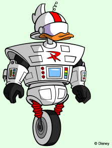
Stability analysis and control design for weakly hard real-time systems

How can richer interfaces, e.g. m-k guarantees, be used for the functional verification of control systems

Steffen Linsenmayer (partially work of and with R. Blind and F. Allgöwer)

Institute for Systems Theory and Automatic Control
University of Stuttgart, Germany

ESWEEK 2017, October 15, 2017



Standing upright | Moving to locations | . . . | Speaking with others

Image source: Disney/Wikimedia Commons

Embedded control systems - Relevant properties

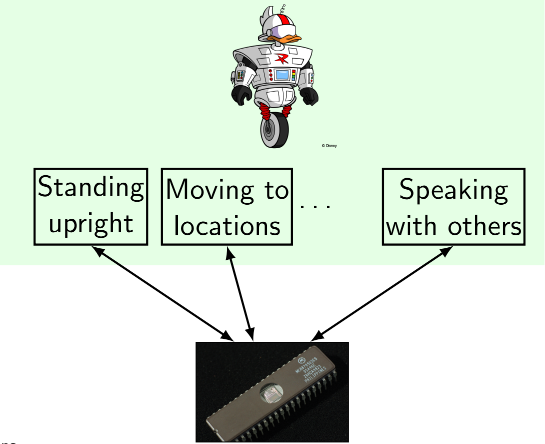


Image source: Disney/Wikimedia Commons

Embedded control systems - Relevant properties

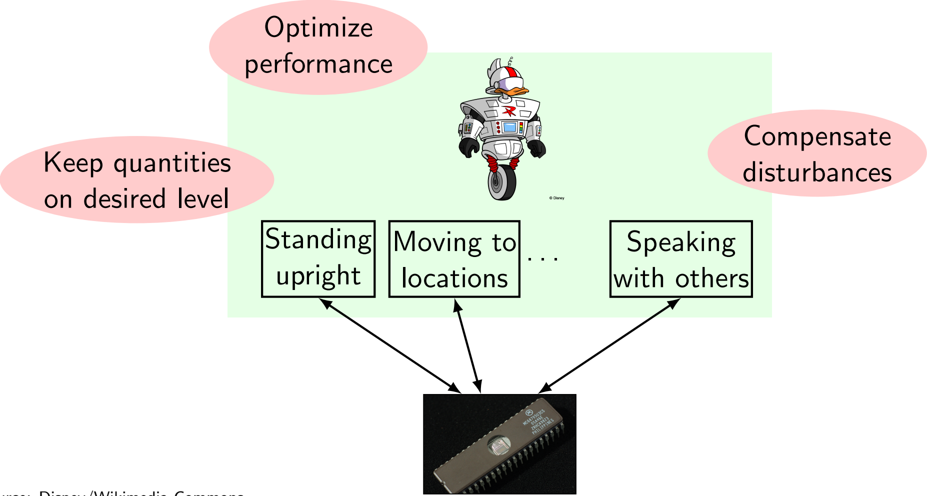


Image source: Disney/Wikimedia Commons

Embedded control systems - Relevant properties

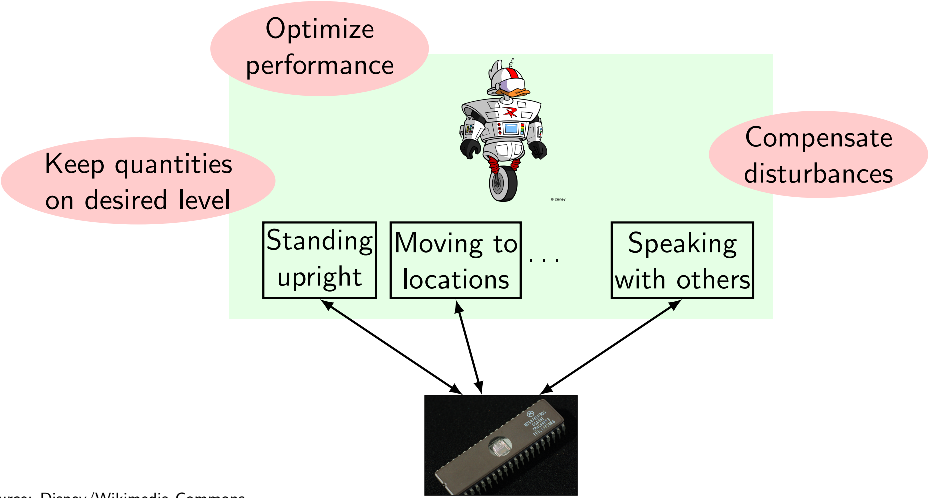


Image source: Disney/Wikimedia Commons

Is it necessary that all deadlines are met to guarantee properties of the control system?

Outline

1. Functional verification of control systems
2. How to use weakly hard real-time constraints in control systems
3. Stability analysis with weakly hard real-time constraints
4. Controller design with weakly hard real-time constraints
5. Summary

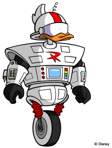
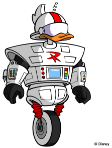


Image source: Disney/Wikimedia Commons



→ Standing upright

Image source: Disney/Wikimedia Commons

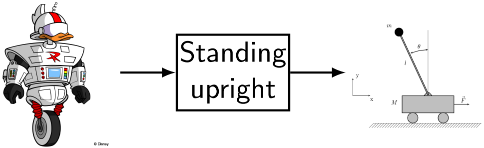


Image source: Disney/Wikimedia Commons

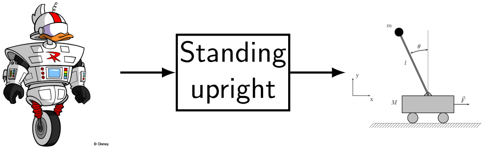


Image source: Disney/Wikimedia Commons

This can be used to motivate/illustrate a major property of control systems:
(Asymptotic) stability

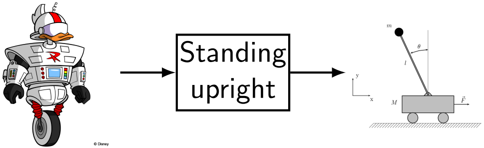
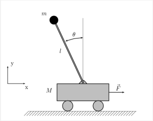


Image source: Disney/Wikimedia Commons

This can be used to motivate/illustrate a major property of control systems:
(Asymptotic) stability

The upright equilibrium is unstable! $\Rightarrow$ One needs a controller to stabilize it.



Goal: Guarantee asymptotic stability of the upright equilibrium

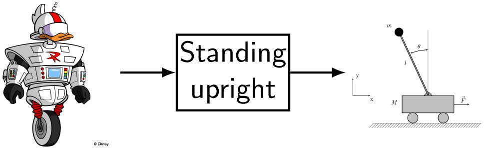
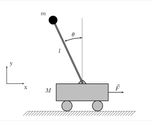


Image source: Disney/Wikimedia Commons

This can be used to motivate/illustrate a major property of control systems:
(Asymptotic) stability

The upright equilibrium is unstable! $\Rightarrow$ One needs a controller to stabilize it.



Goal: Guarantee asymptotic stability of the upright equilibrium *with richer interfaces*.

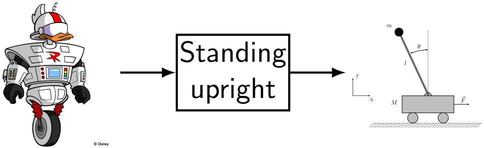


Image source: Disney/Wikimedia Commons

This can be used to motivate/illustrate a major property of control systems:
(Asymptotic) stability

The upright equilibrium is unstable! $\Rightarrow$ One needs a controller to stabilize it.

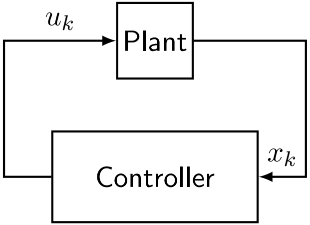
Asymptotic Stability

The equilibrium at the origin of $x_{k+1} = f(x_k)$ is asymptotically stable if it is stable and if δ can be chosen such that $\|x_{k_0}\| < \delta$ implies that $\|x_k\| \to 0$ when $k \to \infty$.

This talk: Linear time-invariant systems with full state available for feedback.

State-space representation of the plant:

$$x_{k+1} = Ax_k + Bu_k$$



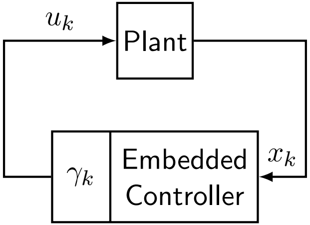
This talk: Linear time-invariant systems with full state available for feedback.

State-space representation of the plant:

$$x_{k+1} = Ax_k + Bu_k$$

Representation of the embedded controller:

$$u_k = f(\gamma_k, x_k, (u_{k-1})), \gamma_k \in \{0, 1\}$$



Stabilization

Can we derive a (static state feedback) controller that renders the origin of the closed loop asymptotically stable?

Stabilization

Can we derive a (static state feedback) controller that renders the origin of the closed loop asymptotically stable?

Performance

Can we derive a controller that minimizes a given cost functional?

Stabilization

Can we derive a (static state feedback) controller that renders the origin of the closed loop asymptotically stable?

Performance

Can we derive a controller that minimizes a given cost functional?

Robustness

Can we derive a controller that is robust to uncertainties and disturbances?

Stabilization

Can we derive a (static state feedback) controller that renders the origin of the closed loop asymptotically stable?

Stabilization

Can we derive a (static state feedback) controller that renders the origin of the closed loop asymptotically stable?

Up to now:

- Stability and Stabilization as important properties and tasks for functional verification of control systems.

Open tasks:

- Show that we can include richer interfaces in control system models.
- Analyze stability with richer interfaces.
- Design controllers with richer interfaces.

Control Problem

$x_{k+1} = Ax_k + Bu_k$

A, B stabilizable

Control Problem	*Embedded Controller*
$x_{k+1} = Ax_k + Bu_k$	$u_k = f(\gamma_k, x_k, (u_{k-1}))$
A, B stabilizable	model of the process γ_k

Control Problem	*Embedded Controller*
$x_{k+1} = Ax_k + Bu_k$	$u_k = f(\gamma_k, x_k, (u_{k-1}))$
A, B stabilizable	max. # of successive deadline misses m

- Worst-case consecutive specification $\Rightarrow$ introduces conservativity

Control Problem	*Embedded Controller*
$x_{k+1} = Ax_k + Bu_k$	$u_k = f(\gamma_k, x_k, (u_{k-1}))$
A, B stabilizable	deadline miss probability p%

- Worst-case consecutive specification $\Rightarrow$ introduces conservativity
- Statistical description $\Rightarrow$ leads to statements for mean and variance

Motivation for using richer interfaces

Control Problem	*Embedded Controller*
$x_{k+1} = Ax_k + Bu_k$	$u_k = f(\gamma_k, x_k, (u_{k-1}))$
A, B stabilizable	max. m deadline misses out of k transmissions

- Worst-case consecutive specification $\Rightarrow$ introduces conservativity
- Statistical description $\Rightarrow$ leads to statements for mean and variance
- Window based description $\Rightarrow$ promising idea to bridge the gap

Control Problem

$x_{k+1} = Ax_k + Bu_k$

A, B stabilizable

Embedded Controller

$u_k = f(\gamma_k, x_k, (u_{k-1}))$

max. m deadline misses out of k transmissions

- Worst-case consecutive specification ⇒ introduces conservativity
- Statistical description ⇒ leads to statements for mean and variance
- Window based description ⇒ promising idea to bridge the gap

Goal

Include window based descriptions in the model of embedded control systems.

Weakly hard real-time (WHRT) constraints to model task executions

Characterization of binary sequences γ by satisfaction of constraints λ "$(\gamma \vdash \lambda)$":

Bernat et al. (2001). Weakly hard real-time systems.

Weakly hard real-time (WHRT) constraints to model task executions

Characterization of binary sequences γ by satisfaction of constraints λ "$(\gamma \vdash \lambda)$":

In any window of m consecutive executions, ...

$\gamma \vdash \binom{n}{m}$

...there are at least n successful executions.

Bernat et al. (2001). Weakly hard real-time systems.

Weakly hard real-time (WHRT) constraints to model task executions

Characterization of binary sequences γ by satisfaction of constraints λ "$(\gamma \vdash \lambda)$":

In any window of m consecutive executions, ...

$\gamma \vdash \binom{n}{m}$

... there are at least n successful executions.

$\gamma \vdash \left\langle {n \atop m} \right\rangle$

... there are at least n consecutive successful executions.

Bernat et al. (2001). Weakly hard real-time systems.

Weakly hard real-time (WHRT) constraints to model task executions

Characterization of binary sequences γ by satisfaction of constraints λ "$(\gamma \vdash \lambda)$":

In any window of m consecutive executions, ...

$\gamma \vdash \binom{n}{m}$

...there are at least n successful executions.

$\gamma \vdash \left\langle {n \atop m} \right\rangle$

...there are at least n consecutive successful executions.

$\gamma \vdash \overline{\left\langle {n \atop m} \right\rangle}$

...there are never n consecutive unsuccessful executions.

Bernat et al. (2001). Weakly hard real-time systems.

Satisfaction Set

The *satisfaction set* of length N of a constraint λ, denoted $\mathcal{S}^N(\lambda)$, is the set of all sequences $\alpha \in \{0,1\}^N$ that satisfy λ. Formally,

$$\mathcal{S}^N(\lambda) := \left\{\alpha \in \{0,1\}^N : \alpha \vdash \lambda\right\}.$$

Bernat et al. (2001). Weakly hard real-time systems.

Satisfaction Set

The *satisfaction set* of length N of a constraint λ, denoted $\mathcal{S}^N(\lambda)$, is the set of all sequences $\alpha \in \{0,1\}^N$ that satisfy λ. Formally,

$$\mathcal{S}^N(\lambda) := \left\{\alpha \in \{0,1\}^N : \alpha \vdash \lambda\right\}.$$

Partial Order

Given two constraints λ', λ. We say that λ' is *harder than* λ (λ is *weaker than* λ'), denoted $\lambda' \preceq \lambda$ if all sequences that satisfy λ' also satisfy λ. Formally,

$$\lambda' \preceq \lambda \Leftrightarrow \mathcal{S}^\infty(\lambda') \subseteq \mathcal{S}^\infty(\lambda).$$

Bernat et al. (2001). Weakly hard real-time systems.

Satisfaction Set

The *satisfaction set* of length N of a constraint λ, denoted $\mathcal{S}^N(\lambda)$, is the set of all sequences $\alpha \in \{0,1\}^N$ that satisfy λ. Formally,

$$\mathcal{S}^N(\lambda) := \left\{\alpha \in \{0,1\}^N : \alpha \vdash \lambda\right\}.$$

Partial Order

Given two constraints λ', λ. We say that λ' is *harder than* λ (λ is *weaker than* λ'), denoted $\lambda' \preceq \lambda$ if all sequences that satisfy λ' also satisfy λ. Formally,

$$\lambda' \preceq \lambda \Leftrightarrow \mathcal{S}^\infty(\lambda') \subseteq \mathcal{S}^\infty(\lambda).$$

Example:
At least 3 *succ. executions in a row* is harder than *at least* 2 *succ. executions in a row.*

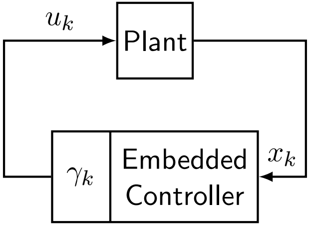
Bernat et al. (2001). Weakly hard real-time systems.

State-space representation of the plant:

$$x_{k+1} = Ax_k + Bu_k$$

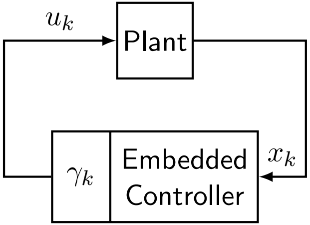
Representation of the embedded controller:

$$u_k = f(\gamma_k, x_k, (u_{k-1}))$$



State-space representation of the plant:

$$x_{k+1} = Ax_k + Bu_k$$

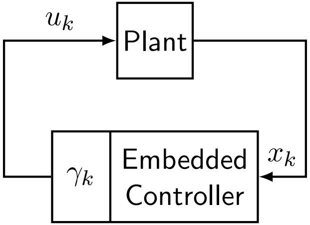


Representation of the embedded controller:

$$u_k = f(\gamma_k, x_k, (u_{k-1})), \gamma \vdash \lambda$$

State-space representation of the plant:

$$x_{k+1} = Ax_k + Bu_k$$



Representation of the embedded controller:

$$u_k = f(\gamma_k, x_k, (u_{k-1})), \gamma \vdash \lambda$$

Actuation strategies:

- zero strategy

$$u_k = \gamma_k K x_k$$

- hold strategy

$$u_k = \gamma_k K x_k + (1 - \gamma_k) u_{k-1}$$

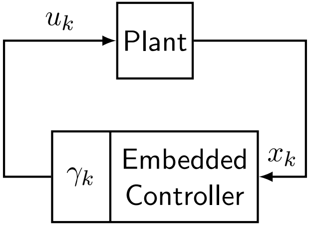
Model of the closed loop with richer interfaces

State-space representation of the plant:

$$x_{k+1} = Ax_k + Bu_k$$

Representation of the embedded controller:

$$u_k = f(\gamma_k, x_k, (u_{k-1})), \gamma \vdash \lambda$$



Up to now:

- Stability and Stabilization as important properties and tasks for functional verification of control systems.
- We can include WHRT constraints in control system models.

Open tasks:

- Analyze stability with richer interfaces.
- Design controllers with richer interfaces.

Plant:

$$x_{k+1} = ax_k + bu_k, \quad x_k \in \mathbb{R}, a > 1 \Rightarrow \textbf{unstable}$$

Embedded Controller:

$$u_k = \gamma_k K x_k$$

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = ax_k + bu_k, \quad x_k \in \mathbb{R}, a > 1 \Rightarrow \textbf{unstable}$$

Embedded Controller:

$$u_k = \gamma_k K x_k$$

Probabilistic description of γ:

Stabilizable in the mean square sense if and only if the success probability is greater than $1 - \frac{1}{a^2}$.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = ax_k + bu_k, \quad x_k \in \mathbb{R}, a > 1 \Rightarrow \textbf{unstable}$$

Embedded Controller:

$$u_k = \gamma_k K x_k$$

Probabilistic description of γ:

Stabilizable in the mean square sense if and only if the success probability is greater than $1 - \frac{1}{a^2}$.

WHRT description of γ:

- Use dead-beat controller $K = -\frac{a}{b}$.
- Then $x(k+1) = 0$ after the first successful execution.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = ax_k + bu_k, \quad x_k \in \mathbb{R}, a > 1 \Rightarrow \textbf{unstable}$$

Embedded Controller:

$$u_k = \gamma_k K x_k$$

Probabilistic description of γ:
Stabilizable in the mean square sense if and only if the success probability is greater than $1 - \frac{1}{a^2}$.

WHRT description of γ:

- Use dead-beat controller $K = -\frac{a}{b}$.
- Then $x(k+1) = 0$ after the first successful execution.

The system is asymptotically stable for all sequences $\gamma \vdash \lambda \preceq \binom{1}{m}$, where m might be arbitrarily large but finite.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

The closed loop can be modeled as a linear discrete-time switched system,

$$\xi_{k+1} = \mathcal{A}_{\gamma_k} \xi_k, \quad \xi_k \in \mathbb{R}^{n_\xi}, \gamma_k \in \{0, 1\}, \gamma \vdash \lambda$$

The closed loop can be modeled as a linear discrete-time switched system,

$$\xi_{k+1} = \mathcal{A}_{\gamma_k}\xi_k, \quad \xi_k \in \mathbb{R}^{n_\xi}, \gamma_k \in \{0,1\}, \gamma \vdash \lambda$$

Zero control:

- $u_k = \gamma_k K x_k$
- $\xi_k = x_k \rightarrow n_\xi = n$
- $\mathcal{A}_0 = A$
- $\mathcal{A}_1 = A + BK$

The closed loop can be modeled as a linear discrete-time switched system,

$$\xi_{k+1} = \mathcal{A}_{\gamma_k}\xi_k, \quad \xi_k \in \mathbb{R}^{n_\xi}, \gamma_k \in \{0,1\}, \gamma \vdash \lambda$$

Zero control:

- $u_k = \gamma_k K x_k$
- $\xi_k = x_k \rightarrow n_\xi = n$
- $\mathcal{A}_0 = A$
- $\mathcal{A}_1 = A + BK$

Hold control:

- $u_k = \gamma_k K x_k + (1-\gamma_k)u_{k-1}$
- $\xi_k = [x_k, u_{k-1}] \rightarrow n_\xi = n + n_u$
- $\mathcal{A}_0 = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \mathcal{A}_1 = \begin{bmatrix} A+BK & 0 \\ K & 0 \end{bmatrix}$

The closed loop can be modeled as a linear discrete-time switched system,

$$\xi_{k+1} = \mathcal{A}_{\gamma_k}\xi_k, \quad \xi_k \in \mathbb{R}^{n_\xi}, \gamma_k \in \{0,1\}, \gamma \vdash \lambda$$

Zero control:

- $u_k = \gamma_k K x_k$
- $\xi_k = x_k \rightarrow n_\xi = n$
- $\mathcal{A}_0 = A$
- $\mathcal{A}_1 = A + BK$

Hold control:

- $u_k = \gamma_k K x_k + (1-\gamma_k)u_{k-1}$
- $\xi_k = [x_k, u_{k-1}] \rightarrow n_\xi = n + n_u$
- $\mathcal{A}_0 = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \mathcal{A}_1 = \begin{bmatrix} A+BK & 0 \\ K & 0 \end{bmatrix}$

Notation

Given a sequence $w \in \{0,1\}^m$, the state evolves as $x_{k+|w|} = \prod_{i=|w|}^{1} \mathcal{A}_{w_i} x_k$.
Hence, we use $\mathcal{A}^w := \prod_{i=|w|}^{1} \mathcal{A}_{w_i}$.
Example: $\mathcal{A}^{1100101} = \mathcal{A}_1\mathcal{A}_0\mathcal{A}_1\mathcal{A}_0\mathcal{A}_0\mathcal{A}_1\mathcal{A}_1$.

Theorem 1

Given the constraint λ and the matrices $\mathcal{A}_0, \mathcal{A}_1 \in \mathbb{R}^{n_\xi \times n_\xi}$. If there exists a symmetric, positive definite matrix $\mathcal{P} \in \mathbb{R}^{n_\xi \times n_\xi}$ such that

$$(\mathcal{A}^w)^\intercal \mathcal{P} \mathcal{A}^w - \mathcal{P} < 0, \forall w \in \mathcal{S}^m(\lambda),$$

then the closed loop is asymptotically stable for all sequences $\gamma \vdash \lambda' \preceq \lambda$.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Theorem 1

Given the constraint λ and the matrices $\mathcal{A}_0, \mathcal{A}_1 \in \mathbb{R}^{n_\xi \times n_\xi}$. If there exists a symmetric, positive definite matrix $\mathcal{P} \in \mathbb{R}^{n_\xi \times n_\xi}$ such that

$$(\mathcal{A}^w)^\intercal \mathcal{P} \mathcal{A}^w - \mathcal{P} < 0, \forall w \in \mathcal{S}^m(\lambda),$$

then the closed loop is asymptotically stable for all sequences $\gamma \vdash \lambda' \preceq \lambda$.

Procedure to check whether the origin of the closed-loop is asy. stable:

- Compute $\mathcal{A}_0, \mathcal{A}_1$ according to the actuation strategy.
- Derive all sequences in $\mathcal{S}^m(\lambda)$.
- Compute all $\mathcal{A}^w$ for $w \in \mathcal{S}^m(\lambda)$.
- Find a positive definite matrix $\mathcal{P}$ such that all LMI's are satisfied.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Theorem 2

Given the constraint λ and the matrices $\mathcal{A}_0, \mathcal{A}_1 \in \mathbb{R}^{n_\xi \times n_\xi}$. If there exist symmetric, positive definite matrices $\mathcal{P}_w \in \mathbb{R}^{n_\xi \times n_\xi}, w \in \mathcal{S}^m(\lambda)$ such that

$$(\mathcal{A}^{w_1})^\intercal \mathcal{P}_{w_1} \mathcal{A}^{w_1} - \mathcal{P}_{w_2} < 0, \forall w_1, w_2 \in \mathcal{S}^m(\lambda), w_1 w_2 \vdash \lambda$$

then the closed loop is asymptotically stable for all sequences $\gamma \vdash \lambda' \preceq \lambda$.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Theorem 2

Given the constraint λ and the matrices $\mathcal{A}_0, \mathcal{A}_1 \in \mathbb{R}^{n_\xi \times n_\xi}$. If there exist symmetric, positive definite matrices $\mathcal{P}_w \in \mathbb{R}^{n_\xi \times n_\xi}, w \in \mathcal{S}^m(\lambda)$ such that

$$(\mathcal{A}^{w_1})^\intercal \mathcal{P}_{w_1} \mathcal{A}^{w_1} - \mathcal{P}_{w_2} < 0, \forall w_1, w_2 \in \mathcal{S}^m(\lambda), w_1 w_2 \vdash \lambda$$

then the closed loop is asymptotically stable for all sequences $\gamma \vdash \lambda' \preceq \lambda$.

Procedure to check whether the origin of the closed-loop is asy. stable:

- Compute $\mathcal{A}_0, \mathcal{A}_1$ according to the actuation strategy.
- Derive all sequences in $\mathcal{S}^m(\lambda)$ and all combinations $w_1 w_2 \vdash \lambda$.
- Compute all $\mathcal{A}^w$ for $w \in \mathcal{S}^m(\lambda)$.
- Find $|\mathcal{S}^m(\lambda)|$ positive definite matrices $\mathcal{P}_w$ such that LMI's are satisfied (combinations s.t. $w_1 w_2 \vdash \lambda$).

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Theorem 2

Given the constraint λ and the matrices $\mathcal{A}_0, \mathcal{A}_1 \in \mathbb{R}^{n_\xi \times n_\xi}$. If there exist symmetric, positive definite matrices $\mathcal{P}_w \in \mathbb{R}^{n_\xi \times n_\xi}, w \in \mathcal{S}^m(\lambda)$ such that

$$(\mathcal{A}^{w_1})^\intercal \mathcal{P}_{w_1} \mathcal{A}^{w_1} - \mathcal{P}_{w_2} < 0, \forall w_1, w_2 \in \mathcal{S}^m(\lambda), w_1 w_2 \vdash \lambda$$

then the closed loop is asymptotically stable for all sequences $\gamma \vdash \lambda' \preceq \lambda$.

Procedure to check whether the origin of the closed-loop is asy. stable:

- Compute $\mathcal{A}_0, \mathcal{A}_1$ according to the actuation strategy.
- Derive all sequences in $\mathcal{S}^m(\lambda)$ and all combinations $w_1 w_2 \vdash \lambda$.
- Compute all $\mathcal{A}^w$ for $w \in \mathcal{S}^m(\lambda)$.
- Find $|\mathcal{S}^m(\lambda)|$ positive definite matrices $\mathcal{P}_w$ such that LMI's are satisfied (combinations s.t. $w_1 w_2 \vdash \lambda$). $\Rightarrow$ increased possibility to find a solution

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Interpretation:

- Theorem 2 takes into account that the concatenation of sequences is actually constrained.

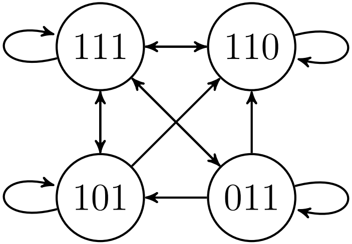
Interpretation:

- Theorem 2 takes into account that the concatenation of sequences is actually constrained.
- This can be represented by a graph.

Interpretation:

- Theorem 2 takes into account that the concatenation of sequences is actually constrained.
- This can be represented by a graph.

Example: $\lambda = \binom{2}{3}$



Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

Goal

Check stability for $\binom{1}{2}$, $\binom{1}{3}$, $\binom{1}{4}$, and $\binom{2}{3}$.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

Goal

Check stability for $\binom{1}{2}$, $\binom{1}{3}$, $\binom{1}{4}$, and $\binom{2}{3}$.

Remark: $\binom{2}{3} \preceq \binom{1}{2} \preceq \binom{1}{3} \preceq \binom{1}{4}$.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ used for zero and hold control

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$		
$\lambda = \binom{1}{3}$		
$\lambda = \binom{1}{4}$		

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$	Theorem 1✓	Theorem 1✗
$\lambda = \binom{1}{3}$		
$\lambda = \binom{1}{4}$		

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$	Theorem 1✓	Theorem 2✓
$\lambda = \binom{1}{3}$		
$\lambda = \binom{1}{4}$		

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$	Theorem 1✓	Theorem 2✓
$\lambda = \binom{1}{3}$	Theorem 2✓	
$\lambda = \binom{1}{4}$		

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$	Theorem 1✓	Theorem 2✓
$\lambda = \binom{1}{3}$	Theorem 2✓	$\mathcal{A}^{100}$, $\mathcal{A}^{010}$, and $\mathcal{A}^{001}$ are unstable✗
$\lambda = \binom{1}{4}$		

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$	Theorem 1✓	Theorem 2✓
$\lambda = \binom{1}{3}$	Theorem 2✓	$\mathcal{A}^{100}$, $\mathcal{A}^{010}$, and $\mathcal{A}^{001}$ are unstable✗
$\lambda = \binom{1}{4}$	e.g. $\mathcal{A}^{1000}$ and $\mathcal{A}^{0001}$ are unstable✗	

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$	Theorem 1✓	Theorem 2✓
$\lambda = \binom{1}{3}$	Theorem 2✓	$\mathcal{A}^{100}$, $\mathcal{A}^{010}$, and $\mathcal{A}^{001}$ are unstable✗
$\lambda = \binom{1}{4}$	e.g. $\mathcal{A}^{1000}$ and $\mathcal{A}^{0001}$ are unstable✗	already unstable for $\lambda = \binom{1}{3}$✗

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	Theorem 1✓	Theorem 1✓
$\lambda = \binom{1}{2}$	Theorem 1✓	Theorem 2✓
$\lambda = \binom{1}{3}$	Theorem 2✓	$\mathcal{A}^{100}$, $\mathcal{A}^{010}$, and $\mathcal{A}^{001}$ are unstable✗
$\lambda = \binom{1}{4}$	e.g. $\mathcal{A}^{1000}$ and $\mathcal{A}^{0001}$ are unstable✗	already unstable for $\lambda = \binom{1}{3}$✗

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix} \text{ used for zero and hold control}$$

Up to now:

- Stability and Stabilization as important properties and tasks for functional verification of control systems.
- We can include WHRT constraints in control system models.
- We can analyze stability with WHRT constraints.

Open tasks:

- Design controllers with richer interfaces.

Blind, Allgöwer (2015). Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process.

Up to now we considered all components as given and checked whether the origin of the closed loop is asymptotically stable.

Up to now we considered all components as given and checked whether the origin of the closed loop is asymptotically stable.

In the remainder:

Static state feedback stabilization

Given

- a weakly hard real-time constraint λ,
- a discrete-time linear control system (A, B stabilizable),
- and an actuation strategy (zero resp. hold),

Up to now we considered all components as given and checked whether the origin of the closed loop is asymptotically stable.

In the remainder:

Static state feedback stabilization

Given

- a weakly hard real-time constraint λ,
- a discrete-time linear control system (A, B stabilizable),
- and an actuation strategy (zero resp. hold),

find K s.t. the origin of the closed loop system with $u_k = \gamma_k K x_k$ resp. $u_k = \gamma_k K x_k + (1 - \gamma_k) u_{k-1}$ is asymptotically stable for all $\gamma \vdash \lambda$.

Why do we need to develop new methods to solve this problem?

Why do we need to develop new methods to solve this problem?

- We do not want to deal with sequences of length m, since K appears nonlinear in the corresponding matrices, i.e. in $\mathcal{A}^w$.

Why do we need to develop new methods to solve this problem?

- We do not want to deal with sequences of length m, since K appears nonlinear in the corresponding matrices, i.e. in $\mathcal{A}^w$.
- For the computation of a controller one needs to take care that the dimension of the controller does not vary with the chosen strategy, i.e., $\dim K$ may not be dependent on $\dim \xi$.

Why do we need to develop new methods to solve this problem?

- We do not want to deal with sequences of length m, since K appears nonlinear in the corresponding matrices, i.e. in $\mathcal{A}^w$.
- For the computation of a controller one needs to take care that the dimension of the controller does not vary with the chosen strategy, i.e., $\dim K$ may not be dependent on $\dim \xi$.

$\Rightarrow$ This is covered by the introduction of an alternative discretization

New sequences are derived based on γ :

- $(\tau_k)_{k\in\mathbb{N}}$ contains all time instants when the computation is executed in time.
- $(\alpha_k)_{k\in\mathbb{N}}$ contains the number of deadline misses after an executed computation.

New sequences are derived based on γ :

- $(\tau_k)_{k\in\mathbb{N}}$ contains all time instants when the computation is executed in time.
- $(\alpha_k)_{k\in\mathbb{N}}$ contains the number of deadline misses after an executed computation.
- Example:

$$\gamma = (0, 1, 0, 0, 1, 0, 1, \dots)$$
$$\tau = (1, 4, 6, \dots)$$
$$\alpha = (2, 1, \dots)$$

New sequences are derived based on γ :

- $(\tau_k)_{k\in\mathbb{N}}$ contains all time instants when the computation is executed in time.
- $(\alpha_k)_{k\in\mathbb{N}}$ contains the number of deadline misses after an executed computation.
- Example:

$$\begin{aligned}\gamma &= (0,1,0,0,1,0,1,\dots)\\ \tau &= (1,4,6,\dots)\\ \alpha &= (2,1,\dots)\end{aligned}$$

Those sequences are used for the restricted dynamical system

$$\tilde{\xi}_{k+1} = \tilde{A}_{\alpha_k}\tilde{\xi}_k$$

with $\tilde{\xi}_k := \xi_{\tau_k}$.

New sequences are derived based on γ :

- $(\tau_k)_{k\in\mathbb{N}}$ contains all time instants when the computation is executed in time.
- $(\alpha_k)_{k\in\mathbb{N}}$ contains the number of deadline misses after an executed computation.
- Example:

$$\gamma = (0,1,0,0,1,0,1,\dots)$$
$$\tau = (1,4,6,\dots)$$
$$\alpha = (2,1,\dots)$$

Those sequences are used for the restricted dynamical system

$$\tilde{\xi}_{k+1} = \tilde{A}_{\alpha_k}\tilde{\xi}_k$$

with $\tilde{\xi}_k := \xi_{\tau_k}$.

- Stability of the original closed loop system can be deduced from the $\tilde{x}$ dynamics of the restricted system.
- The $\tilde{x}$ dynamics are in the form $A_l^x + B_l^x K$.

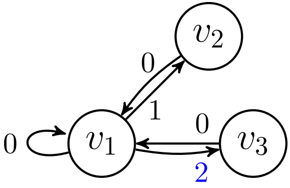
Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

Graph representation (Example $\gamma \vdash \left\langle {2 \atop 5} \right\rangle$):

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

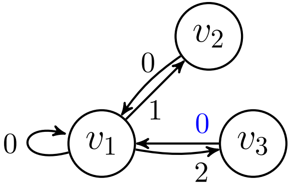
Graph representation (Example $\gamma \vdash \langle {2 \atop 5} \rangle$):



$$\gamma = (1, 0, 0, 1,$$
$$\alpha = (2,$$

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

Graph representation (Example $\gamma \vdash \langle {2 \atop 5} \rangle$):

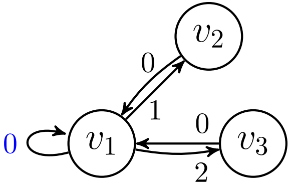


$$\gamma = (1, 0, 0, 1, 1,$$

$$\alpha = (2, 0,$$

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

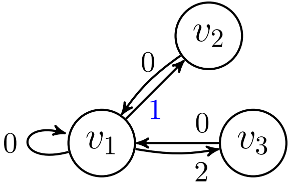
Graph representation (Example $\gamma \vdash \langle {2 \atop 5} \rangle$):



$$\gamma = (1, 0, 0, 1, 1, 1,$$
$$\alpha = (2, 0, 0,$$

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

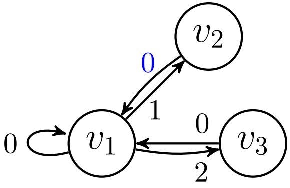
Graph representation (Example $\gamma \vdash \left\langle {2 \atop 5} \right\rangle$):



$$\gamma = (1, 0, 0, 1, 1, 1, 0, 1,$$
$$\alpha = (2, 0, 0, 1,$$

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

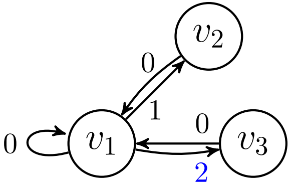
Graph representation (Example $\gamma \vdash \langle {2 \atop 5} \rangle$):



$$\gamma = (1, 0, 0, 1, 1, 1, 0, 1, 1,$$
$$\alpha = (2, 0, 0, 1, 0,$$

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

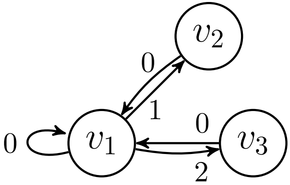
Graph representation (Example $\gamma \vdash \left\langle {2 \atop 5} \right\rangle$):



$$\gamma = (1,0,0,1,1,1,0,1,1,0,0,1,\dots)$$
$$\alpha = (2,0,0,1,0,2,\dots)$$

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

Graph representation (Example $\gamma \vdash \langle{2 \atop 5}\rangle$):



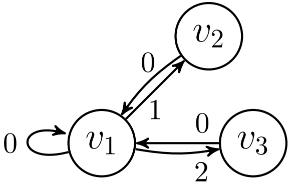
$$\gamma = (1,0,0,1,1,1,0,1,1,0,0,1,\dots)$$
$$\alpha = (2,0,0,1,0,2,\dots)$$

Recall previous example:

$$\gamma = (0,1,0,0,1,0,1,\dots)$$
$$\tau = (1,4,6,\dots)$$
$$\alpha = (2,1,\dots)$$

Derive a graph, s.t. the concatenation of its labels represents all possible sequences α that correspond to all $\gamma \vdash \lambda$.

Graph representation (Example $\gamma \vdash \langle {2 \atop 5} \rangle$):



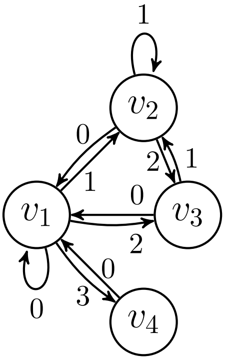
$$\gamma = (1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, \dots)$$
$$\alpha = (2, 0, 0, 1, 0, 2, \dots)$$

Recall previous example:

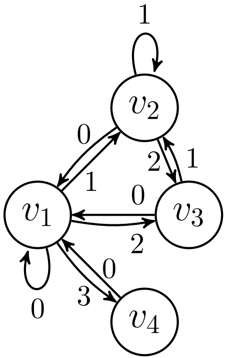
$$\gamma = (0, 1, 0, 0, 1, 0, 1, \dots)$$
$$\tau = (1, 4, 6, \dots)$$
$$\alpha = (2, 1, \dots)$$

This sequence does not satisfy the constraint!

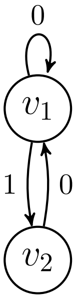
Example $\gamma \vdash \binom{2}{5}$:



Example $\gamma \vdash \binom{2}{5}$:



Example $\gamma \vdash \overline{\left\langle{2 \atop 5}\right\rangle}$:



Theorem 3

Given a WHRT constraint λ and its graph representation $\mathcal{G}$,

Linsenmayer, Allgöwer (2017). Stabilization of Networked Control Systems with weakly hard real-time dropout description.

Theorem 3

Given a WHRT constraint λ and its graph representation $\mathcal{G}$, the discrete-time linear control system is state feedback stabilizable under the zero (resp. hold) strategy w.r.t. all $\lambda' \preceq \lambda$

Linsenmayer, Allgöwer (2017). Stabilization of Networked Control Systems with weakly hard real-time dropout description.

Theorem 3

Given a WHRT constraint λ and its graph representation $\mathcal{G}$, the discrete-time linear control system is state feedback stabilizable under the zero (resp. hold) strategy w.r.t. all $\lambda' \preceq \lambda$, if there exist n_V symmetric matrices $S_1, \ldots, S_{n_V} \in \mathbb{R}^{n \times n}$ and matrices $G \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n_u \times n}$, satisfying

$$\begin{bmatrix} G + G^\intercal - S_i & (A_l^x G + B_l^x F)^\intercal \\ A_l^x G + B_l^x F & S_j \end{bmatrix} > 0, \ \forall (i, j, l) \in E.$$

Linsenmayer, Allgöwer (2017). Stabilization of Networked Control Systems with weakly hard real-time dropout description.

Theorem 3

Given a WHRT constraint λ and its graph representation $\mathcal{G}$, the discrete-time linear control system is state feedback stabilizable under the zero (resp. hold) strategy w.r.t. all $\lambda' \preceq \lambda$, if there exist n_V symmetric matrices $S_1, \ldots, S_{n_V} \in \mathbb{R}^{n\times n}$ and matrices $G \in \mathbb{R}^{n\times n}$, $F \in \mathbb{R}^{n_u\times n}$, satisfying

$$\begin{bmatrix} G + G^\intercal - S_i & (A_l^x G + B_l^x F)^\intercal \\ A_l^x G + B_l^x F & S_j \end{bmatrix} > 0, \ \forall (i,j,l) \in E.$$

A stabilizing state feedback controller is given by $K = FG^{-1}$.

Linsenmayer, Allgöwer (2017). Stabilization of Networked Control Systems with weakly hard real-time dropout description.

Procedure to derive stabilizing controller:

Procedure to derive stabilizing controller:

1. Derive a labeled graph $\mathcal{G}$ for the constraint λ.

Procedure to derive stabilizing controller:

1. Derive a labeled graph $\mathcal{G}$ for the constraint λ.
2. Compute the matrices A_l, B_l for all labels,

$$A_l = A^{l+1}$$
$$B_l = \begin{cases} A^l B & \text{zero strategy} \\ \sum_{i=0}^{l} A^i B & \text{hold strategy.} \end{cases}$$

Procedure to derive stabilizing controller:

1. Derive a labeled graph $\mathcal{G}$ for the constraint λ.
2. Compute the matrices A_l, B_l for all labels,

$$A_l = A^{l+1}$$
$$B_l = \begin{cases} A^l B & \text{zero strategy} \\ \sum_{i=0}^{l} A^i B & \text{hold strategy.} \end{cases}$$

3. Find matrices $F, G, S_1, \ldots, S_{n_V}$, satisfying

$$\begin{bmatrix} G + G^\intercal - S_i & (A_l G + B_l F)^\intercal \\ A_l G + B_l F & S_j \end{bmatrix} > 0, \quad \forall (i, j, l) \in E.$$

Procedure to derive stabilizing controller:

1. Derive a labeled graph $\mathcal{G}$ for the constraint λ.
2. Compute the matrices A_l, B_l for all labels,

$$A_l = A^{l+1}$$
$$B_l = \begin{cases} A^l B & \text{zero strategy} \\ \sum_{i=0}^{l} A^i B & \text{hold strategy.} \end{cases}$$

3. Find matrices $F, G, S_1, \ldots, S_{n_V}$, satisfying

$$\begin{bmatrix} G + G^\intercal - S_i & (A_l G + B_l F)^\intercal \\ A_l G + B_l F & S_j \end{bmatrix} > 0, \quad \forall (i,j,l) \in E.$$

4. Compute a stabilizing state feedback controller for all $\lambda' \preceq \lambda$,

$$K = F G^{-1}.$$

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ was used for zero and hold control

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ was used for zero and hold control

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{2}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	unstable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$
$\lambda = \binom{1}{4}$	unstable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$	unstable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ was used for zero and hold control

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{2}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stabilization with $K = \begin{bmatrix} -0.33 & -0.54 \end{bmatrix}$ ✓
$\lambda = \binom{1}{4}$	unstable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$	unstable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ was used for zero and hold control

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{2}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stabilization with $K = \begin{bmatrix} -0.33 & -0.54 \end{bmatrix}$ ✓
$\lambda = \binom{1}{4}$	no stabilization with Thm 4 ✗	unstable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$

Plant:

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k$$

Embedded Controller:

$K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ was used for zero and hold control

	Zero strategy	Hold strategy
$\lambda = \binom{2}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{2}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓
$\lambda = \binom{1}{3}$	stable with $K = \begin{bmatrix} -0.35 & -0.85 \end{bmatrix}$ ✓	stabilization with $K = \begin{bmatrix} -0.33 & -0.54 \end{bmatrix}$ ✓
$\lambda = \binom{1}{4}$	no stabilization with Thm 4 ✗	stabilization with $K = \begin{bmatrix} -0.29 & -0.48 \end{bmatrix}$ ✓

Outline

In this talk:

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.
- We presented a possibility to include WHRT constraints in control system models.

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.
- We presented a possibility to include WHRT constraints in control system models.
- We showed procedures for analyzing stability with WHRT constraints.

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.
- We presented a possibility to include WHRT constraints in control system models.
- We showed procedures for analyzing stability with WHRT constraints.
- We derived a procedure to design a controller for a given WHRT constraint.

Summary

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.
- We presented a possibility to include WHRT constraints in control system models.
- We showed procedures for analyzing stability with WHRT constraints.
- We derived a procedure to design a controller for a given WHRT constraint.

Possible other directions:

- Design optimal/robust controllers for systems with WHRT constraints.

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.
- We presented a possibility to include WHRT constraints in control system models.
- We showed procedures for analyzing stability with WHRT constraints.
- We derived a procedure to design a controller for a given WHRT constraint.

Possible other directions:

- Design optimal/robust controllers for systems with WHRT constraints.

Answers to main questions

Q: Is it necessary that all deadlines are met to guarantee properties of the control system?

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.
- We presented a possibility to include WHRT constraints in control system models.
- We showed procedures for analyzing stability with WHRT constraints.
- We derived a procedure to design a controller for a given WHRT constraint.

Possible other directions:

- Design optimal/robust controllers for systems with WHRT constraints.

Answers to main questions

Q: Is it necessary that all deadlines are met to guarantee properties of the control system?
A: No. We showed that we can also integrate richer interfaces, as WHRT constraints, such that we can e.g. guarantee stability although not all deadlines are met.

In this talk:

- We motivated why stability and stabilization are important properties and tasks in the context of functional verification of control systems.
- We presented a possibility to include WHRT constraints in control system models.
- We showed procedures for analyzing stability with WHRT constraints.
- We derived a procedure to design a controller for a given WHRT constraint.

Possible other directions:

- Design optimal/robust controllers for systems with WHRT constraints.

Answers to main questions

Q: Is it necessary that all deadlines are met to guarantee properties of the control system?
A: No. We showed that we can also integrate richer interfaces, as WHRT constraints, such that we can e.g. guarantee stability although not all deadlines are met.

Thank you for your attention!

Stability analysis and control design for weakly hard real-time systems

How can richer interfaces, e.g. m-k guarantees, be used for the functional verification of control systems

Steffen Linsenmayer (partially work of and with R. Blind and F. Allgöwer)

Institute for Systems Theory and Automatic Control
University of Stuttgart, Germany

ESWEEK 2017, October 15, 2017