Module 08
Introduction to Networked Control Systems: The *Not-So-Fancy* Term for CPSs

Ahmad F. Taha

EE 5243: Introduction to Cyber-Physical Systems

Email: ahmad.taha@utsa.edu

Webpage: http://engineering.utsa.edu/~taha/index.html

October 19, 2015

Module 8 Outline

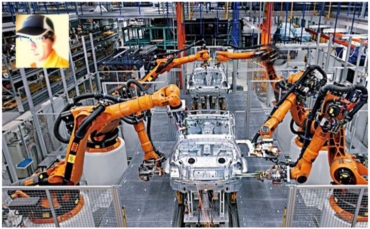
In this module, we introduce *Networked Control Systems*. Outline of this module:

1. NCS: introduction, role, and relevance
2. NCS applications, design challenges, and requirements
3. NCS Model Description
4. Scheduling protocols for NCSs
5. Stability analysis of NCSs
6. Examples

Networked Dynamical Systems

- In many complex systems, communication networks are employed to exchange information between spatially distributed system components
- Dynamical systems are becoming inherently more networked
- **Pros**: efficient monitoring, distributed computation, low cost, ease of use

Industrial Automation



Power Plants

- **Cons**: vulnerability, networked-induced delays and perturbations, attacks
- **Examples**: power-grids, transportation networks, water networks, vehicles, aircrafts...
- **NCSs control and monitor most Cyber-Physical Systems (CPS)**

What are Networked Control Systems?

- **NCS**: control system that closes its control loops through a network
- Exchange of information between plant and controller via a network
- The u's and the y's are sent to controllers through a comm. network

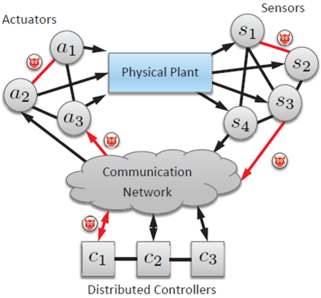
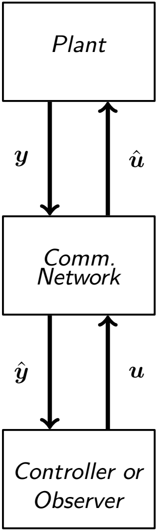


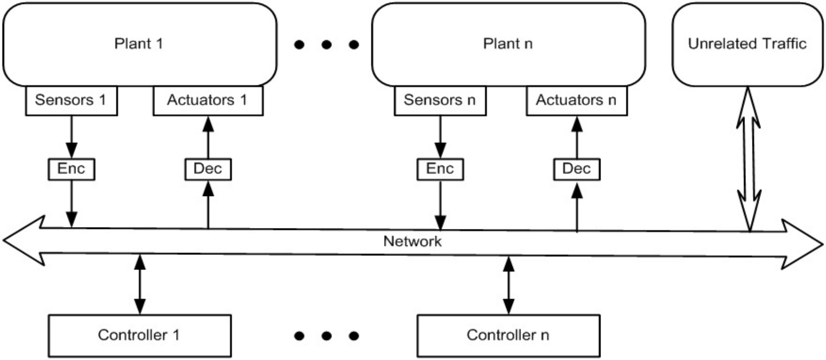
Figure: A. Texiera, "Toward Secure and Reliable Networked Control Systems". Licentiate thesis, KTH, 2011.



More on NCSs

- Recent literature on LTI NCSs
- NCSs: the so-called 3RD GEN OF CONTROL SYSTEMS
- *What are the first two generations of control systems?*
- **Advantages**: modularity, flexibility and simplicity of implementation and reduced system wiring
- All modern cars have NCSs
- Network effect can be modeled as perturbation or time-delay to the exchanged signals
- One node can only use the shared medium at any given time

Typical NCS Setup



NCS Objectives

High-Level NCS Design Objectives

1. Develop a system that efficiently uses the finite bus capacity while maintaining reasonable closed-loop control system performance.
2. The addition of network to the system will improve system reliability, reduce weight, space, power and wiring requirements. Nevertheless, there are constraints that somewhat limits its applications. Hence, satisfying these physical constraints is another NCS objective.

NCS Applications

- Manufacturing plants, automobiles, air conditioning/cooling systems, elevators
- Building automation, medical equipment, remote surgery, mobile sensor networks
- Robotics, UAVs, drones
- Transportation systems, power systems, communication networks
- Economic systems, spacecraft dynamics
- ..And many more...

NCS Limitations

- Modern control systems are increasingly adopting the networked control framework
- Another NCS objective is design systems with improved QoS
- QoS of the network includes the network's fairness, tolerance to packet losses, etc...
- Insertion of a communication network in the feedback control loop makes the analysis and design of an NCS more complex
- Hence, there are disadvantages and challenges that come from adopting the networked control configuration
- Important issues that an NCS designers address are: network's fairness, stability, delays, and error analysis
- Crucial design challenges: network-induced perturbations, attacks, and time-delays

NCS and Control of Delays/Perturbations

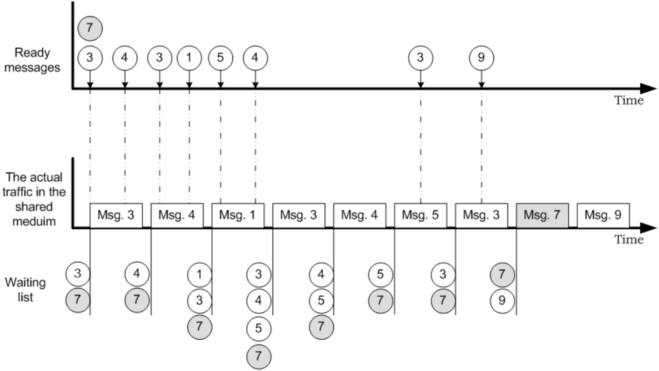
- Control systems have different types of delays (e.g. plant natural delay, control-induced time delay, network-induced time delay)
- Two classes of control mechanisms:
  - *Control **of** Network*
    The network-induced time delay comes exclusively from the network existence in the feedback-loops of the control system. Therefore, taking control actions in the network level can directly improve the system performance
  - *Control **over** Network*
    In the controller design stage, it is very important to choose the control strategy that helps to reduce the network-induced delay

Overview of scheduling protocols

- For many NCSs, one node can only use the shared medium at any given time
- This affects the stability and performance of system
- Requires a scheduling protocol for NCS
- Two kinds of scheduling protocols: static and dynamic
  - Static: predetermined channel access (e.g., Round-Robin, Toke-Ring)
  - Dynamic: channel access determined during network access (e.g., MEF-TOD, CAN)
- Maximum Error First-Try Once Discard (MEF-TOD) protocol [Walsh et al., 2002]
  - Arbitrates between multiple nodes
  - Does not give equal access to all nodes

More about MEF/TOD

- Message with largest error wins the medium access
- Error due to the network-induced delay
- Packet discarded if it fails to win the competition
- One node may hog the network for long period of time
- **Example:**



Scheduling Protocol Design Requirements

Limitations:

- Scheduling protocols work properly in low traffic situations
- Network-induced time delay increases as network load increases
- Low priority messages will be delayed more than once
- **Main objective:** a hybrid static-dynamic scheduling protocol that improves QoS

Design Objectives:

- Protocol that provides more even access for all nodes
- Addresses network fairness issue
- Maintains closed-loop system stability
- Targets different types of systems

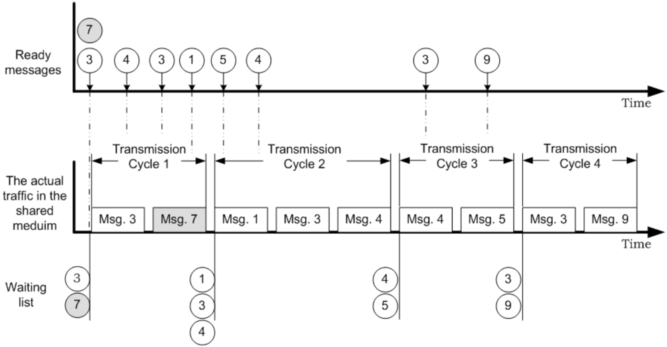
Traffic Division Arbitration (TDA) Protocol

TDA operation [Elmahdi, 2014]:

1. Combination of static and dynamic scheduling protocols
2. Guarantees access to the network even for low priority messages
3. Maintains closed loop system stability
4. Can be used for various kinds of systems

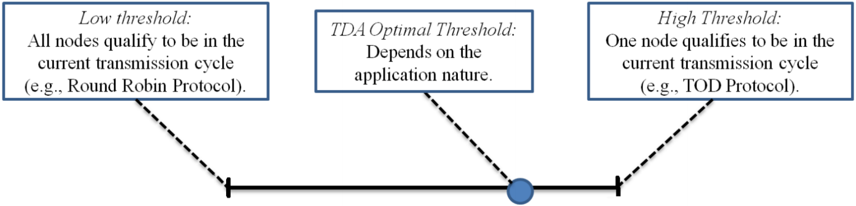
TDA Description

- TDA has two arbitration levels; dynamic and static
- Dynamic portion: network traffic divided into transmission cycles
- Threshold determines which messages passes to second level
- Static portion: messages from the first level transmitted according to a pre-determined error priority
- **Example:**



Error Threshold

- Error threshold depends on the system nature
- Determining this error threshold via a heuristic algorithm, or via a formulation of an optimization framework to find an **optimal threshold**



Low threshold:
All nodes qualify to be in the current transmission cycle (e.g., Round Robin Protocol).

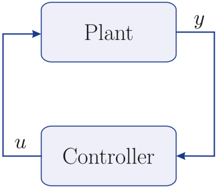
TDA Optimal Threshold:
Depends on the application nature.

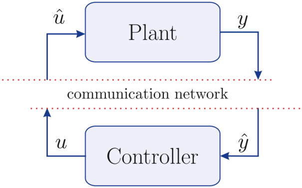
High Threshold:
One node qualifies to be in the current transmission cycle (e.g., TOD Protocol).

Network-Induced Imperfections

Roughly speaking, the network-induced imperfections and constraints can be categorized in five types [Bemporad et al., 2010]:

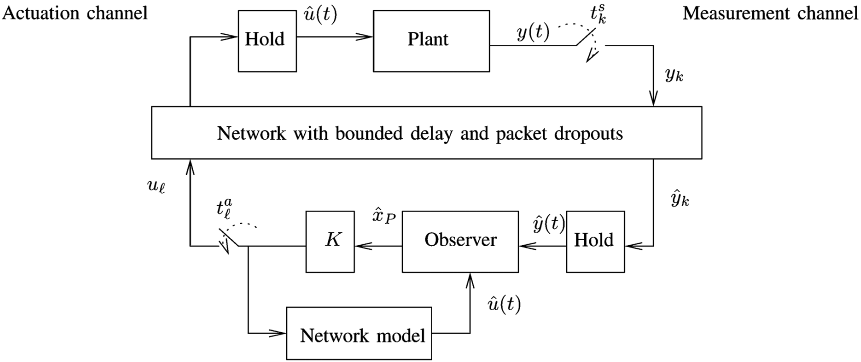
1. Variable sampling/transmission intervals + communication delays
2. Packet dropouts caused by the unreliability of the network
3. Communication constraints caused by the sharing of the network by multiple nodes and the fact that only one node is allowed to transmit its packet per transmission
4. Quantization errors in the signals transmitted over the network due to the finite word length of the packets





Therefore, u **becomes** $\hat{u}$ **and** y **becomes** $\hat{y}$

NCS Signals: Quantization and ZOH [Hespanha et al., 2007]



Zero-Order-Hold (ZOH): model of signal reconstruction done by a conventional digital-to-analog converter (DAC).

Example:

$$\boxed{\hat{y}(t) = y_{\text{ZOH}}(t) = \sum_{k=-\infty}^{\infty} y_k \cdot \text{rect}\left(\frac{t - T/2 - nT}{T}\right)}$$

Notation and NCS Model Description

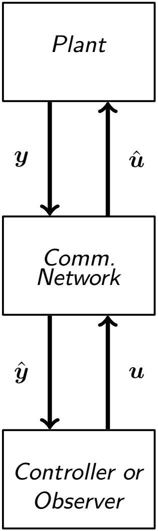
- To analyze the stability/performance of NCS given a schedule protocol, we need to derive dynamics of the closed loop system
- Precisely, we need to derive dynamics of the error due to the network
- The LTI controller model is described as:

$$\sum_c (A_c,\, B_c,\, C_c,\, D_c)$$

- The LTI plant model is described as:

$$\sum_p (A_p,\, B_p,\, C_p,\, 0)$$

- Outputs measured at an actuator can be incorporated directly into the controller and do not require treatment in our model



NCS Model Description (Cont'd)

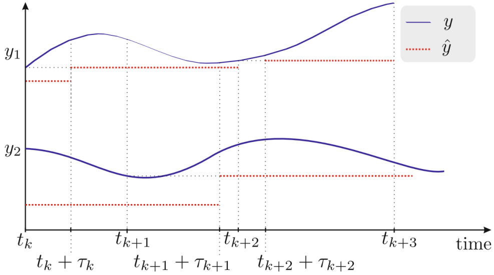
- Overall state:
$$x(t)^\top = [x_p(t)^\top x_c(t)^\top]$$
- Without a network, $e(t) = 0$, dynamics reduced to [Walsh et al., 2002]:
$$\dot{x}(t) = A_{11}x(t), \quad A_{11} = \begin{bmatrix} A_p + B_pD_cC_p & B_pC_c \\ B_cC_p & A_c \end{bmatrix}$$
- With the network, $e(t) \neq 0$, define network-induced error state:
$$e(t)^\top = [\hat{y}(t)^\top \hat{u}(t)^\top] - [y(t)^\top u(t)^\top]$$
- Combined NCS state:
$$z(t)^\top = [x(t)^\top e(t)^\top]$$

$$\boxed{\dot{z}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = Az(t)}$$

$$A_{12} = \begin{bmatrix} B_pD_c & B_p \\ B_c & 0 \end{bmatrix}, A_{21} = -\begin{bmatrix} C_p & 0 \\ 0 & C_c \end{bmatrix} A_{11}, A_{22} = -\begin{bmatrix} C_p & 0 \\ 0 & C_c \end{bmatrix} A_{12}$$

More on the Network-Induced Error

- Behavior of network-induced error, $e(t)$, is mainly determined by NCS architecture + scheduling protocol
- Special case: one-packet transmission: $e(t) = 0$ — there's no competition b/w nodes
- For multiple nodes (MIMO systems), scheduling decides which components of $e(t)$ are set to zero during transmission times
- Note: $\dot{\hat{y}} = \dot{\hat{u}} = 0$ during transmission [Bemporad et al., 2010] (due to ZOH)



Assumptions

To derive an error bound related to a NCS, we consider the following assumptions for the TDA-based system:

1. Closed loop system is asymptotically stable without a network, i.e., eig(A_{11})< 0
2. Controller is continuous-time, i.e., $u(t) = h(x(t))$
3. Sampling delay is negligibly small
4. Communication medium is error-free, observation noise is negligible, i.e., $\hat{y}$ can be computed without noise
5. Each node grants the medium access one time every k transmissions starting at time t_0

Notation and Error Bound

- NCS consists of multiple independent sensors and actuators competing for network access
- Since nodes act asynchronously, we allow access to the network at anytime given a maximum bound
- Bound is defined as: maximum allowable transfer interval (MATI) — τ_m
- Number of nodes: k
- Maximum growth in error: β, i.e., $\|e_i(t+\tau_m) - e_i(t)\| < \beta$

Theorem (Bound on $e(t)$)

Given the above assumptions, the network-induced error bounded by:

$$||e(t)|| < \beta \frac{k}{2}(k+1)$$

NCS Stability and τ_m Bound

- Non-networked system $\dot{x}(t) = A_{11}x(t)$ is asymptotically stable, then, we can find a matrix $P = P^\top$, a solution to this LMI:

$$A_{11}^\top P + PA_{11} \prec 0$$

- **Note:** Different dynamical system representation, would result in a different LMI (and different bound)

Theorem (Same bounds as in Walsh, Ye, and Bushnell, 2002)

If τ_m is upper bounded by the minimum of

$$\frac{1}{4||A||\left(\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}+1\right)k(k+1)},$$

and

$$\frac{1}{8\lambda_{\max}(P)\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}||A||^2\left(\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}+1\right)k(k+1)},$$

then the TDA/NCS is globally exponentially stable.

System Example Description

- Objective: analyze the effect of increasing traffic
- Simulate 3 and 12 subsystems
- Each subsystem is an armature-controlled DC motor:

$$\begin{cases} \dot{x}(t) = A_p x(t) + B_p u(t) \\ y(t) = C_p x(t) \\ x(t) \in \mathbb{R}^3, u(t) \in \mathbb{R}, \text{ and } y(t) \in \mathbb{R} \end{cases}$$

- State-feedback and PID controllers to stabilize the system and eliminate transient response

Error Decision Functions

Two decision functions for the schedulers to prioritize medium access

- Both decision functions use the tracking error $\varepsilon_k(n)$
- First decision function computes the relative error growth rate:

$$d_k(n) = \frac{|\varepsilon_k(n) - \varepsilon_k(n-1)|}{|\varepsilon_k(n-1)| + \delta}$$

- d_k slows the rate of convergence as the error gets small
- Denominator included to normalize tracking errors that come from different types of sub-systems

Second Decision Function

- Second decision function:

$$\tilde{d}_k(n) = 2\varepsilon_k(n) - \varepsilon_k(n-1)$$

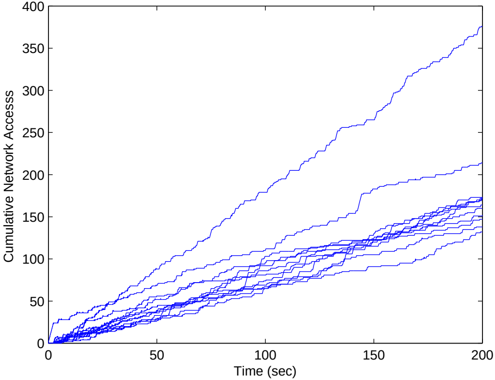
- One-step predictor of the tracking error

$$\begin{aligned} \varepsilon_k(n+1) &\approx \varepsilon_k(n) + \varepsilon_k^{'}(n) \\ &\approx \varepsilon_k(n) + (\varepsilon_k(n) - \varepsilon_k(n-1)) \\ &= 2\varepsilon_k(n) - \varepsilon_k(n-1) \end{aligned}$$

- Approximates future tracking error

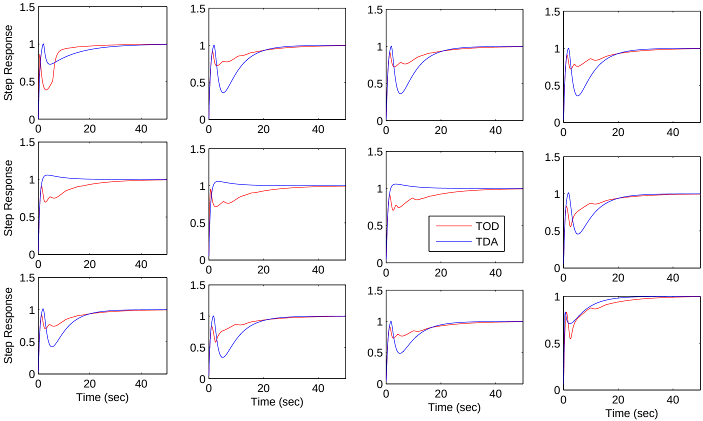
TDA Simulation

- Cumulative number of network access instances
- Access distribution comparable



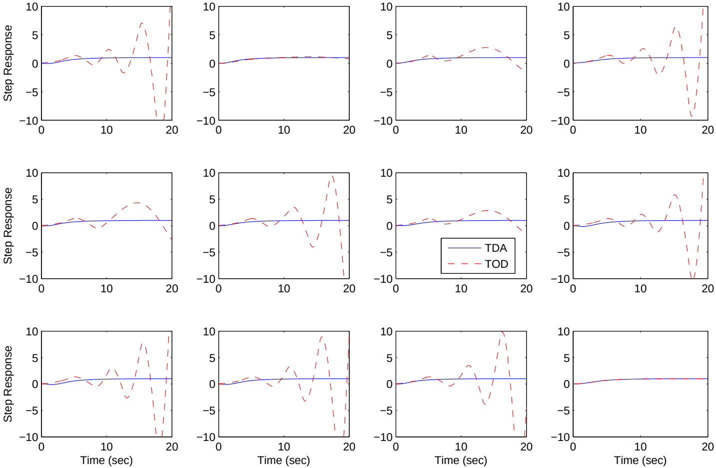
Step Response Comparison Using d_k — TOD vs TDA

- Similar TOD and TDA performance



Step Response Comparison Using $\tilde{d}_k$ — TOD vs TDA

- TDA outperforms TOD



Module 08 Summary

In this module, we:

- Introduced NCSs
- Presented their applications, merits, demerits, limitations, challenges
- Summarized scheduling protocols for NCSs
- Formulated a dynamical model given networked induced error
- Discussed error bounds
- Illustrated a numerical example

Questions And Suggestions?

Thank You!

Please visit

engineering.utsa.edu/~taha

IFF you want to know more ☺

References I

Bemporad, A., Heemels, M., & Johansson, M. (2010). *Networked control systems*, vol. 406. Springer.

Elmahdi, A. M. (2014). *Network-induced time delay control to improve performance in networked dynamical systems*. Ph.D. thesis, Purdue University, West Lafayette.
URL `http://search.proquest.com/docview/1615069742?accountid=13360`

Hespanha, J. P., Naghshtabrizi, P., & Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, *95*(1), 138–162.

Walsh, G. C., Ye, H., & Bushnell, L. G. (2002). Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, *10*(3), 438–446.