Module 9
Decentralized Networked Control Systems: Battling Time-Delays and Perturbations

Ahmad F. Taha

EE 5243: Optimization and Control of Cyber-Physical Systems

Email: ahmad.taha@utsa.edu

Webpage: http://engineering.utsa.edu/ataha

April 14, 2021

Module 9 Outline

We discuss the following topics in this module:

1. Decentralized control: intro and definition
2. Applications of decentralized control
3. Decentralized control + NCSs = DNCS
4. Observer-based decentralized control (OBDC) architecture
5. OBDC + networked control
6. Time-delay modeling and bound-derivation
7. Stability analysis + examples

Introduction to Decentralized Control?

- Decentralized control (DC): used when there is a large scale system (LSS) whose subsystems have interconnections
- Constrained DC: existing constraints on data transfer between subsystems
- Unlike centralized control, DC can be robust and scalable
- Even more robust for systems that are distributed over a large geographical area
- DC algorithms use only local information to produce control laws

Why Decentralized Control?

Decentralized Control: utilization of **local** information to achieve global results

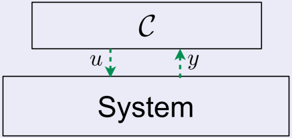
- Replaces *centralized control:* the *orthodox* concept of high performance system driven by a central computer has become *obsolete*
- Very viable and efficient for large-scale interconnected systems
- Examples: transportation systems, communication networks, power systems, economic systems, manufacturing processes
- Emerging synonyms from decentralized control: subsystems, distributed computing, neural networks, parallel processing, etc...
- DC connects graph theory with control & optimization theory
- Very active research area, *overkill?*

Centralized vs. Decentralized Control

Centralized Control

- One system, one control, simple framework
- Classical control, rich history
- Pros: so much theory $\Rightarrow$ so much methods to use
- Cons:

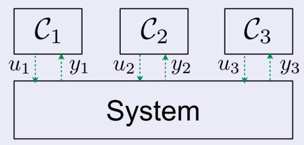
1. Expensive, difficulty to transmit all control output to all actuators at the same time
2. Hard to send all data from sensors to controllers at the same time, for short sampling periods
3. Computationally inefficient for MIMO LSSs



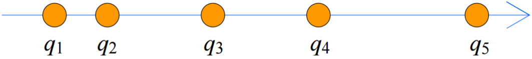
Centralized vs. Decentralized Control

Decentralized Control

- One (or many) system(s), many controls, working in parallel
- Classical control, rich history
- Pros: easier communication, efficient computations
- Cons: more vulnerable to communication networks, network's limitations



DC Motivating Example — Vehicle Spacing [Swigart & Lall, 2010]



- N vehicles in a line, with vehicle i located at position q_i
- Each vehicle is displaced a distance x_i from its original position
- Each vehicle has sensors measuring the relative displacements of its neighbors plus noise
  - Example: $y_1 = \begin{bmatrix} x_1 \\ x_2 - x_1 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$, $y_2 = \begin{bmatrix} x_2 - x_1 \\ x_3 - x_2 \end{bmatrix} + \begin{bmatrix} w_3 \\ w_4 \end{bmatrix}$, etc...
  - System dynamics for each car: $\dot{x}_i = f_i(x_i, u_i, w_i, t)$, $\forall i$
  - How can we design decentralized, local control actions, u_i, such that a certain spacing is maintained?
  - Difference between a global control signal and local one

DC Motivating Example (Cont'd)

- Vehicles can communicate with other vehicles their sensor data:

1. Every vehicle receives the output of every sensor
2. Every vehicle sees only its own sensor data
3. Each vehicle i receives the sensor data of vehicles $i-1$, i, and $i+1$

- Information structure 1. would be considered centralized
- 2. and 3. patterns are decentralized: local controls and data exchanged
- Potential control objectives:

(a) Is there a strategy that will restore unit spacing between the vehicles?

(b) If not, Is a strategy which minimizes mean square relative position error?

$$\mathbb{E}\sum_{i=1}^{N-1}(x_{i+1}-x_i)^2$$

(c) Can we trade-off position error with the mean square distance traveled?

$$\mathbb{E}\sum_{i=1}^{N-1}(x_{i+1}-x_i)^2+\lambda\cdot\mathbb{E}\sum_{i=1}^{N-1}u_i^2$$

Decentralized Networked Control Systems — Why?

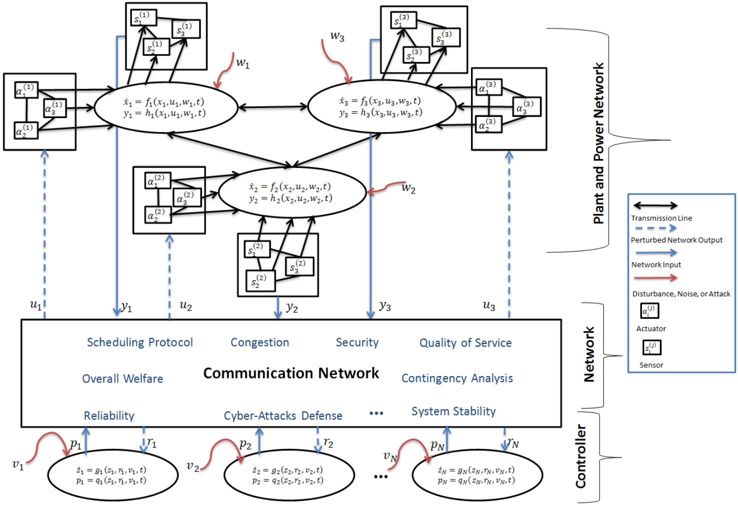
- In many DC applications, data exchanged locally is transmitted through communication networks
- However, it's common to ignore the effect that networks might have on decentralized control strategies
- Hence, studying network effect is very important
- Why?
  - Perturbations caused to exchanged data can influence the decentralized control strategy
  - Privacy issues
  - Time-delays can lead to asynchrony in control actions (think of the moving cars example)

So, what now? DNCS System Description

Module plan:

1. Study a generic decentralized control law for dynamical systems
2. Understand the solution of such DC law
3. Insert a communication network
4. Map DC to NCSs
5. Study system description and dynamics
6. Analyze effect of time-delays and perturbations on DNCSs

Decentralized Netwokred Control Systems — Example



Decentralized Control + Networked Control

= *Decentralized Networked Control System (DNCS)*

Outline

1. Review an OBDC design for non-networked systems
2. Derive dynamics of the OBDC with a network
3. Map the DNCS formation to a typical NCS setup
4. Time-delay analysis of the the DNCS
5. Stability Analysis – Bounds on the time-delay
6. Numerical Results

Observer-Based Decentralized Control (OBDC)

- Different decentralized control strategies have been developed
- An important class of DC architectures is Observer-Based Decentralized Control (OBDC)
- **Basic idea:** develop decentralized state-observers that use local information and define a control law based on the estimate
- OBDC helps in reducing the number of sensors needed for estimation & control
- Authors in [Ha & Trinh, 2004] developed an OBDC for multi-agent systems such that:
  - No information transfer between controllers is required
  - Under certain conditions, closed-loop system is stable
  - Observer's order can be arbitrarily selected

OBDC Plant Dynamics & Objective

- Large-scale system where the plant dynamics are described as follow:

$$\begin{cases} \dot{x} = Ax + \sum_{i=1}^{N} B_i u_i \\ y_i = C_i x, \;\; i = 1, 2, \ldots, N \end{cases}$$

$$\begin{cases} u = \begin{bmatrix} u_1^\top & \ldots & u_N^\top \end{bmatrix}^\top, \; y = \begin{bmatrix} y_1^\top & \ldots & y_N^\top \end{bmatrix}^\top \\ B = \begin{bmatrix} B_1 & \ldots & B_N \end{bmatrix}, \; C = \begin{bmatrix} C_1^\top & \ldots & C_N^\top \end{bmatrix}^\top. \end{cases}$$

- N local control stations & no information flow between controllers
- Then the plant can be written in the following compact form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

OBDC Objective

Design N local decentralized controllers to generate local control laws for all subsystems, given that we do not have access to the full plant-state.

OBDC Design

- Authors in [Ha & Trinh, 2004] proposed the following controller:

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = -\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} x$$

- Then, $u_i = -F_i x$, $\forall i = 1, \ldots, N$

* Since x is not available, let $F_i = K_i L_i + W_i C_i$, then

$$u_i = -F_i x = -(K_i L_i + W_i C_i)x \approx -K_i z_i - W_i y_i$$

* If $z_i \to L_i x$, then above equation is valid

- Let z_i have the following dynamics:

$$\dot{z}_i = E_i z_i + L_i B_i u_i + G_i y_i$$

* **Design objective:** find E_i, L_i, G_i, W_i, K_i such that:

1. Estimation error converges to zero
2. Local control actions stabilize the system

OBDC Design (Cont'd)

$$\boxed{\dot{z}_i = E_i z_i + L_i B_i u_i + G_i y_i}$$

- The observation error: $e_{o_i} = z_i - L_i x, \qquad i = 1, 2, \ldots, N$
- Plant dynamics with control u_i:

$$\boxed{\dot{x} = Ax + B_i u_i + B_{r_i} u_{r_i}}$$

* u_{r_i} contains $(N-1)$ inputs of the remaining $(N-1)$ subsystems

- Hence, we can write the observation error dynamics as:

$$\begin{aligned}
\dot{e}_{o_i} &= \dot{z}_i - L_i \dot{x} \\
&= E_i z_i + L_i B_i u_i + G_i y_i - L_i \left(Ax + B_i u_i + B_{r_i} u_{r_i}\right) \\
&= E_i z_i + L_i B_i u_i + G_i C_i x - L_i \left(Ax + B_i u_i + B_{r_i} u_{r_i}\right) \\
&\quad + E_i L_i x - E_i L_i x \\
\dot{e}_{o_i} &= E_i e_{o_i} + \left(G_i C_i - L_i A + E_i L_i\right) x - L_i B_{r_i} u_r
\end{aligned}$$

- We want to find design parameters K_i, L_i, G_i, W_i such that $e_{o_i} \to 0$
- *How?*

OBDC Design — Matrix Equations

$$\dot{e}_{o_i} = E_i e_{o_i} + (G_i C_i - L_i A + E_i L_i)\, x - L_i B_{r_i} u_r$$

- We want to find design parameters K_i, L_i, G_i, W_i such that $e_{o_i} \to 0$
- How? Set unwanted terms in the above equations to zero and obtain matrix equations
- Precisely:

$$\begin{aligned} L_i B_{r_i} &= 0 \\ K_i L_i + W_i C_i &= F_i \\ G_i C_i - L_i A + E_i L_i &= 0 \end{aligned}$$

- How can we solve the above nonlinear system of matrix-equations? *Kronecker Products*
- Assumptions:

1. (A, B, C) is controllable and observable
2. (A, B_i, C_i) are stabilizable and detectable
3. Global state feedback control $u = -Fx$ exists, F_i is given

Kronecker Products — A Quick Intro (Thanks Wiki)

- If $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times q}$, then $A \otimes B$ is $mp \times nq$ block matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix},$$

- Precisely:

$$A \otimes B = \begin{bmatrix}
a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\
a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\
\vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\
a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\
\vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\
a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\
a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\
\vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\
a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq}
\end{bmatrix}$$

Properties of Kronecker Products

- Some useful properties:

$$
\begin{aligned}
A \otimes (B + C) &= A \otimes B + A \otimes C \\
(A + B) \otimes C &= A \otimes C + B \otimes C \\
(kA) \otimes B &= A \otimes (kB) = k(A \otimes B) \\
(A \otimes B) \otimes C &= A \otimes (B \otimes C) \\
(A \otimes B)(C \otimes D) &= (AC) \otimes (BD) \\
(A \otimes B)^{\mathrm{T}} &= A^{\mathrm{T}} \otimes B^{\mathrm{T}} \\
(A \otimes B)^{*} &= A^{*} \otimes B^{*}
\end{aligned}
$$

- Solve for matrix X if $AXB = C$ using $\otimes$ product:

$$(B^{\top} \otimes A)\ \mathrm{vec}(X) = \mathrm{vec}(AXB) = \mathrm{vec}(C)$$

* $\mathrm{vec}(X)$ denotes the vectorization of the matrix X formed by stacking the columns of X into a single column vector

* $\boxed{AX + YB = C} \Leftrightarrow \boxed{(I \otimes A)\,\mathrm{vec}(X) + (B^{\top} \otimes I)\,\mathrm{vec}(Y) = \mathrm{vec}(C)}$

- Important property if A, B are square matrices of sizes m and n:

$$A \otimes B = (I_n \otimes A) + (B \otimes I_m)$$

Back to the OBDC Design Problem

- Solve the following system of matrix equations

$$L_iB_{r_i} = 0 \;\Rightarrow\; \boxed{L_i = \left(\text{Null}(B_{r_i}^\top)\right)^\top}$$

$$K_iL_i + W_iC_i = F_i$$

$$G_iC_i - L_iA + E_iL_i = 0$$

- The second equation can be written as (see $\otimes$ properties):

$$(L_i^\top \otimes I_{m_i})\,\text{vec}(K_i) + (C_i^\top \otimes I_{m_i})\,\text{vec}(W_i) = \text{vec}(F_i) \quad (*)$$

- Also, third equation has only one unknown now, G_iC_i

$$(C_i^\top \otimes I_{o_i})\,\text{vec}(G_i) = \text{vec}(L_iA - E_iL_i) = \text{vec}(V_i) \quad (**)$$

- Combining $(*)$ and $(**)$, we get:

$$\boxed{\underbrace{\begin{bmatrix} L_i^\top \otimes I_{m_i} & C_i^\top \otimes I_{m_i} & 0 \\ 0 & 0 & C_i^\top \otimes I_{o_i} \end{bmatrix}}_{\Psi} \begin{bmatrix} \text{vec}(K_i) \\ \text{vec}(W_i) \\ \text{vec}(G_i) \end{bmatrix} = \begin{bmatrix} \text{vec}(F_i) \\ \text{vec}(V_i) \end{bmatrix}}$$

- Hence, we can find K_i, W_i, G_i, as LHS and RHS are both given

Involving the Network

- Given: a communication network exists between local controllers and plants
- Hence, instead of OBDC, we have an OBDC-NCS, or a DNCS
- First, can we map the overall system dynamics to a typical NCS dynamics?
- If yes, can we analyze the stability of NCS (that includes the OBDC architecture)?
- What is a bound the maximum allowable time-delay due to the network?
- First, we start by constructing a mapping between DNCS dynamics and NCS ones

Mapping the DNCS to NCS Setup

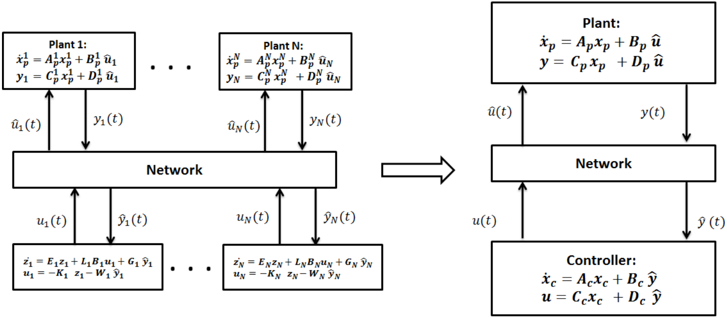
- Plant Dynamics:

$$\begin{cases} \dot{x}_p = A_p x_p + B_p \hat{u} \\ y = C_p x_p + D_p \hat{u}, \end{cases} \quad (1)$$

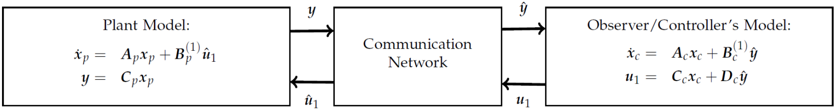
- Controller Dynamics:

$$\begin{cases} \dot{x}_c = A_c x_c + B_c \hat{y} \\ u = C_c x_c + D_c \hat{y}, \end{cases} \quad (2)$$

- Given the OBDC parameters (E, L, K, W, G), find (A_c, B_c, C_c, D_c)



DNCS — Problem Formulation



- The communication network effect can be modeled as
  - Pure-time delay:

$$\boxed{\hat{y} = y(t-\tau), \;\; \hat{u}_1 = u_1(t-\tau)}$$

  - Signals perturbation:

$$\boxed{e_y = y - \hat{y}, \;\; e_{u_1} = u_1 - \hat{u}_1}$$

- Network perturbation effect in [Elmahdi et al., 2015]
- Under unknown inputs, we addressed the time delay + perturbation problem in [Taha et al., 2015]
- This module, we study the network effect as time-delay for LTI NCSs without unknown inputs — simpler case than the one in [Taha et al., 2015]
- **Research Question:** how can we design an observer-based controller for NCSs such that the closed-loop stability is guaranteed?

Time-Delay Analysis for DNCS

- We now convert the DNCS setup to the general setup of the NCS
- The controller's output ($u(t)$) and input ($\hat{y}(t)$) are defined as:

$$\begin{aligned} u(t) &= C_c x_c(t) + D_c C_p x_p(t-\tau) \\ \hat{y}(t) &= y(t-\tau) = C_p x_p(t-\tau) \end{aligned}$$

- Hence, plant & controller state dynamics can be written as:

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p C_c x_c(t) + B_p D_c C_p x_p(t-\tau) \\ \dot{x}_c(t) &= A_c x_c(t) + B_c C_p x_p(t-\tau) \end{aligned}$$

- We use the following Taylor series expansion for $x(t-\tau)$:

$$x(t-\tau) = \sum_{n=0}^{\infty} (-1)^n \frac{\tau^n}{n!} x^{(n)}(t),$$

  where $x(t) = \begin{bmatrix} x_p(t)^\top & x_c(t)^\top \end{bmatrix}^\top$

- *Study closed-loop system stability?* **Derive augmented dynamics of $x(t)$**
- Recall that given the OBDC parameters (E, L, K, W, G), we can find (A_c, B_c, C_c, D_c)

Time-Delay System Dynamics Construction

- Neglecting the higher order terms, we get an approximated expression of $\dot{x}(t)$ in terms of only $x(t)$ and τ as follows:

$$x(t-\tau) = x(t) - \tau\dot{x}(t) + \frac{\tau^2}{2}\ddot{x}(t). \tag{3}$$

- Combining $\dot{x}_p(t)$ and $\dot{x}_c(t)$ to find $\dot{x}(t)$,

$$\begin{bmatrix}\dot{x}_p(t)\\ \dot{x}_c(t)\end{bmatrix} = \begin{bmatrix}A_p & B_pC_c\\ 0 & A_c\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix} + \begin{bmatrix}B_pD_cC_p & 0\\ B_cC_p & 0\end{bmatrix}\begin{bmatrix}x_p(t-\tau)\\ x_c(t-\tau)\end{bmatrix}.$$

- Let $\Gamma_0 = \begin{bmatrix}A_p & B_pC_c\\ 0 & A_c\end{bmatrix}$ and $\Gamma_1 = \begin{bmatrix}B_pD_cC_p & 0\\ B_cC_p & 0\end{bmatrix}$
- We can write $\dot{x}(t)$ as:

$$\dot{x}(t) = \Gamma_0 x(t) + \Gamma_1 x(t-\tau) \tag{4}$$

- Taking the second derivative of $x_p(t)$ and $x_c(t)$:

$$\ddot{x}(t) = \begin{bmatrix}\ddot{x}_p(t)\\ \ddot{x}_c(t)\end{bmatrix} = \begin{bmatrix}A_p\dot{x}_p(t) + B_pC_p\dot{x}_c(t) + B_pD_cC_p\dot{x}_p(t-\tau)\\ A_c\dot{x}_c(t) + B_cC_p\dot{x}_p(t-\tau)\end{bmatrix}$$

Closed-Loop Augmented State Dynamics

- $x_p(t-\tau)$ is piecewise-constant because it changes value at transmission times only, hence:
$$\dot{x}_p(t-\tau) = \dot{x}_c(t-\tau) = 0$$
- Substituting the above approximation in $\ddot{x}(t)$, we get,
$$\ddot{x}(t) = \Gamma_0 \dot{x}(t) \qquad (5)$$
- After a series of algebraic manipulations, we get the closed-loop dynamics:
$$\begin{aligned} \dot{x}(t) &= (I + \tau\Gamma_1 - \frac{\tau^2}{2}\Gamma_1\Gamma_0)^{-1}(\Gamma_0 + \Gamma_1)x(t) \\ \dot{x}(t) &= \Omega(\tau, \tau^2)x(t) \end{aligned}$$
where
$$\Omega(\tau,\tau^2) = \begin{bmatrix} I + \tau B_p D_c C_p - \frac{\tau^2}{2} B_p D_c C_p A_p & -\frac{\tau^2}{2} B_p D_c C_p B_p B_c \\ \tau B_c C_p - \frac{\tau^2}{2} B_c C_p A_p & I - \frac{\tau^2}{2} B_c C_p B_p B_c \end{bmatrix}^{-1} \cdot \begin{bmatrix} A_p + B_p D_c C_p & B_p B_c \\ B_c C_p & A_c \end{bmatrix}$$
- **Sanity check:** set $\tau = 0$ (i.e., nullify the network effect), do we get the dynamics of the non-networked OBDC? **Yes, we do!**

DNCS Stability Analysis

- We now have closed-loop dynamics of the system that can be analyzed using traditional stability analysis techniques.
- The key challenge is the quadratic presence of τ in the dynamics of the system $\Rightarrow$ *couple research questions*
- **Research Question 1:** What is the upper *bound* on the time-delay τ that would drive the system unstable?
- *The notion of instability here implies that the state-estimation fails to track the actual state.*
- **Research Question 2:** What is the maximum allowable disturbance or unknown input bound that guarantees an acceptable state-estimation?

Main Result — Time-Delay Bound

- By the design of the non-networked OBDC, the non-networked system

$$\dot{x}(t) = \Gamma x(t) = (\Gamma_0 + \Gamma_1)x(t)$$

is asymptotically stable ($\texttt{eig}(\Gamma) < 0$)

- For a Hurwitz Γ, we have $P = P^\top \succ 0$, is the solution to the Lyapunov matrix equation

$$\Gamma^\top P + P\Gamma = -2Q,$$

for a given $Q = Q^\top \succ 0$

Theorem (Stability of Time-Delay Based NCSs)

If the network induced delay satisfies the following inequality,

$$\left(\|P\Gamma_1\Gamma_0\Gamma\| + 2\|P\Gamma_1^2\Gamma\|\right)\tau^2 + \left(-2\|P\Gamma_1\Gamma\|\right)\tau + \left(-2\lambda_{min}(Q)\right) < 0$$

then then the observer-based networked control system is asymptotically stable.

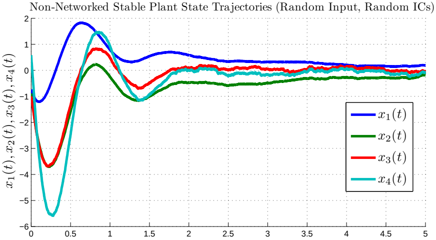
Numerical Results for the Non-Networked System

- Consider a 4^{th} order unstable plant with the following SS representation:

$$\begin{cases} \dot{x}_p(t) = A_p x_p(t) + B_p u(t) \\ y_p(t) = C_p x_p(t), \end{cases} \tag{6}$$

$$A_p = \begin{bmatrix} 1 & 2 & 3 & -4 \\ 5 & 6 & 7 & -8 \\ 9 & 10 & 11 & -12 \\ 13 & 14 & 15 & -16 \end{bmatrix}, \; B_p = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & -1 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 1 & 2 & 5 \end{bmatrix}, \; C_p = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 2 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- First, we design the non-networked observer-based control
- States trajectories for $\tau = 0$ and random initial conditions
- Stabilized state trajectories through the OBDC



Time-delay Bound Testing Algorithm

We follow this algorithm to test the usefulness of the derived bound:

Algorithm 1 Time-Delay DNCS Design and Stability Analysis

1: Solve for the observer-based control parameters (K, L, G, W)

$$L_i B_{r_i} = 0$$
$$K_i L_i + W_i C_i = F_i$$
$$G_i C_i - L_i A + E_i L_i = 0,$$

2: Given $A_p, A_c, B_p, B_c, C_p, C_c$ and D_c, compute $\Gamma, \Gamma_0, \Gamma_1$

3: Find a matrix $P = P^\top \succ \boldsymbol{O}$, a solution to the Lyapunov matrix equation

$$\Gamma^\top P + P\Gamma = -2Q$$

4: Analyze the stability of the networked system:

$$\dot{x}(t) = \Omega\left(\tau, \tau^2\right) x(t) = (I + \tau\Gamma_1 - \frac{\tau^2}{2}\Gamma_1\Gamma_0)^{-1}(\Gamma_0 + \Gamma_1)x(t)$$

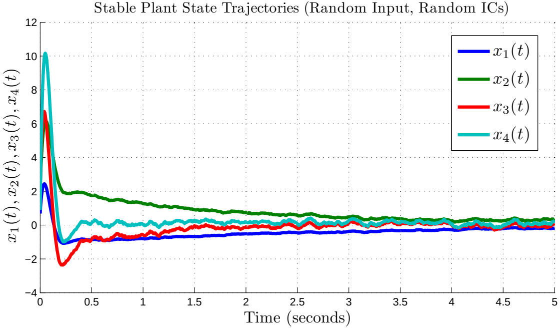
by varying the time-delay (τ)

5: Establish an experimental bound on τ that guarantees the stability of the DNCS

6: Compare the theoretical bound on τ given by the quadratic polynomial in Theorem 1 and the experimental one computed in Step 5

Numerical Results

- After finding the parameters for the non-networked system, we apply Algorithm 1.
- Experimental bound: $0 < \tau < \tau_{\text{exper}}^{\text{max}} = 0.231$ sec
- Evaluating the coefficients for the second degree bound polynomial for τ, we get the theoretical bound: $0 < \tau < \tau_{\text{theor}}^{\text{max}} = 0.202$ sec
- The derived upper bound for the time-delay that guarantees the stability of the NCS is not too conservative

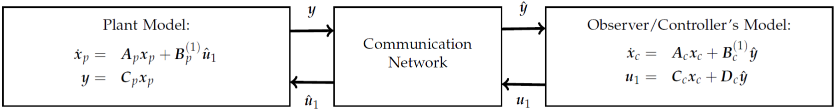


Significance of the Derived Bound on τ

So why is it important to compute the bound on τ?

- The determination of an upper bound on τ is *significantly* important in the design of a NCS so that a suitable sampling period is chosen
- Traditionally, the sampling period h should satisfy: $0 < \tau < \tau_{\max} < h$
- When the time-delay is greater than the sampling period, the global stability of the overall NCS can not be guaranteed
- Can be applied to different kind of applications where communication network is replaced with physical networks (supply-chain networks, air traffic systems, transportation networks)
- **Derived bounds in the literature are very conservative!**

N_{et}UIO — Problem Formulation



- The communication network effect can be modeled as
  - Pure-time delay:

$$\boxed{\hat{y} = y(t-\tau), \;\; \hat{u}_1 = u_1(t-\tau)}$$

  - Signals perturbation:

$$\boxed{e_y = y - \hat{y}, \;\; e_{u_1} = u_1 - \hat{u}_1}$$

- We addressed the time-delay effect in our previous work, where we derived a bound on τ that guarantees the system's asymptotic stability[1]
- In this paper, we study the network effect as perturbation, e is the networked induced error
- **Research Question:** how can we design a N_{et}UIO for NCSs such that the closed-loop stability is guaranteed?

[1] A. Taha *et al.*, "Networked Unknown Input Observer Analysis and Design for Time-Delay Systems", *IEEE SMC 2014*, San Diego, CA, October 2014.

Perturbation Bounds Derivation

- The dynamics of the perturbed system can be written as follows:

$$\dot{x}(t) = \boldsymbol{\Lambda_1} x(t) + \boldsymbol{\Lambda_2} e(t) + \boldsymbol{\Omega_1} u_2(t), \tag{7}$$

$$\boldsymbol{\Lambda}_2 = \begin{bmatrix} -\boldsymbol{B}_p^{(1)}\boldsymbol{D}_c & -\boldsymbol{B}_p^{(1)} \\ -\left(\boldsymbol{B}_c^{(1)} + \boldsymbol{B}_c^{(2)}\boldsymbol{D}_c\right) & -\boldsymbol{B}_c^{(2)} \end{bmatrix}, \; \boldsymbol{\Omega_1} = \begin{bmatrix} \boldsymbol{B}_p^{(2)} \\ \boldsymbol{O} \end{bmatrix},$$

- By the UIO design for the non-networked system, the matrix

$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} \boldsymbol{\Lambda}_{1,1} & \boldsymbol{B}_p^{(1)}\boldsymbol{C}_c \\ \boldsymbol{\Lambda}_{2,1} & \left(\boldsymbol{A}_c + \boldsymbol{B}_c^{(2)}\boldsymbol{C}_c\right) \end{bmatrix}$$

  is asymptotically stable.
- $\boldsymbol{\Lambda_2} e(t)$: perturbation due to the network
- $\boldsymbol{\Omega_1} u_2(t)$: perturbation due to the unknown input

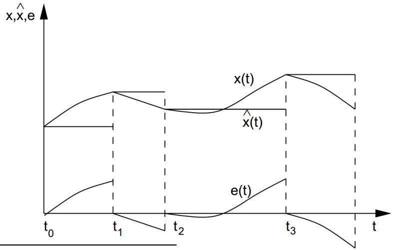
Perturbation Analysis (Cont'd)

- Behavior of the network-induced error, $e(t)$, is mainly determined by the architecture of the NCS and the scheduling protocol [4][1]
- Hence, we need to derive a state-bounded error quantity $e_x(t)$
- Consider the time interval between transmissions $t \in [t_i, t_{i+1}]$:

$$\hat{y}(t) = y(t_i), \;\; \hat{u}_1(t) = u_1(t_i), \;\; e_x(t) = x(t) - x(t_i) = x(t) - \hat{x}(t)$$

$$\Rightarrow \Lambda_2 e(t) = \Lambda_2 \begin{bmatrix} \boldsymbol{C}_p & \boldsymbol{O} \\ \boldsymbol{D}_c\boldsymbol{C}_p & \boldsymbol{C}_c \end{bmatrix} e_x(t) = \boldsymbol{E} e_x(t)$$

$$\Rightarrow \boxed{\dot{x}(t) = \Lambda_1 x(t) + \boldsymbol{E} e_x(t) + \Omega_1 u_2(t)} \quad (8)$$



[1] [4] Walsh *et al.*, "Stability Analysis of Networked Control Systems", *IEEE Transactions on Control Systems Technology*, 10(3), 2002.

Main Result

For the perturbed N$_{\text{et}}$UIO in (8) and for a Hurwitz Λ_1, we have $\boldsymbol{P} = \boldsymbol{P}^\top \succ \boldsymbol{O}$, is the solution to the Lyapunov matrix equation

$$\Lambda_1^\top \boldsymbol{P} + \boldsymbol{P}\Lambda_1 = -2\boldsymbol{Q},$$

for a given $\boldsymbol{Q} = \boldsymbol{Q}^\top \succ \boldsymbol{O}$.

Theorem (N$_{\text{et}}$UIO Perturbation Bound)

If $\|u_2(t)\| < \mu_x \|x(t)\|$ and if the norm of the network induced perturbation $\|e_x\|$ satisfies

$$\boxed{\|e_x\| < \zeta \|x\|}$$

where

$$\boxed{\zeta \leq \frac{\lambda_{min}(\boldsymbol{Q}) - \mu_x \|\Omega_1\| \lambda_{max}(\boldsymbol{P})}{\lambda_{max}(\boldsymbol{P})\|\boldsymbol{E}\|}}$$

then the origin is a globally exponentially stable equilibrium point of the perturbed N$_{\text{et}}$UIO.

Why is this bound significant?

- The problem of designing UIOs for NCSs received minor attention
- The bound can be applied for any UIO architecture by simply changing the state-space matrices corresponding to the LTI UIO
- N_{et}UIO can be designed such that this bound is satisfied
- Similar bounds have been developed for NCS with no unknown inputs
- Walsh *et al.*[2] show that the NCS with no unknown inputs is asymptotically stable if $\|e_x(t)\| \leq \gamma \|x(t)\|$ where

$$\gamma = \frac{||\Lambda_1|| \; ||\Lambda_1 + \boldsymbol{E}||^{-1}(e^{||\Lambda_1 + \boldsymbol{E}||\tau_m} - 1)e^{||\Lambda_1 + \boldsymbol{E}||\tau_m}}{1 - ||\boldsymbol{E}|| \; ||\Lambda_1 + \boldsymbol{E}||^{-1}(e^{||\Lambda_1 + \boldsymbol{E}||\tau_m} - 1)},$$

  where

$$\tau_m < \frac{\lambda_{min}(\boldsymbol{Q})}{16\lambda_2 \sqrt{\frac{\lambda_2}{\lambda_1}} \|\Lambda\|^2 \left(1 + \sqrt{\frac{\lambda_2}{\lambda_1}}\right) \sum_{i=1}^{p} i},$$

  is the maximum allowable transfer interval
- ***How do γ and ζ compare? Which one is more conservative?***

[2] G. C. Walsh, H. Ye, L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, May 2002.

Hui and Zak's Non-N_{et} UIO

- To test the main result, we apply a non-networked UIO
- UIO Dynamics[1]:

$$\dot{x}_c = A_c x_c + B_c^{(1)} y + B_c^{(2)} u_1,$$
$$\hat{x}_p = x_c + My,$$

where

$$A_c = (I - MC_p)(A_p - LC_p), B_c^{(2)} = (I - MC_p)B_P^{(1)}$$

$$B_c^{(1)} = (I - MC_p)(A_p M + L - LC_p M).$$

- Given the SS plant parameters $A_p, B_p^{(1)}, B_p^{(2)}, C_p$, the UIO design parameter M is derived such that the observation error converges to zero
- Precisely, $M \in \mathbb{R}^{n \times p}$ is chosen such that $(I - MC_p)B_p^{(2)} = 0$
- L is an added gain to improve the convergence of the estimated state $(\hat{x}_p)$

[1] Hui, S., Żak, S. H. (2005). "Observer design for systems with unknown inputs." *International Journal of Applied Mathematics and Computer Science*, vol. 15(4), pp. 431–446, 2005

Numerical Results for the Non-N_{et}UIO

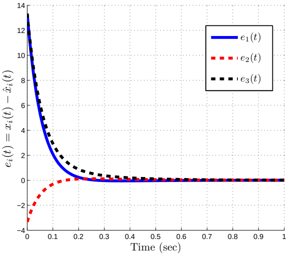
- We test the derived bound on the networked-induced perturbation for the N_{et}UIO
- The system is modeled by

$$\boldsymbol{A}_p = \begin{bmatrix} -5 & 3 & 0 \\ 4 & -10 & 4 \\ 0 & 0 & -4 \end{bmatrix}, \boldsymbol{B}_p^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \boldsymbol{B}_p^{(2)} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, \boldsymbol{C}_p = \begin{bmatrix} 2 & 4 & -1 \end{bmatrix}.$$

- Unknown input includes disturbances & actuator faults:

$$u_2(t) = \sin(t) + \cos(t) + \max\{0, t - 0.5\}$$

- Non-networked state-estimation error converges rapidly to zero



Numerical Results for the N_{et}UIO

- Using the parameters for the plant and UIO, we compute the perturbed dynamics of the N_{et}UIO:

$$\dot{x}(t) = \Lambda_1 x(t) + E e_x(t) + \Omega_1 u_2(t),$$

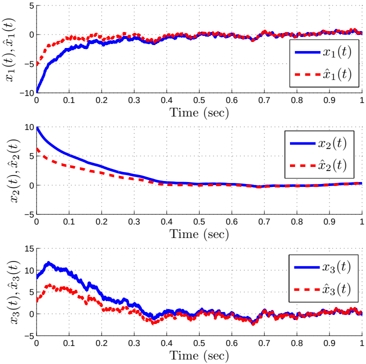
- Find $P = P^\top \succ 0$, a solution to lyap(Λ_1, Q), $Q = Q^\top \succ 0$
- The theoretical bound that would render the system unstable is

$$\boxed{\|e_x\| < 1.2097 \cdot \|x\|}$$

- The bound derived by Walsh *et al.* is

$$\boxed{\|e_x\| < 1.42 \cdot 10^{-4} \cdot \|x\|}$$

- For different LTI systems, our derived bound provides much less conservative bound than the one in the literature



Conclusions and Future Work

Recap of the problem addressed in this talk

- In many control systems, state estimators' inputs are usually sent through a communication network
- Objective: analyze the effects of plant unknown inputs and network disturbances

How could this bound be used? Challenges and Possibilities?

- Derivation of network perturbation bounds would assist in the design of controllers and observers
- Example: state-feedback gain and UIO gain matrices can be designed to reduce the disturbance effects of unknown inputs and network perturbation
- **Fault detection and isolation** techniques can be jointly analyzed under a $\mathrm{N_{et}UIO}$ scheme
- **Unknown input estimation?**
- **Optimal decentralized control problem for the $\mathrm{N_{et}UIO}$**

Future Work

- The need to look at more applications for Observer-Based Control in networked dynamical systems
- Derivation of network delay and perturbation bounds would assist in the design of controllers and observers
- Example: state-feedback & OBDC gain matrices can be *designed* to reduce the disturbance effects of unknown inputs & network-induced perturbations
- **Fault detection and isolation** techniques can be jointly analyzed under a DNCS scheme
- **Optimal decentralized networked control problem for systems with unknown inputs?**

Questions And Suggestions?

Thank You!

Please visit

engineering.utsa.edu/ataha

IFF you want to know more ☺

References I

Elmahdi, A., Taha, A. F., Sun, D., & Panchal, J. H. (2015). Decentralized control framework and stability analysis for networked control systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, *137*(5), 051006–051006–11.

Ha, Q. P., & Trinh, H. (2004). Observer-based control of multi-agent systems under decentralized information structure. *International journal of systems science*, *35*(12), 719–728.

Swigart, J., & Lall, S. (2010). Decentralized control. In A. Bemporad, M. Heemels, & M. Johansson (Eds.) *Networked Control Systems*, vol. 406 of *Lecture Notes in Control and Information Sciences*, (pp. 179–201). Springer London.
URL http://dx.doi.org/10.1007/978-0-85729-033-5_6

Taha, A. F., Elmahdi, A., Panchal, J. H., & Sun, D. (2015). Unknown input observer design and analysis for networked control systems. *International Journal of Control*, *88*(5), 920–934.
URL http://dx.doi.org/10.1080/00207179.2014.985718