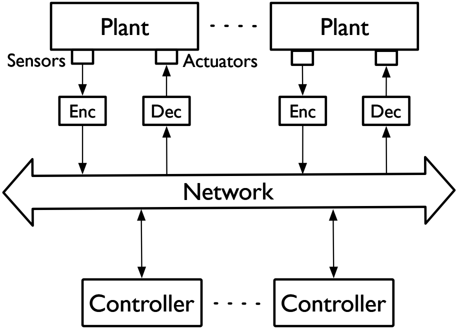
Lecture 3

Stability of NCS: sampling and delay

Giancarlo Ferrari Trecate[1]

[1]Dependable Control and Decision Group
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
giancarlo.ferraritrecate@epfl.ch

Recap from the last lecture



Control networks

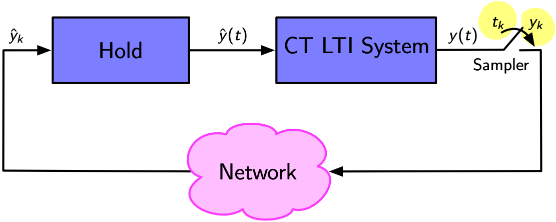
- Packet-networks designed for real-time operations
- Delays induced by: the physical layer, the transmission of complete packets, queuing at source nodes, decoding at the destination nodes, the MAC protocol, and the network load
  - Time-varying, often stochastic delays
- Packet dropout due to collisions + no retransmission of old packets

Outline

How much sampling time and network delays can deteriorate stability and performance?

- Analysis
  - Discrete-time (DT) models of NCS with linear dynamics
  - Examples of the effect of sampling and delays on stability
  - The Maximum Allowable Transfer Interval (MATI): definition and estimation
- Control Design
  - Delay compensation for remote control

NCS - collocated control

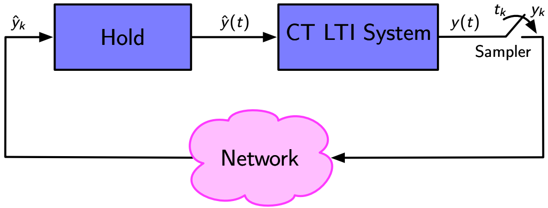


Previous lecture: “encoder”=sampler, “decoder”=hold

Assumption: One-plant-one-controller setting. The controller and system are

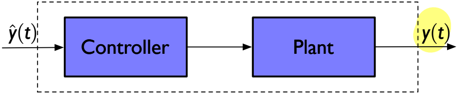
- collocated
- represented by a continuous-time (CT) LTI system

NCS - collocated control



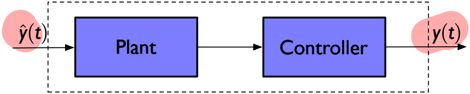
Previous lecture: "encoder" = sampler, "decoder" = hold

- Two possible arrangements:



- y: plant measurements
- $\hat{y}$: input to controller
- Controller collocated with actuators

- Controller gets "corrupted" measurements



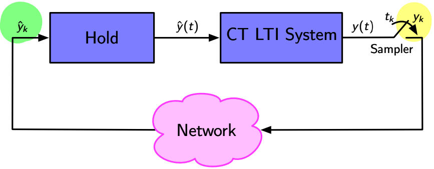
- y: control signals
- $\hat{y}$: input to the actuators
- controller collocated with sensors

- Plant gets "corrupted" control actions

Model of sample and hold

- System CT output $y(t)$ sampled at $\{t_k, k \in \mathbb{N}\}$
  - $y_k = y(t_k)$, $T_k \triangleq t_{k+1} - t_k$
- Hold + ideal network (no delay)

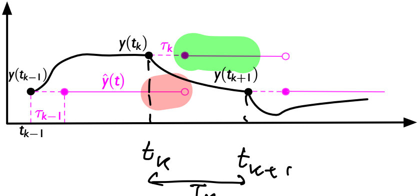
$$\hat{y}_k = y_k \qquad k \in \mathbb{N}$$
$$\hat{y}(t) = \hat{y}_k \quad t \in [t_k, t_{k+1})$$



- Hold + network time varying delay: y_k arrives at $t_k + \tau_k$

$\rightarrow$ Assumption (for simplicity) : $\tau_k < T_k$

$$\hat{y}_k = y_k, \ k \in \mathbb{N}$$
$$\hat{y}(t) = \begin{cases} \hat{y}_{k-1} & t \in [t_k, t_k + \tau_k) \\ \hat{y}_k & t \in [t_k + \tau_k, t_{k+1}) \end{cases}$$

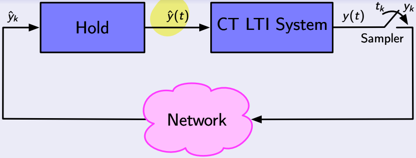


Discrete-time model of the LTI system (1/2)

Goal: compute the discrete-time dynamics for x_k

LTI system

$$\dot{x} = Ax + Bu, \quad u = \hat{y}$$
$$y = Cx$$



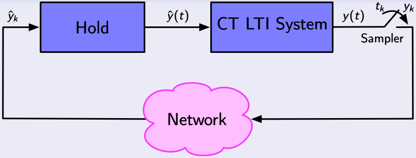
- Set $x_k = x(t_k)$, $y_k = y(t_k)$ etc.
- Recall the Lagrange formula for the above system with $x(t_0) = x_0$

$$x(t) = e^{A(t-t_0)}x_0 + \underbrace{\int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau}_{(b)}$$

State transition operator e^{As}: pushes x_0 ahead by s seconds

Discrete-time model of the LTI system (1/2)

Goal: compute the discrete-time dynamics for x_k



LTI system

$$\dot{x} = Ax + Bu, \quad u = \hat{y}$$
$$y = Cx$$

- Set $x_k = x(t_k)$, $y_k = y(t_k)$ etc.
- Recall the Lagrange formula for the above system with $x(t_0) = x_0$

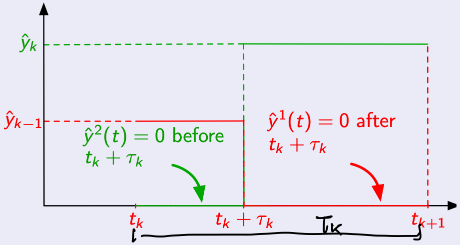
$$x(t) = e^{A(t-t_0)}x_0 + \underbrace{\int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau}_{(b)}$$

Constant-input transmission operator $\Gamma(s) \triangleq \int_0^s e^{Az}dz$

- (b)$= \Gamma(t-t_0)B\bar{u}$ if $u(.) = \bar{u}$ on $[t_0, t]$.
  Proof: (b)$=\int_{t_0}^{t} e^{A(t-\tau)}d\tau B\bar{u} = -\int_{t-t_0}^{0} e^{Az}dzB\bar{u} = \int_0^{t-t_0} e^{Az}dzB\bar{u}$, where we have set $z = t-\tau$.
- $\Gamma(t-t_0)B$ "pushes" $\bar{u}$ ahead by $t-t_0$ seconds

Discrete-time model of the LTI system (2/2)

Represent $u(t) = \hat{y}(t)$ on $[t_k, t_{k+1}]$ as $\hat{y}^1(t) + \hat{y}^2(t)$



For computing x_{k+1}, use the superposition principle with three causes: x_k, $\hat{y}^1(t)$, and $\hat{y}^2(t)$. Three experiments on $[t_k, t_{k+1}]$:

- $\phi(t_{k+1}, t_k, x_k, 0) = e^{AT_k} x_k$
- $\phi(t_{k+1}, t_k, 0, \hat{y}^2(t)) = e^{A(T_k - \tau_k)} \Gamma(\tau_k) B \cdot 0 + \Gamma(T_k - \tau_k) B \hat{y}_k = \Gamma(T_k - \tau_k) B C x_k$
- $\phi(t_{k+1}, t_k, 0, \hat{y}^1(t)) = e^{A(T_k - \tau_k)} x(t_k + \tau_k) = e^{A(T_k - \tau_k)} \Gamma(\tau_k) B \hat{y}_{k-1}$

x_{k+1} is the linear combination (with unit coefficients) of the three experiments:

$$x_{k+1} = e^{AT_k} x_k + e^{A(T_k - \tau_k)} \Gamma(\tau_k) B \hat{y}_{k-1} + \Gamma(T_k - \tau_k) B C x_k$$

Alternative derivation of the NCS (check @ home)

Use only the Lagrange formula on $[t_k, t_{k+1}]$

$$x_{k+1} = e^{A(t_{k+1}-t_k)}x_k + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}B\hat{y}(s)ds$$

$$= e^{A(t_{k+1}-t_k)}x_k + \left(\int_{t_k}^{t_k+\tau_k} e^{A(t_{k+1}-s)}ds\right)B\hat{y}_{k-1}ds$$

$$+ \left(\int_{t_k+\tau_k}^{t_{k+1}} e^{A(t_{k+1}-s)}ds\right)B\hat{y}_k$$

$$= e^{A(t_{k+1}-t_k)}x_k + e^{A(t_{k+1})}e^{-A(t_k+\tau_k)}\int_{t_k}^{t_k+\tau_k} e^{A(t_k+\tau_k-s)}dsB\hat{y}_{k-1}$$

$$+ \left(\int_{t_k+\tau_k}^{t_{k+1}} e^{A(t_{k+1}-s)}ds\right)B\hat{y}_k$$

Changing variables in the integrals, so as to make $\Gamma(\cdot)$ appear:

$$x_{k+1} = e^{AT_k}x_k + e^{A(T_k-\tau_k)}\Gamma(\tau_k)B\hat{y}_{k-1} + \Gamma(T_k-\tau_k)BCx_k$$

NCS global model

Need x_k and $\hat{y}_{k-1}$ for computing x_{k+1}. Define the augmented state

$$z_k \triangleq \left[x(t_k)^T \quad \hat{y}_{k-1}^T \right]^T .$$

NCS dynamics

$$z_{k+1} = \Psi(T_k, \tau_k) z_k$$

$$\Psi(T_k, \tau_k) = \begin{bmatrix} e^{AT_k} + \Gamma(T_k - \tau_k)BC & e^{A(T_k - \tau_k)}\Gamma(\tau_k)B \\ C & 0 \end{bmatrix}$$

Remarks

- The second line of Ψ represents $\hat{y}_k = Cx(t_k)$
- Ψ embodies the effect of sampling, network delay, and feedback interconnection
- Ideal network: $\tau_k = 0 \rightarrow \Gamma(0) = 0 \rightarrow$ The red part disappears
  - Simplified dynamics: $x_{k+1} = (e^{AT_k} + \Gamma(T_k)BC)x_k$

NCS global model

Need x_k and $\hat{y}_{k-1}$ for computing x_{k+1}. Define the augmented state

$$z_k \triangleq \left[x(t_k)^T \quad \hat{y}_{k-1}^T \right]^T .$$

NCS dynamics

$$z_{k+1} = \Psi(T_k, \tau_k) z_k$$

$$\Psi(T_k, \tau_k) = \begin{bmatrix} e^{AT_k} + \Gamma(T_k - \tau_k)BC & e^{A(T_k - \tau_k)}\Gamma(\tau_k)B \\ C & 0 \end{bmatrix}$$

Problems

- How to calculate block matrices in Ψ?

Computations of e^{As} $= I + As + \frac{(As)^2}{2!} + \dots$

$$\Psi(T_k, \tau_k) = \begin{bmatrix} e^{AT_k} + \Gamma(T_k - \tau_k)BC & e^{A(T_k - \tau_k)}\Gamma(\tau_k)B \\ C & 0 \end{bmatrix}$$

- Closed-form of e^{As}: simple for only special A (e.g. diagonal)
- Symbolic computations. In MatLab (requires the symbolic toolbox)

  ```
  syms s
  A = [-1 0.9; 0 -0.2]
  E = expm(A*s)
  ```

  gives

  $$E = \begin{bmatrix} e^{-s} & \frac{9}{8}(e^{\frac{s}{5}} - e^{-s}) \\ 0 & e^{\frac{s}{5}} \end{bmatrix}$$

- Numerical computation for given A and s

  ```
  A = [-1 0.9; 0 -0.2]
  s=0.5
  E = expm(A*s)
  ```

Computation of $\Gamma(s) = \int_0^s e^{A\tau} d\tau$

- If $det(A) \neq 0$, then $\Gamma(s) = A^{-1}(e^{As} - I)$
  Proof:

$$\int_0^s e^{A\tau} d\tau = \int_0^s (I + A\tau + \frac{(A\tau)^2}{2!} + ..)d\tau = (sI + \frac{As^2}{2!} + \frac{A^2 s^3}{3!} + ..)$$

  Then,

$$A \int_0^s e^{A\tau} d\tau = (e^{As} - I)$$

$e^{As} = I + As + \frac{(As)^2}{2!} + \cdots$

- If $\det(A) = 0$, other methods exist
  In MatLab, compute $\Gamma(0.5)$ as

```
s=0.5
EXPO=(@(X)(expm(A*X))
Gamma=integral(EXPO,0,0,s,'ArrayValued',time)
```

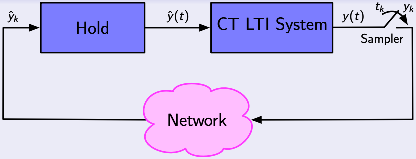
2h13 ↓

Summary of the model (1/2)

NCS with collocated control

LTI system

$$\dot{x} = Ax + B\hat{y}$$
$$y = Cx$$

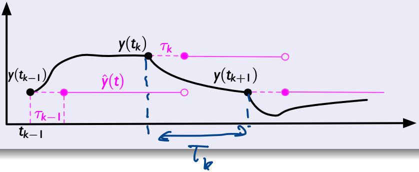


- $T_k = t_{k+1} - t_k$

Network model with delays: y_k arrives at $t_k + \tau_k$ ($\tau_k < T_k$ by assumption)

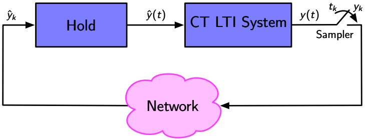
$$\hat{y}_k = y_k, \;\; k \in \mathbb{N}$$

$$\hat{y}(t) = \begin{cases} \hat{y}_{k-1} & t \in [t_k, t_k + \tau_k) \\ \hat{y}_k & t \in [t_k + \tau_k, t_{k+1}) \end{cases}$$



Augmented state: $z_k \triangleq \left[x(t_k)^T \quad \hat{y}_{k-1}^T \right]^T$

Summary of the model 2/2



The NCS DT system

$$z_{k+1} = \Psi(T_k, \tau_k) z_k$$

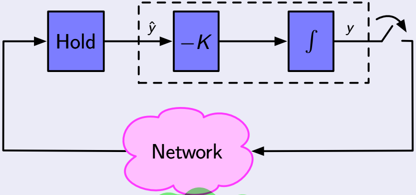
$$\Psi(T_k, \tau_k) = \begin{bmatrix} e^{AT_k} + \Gamma(T_k - \tau_k)BC & e^{A(T_k - \tau_k)}\Gamma(\tau_k)B \\ C & 0 \end{bmatrix}$$

- $\Gamma(s) = \int_0^s e^{A\tau} d\tau$
- The NCS is an LTV system
- Nonlinear ans nontrivial dependence of Ψ on sampling intervals T_k and delays τ_k
- **Effect of T_k and τ_k on stability?**

Examples: effect of sampling and delays on NCS stability

Effect of sampling and delay on stability (1/3)

Example: controlled integrator



LTI system in the box

$$\begin{cases} \dot{x} = 0 \cdot x - K\hat{y} & A = 0, B = -K < 0 \\ y = x & C = 1 \end{cases}$$

Assumptions: $T_k = T$, $\tau_k = \tau$, $k = 0, 1, \cdots$

One has: $e^{AT} = 1$, $\Gamma(s) = \int_0^s e^{A\tau} d\tau = s$

$$\Psi(T,\tau) = \begin{bmatrix} 1 - K(T-\tau) & -K\tau \\ 1 & 0 \end{bmatrix}$$

The NCS is a DT LTI system. Check stability using the eigenvalues of Ψ

$$\chi(\lambda) = \det(\lambda I - \Psi(T,\tau)) = \det\left(\begin{bmatrix} \lambda - 1 + K(T-\tau) & K\tau \\ -1 & \lambda \end{bmatrix}\right)$$

$$= \lambda^2 - \lambda(1 - K(T-\tau)) + K\tau$$

Effect of sampling and delay on stability (2/3)

- Recall: Jury's criterion for $\chi(\lambda) = \lambda^2 + \alpha\lambda + \beta$

  All roots of $\chi(\lambda)$ have modulus $< 1 \Leftrightarrow \begin{cases} \beta > -\alpha - 1 \\ \beta > \alpha - 1 \\ \beta < 1 \end{cases}$

For $\chi(\lambda) = \lambda^2 - \lambda\overbrace{(1 - K(T - \tau))}^{-\alpha} + \overbrace{K\tau}^{\beta}$ we have the conditions

$$+K\tau > (1 - K(T - \tau)) - 1 \quad \rightarrow \quad K\tau > -K(T - \tau) \tag{1}$$

$$+K\tau > -(1 - K(T - \tau)) - 1 \quad \rightarrow \quad K\tau > -2 + K(T - \tau) \tag{2}$$

$$K\tau < 1 \quad \xrightarrow{K>0} \quad \tau < \frac{1}{K} \tag{3}$$

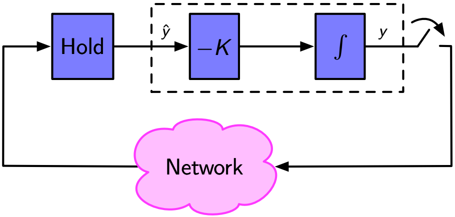
Since $K > 0$

(1) $\rightarrow \tau > -T + \tau \rightarrow 0 > -T$ always OK

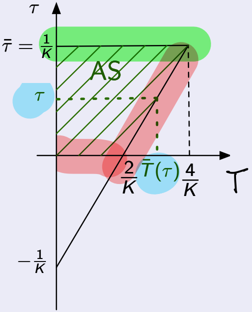
(2) $\rightarrow \tau > -\frac{2}{K} + T - \tau \rightarrow \tau > -\frac{1}{K} + \frac{T}{2}$

Conclusion:
From (2), (3), and $\tau > 0$, the NCS is AS iff $\max\left(0, -\frac{1}{K} + \frac{T}{2}\right) < \tau < \frac{1}{K}$

Effect of sampling and delay on stability (3/3)



Region of asymptotic stability in the (T, τ)-plane

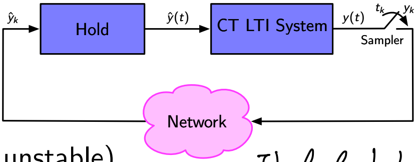


- Maximal tolerable delay: $\bar{\tau} = 1\backslash K$
  - Aggressive controller (K big) implies small $\bar{\tau}$
- For a given $\tau \in [0, \bar{\tau}]$, there is a Maximum Allowable Transfer Interval (MATI), i.e. a maximal $\bar{T}(\tau)$

Other examples (constant T_k and τ_k): first-order system

LTI system

$$\dot{x} = Ax + Bu$$
$$y = Cx$$



- $A = 1, B = -2, C = 1$ (open-loop unstable)

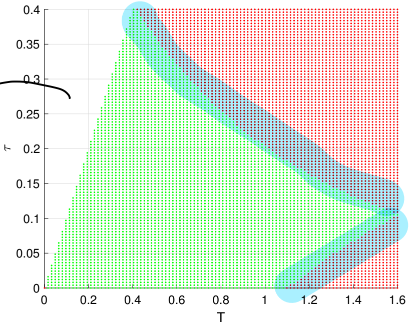
Ideal closed-loop system

$\dot{x} = Ax + BCx$

$\dot{x} = \underbrace{(A + BC)}_{-1} x$

↳ As. Stabl

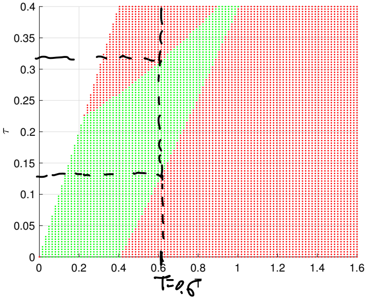
$T < \tau$



- Green$=\Psi(T, \tau)$ is Schur. Non-obvious shape of the region ...
  - still for $\tau \in [0, 0.4]$, there is a MATI

Other examples (constant T_k and τ_k): first-order system

- $A = -1, B = -5, C = 1$ (open-loop stable)



- There is a MATI but also a lower bound for T for high enough τ.

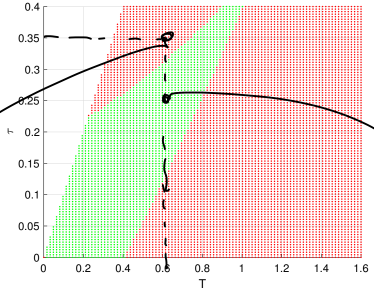
Remark

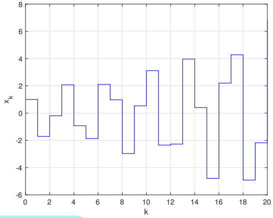
For $T = 0.6$, a delay large enough (but not too much) is stabilizing
$\rightarrow$ not obvious

E. Fridman, "Introduction to time-delay and sampled-data systems," 2014 European Control Conference (ECC), Strasbourg, 2014, pp. 1428-1433.

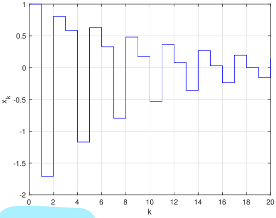
Simulations

- $A = -1, B = -5, C = 1$ (open-loop stable)





$T = 0.6, \tau = 0.35$: unstable

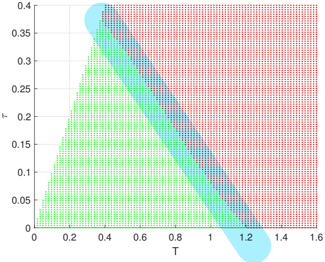


$T = 0.6, \tau = 0.25$: stable

Other examples : second-order system

$$A = \begin{bmatrix} -1 & 0.9 \\ 0.2 & -0.2 \end{bmatrix} \quad B = \begin{bmatrix} -5 \\ -0.5 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.1 & 1 \end{bmatrix} \quad \text{spec}(A) = \{-1.183, -0.017\}$$



Conclusions

The analysis of pairs (T, τ) guaranteeing asymptotic stability is not trivial

- In all cases, there is a MATI
- formal methods needed: see next!

Estimation of the MATI

The MATI

Definition

For a given $\tau_{min}, \tau_{max} \in \mathbb{R}$, the MATI is the largest $T \in \mathbb{R}$ such that

$$T > T_k, k = 0, 1, \ldots \Rightarrow \text{the NCS is AS for all } \tau_k \in [\tau_{min}, \tau_{max}]$$

Remarks

Knowledge of MATI allows one to set the sampler for

- preserving stability
- avoiding small T_k, which increases the network load

MATI estimation - constant T, τ

Assumption 1 : $\tau_k = \tau,\ T_k = T,\ \forall k \geq 0$ and $\tau < T$

- Realistic for:
  - Controlled Area Network (CAN) protocol (the maximal τ_k is constant for high-priority messages) and token-ring bus
  - protocols where τ_k are equalized using a buffer at the receiver $\rightarrow$ careful: all messages will appear to have the worst-case delay

Under Assumption 1, one can compute a stability region in the plane (T, τ), as done in previous examples.

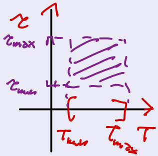
MATI estimation - variable T_k, τ_k

Sufficient stability condition using the candidate Lyapunov function $V(z) = z^T P z$ for the NCS dynamics

$$z_{k+1} = \Psi(T_k, \tau_k) z_k$$

Theorem: Assume that $\forall k \in \mathbb{N}$

$$T_k \in [T_{min}, T_{max}] \text{ and } \tau_k \in [\tau_{min}, \tau_{max}] \tag{4}$$



where $T_{min} > \tau_{max}$. The NCS is exponentially stable if $\exists P = P^T > 0$ and $\nu > 0$ such that

$$\Psi(T,\tau)^T P \Psi(T,\tau) - P \leq -\nu I \quad \forall T \in [T_{min}, T_{max}] \; \forall \tau \in [\tau_{min}, \tau_{max}] \tag{5}$$

- Checking if there is P such that (5) holds amounts to solving a set of LMIs, after gridding the box $[T_{min}, T_{max}] \times [\tau_{min}, \tau_{max}]$
- $V(z)$ is a *common* Lyapunov function for all (T, τ) in the box $\rightarrow$ **sufficient** condition only.

1h44

MATI estimation - variable T_k, τ_k

Sufficient stability condition using the candidate Lyapunov function $V(z) = z^T P z$ for the NCS dynamics

$$z_{k+1} = \Psi(T_k, \tau_k) z_k$$

Theorem: Assume that $\forall k \in \mathbb{N}$

$$T_k \in [T_{min}, T_{max}] \text{ and } \tau_k \in [\tau_{min}, \tau_{max}] \tag{4}$$

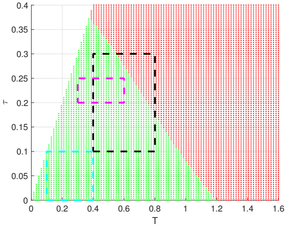
where $T_{min} > \tau_{max}$. The NCS is exponentially stable if $\exists P = P^T > 0$ and $\nu > 0$ such that

$$\Psi(T, \tau)^T P \Psi(T, \tau) - P \leq -\nu I \quad \forall T \in [T_{min}, T_{max}] \; \forall \tau \in [\tau_{min}, \tau_{max}] \tag{5}$$

- For estimating the MATI
  - $\tau_{min}, \tau_{max}, T_{min}$ are given by the network technology
  - Reduce T_{max} until (5) is feasible $\rightarrow$ (conservative) estimate of MATI

Example: second-order system (ctd.)

Green/red regions=stable/unstable NCS for constant T and τ



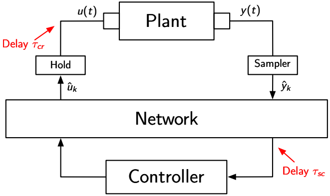
Conservativity of the theorem

Through numerical computations one finds that the LMIs are:

- unfeasible for (T, τ) in the black region (expected as $T_k = T, \tau_k = \tau$, and $(T, \tau) \in$(red region) causes instability)
- unfeasible for (T, τ) in the magenta region
  - LMIs are just sufficient for stability
  - stability condition for variable T_k, τ_k are expected to be more restrictive than those for constant T_k and τ_k
- feasible in the cyan region

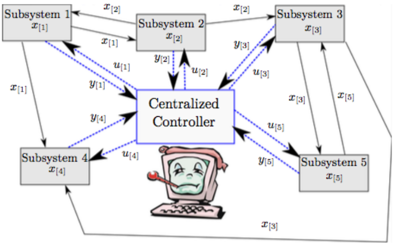
Compensation of network-induced delay (remote control setting)

NCS - remote control setting



- Plant and controller on different sides of the network

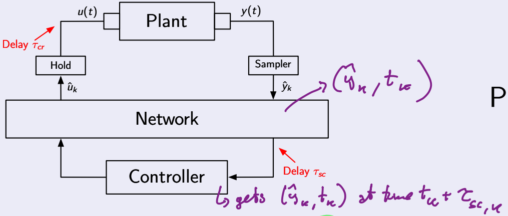
Centralized control



Remote control by human

Compensation for network-induced delay

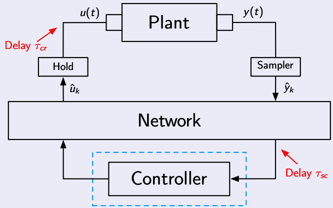
NCS - remote control setting



$$\text{Plant :} \begin{cases} \dot{x} = Ax + Bu \\ y = x \end{cases}$$

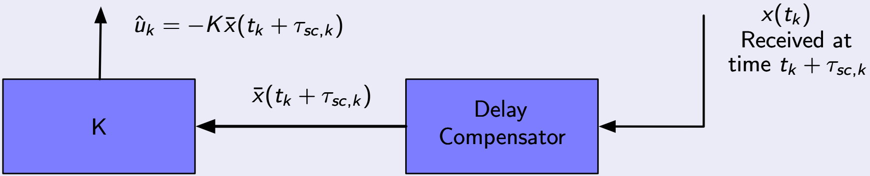
- The controller gets delayed (full-state) measurements and the plant gets delayed control actions.
- τ_{sc} : sensor-to-controller delay. Using time-stamped measurements the controller CAN KNOW τ_{sc} at the time of computation of $\hat{u}_k$. Idea: compensate for it!
- τ_{cr} : controller-to-actuator delay. Unknown to the controller at the time of computation of $\hat{u}_k$

Timing diagram and structure of the controller

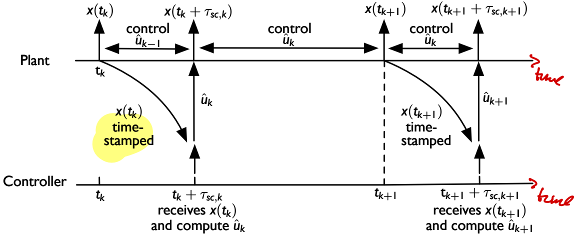


$$\text{Plant}: \begin{cases} \dot{x} = Ax + Bu \\ y = x \end{cases}$$

Zoom of the controller



Timing diagram and structure of the controller



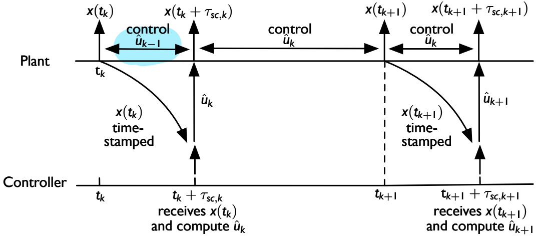
Assumptions (for simplicity)

- No controller-to-actuator delay: $\tau_{cr,k} = 0$
- Constant sampling period T and $\tau_{sc,k} < T, \ k = 0, 1, \dots$

How the controller computes $\hat{u}_k$?

- Option 1 : $\hat{u}_k = -Kx(t_k)$, but the state is already $x(t_k + \tau_{sc,k}) \rightarrow$ delay
- Option 2 : $\hat{u}_k = -K\bar{x}(t_k + \tau_{sc,k})$, where $\bar{x}(t_k + \tau_{sc,k})$ is an estimate of $x(t_k + \tau_{sc,k}) \rightarrow$ much better

Compensation of τ_{sc} and computation of $\hat{u}_k$

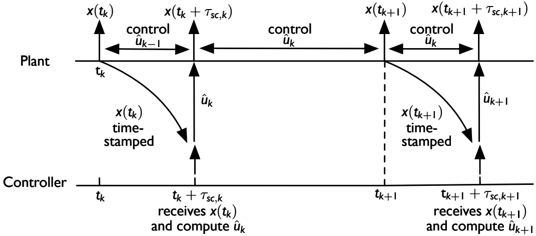


- Compute the prediction $\bar{x}(t_k + \tau_{sc,k})$ of $x(t_k + \tau_{sc,k})$
  - Setting $\Gamma(\bar{t}) = \int_0^{\bar{t}} e^{As} ds$, we define

$$\bar{x}(t_k + \tau_{sc,k}) = e^{A\tau_{sc,k}} x(t_k) + \Gamma(\tau_{sc,k}) B \hat{u}_{k-1}$$

  - x_{t_k} is received at $t_k + \tau_{sc,k}$
  - $\hat{u}_{k-1}$ is the known constant input over $t \in [t_k, t_k + \tau_{sc,k}]$
  - $\bar{x}(t_k + \tau_{sc,k})$ coincides with the true state $x(t_k + \tau_{sc,k})$

Compensation of τ_{sc} and computation of $\hat{u}_k$



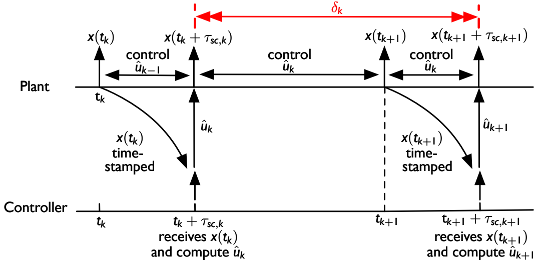
- Control law computed at $t_k + \tau_{sc,k}$

$$\hat{u}_k = -K\bar{x}(t_k + \tau_{sc,k})$$

Remark: We use the notation $\hat{u}_k$ even if the input is not computed at t_k (but it is the k^{th} computed value of $\hat{u}$)

Hold $u(t) = \hat{u}_k$ for $t \in [t_k + \tau_{sc,k}, t_{k+1} + \tau_{sc,k+1}]$

Closed-loop system from $t_k + \tau_{sc,k}$ to $t_{k+1} + \tau_{sc,k+1}$

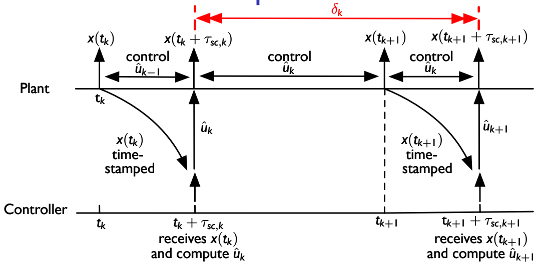


For $\delta_k = T + \tau_{sc,k+1} - \tau_{sc,k}$, one obtains

$$x(t_{k+1} + \tau_{sc,k+1}) = \underbrace{\left(e^{A\delta_k} - \Gamma(\delta_k)BK\right)}_{\tilde{A}_k} x(t_k + \tau_{sc,k})$$

- Defining the state $\tilde{x}_k = x(t_k + \tau_{sc,k})$, one has the LTV system $\tilde{x}_{k+1} = \tilde{A}_k \tilde{x}_k$
- If $\tau_{sc,k}$ is constant, $\delta_k = T$. Hence, $\tilde{A}_k = \tilde{A} = (e^{AT} - \Gamma(T)BK)$ and the NCS dynamics are LTI

Nominal controller with compensation



Nominal design of K

Assume $\tau_{sc,k} = \tau_{sc}$ (constant) and compute K such that $e^{AT} - \Gamma(T)BK$ is Schur

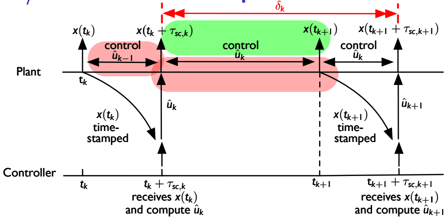
- use, e.g., eigenvalue assignment
- the exact value of τ_{sc} is irrelevant. Closed-loop stability is guaranteed if delays are constant in the real network

Summary: nominal controller with compensation

$$\bar{x}(t_k + \tau_{sc,k}) = e^{A\tau_{sc,k}} x(t_k) + \Gamma(\tau_{sc,k}) B \hat{u}_{k-1}$$

$$\hat{u}_k = -K\bar{x}(t_k + \tau_{sc,k})$$

Comparison with/without compensation



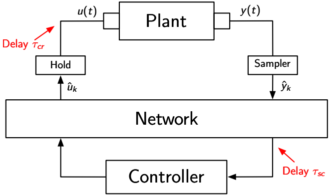
Comparison with the uncompensated controller: $\hat{u}_k = -Kx(t_k)$

- Compensated controller:
  - "wrong" (=old) control action on $[t_k, t_k + \tau_{sc,k}]$
  - best possible control action on $[t_k + \tau_{sc,k}, t_{k+1}]$
- Uncompensated controller:
  - "wrong" (=old) control action on $[t_k, t_k + \tau_{sc,k}]$
  - "wrong" (=based on past state) control action on $[t_k + \tau_{sc,k}, t_{k+1}]$

Performance and stability

- The compensated controller always outperforms the uncompensated one
- If, in the real network, $\tau_{sc,k}$ is not constant, closed-loop stability is not guaranteed using either controller

Example: delay compensation vs no compensation (1/2)



Plant dynamics

$$\dot{x} = -2x + u, \quad T = 0.2$$

Controller with compensation

closed-loop eigenvalue

- Nominal design of K

  $$e^{-2T} - \Gamma(T) \cdot 1 \cdot K = -0.9$$

  → arbitrarily set in the unit disk

  $$0.6703 - 0.1648 \cdot K = -0.9 \rightarrow K = 9.5263$$

- Predictor

  $$\bar{x}(t_k + \tau_{sc,k}) = e^{-2\cancel{T}\,\tau_{sc,k}} x(t_k) + \Gamma(\tau_{sc,k}) B \hat{u}_{k-1}$$

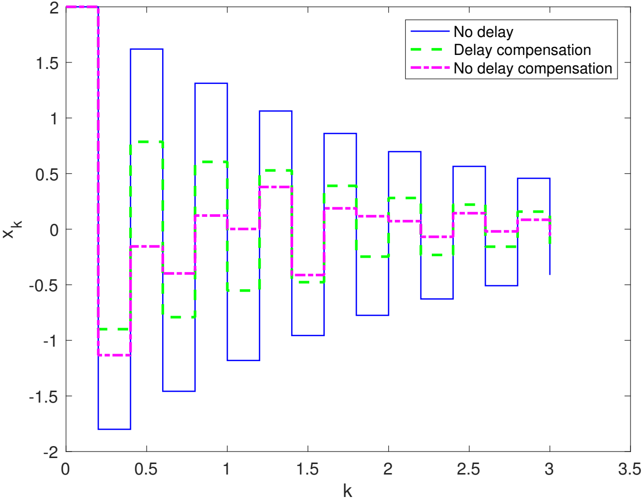
- Control action

  $$\hat{u}_k = -K\bar{x}(t_k + \tau_{sc,k})$$

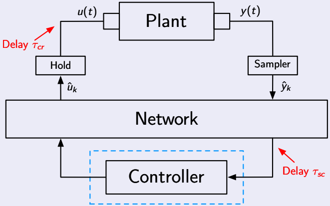
- Controller without compensation $\hat{u}_k = -Kx(t_k)$

Example: delay compensation vs no compensation (2/2)

Simulation with delays chosen randomly in $[0, 0.07]$

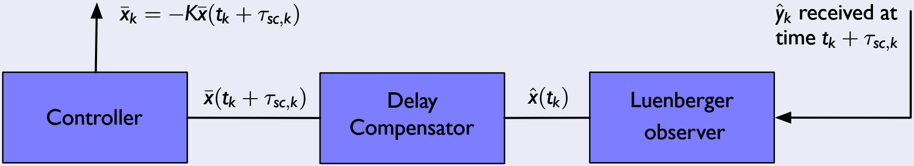


Generalization: NCS with output feedback



$$\text{Plant}: \begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$

Zoom of the controller



- $\bar{x}(t_k + \tau_{sc,k})$ is an estimate of the state $x(t_k + \tau_{sc,k})$
- No further details on this scheme

Take-home messages

- The effect of time-varying sampling intervals and delays can be VERY difficult to analyze
  - Estimate MATI using simulations or Lyapunov theory
- For remote control and time-stamped sensor measurements, it is always beneficial to compensate for known delays

$2h < \zeta$