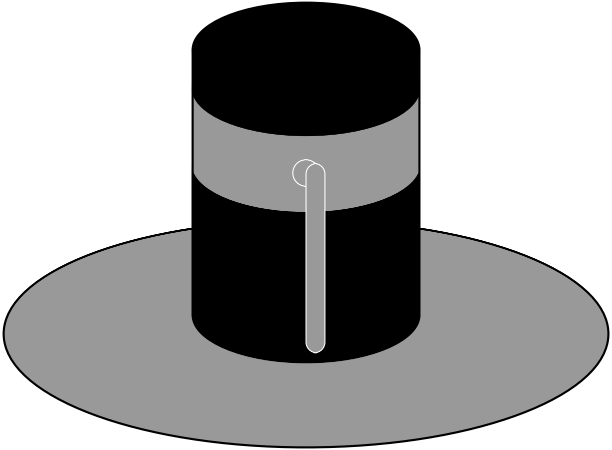
- Controllers are software programs that run on hardware
- As such, they can experience computational problems
- For the rest of this talk: faults causes *deadline misses*

In a nutshell...

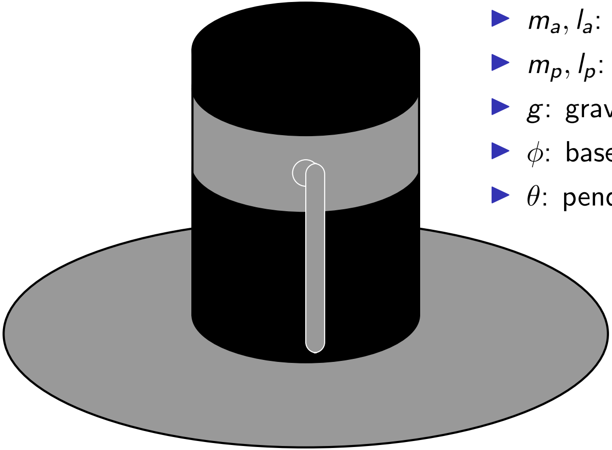
- Controllers are software programs that run on hardware
- As such, they can experience computational problems
- For the rest of this talk: faults causes *deadline misses*
- If we run these controller *in practice* we see that very often deadline misses are not a problem – but: can we *certify* that the system "will not misbehave" despite the presence of deadline misses?

Control Design

Modelling the Physical Phenomena[2]

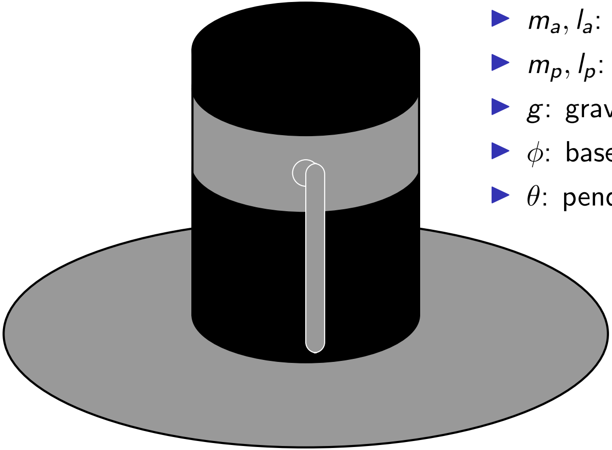


Modelling the Physical Phenomena[2]



- J: moment of inertia of the center pillar
- m_a, l_a: mass and length of first arm
- m_p, l_p: mass and length of pendulum arm
- g: gravitational acceleration constant
- ϕ: base angle
- θ: pendulum angle

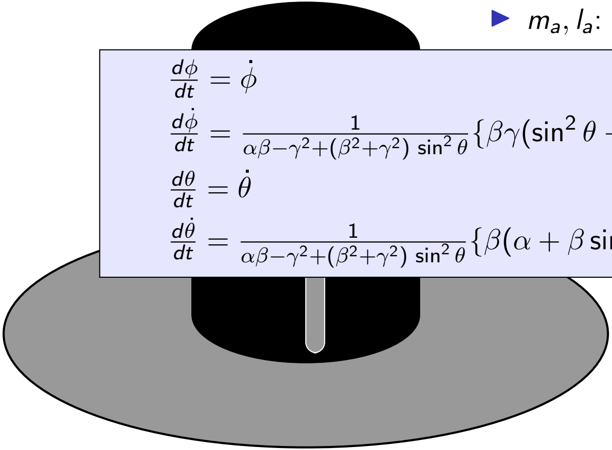
Modelling the Physical Phenomena[2]



- J: moment of inertia of the center pillar
- m_a, l_a: mass and length of first arm
- m_p, l_p: mass and length of pendulum arm
- g: gravitational acceleration constant
- ϕ: base angle
- θ: pendulum angle

- $\alpha := J + (\frac{1}{3} m_a + m_p)\, l_a^2$
- $\beta := \frac{1}{3} m_p\, l_p^2$
- $\gamma := \frac{1}{2} m_p\, l_a\, l_p$
- $\delta := \frac{1}{2} m_p\, l_p\, g$

Modelling the Physical Phenomena[2]



$$\frac{d\phi}{dt} = \dot{\phi}$$

$$\frac{d\dot{\phi}}{dt} = \frac{1}{\alpha\beta - \gamma^2 + (\beta^2 + \gamma^2)\sin^2\theta}\{\beta\gamma(\sin^2\theta - 1)\sin\theta\dot{\phi}^2 + \dots\}$$

$$\frac{d\theta}{dt} = \dot{\theta}$$

$$\frac{d\dot{\theta}}{dt} = \frac{1}{\alpha\beta - \gamma^2 + (\beta^2 + \gamma^2)\sin^2\theta}\{\beta(\alpha + \beta\sin^2\theta)\cos\theta\sin\theta\dot{\phi}^2 + \dots\}$$

- J: moment of inertia of the center pillar
- m_a, l_a: mass and length of first arm
- n arm
- nt
- $\alpha := J + (\frac{1}{3}m_a + m_p)\, l_a^2$
- $\beta := \frac{1}{3}m_p\, l_p^2$
- $\gamma := \frac{1}{2}m_p\, l_a\, l_p$
- $\delta := \frac{1}{2}m_p\, l_p\, g$

[2] For the full derivation, see Magnus Gäfvert, Modelling the Furuta Pendulum, ISSN 0280–5316

Modelling the Physical Phenomena

- Identifying system state, input, and output
- Non-linear resulting model
- Determining the system *equilibria* and linearizing the model

Modelling the Physical Phenomena

- Identifying system state, input, and output
- Non-linear resulting model
- Determining the system *equilibria* and linearizing the model

$$
\begin{aligned}
\dot{x}\left(t\right) &= A_c\, x\left(t\right) + B_c\, u\left(t\right) \\
y\left(t\right) &= C_c\, x\left(t\right) + D_c\, u\left(t\right)
\end{aligned}
$$

Modelling the Physical Phenomena

- Identifying system state, input, and output
- Non-linear resulting model
- Determining the system *equilibria* and linearizing the model
- Discretizing with time step T

$$\begin{aligned} x_{k+1} &= A_d\, x_k + B_d\, u_k \\ y_k &= C_d\, x_k + D_d\, u_k \end{aligned}$$

Example: Furuta Pendulum model

$$
\begin{aligned}
x_{k+1} &= A_d \; x_k + B_d \; u_k \\
y_k &= C_d \; x_k + D_d \; u_k
\end{aligned}
$$

- $x = \begin{bmatrix} \theta & \dot{\theta} & \dot{\phi} \end{bmatrix}^T$, $y = x$, $T = 5\,ms$
- u is the torque applied at the base level

Example: Furuta Pendulum model

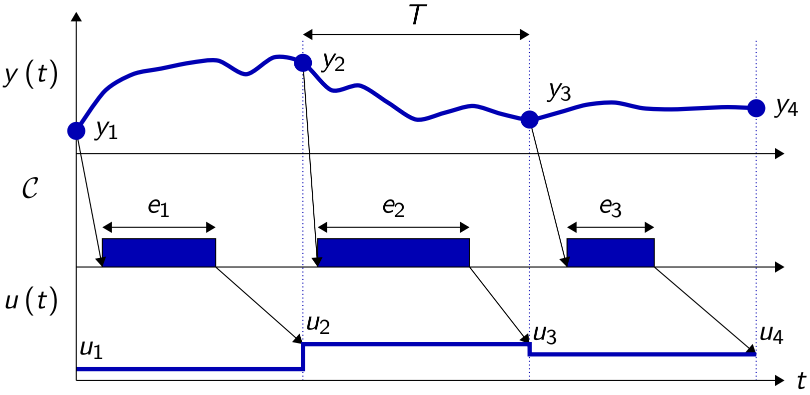
$$\begin{aligned} x_{k+1} &= A_d \; x_k + B_d \; u_k \\ y_k &= C_d \; x_k + D_d \; u_k \end{aligned}$$

- $x = \begin{bmatrix} \theta & \dot{\theta} & \dot{\phi} \end{bmatrix}^T$, $y = x$, $T = 5\,ms$
- u is the torque applied at the base level

Around the *upright* equilibrium point:

$$A_d = \begin{bmatrix} 1.001 & 0.005 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\; B_d = \begin{bmatrix} -0.083 \\ -33.2 \\ 38.6 \end{bmatrix},\; C_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\; D_d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Controller Nominal Execution



Synthesizing the Controller

- Based on objectives (like speed of convergence and ability to reject disturbances) we can pick a control algorithm (which executes periodically inside e_k)
  - many alternatives: state/output feedback, PID, LQR, LQG, MPC, ...
- and verify that the closed-loop behaves in the desired way.

Example: Furuta Pendulum control synthesis

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = K\, y_k = K\, x_k = \begin{bmatrix} 0.375 & 0.025 & 0.0125 \end{bmatrix} x_k$$

- Output feedback controller (but $y = x$, hence state feedback)
- At the beginning of every iteration we sense y, and calculate the next u
- Autonomous behavior: $x_{k+1} = A_d\, x_k + B_d\, K\, x_{k-1}$

Example: Furuta Pendulum control synthesis

$$
\begin{aligned}
x_{k+1} &= A_d\, x_k + B_d\, u_k \\
u_{k+1} &= K\, y_k = K\, x_k = \begin{bmatrix} 0.375 & 0.025 & 0.0125 \end{bmatrix} x_k
\end{aligned}
$$

- Output feedback controller (but $y = x$, hence state feedback)
- At the beginning of every iteration we sense y, and calculate the next u
- Autonomous behavior: $x_{k+1} = A_d\, x_k + B_d\, K\, x_{k-1}$

$$
\tilde{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}, \; \tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d\, K \\ I & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} = A\, \tilde{x}_k
$$

Verifying the Control Design

- Typical assumptions in terms of computation:
  - instantaneous sensing and actuation
  - instantaneous computation
  - no communication overhead
- The design framework that we used is already employing a *one-step delay* paradigm, to take advantage of *predictable communication and execution times*

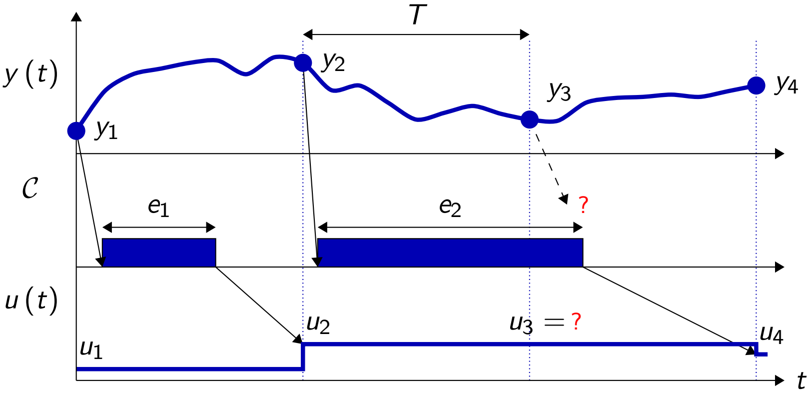
Verifying the Control Design

- Typical assumptions in terms of computation:
  - instantaneous sensing and actuation
  - instantaneous computation
  - no communication overhead
- The design framework that we used is already employing a *one-step delay* paradigm, to take advantage of *predictable communication and execution times*
- If the spectral radius $\rho(A)$ is less than 1, the closed-loop system is stable,

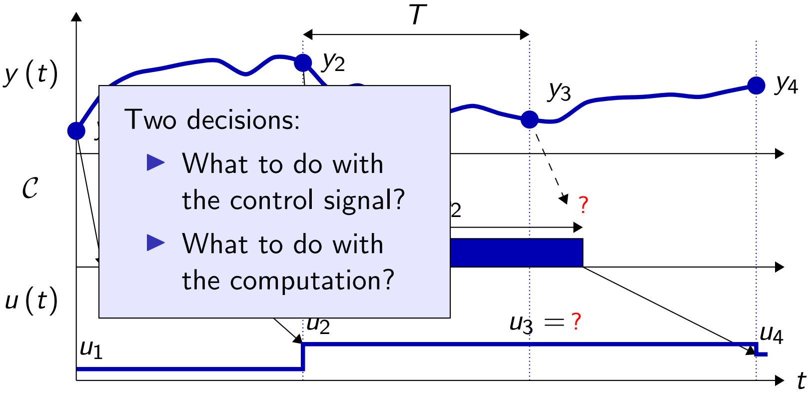
$$\rho(A) = \max |\lambda(A)|$$

What if there are deadline misses?

Missing a Deadline



Missing a Deadline



Two decisions:

- What to do with the control signal?
- What to do with the computation?

Missing a Deadline

For the control signal[3]

- **Hold**: keeping the previous value
- **Zero**: set the control signal to zero

[3] Steffen Linsenmayer and Frank Allgöwer, CDC 2017
"Stabilization of networked control systems with weakly hard real-time dropout description"

Missing a Deadline

For the control signal[3]

- **Hold**: keeping the previous value
- **Zero**: set the control signal to zero

For the computation[4]

- **Kill**: kill the current task with a clean reset, nothing happened
- **Skip-Next**: let the current task continue but do not start a new one in the next period and wait for the following activation

[3]Steffen Linsenmayer and Frank Allgöwer, CDC 2017
"Stabilization of networked control systems with weakly hard real-time dropout description"

[4]Anton Cervin, IFAC World Congress 2005
"Analysis of overrun strategies in periodic control tasks."

Kill&Zero

Kill&Zero

Hit

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = K\, x_k$$

Kill&Zero

Hit

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = K\, x_k$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ K & 0 \end{bmatrix}}_{A_H} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

Kill&Zero

Hit

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = K\, x_k$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ K & 0 \end{bmatrix}}_{A_H} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

Miss

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = 0$$

Kill&Zero

Hit

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = K\, x_k$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ K & 0 \end{bmatrix}}_{A_H} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

Miss

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = 0$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix}}_{A_M} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

Kill&Zero

Hit

$$x_{k+1} = A_d\,x_k + B_d\,u_k$$
$$u_{k+1} = K\,x_k$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ K & 0 \end{bmatrix}}_{A_H} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

The closed-loop system switches arbitrarily between A_H and A_M

Miss

$$x_{k+1} = A_d\,x_k + B_d\,u_k$$
$$u_{k+1} = 0$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix}}_{A_M} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

Kill&Zero

Hit

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = K\, x_k$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ K & 0 \end{bmatrix}}_{A_H} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

The closed-loop system switches arbitrarily between A_H and A_M

Miss

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = 0$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix}}_{A_M} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

We don't have much hope to guarantee stability...

Kill&Zero

Hit

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = K\, x_k$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ K & 0 \end{bmatrix}}_{A_H} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

The closed-loop system switches arbitrarily between A_H and A_M

Miss

$$x_{k+1} = A_d\, x_k + B_d\, u_k$$
$$u_{k+1} = 0$$
$$\Downarrow$$
$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix}}_{A_M} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

We don't have much hope to guarantee stability...
...unless we add constraints!

Constraint Example

"We cannot miss more than n consecutive deadlines"[5]
...means that the system switches arbitrarily between matrices in Σ:

$$\Sigma = \{A_H \, A_M^i \mid i \in \mathbb{Z}, 0 \leq i \leq n\}$$

[5] Martina Maggio, Arne Hamann, Eckart Mayer-John, Dirk Ziegenbein, ECRTS 2020 "Control System Stability under Consecutive Deadline Misses Constraints"

Constraint Example

"We cannot miss more than n consecutive deadlines"[5]
...means that the system switches arbitrarily between matrices in Σ:

$$\Sigma = \{A_H \, A_M^i \mid i \in \mathbb{Z}, 0 \leq i \leq n\}$$

corresponds to i misses
followed by 1 hit

[5] Martina Maggio, Arne Hamann, Eckart Mayer-John, Dirk Ziegenbein, ECRTS 2020
"Control System Stability under Consecutive Deadline Misses Constraints"

Joint Spectral Radius

We can use a result[6] on switching systems, that states that the system that arbitrarily switches among matrices in Σ is asymptotically stable if and only if the joint spectral radius[7] $\rho(\Sigma)$ is less than 1

$$\begin{aligned} \rho_\mu(\Sigma) &= \sup\{\rho(A)^{\frac{1}{\mu}} : A \in \Sigma^\mu\} \\ \rho(\Sigma) &= \limsup_{\mu\to\infty} \rho_\mu(\Sigma) \end{aligned}$$

[6] Gian-Carlo Rota and Gilbert Strang, Indagationes Mathematicae, 63:379-381, 1960 "A note on the joint spectral radius"

[7] Raphael Junger, Lecture Notes in Control and Information Sciences, 2009 "The Joint Spectral Radius: Theory and Applications"

Joint Spectral Radius

- The problem of determining if the joint spectral radius is less than 1 is undecidable[8] even for "simple" set of matrices Σ
- But lower and upper bounds $\{\rho_\ell(\Sigma), \rho_u(\Sigma)\}$ can be found via many[9] different analytical methods
- So if $\rho_u(\Sigma) < 1$ the stability of the system with (constrained) deadline misses is guaranteed

[8] Vincent Blondel and John Tsitsiklis, Systems & Control Letters, 41(2):135–140, 2000 "The boundedness of all products of a pair of matrices is undecidable"

[9] Guillaume Vankeerberghen, Julien Hendrickx, and Raphaël M. Jungers, HSCC 2014 "JSR: a toolbox to compute the joint spectral radius"

Fault Models

- Probabilistic
- Constrained, or *weakly-hard*[10]

[10] Guillem Bernat, Alan Burns, Albert Liamosí, IEEE Transactions on Computers, 2001, “Weakly hard real-time systems”

Fault Models

- Probabilistic
- Constrained, or *weakly-hard*[10]

1. $\tau \vdash \binom{x}{k}$, *AnyHit*
2. $\tau \vdash \left\langle {x \atop k} \right\rangle$, *RowHit*
3. $\tau \vdash \overline{\binom{x}{k}}$, *AnyMiss*
4. $\tau \vdash \overline{\left\langle {x \atop k} \right\rangle} = \overline{\langle x \rangle}$, *RowMiss*

with $x \in \mathbb{N}^{\geq}$, $k \in \mathbb{N}^{>}$, *where* $x \leq k$

[10] Guillem Bernat, Alan Burns, Albert Liamosí, IEEE Transactions on Computers, 2001, "Weakly hard real-time systems"

Fault Models

- Probabilistic
- Constrained, or *weakly-hard*[10]

1. $\tau \vdash \binom{x}{k}$, $\mathtt{AnyHit}$
2. $\tau \vdash \left\langle {x \atop k} \right\rangle$, $\mathtt{RowHit}$
3. $\tau \vdash \overline{\binom{x}{k}}$, $\mathtt{AnyMiss}$
4. $\tau \vdash \overline{\left\langle {x \atop k} \right\rangle} = \overline{\langle x \rangle}$, $\mathtt{RowMiss}$

with $x \in \mathbb{N}^{\geq}$, $k \in \mathbb{N}^{>}$, *where* $x \leq k$

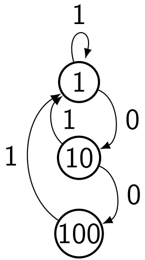
$$\ldots 0\, \underbrace{1\,0\,1\,1\,1}_{k}\, 0\,1\,1\,0 \ldots$$

[10] Guillem Bernat, Alan Burns, Albert Liamosí, IEEE Transactions on Computers, 2001, “Weakly hard real-time systems”

Weakly-Hard Constraints as Automata

- Any weakly-hard constraint can be transformed into a corresponding finite state machine[a]
- The transformation enables the analysis via joint spectral radius[b]

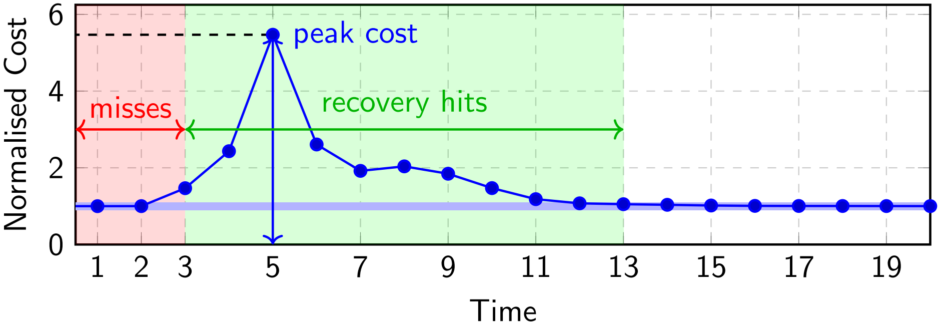
$\binom{x}{k} = \binom{1}{3}$



[a]Nils Vreman, Richard Pates, and Martina Maggio, RTAS 2022 "WeaklyHard.jl: Scalable Analysis of Weakly-Hard Constraints" `https://github.com/NilsVreman/WeaklyHard.jl`

[b]Nils Vreman, Paolo Pazzaglia, Victor Magron, Jie Wang, Martina Maggio, CDC & Letters 2022 "Stability of Linear Systems Under Extended Weakly-Hard Constraints"

Performance Analysis[11]



[11]Nils Vreman, Anton Cervin and Martina Maggio, ECRTS 2021
"Stability and Performance Analysis of Control Systems Subject to Bursts of Deadline Misses"

Conclusion

- Stability and performance analysis of control systems subject to deadline misses
- Sometimes when control software experiences faults (missing deadlines) there is no need to worry!

maggio@cs.uni-saarland.de