Pure Time Delay Analysis for Decentralized Networked Control Systems

Ahmad Taha (Purdue, ECE)

Joint work with
J. Panchal (Purdue, MechE), A. Elmahdi & D. Sun (Purdue, AAE),

Design Engineering Lab @ Purdue (DELP)
School of Mechanical Engineering
Purdue University, West Lafayette, IN

November 12, 2014
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?*
Decentralized Networked Control Systems — Example

Decentralized Control + Networked Control = Decentralized Networked Control System (DNCS)
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.

Ha and Trinh [1] 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.

---

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 DNCS
5. Stability Analysis – Bounds on the time-delay
6. Numerical Results
OBDC Plant Dynamics & Objective

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

\[
\begin{aligned}
\dot{x} &= Ax + \sum_{i=1}^{N} B_i u_i \\
y_i &= C_i x, \quad i = 1, 2, \ldots, N
\end{aligned}
\]

- \(N\) 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

- Ha and Trinh proposed the following controller:

\[ u_i = -F_i x = -(K_i L_i + W_i C_i)x \]

\[ = -K_i z_i - W_i y_i, \quad z_i \rightarrow L_i x \]

- Let \( z_i \) have the following dynamics:

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

- The observation error:

\[ e_{o_i} = z_i - L_i x, \quad i = 1, 2, \ldots, N \]

- Observation error dynamics:

\[ \dot{e}_{o_i} = \dot{z}_i - L_i \dot{x} = E_i e_{o_i} + (G_i C_i - L_i A + E_i L_i)x - L_i B r_i u_r \]

- We want to find design parameters \( K_i, L_i, G_i, W_i \) such that:

\[ L_i B r_i = 0 \quad \Rightarrow \quad L_i = \left( \text{Null}(B r_i) \right)^T \]

\[ K_i L_i + W_i C_i = F_i \quad \Rightarrow \quad \begin{bmatrix} L_i^T \otimes I_{m_i} & C_i^T \otimes I_{m_i} & 0 \\ 0 & 0 & C_i^T \otimes I_{o_i} \end{bmatrix} \begin{bmatrix} \text{vec}(K_i) \\ \text{vec}(W_i) \\ \text{vec}(G_i) \end{bmatrix} = \begin{bmatrix} \text{vec}(F_i) \\ \text{vec}(V_i) \end{bmatrix} \]
Mapping the DNCS to NCS Setup

- **Plant Dynamics:**
  \[
  \begin{align*}
  \dot{x}_p &= A_p x_p + B_p \hat{u} \\
  y &= C_p x_p + D_p \hat{u},
  \end{align*}
  \tag{1}
  \]

- **Controller Dynamics:**
  \[
  \begin{align*}
  \dot{x}_c &= A_c x_c + B_c \hat{y} \\
  u &= C_c x_c + D_c \hat{y},
  \end{align*}
  \tag{2}
  \]

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

---

© Ahmad Taha (Purdue, ECE)

Time-Delay Analysis for DNCS — ASME DSCC 2014 — Session: Time-Delay Systems & Stability
DNCS — Problem Formulation

The communication network effect can be modeled as
- Pure-time delay:
  $$\hat{y} = y(t - \tau), \quad \hat{u}_1 = u_1(t - \tau)$$
- Signals perturbation:
  $$e_y = y - \hat{y}, \quad e_{u_1} = u_1 - \hat{u}_1$$

We addressed the network perturbation effect in our previous work [2]
In this paper, we study the network effect as time-delay

**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:
  \[
  u(t) = C_c x_c(t) + D_c C_p x_p(t - \tau)
  \]
  \[
  \hat{y}(t) = y(t - \tau) = C_p x_p(t - \tau)
  \]
- Hence, plant & controller state dynamics can be written as:
  \[
  \dot{x}_p(t) = A_p x_p(t) + B_p C_c x_c(t) + B_p D_c C_p x_p(t - \tau)
  \]
  \[
  \dot{x}_c(t) = A_c x_c(t) + B_c C_p x_p(t - \tau)
  \]
- 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 \( x(t) = \begin{bmatrix} x_p(t) \ x_c(t) \end{bmatrix}^T \)
- 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 \((A_c, B_c, C_c, D_c)\)
Neglecting the higher order terms, we get an approximated expression of $\dot{x}(t)$ in terms of only $x(t)$ and $\tau$ as follows:

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

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

$$\begin{bmatrix} \dot{x}_p(t) \\ \dot{x}_c(t) \end{bmatrix} = \begin{bmatrix} A_p & B_p C_c \\ O & A_c \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} + \begin{bmatrix} B_p D_c C_p & O \\ B_c C_p & O \end{bmatrix} \begin{bmatrix} x_p(t - \tau) \\ x_c(t - \tau) \end{bmatrix}. \quad (4)$$

Let $\Gamma_0 = \begin{bmatrix} A_p & B_p C_c \\ O & A_c \end{bmatrix}$ and $\Gamma_1 = \begin{bmatrix} B_p D_c C_p & O \\ B_c C_p & O \end{bmatrix}$

We can write $\dot{x}(t)$ as:

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

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

$$\ddot{x}(t) = \begin{bmatrix} \ddot{x}_p(t) \\ \ddot{x}_c(t) \end{bmatrix} = \begin{bmatrix} A_p \dot{x}_p(t) + B_p C_p \dot{x}_c(t) + B_p D_c C_p \dot{x}_p(t - \tau) \\ A_c \dot{x}_c(t) + B_c C_p \dot{x}_p(t - \tau) \end{bmatrix}$$
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,
  $$\ddot{x}(t) = \Gamma_0 \dot{x}(t) \quad (5)$$

- After a series of algebraic manipulations, we get the closed-loop dynamics:
  $$\dot{x}(t) = \Omega(\tau, \tau^2) x(t)$$

where

$$\Omega(\tau, \tau^2) = \left[ I + \tau B_p D_c C_p - \frac{\tau^2}{2} B_p D_c C_p A_p \right]^{-1} \left[ A_p + B_p D_c C_p B_p B_c \right]$$

**Sanity check:** set $\tau = 0$ (i.e., nullify the network effect), do we get the dynamics of the non-networked OBDC? **Yes, we do!**
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 \( \tau \) in the dynamics of the system \( \Rightarrow \) couple research questions

- **Research Question 1:** What is the upper *bound* on the time-delay \( \tau \) 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 (\( \text{eig}(\Gamma) < 0 \))

- For a Hurwitz \( \Gamma \), we have \( P = P^T > 0 \), is the solution to the Lyapunov matrix equation
  \[ \Gamma^T P + P \Gamma = -2Q, \]
  for a given \( Q = Q^T > 0 \)

Theorem (Stability of Time-Delay Based NCSs)

*If the network induced delay satisfies the following inequality,

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

then the observer-based networked control system is asymptotically stable.*
Numerical Results for the Non-Networked System

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

\[
\begin{align*}
\dot{x}_p(t) &= A_p x_p(t) + B_p u(t) \\
y_p(t) &= C_p x_p(t),
\end{align*}
\]

\(\mathbf{A}_p = \begin{bmatrix} 1 & 2 & 3 & -4 \\ 5 & 6 & 7 & -8 \\ 9 & 10 & 11 & -12 \\ 13 & 14 & 15 & -16 \end{bmatrix}, \quad \mathbf{B}_p = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & -1 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 1 & 2 & 5 \end{bmatrix}, \quad \mathbf{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:

![Non-Networked Stable Plant State Trajectories](image)

- 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 \((K, L, G, W)\)

\[
L_iB_{r_i} = 0
\]

\[
K_iL_i + W_iC_i = F_i
\]

\[
G_iC_i - L_iA + E_iL_i = O,
\]

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 \(P = P^T > 0\), a solution to the Lyapunov matrix equation

\[
\Gamma^T P + P \Gamma = -2Q
\]

4: Analyze the stability of the networked system:

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

by varying the time-delay \(\tau\)

5: Establish an experimental bound on \(\tau\) that guarantees the stability of the DNCS

6: Compare the theoretical bound on \(\tau\) 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: $0 < \tau < \tau_{\text{max \ exper}} = 0.231$ sec
- Evaluating the coefficients for the second degree bound polynomial for $\tau$, we get the theoretical bound: $0 < \tau < \tau_{\text{max \ theor}} = 0.202$ sec
- The derived upper bound for the time-delay that guarantees the stability of the NCS is not too conservative

States trajectories for $\tau = 0.2$ sec with random initial conditions
So why is it important to compute the bound on $\tau$?

- The determination of an upper bound on $\tau$ is significantly important in the design of a NCS so that a suitable sampling period is chosen.
- Traditionally, the sampling period $h$ should satisfy: $0 < \tau < \tau_{\text{max}} < h$.
- When the time-delay is greater than the sampling period, the global stability of the overall NCS cannot 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!
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?

Any questions?

Thank You!
Please visit
web.ics.purdue.edu/~tahaa
IFF you want to know more 😊