Networked Control Systems: a Discrete-time Approach

Nathan van de Wouw
*Dynamics and Control Group,*
*Department of Mechanical Engineering,*
*Eindhoven University of Technology,*
*the Netherlands*

Université Catholique de Louvain, October 5th, 2010
Acknowledgement of Collaborators

- Maurice Heemels, Tijs Donkers, Henk Nijmeijer (Eindhoven University of Technology, the Netherlands)
- Marieke Cloosterman (ASML Research, the Netherlands)
- Laurentiu Hetel (University of Lille, France)
- Jamal Daafouz (University of Nancy, France)
- Payam Naghsthabrizi (Ford Research, U.S.A.)
- Joao Hespanha (University of California, Santa Barbara, U.S.A.)
Contents

► Introduction on Networked Control Systems (NCS)

► Discrete-time Modelling of linear NCS:
  • Time-varying sampling intervals
  • Communication delays
  • Packet dropouts

► Stability analysis of linear NCS

► Tracking control of linear NCS

► NCS including communication protocols

► Conclusions & Outlook on Future Work
Introduction

- Cooperative Adaptive Cruise Control
- Cooperative robotics
- Wireless Sensor Networks
- Wireless/distributed control of water distribution networks (EU-project WIDE)
- Wireless Motion Control
- Etc., etc.

NCS: Control systems in which controllers, sensors and actuators are communicating over a network
Introduction

To network ...

- Ease of installation and maintenance
- Large flexibility (especially with WSN)
- Lower costs
- Less wires (less wear, less disturbances, less weight!) in case of WSN
- Control of physically distributed systems
...or not to network:

- Varying sampling/transmission interval
- Varying communication delays
- Packet loss
- Communication constraints through shared network
- Quantization
Existing approaches towards modelling/stability analysis:

1. Emulation approach (Nešić, Teel, Carnevale, Tabarra, Heemels, van de Wouw):
   - Time-varying sampling intervals, SMALL delays
   - Communication constraints, general classes of scheduling protocols
   - Nonlinear systems
   - Continuous-time control synthesis based on continuous-time model
Existing approaches towards modelling/stability analysis:

1. **Emulation approach** (Nešić, Teel, Carnevale, Tabarra, Heemels, van de Wouw):
   - Time-varying sampling intervals, SMALL delays
   - Communication constraints, general classes of scheduling protocols
   - Nonlinear systems
   - Continuous-time control synthesis based on continuous-time model

2. **Modelling in terms of delay-impulsive differential equations** (Naghsthabrizi, Hespanha, Teel, van de Wouw):
   - Time-varying sampling intervals, LARGE delays
   - Linear systems
   - LMI-based stability analysis and controller synthesis
Existing approaches towards modelling/stability analysis:

1. **Emulation approach** (Nešić, Teel, Carnevale, Tabarra, Heemels, van de Wouw):
   - Time-varying sampling intervals, SMALL delays
   - Communication constraints, general classes of scheduling protocols
   - Nonlinear systems
   - Continuous-time control synthesis based on continuous-time model

2. **Modelling in terms of delay-impulsive differential equations** (Naghsthabrizi, Hespanha, Teel, van de Wouw):
   - Time-varying sampling intervals, LARGE delays
   - Linear systems
   - LMI-based stability analysis and controller synthesis

3. **Discrete-time modelling** (Zhang, Hetel, Fujioka, Garcia, Cloosterman, van de Wouw, Heemels, Donkers):
   - Time-varying sampling intervals, LARGE delays, packet dropouts
   - Communication constraints, particular classes of scheduling protocols
   - Linear systems
   - LMI-based stability analysis and controller synthesis
Network Control Systems: Modelling

Assumptions:

- Time-driven sensor (sampling times: $s_k$)
- Event-driven controller
- Event-driven actuator

Time-delays:

- Sensor-to-controller $\tau_{sc,k}$
- Controller-to-actuator $\tau_{ca,k}$
- Computational delay $\tau_{c,k}$
- $\tau_k = \tau_{sc,k} + \tau_{ca,k} + \tau_{c,k}$
Network-induced uncertainties:

- **Time-varying delays:** \( \tau_k \in [\tau_{\text{min}}, \tau_{\text{max}}] \)

- **Time-varying sampling intervals:**
  \( h_k = s_{k+1} - s_k \in [h_{\text{min}}, h_{\text{max}}] \)

- **Packet dropouts:**
  \[
  m_k = \begin{cases} 
  1, & \text{if } u_k \text{ is dropped} \\
  0, & \text{if } u_k \text{ is not dropped}
  \end{cases}
  \]

- **Maximum of \( \bar{\delta} \) subsequent dropouts:**
  \[
  \sum_{v=k-\bar{\delta}}^{k} m_v \leq \bar{\delta}
  \]
Network Control Systems: Modelling

Assumptions:

- **Time-varying delays:** \( \tau_k \in [\tau_{\text{min}}, \tau_{\text{max}}] \)
- **Time-varying sampling intervals:**
  \( h_k = s_{k+1} - s_k \in [h_{\text{min}}, h_{\text{max}}] \)
- **Packet dropouts:**
  \[
  m_k = \begin{cases} 
  1, & u_k \text{ is dropped} \\
  0, & u_k \text{ is not dropped}
  \end{cases}
  \]

Problem:
How to guarantee stability in the face of these network-induced uncertainties
Continuous-time (sampled-data) dynamics of the linear plant:

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu^*(t) \\
u^*(t) &= u_{k+j-d-\delta} \quad \text{for} \ t \in [s_k + t^k_j, s_k + t^k_{j+1}),
\end{align*}
\]

where \( \underline{d} := \lfloor \frac{\tau_{\min}}{h_{\max}} \rfloor \), \( \overline{d} := \lceil \frac{\tau_{\max}}{h_{\min}} \rceil \) and \( t^k_j \in [0, h_k] \) the actuation update instants.
Discrete-time Model

- Use an extended state vector: \( \xi_k = \begin{pmatrix} x_k^T & u_{k-1}^T & \ldots & u_{k-d-\delta}^T \end{pmatrix}^T \),

\( x_k := x(s_k) \)

- Uncertain parameters: \( \theta_k := (h_k, t_{1}^{k}, \ldots, t_{d+\delta-d}^{k}) \)

- Discrete-time uncertain NCS model:

\[
\xi_{k+1} = \tilde{A}(\theta_k)\xi_k + \tilde{B}(\theta_k)u_k,
\]

where

\[
\tilde{A}(\theta_k) = \begin{pmatrix}
\Lambda(\theta_k) & M_{d+\delta-1}(\theta_k) & M_{d+\delta-2}(\theta_k) & \ldots & M_1(\theta_k) & M_0(\theta_k) \\
0 & 0 & I & 0 & \ldots & 0 & 0 \\
0 & I & 0 & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & 0 & 0 & 0 \\
0 & \ldots & \ldots & \ldots & 0 & I & 0
\end{pmatrix}
\]

and

\[
\Lambda(\theta_k) = e^{Ah_k}, \quad \tilde{B}(\theta_k) = \begin{pmatrix}
M_{d+\delta}(\theta_k) \\
I \\
0 \\
\vdots \\
0
\end{pmatrix}, \quad M_j(\theta_k) = \begin{cases} 
\int_{h_k-t_j^{k}}^{h_k-t_{j+1}^{k}} e^{As}dsB & \text{if } 0 \leq j \leq d + \delta - d, \\
0 & \text{if } d + \delta - d < j \leq d + \delta
\end{cases}
\]
BUT.....LET’S KEEP IT SIMPLE...

Consider the small-delay case, with a constant sampling interval and no packet dropouts
Continuous-time (sampled-data) dynamics of the linear plant:

\[
\dot{x}(t) = Ax(t) + Bu^*(t)
\]

\[
u^*(t) = u_k \quad \text{for } t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1})
\]
Discrete-time Model

- Use an extended state vector: $\xi_k = \begin{pmatrix} x_k^T & u_{k-1}^T \end{pmatrix}^T$, $x_k := x(s_k)$

- Uncertain parameters: $\theta_k := \tau_k$

- Discrete-time uncertain NCS model:

$$\xi_{k+1} = \tilde{A}(\tau_k)\xi_k + \tilde{B}(\tau_k)u_k$$

where

$$\tilde{A}(\tau_k) = \begin{pmatrix} e^{Ah} & \int_{h-\tau_k}^h e^{As} ds B \\ 0 & \int_0^{h-\tau_k} e^{As} ds B \end{pmatrix}, \quad \tilde{B}(\tau_k) = \begin{pmatrix} \int_0^{\tau_k} e^{As} ds B \\ I \end{pmatrix}$$
Discrete-time Closed-loop Model

- Discrete-time uncertain NCS model:

\[
\xi_{k+1} = \tilde{A}(\tau_k)\xi_k + \tilde{B}(\tau_k)u_k
\]

- In closed-loop with the static discrete-time extended-state feedback controller \( u_k = -K\xi_k = -\bar{K}x_k - K_uu_{k-1} \):

\[
\xi_{k+1} = \left( \tilde{A}(\tau_k) - \tilde{B}(\tau_k)K \right)\xi_k
= \begin{bmatrix}
    e^{Ah} - \int_{0}^{h-\tau_k} e^{As}ds B \bar{K} & \int_{h-\tau_k}^{h} e^{As}ds B - \int_{0}^{h-\tau_k} e^{As}ds B K_u \\
    -\bar{K} & -K_u
\end{bmatrix} \xi_k
\]

- Discrete-time linear system with exponential uncertainty: varying delay \( \tau_k \)
Discrete-time uncertain closed-loop NCS model:

\[ \xi_{k+1} = \left( \tilde{A}(\tau_k) - \tilde{B}(\tau_k) K \right) \xi_k =: H(\tau_k) \xi_k \]

A first approach using a common quadratic Lyapunov function:

\[ V(\xi) = \xi^T P \xi, \quad P = P^T > 0 \]

Closed-loop NCS is globally asymptotically stable if there exists \( P = P^T > 0, \ 0 < \gamma < 1 \) such that

\[ H^T(\tau) PH(\tau) - P < -\gamma P, \quad \forall \tau \in [\tau_{\text{min}}, \tau_{\text{max}}] \]

Infinite set of Linear Matrix Inequalities (LMIs)

How to arrive at a finite number of LMIs?
- **Basic idea:** embed the uncertainty matrix set \( H(\tau), \tau \in [\tau_{\text{min}}, \tau_{\text{max}}] \) in a polytopic set with generators (vertices) \( H_i, i = 1, \ldots, N \):

\[
\{ H(\tau) \mid \tau \in [\tau_{\text{min}}, \tau_{\text{max}}] \} \subseteq \text{convex hull}(H_1, \ldots, H_N)
\]

- **Discrete-time uncertain closed-loop NCS model**

\[
\xi_{k+1} = H(\tau_k)\xi_k
\]

is **globally asymptotically stable** if there exist \( P = P^T > 0, 0 < \gamma < 1 \) such that the following finite set of LMIs are satisfied:

\[
H_i^T P H_i - P < -\gamma P, \quad \forall i = 1, \ldots, N
\]
Basic idea: embed the uncertainty matrix set $H(\tau)$, $\tau \in [\tau_{min}, \tau_{max}]$ in a polytopic set with generators (vertices) $H_i$, $i = 1, \ldots, N$:

$$\{H(\tau) \mid \tau \in [\tau_{min}, \tau_{max}]\} \subseteq \text{convex hull}(H_1, \ldots, H_N)$$

How to get such a polytopic overapproximation?
Methods for obtaining polytopic overapproximations based on

- **Interval matrices**
  (Cloosterman, van de Wouw, Heemels, Nijmeijer, CDC 2006)

- **Taylor series**
  (Hetel, Daafouz, Iung, TAC 2007)

- **Real Jordan form**
  (Cloosterman, van de Wouw, Heemels, Nijmeijer, CDC 2007, ACC 2008, TAC 2009)

- **Gridding (and norm bounding of approximation error)**
  (Fujioka, ACC 2008), (Suh, Automatica 2008),
  (Donkers, Hetel, Heemels, van de Wouw, HSCC 2009, TAC 2010), (Skaf, Boyd, TAC 2009)

- **Cayley-Hamilton theorem**
  (Gielen et. al. Automatica 2010)

- **Comparison of methods for polytopic overapproximation for NCS**
  (Heemels et. al., HSCC2010)
Extensions & Remarks

(Cloosterman et al., CDC 2007, ACC 2008, Automatica 2010), (Hetel et al., CDC 2009)

- Stability of the intersample behavior (and therefore the sampled-data NCS) is also guaranteed

- Variations in the sampling interval $h_k \in [h_{\min}, h_{\max}]$ and the large delay case $\tau_k > h_k$

- Packet Dropouts: modeled as prolongation of maximal sampling interval, maximal delay or separate (hybrid) model

- Usage of parameter-dependent Lyapunov functions, which can be proven to be more general than discrete-time Lyapunov-Krasovskii methods

- LMI-based controller synthesis LMIs for (augmented) state feedbacks: $u_k = -K_{\xi} \xi_k$ and $u_k = -\bar{K}x_k$
Example from the document printing domain with continuous-time dynamics:

\[
\dot{x} = Ax + Bu^*(t), \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ b \end{bmatrix},
\]

with \( b := \frac{nrR}{J_M + n^2J_R} \), \( x = [x_s(t) \quad \dot{x}_s(t)]^T \)

State-feedback \( u_k = -\bar{K}x_k \)
Example with periodic delays

- Constant sampling interval: $h = 1 \text{ ms}$
- Periodically varying delays: $\tau_a, \tau_b, \tau_a, \tau_b, \ldots$ ($\tau_a = 0.2h, \tau_b = 0.6h$)
- $\bar{K} = [50 \ 1.18]$

- Also possible for varying sampling intervals $h_k$ showing similar effects
Example with large delays

- Constant sampling interval: $h = 1$ ms
- $\tau_{min} = 0$, $\tilde{K} = [50 \quad K_2]$
- Two methods for polytopic overapproximation:
  - Method 1: based on interval matrices
  - Method 2: based on the Jordan form approach
Example with delays and packet dropouts

- Constant sampling interval: \( h = 1 \text{ ms} \)
- \( \tau_{\text{min}} = 0, \bar{K} = [50 \ K_b] \)
- Packet dropouts, with the maximum number of subsequent dropouts \( \bar{\delta} = 0, 1, 2 \)
Approximate Tracking Control Problem

- Continuous-time (sampled-data) dynamics of the plant:
  \[
  \begin{align*}
  \dot{x}(t) &= Ax(t) + Bu^*(t), \quad x(0) = x_0 \\
  u^*(t) &= u_k, \quad \text{for } t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1})
  \end{align*}
  \]

- Time-varying sampling intervals, small delays, no packet dropouts

- Desired trajectory: \( x^d(t) \)

- Approximate tracking control problem:
  
  Design a discrete-time tracking controller for \( u_k \) such that \( \lim_{t \to \infty} |x(t) - x^d(t)| \leq \varepsilon \) for some small \( \varepsilon > 0 \)
  
  Performance is about making \( \varepsilon \) small
Controller Design (Feedforward + Feedback):

\[ u_k := u_{e}^{ff}(s_k) - \bar{K} \left( x(s_k) - x^d(s_k) \right) \]

- Exact feedforward \( u_{e}^{ff}(t) \) induces the desired solution:

\[ \dot{x}^d(t) = Ax^d(t) + B u_{e}^{ff}(t) \]
Controller Design for Tracking

- **Controller Design** (Feedforward + Feedback):
  \[ u_k := u_{eq}^{ff}(s_k) - \bar{K} \left( x(s_k) - x^d(s_k) \right) \]

  - Exact feedforward \( u_{eq}^{ff}(t) \) induces the desired solution:
    \[ \dot{x}^d(t) = Ax^d(t) + Bu_{eq}^{ff}(t) \]

- Sampled feedforward in face of ZOH, time-varying sampling times and delays:
  \[ \Delta u^{ff}(t) = u_{eq}^{ff}(s_k) - u_{eq}^{ff}(t) \text{ for } t \in \left[ s_k + \tau_k, s_{k+1} + \tau_{k+1} \right) \]

Error due to:
- Zero-order hold
- Unknown delays
Sampled-data tracking error dynamics \( (e = x - x^d) \):

\[
\dot{e}(t) = Ae(t) - B \bar{K}e(s_k) + B \Delta u^{ff}(t) \quad (\Delta u^{ff}(t) = \text{feedforward error})
\]

for \( t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1}) \)

We aim at \textit{input-to-state stability (ISS)}, i.e.

\[
|e(t)| \leq \beta(|\bar{e}(0)|, t) + \gamma \left( \sup_{0 \leq s \leq t} |\Delta u^{ff}(s)| \right)
\]
Sampled-data tracking error dynamics \((e = x - x^d)\):

\[
\dot{e}(t) = Ae(t) - B \tilde{K}e(s_k) + B \Delta u^{ff}(t)
\]

\((\Delta u^{ff}(t) = \text{feedforward error})\)

for \(t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1}]\)

We aim at input-to-state stability (ISS), i.e.

\[
|e(t)| \leq \beta(|\tilde{e}(0)|, t) + \gamma \left( \sup_{0 \leq s \leq t} |\Delta u^{ff}(s)| \right)
\]
Sampled-data tracking error dynamics ($e = x - x^d$):

$$\dot{e}(t) = Ae(t) - B \bar{K} e(s_k) + B \Delta u^{ff}(t) \quad (\Delta u^{ff}(t) = \text{feedforward error})$$

for $t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1}]$

We aim at input-to-state stability (ISS), i.e.

$$|e(t)| \leq \beta(|\bar{e}(0)|, t) + \gamma \left( \sup_{0 \leq s \leq t} |\Delta u^{ff}(s)| \right)$$
Tracking Performance

- Sampled-data tracking error dynamics ($e = x - x^d$):

$$\dot{e}(t) = Ae(t) - B \bar{K} e(s_k) + B \Delta u^{ff}(t) \quad (\Delta u^{ff}(t) = \text{feedforward error})$$

for $t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1})$

- We aim at input-to-state stability (ISS), i.e.

$$|e(t)| \leq \beta(|\bar{e}(0)|, t) + \gamma (\sup_{0 \leq s \leq t} |\Delta u^{ff}(s)|)$$
Sampled-data tracking error dynamics \((e = x - x^d)\):

\[
\dot{e}(t) = Ae(t) - B \bar{K}e(s_k) + B \Delta u^{ff}(t) \quad (\Delta u^{ff}(t) = \text{feedforward error})
\]

for \(t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1})\)

We aim at input-to-state stability (ISS), i.e.

\[
|e(t)| \leq \beta(|\bar{e}(0)|, t) + \gamma(\sup_{0 \leq s \leq t} |\Delta u^{ff}(s)|)
\]

ISS gain function \(\gamma(\cdot)\) determines ultimate bound on the tracking error depending on 1) plant properties, 2) network properties, 3) controller

Stability LMIs presented before also guarantee ISS!
Ultimate bound for the steady-state tracking error:

\[ |e(t)| \leq \gamma \sup |\Delta u^{ff}(t)| \]

based on:

1. The gain \( \gamma \) amplifying feedforward errors to tracking errors
   \( \gamma \) depends on 1. plant, 2. network, 3. controller

2. Bound on the feedforward error:

\[ \sup |\Delta u^{ff}(t)| \leq (\tau_{\max} + h_{\max}) \max_{t \in \mathbb{R}} \left| \frac{\partial u^{ff}_e(t)}{\partial t} \right| \]

depends on 1. network, 2. plant+desired trajectory
Ultimate bound for the steady-state tracking error:

\[ |e(t)| \leq \gamma \sup |\Delta u^{ff}(t)| \]

based on:

1. The gain \( \gamma \) amplifying feedforward errors to tracking errors
   \( \gamma \) depends on 1. plant, 2. network, 3. controller

2. Bound on the feedforward error:

\[ \sup |\Delta u^{ff}(t)| \leq (\tau_{\text{max}} + h_{\text{max}}) \max_{t \in \mathbb{R}} \left| \frac{\partial u^{ff}_e(t)}{\partial t} \right| \]

depends on 1. network, 2. plant+desired trajectory

Steady-state performance depends on:

1. plant properties
2. network properties
3. controller
4. desired trajectory
Example from the document printing domain with continuous-time dynamics:

\[
\dot{x} = Ax + B u^*(t), \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ b \end{bmatrix},
\]

with \( b := \frac{nr_R}{J_M + n^2 J_R} \), \( x = [x_s(t) \quad \dot{x}_s(t)]^T \)

Desired trajectory: \( x_d(t) = \begin{bmatrix} A_d \sin(\omega t) \\ A_d \omega \cos(\omega t) \end{bmatrix}^T \),

with \( A_d = 0.01 \) and \( \omega = 2\pi \)

Exact feedforward is given by \( u_{eff}^f(t) = -\frac{A_d \omega^2}{b} \sin(\omega t) \)

Feedback gain matrix: \( \bar{K} = \begin{bmatrix} 50 & 1.18 \end{bmatrix} \)
Motion Control Example

- Constant sampling interval:
  \[ h = 5 \times 10^{-3} \text{ s} \]

- Network delays: \( \tau \in [0, \tau_{\text{max}}] \)

Tracking error bounds
Motion Control Example

- Constant sampling interval: 
  \[ h = 5 \times 10^{-3} \text{ s} \]

- Network delays: \( \tau \in [0, \tau_{\text{max}}] \)

Such plot can help to obtain insight in the tradeoff between

- Network Specs
- Control performance specs

For details on the delay-impulsive approach, see van de Wouw et. al. IJRNC 2010

Tracking error bounds
NCS with communication constraints

- Communication constraints:
  - Network is divided into sensor and actuator nodes
  - Only one node can access the network simultaneously
  - This gives rise to the problem of scheduling (communication protocols)
NCS with communication constraints

Two approaches exist for NCS with communication constraints:

- Continuous-time approach:
  - Work of Walsh, Nešić, Teel, Carnevale, Tabarra, Heemels, van de Wouw
  - General nonlinear plants and (UGES) protocols
  - Continuous-time controllers
  - $h_{\text{min}} = 0 = \tau_{\text{min}} = 0$

- Discrete-time approach:
  - Exploits linearity of plants and controllers
  - Both continuous-time and discrete-time controllers
  - $h_k \in [h_{\text{min}}, h_{\text{max}}]$, $\tau_k \in [\tau_{\text{min}}, \tau_{\text{max}}]$ with $h_{\text{min}} \neq 0$, $\tau_{\text{min}} \neq 0$
  - Specific protocols
Results in a discrete-time switched linear uncertain system

\[ \tilde{x}_{k+1} = \tilde{A}_{\sigma_k, h_k, \tau_k} \tilde{x}_k \]

with state \( \tilde{x} = (x^p, x^c, e^y, e^u) \) consisting of the plant state, controller state, and the network-induced errors on the sensor readings and actuator commands

- **Uncertainty:** unknown time-varying transmission intervals and delays leads to exponential uncertainty terms
- **Switching due to scheduling:** e.g. Round Robin or Try-Once-Discard protocols

- Polytopic overapproximations based on gridding method to embed exponential uncertainties in a polytopic model
- LMI-based stability conditions based on parameter-dependent Lyapunov functions
Illustrative Example (batch reactor without delays)
Linear plant, 2 sensor nodes

- Results on bounds on the transmission interval (given for TOD protocol only)

<table>
<thead>
<tr>
<th>Method</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation based, obtained in [1]</td>
<td>( h_k \in (\varepsilon, 0.07] )</td>
</tr>
<tr>
<td>Theoretical, obtained in [1]</td>
<td>( h_k \in (\varepsilon, 10^{-5}] )</td>
</tr>
<tr>
<td>Theoretical, obtained in [2]</td>
<td>( h_k \in (\varepsilon, 0.01] )</td>
</tr>
<tr>
<td>Theoretical, obtained in [3]</td>
<td>( h_k \in (\varepsilon, 0.0108] )</td>
</tr>
<tr>
<td>Newly obtained theoretical bound</td>
<td>( h_k \in [10^{-4}, 0.066] )</td>
</tr>
</tbody>
</table>

---

[1] Walsh, et al., Trans. CST ’02
[2] Nešić & Teel, TAC ’04
[3] Carnevale, et al., TAC ’07
Illustrative Example (batch reactor with delays)

- Now: varying delays, varying transmission intervals & comm. constraints

Conclusions

- Discrete-time modelling framework for linear networked control systems with
  - time-varying sampling intervals
  - time-varying delays
  - packet dropouts
  - communication constraints

- LMI-based conditions for stability

- LMI-based conditions for controller synthesis

- Results less generic than emulation framework by Nešić, Teel (nonlinear systems, generic classes of protocols)

- Approach is tailored to linear systems, discrete-time controller design, particular protocols and can provide less conservative results
Future Work

- Discrete-time approach for nonlinear NCS (CDC, 2010, joint work with Dragan Nesic)
- Including quantisation in the discrete-time framework
- Decentralised/distributed controllers in a networked setting (EU-Project WIDE)
- Application: Project ‘Connect & Drive’ on the Cooperative Adaptive Cruise Control
- Work on a Matlab Toolbox for Networked Control Systems