EE185524

A Survey of Recent Results in NCS (before 2007)

Yurid Eka Nugraha

TSP DTE FTEIC ITS

INVITED PAPER

A Survey of Recent Results in Networked Control Systems

When sensors and actuators communicate with a remote controller over a multi-purpose network, improved techniques are needed for state estimation, determination of closed-loop stability and controller synthesis.

By João P. Hespanha, *Senior Member IEEE*, Payam Naghshtabrizi, *Student Member IEEE*, and Yonggang Xu, *Member IEEE*

Control & communication

- NCSs lie at the intersection of control and communication theories.
- Traditionally, control theory focuses on the study of interconnected dynamical systems linked through **ideal** channels
- Communication theory studies the transmission of information over **imperfect** channels
- **A combination of these two frameworks** is needed to model NCSs.

What make NCS distinct from other CS?

- Band-limited channels
  - in most digital networks, data is transmitted in atomic units called **packets** and sending a single bit or several hundred bits consumes the same amount of network resources
- Sampling and delay
  - The overall delay between sampling and eventual decoding at the receiver can be highly variable because both the **network access delays** and the **transmission delays** depend on highly variable network conditions such as congestion and channel quality.

What make NCS distinct from other CS?

- Packet dropout
  - result from transmission errors in physical network links
- System architecture
  - existence of encoders and decoders
  - however, the boundaries between a digital controller and encoder/decoder blocks are often blurry.

NCS architecture

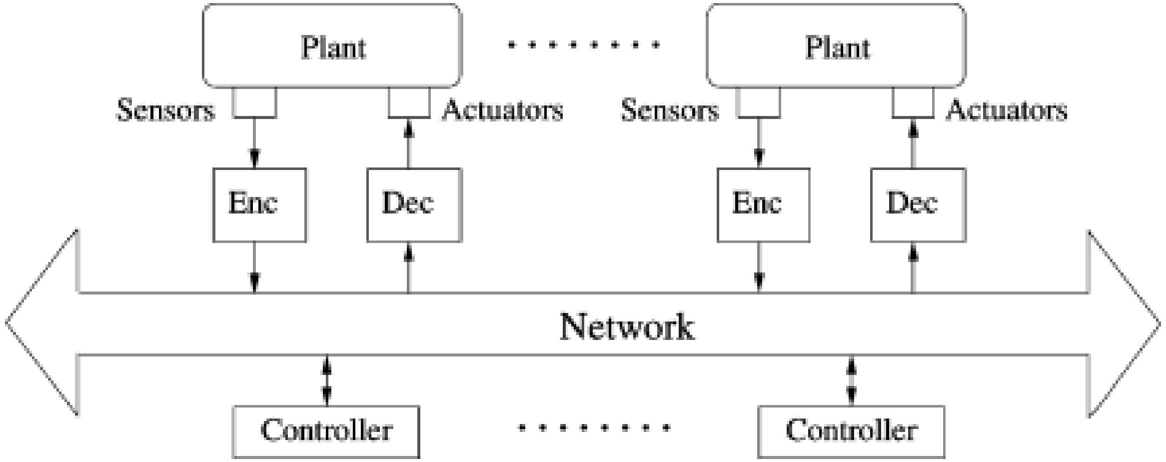


Fig. 1. ***General NCS architecture.***

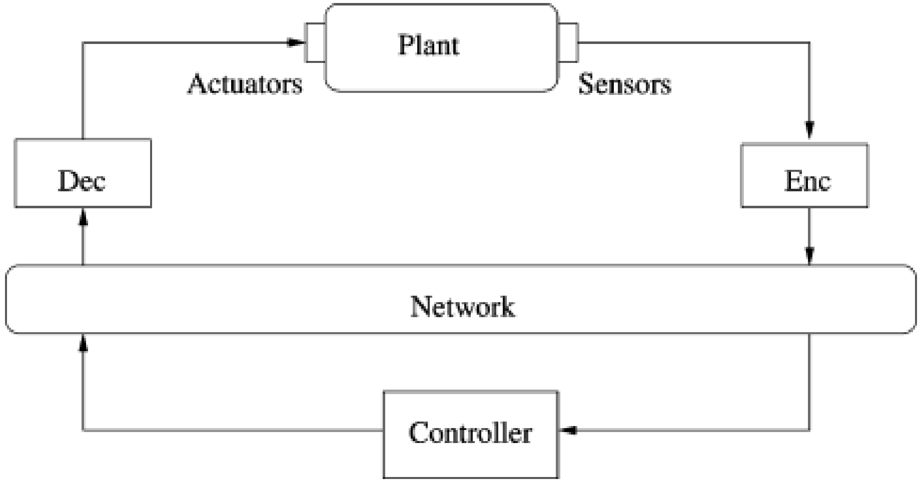
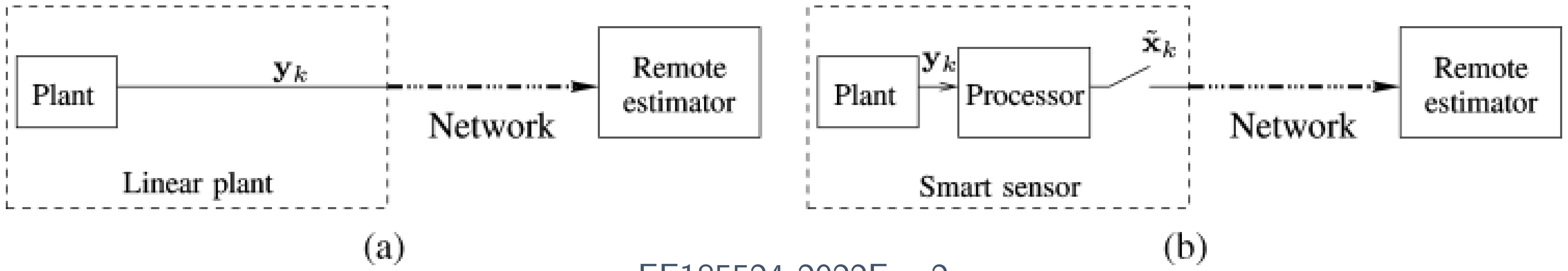


Fig. 2. ***Single-loop NCS.***

- Most of the research on NCSs considers structures simpler than the general one depicted in Fig. 1.
- For example, some controllers may be collocated (and therefore can communicate directly) with the corresponding actuators.

2. Estimation over lossy network

- Assumption: Network can be viewed as a channel that can carry **real numbers** without distortion but that some of the messages can be lost
- Important in remote sensing, sensor networks, etc.
- Consider two scenarios:
  a) raw measurement sent
  b) measurement processed locally first



2. Estimation over lossy network

Plant model:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k$$ Gaussian white noise

$$\mathbf{y}_k = C\mathbf{x}_k + \mathbf{v}_k$$ Gaussian white noise

$$\forall k \in \mathbb{N}, \ \ \mathbf{x}_k, \mathbf{w}_k \in \mathbb{R}^n, \quad \mathbf{y}_k, \mathbf{v}_k \in \mathbb{R}^p$$

2. Estimation over lossy network

Plant model:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k$$

Gaussian white noise

$$\mathbf{y}_k = C\mathbf{x}_k + \mathbf{v}_k$$

Gaussian white noise

$$\forall k \in \mathbb{N},\ \ \mathbf{x}_k, \mathbf{w}_k \in \mathbb{R}^n, \quad \mathbf{y}_k, \mathbf{v}_k \in \mathbb{R}^p$$

White noise: equal intensity at different frequencies

Optimal estimation for Bernoulli dropouts (Sinopoli et al.)

- Lossy channel modeled by $\theta_k \in \{0, 1\}$

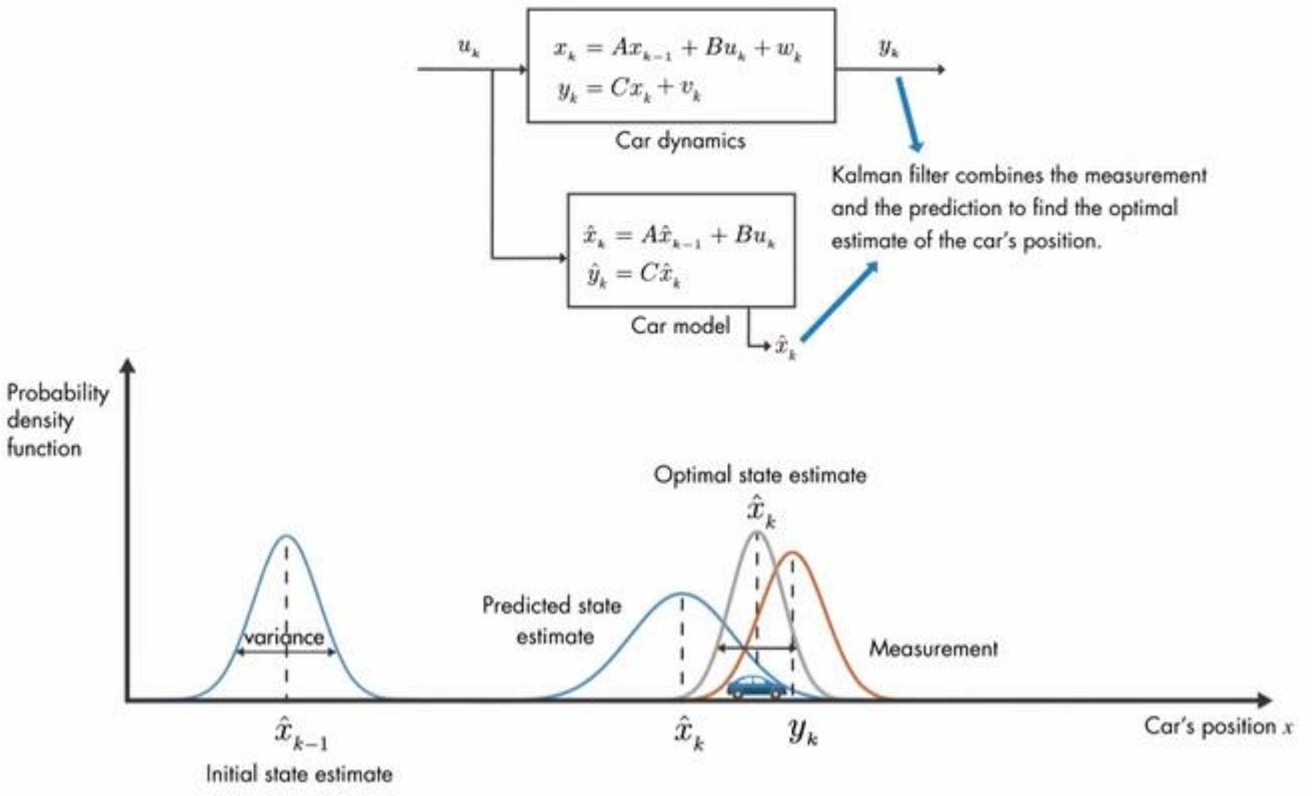
- $\theta_k = 1$ if successful, $\theta_k = 0$ if not

- **Bernoulli** process: a discrete-time stochastic process that takes only **two values**, canonically 0 and 1

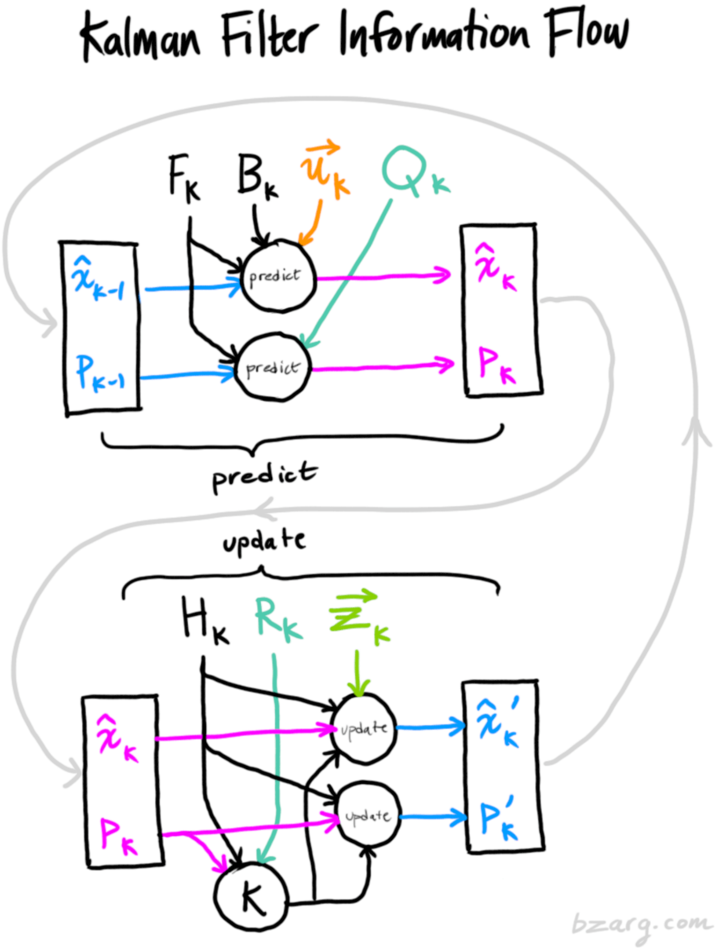
- Optimal estimate:
  $$\hat{\mathbf{x}}_{k|k-1} = E[\hat{\mathbf{x}}_k | \theta_l, \forall l \leq k-1; \mathbf{y}_l, \forall l \leq k-1 \ s.t. \ \theta_l = 1].$$

Kalman filter?

- Optimal state estimator



Kalman filter?



Optimal estimation for Bernoulli dropouts

- Time varying Kalman filter is used:

$$
\begin{aligned}
\hat{\mathbf{x}}_{0|-1} &= 0 \\
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \theta_k F_k(\mathbf{y}_k - Cx\hat{\mathbf{x}}_{k|k-1}) \\
\hat{\mathbf{x}}_{k+1|k} &= A\hat{\mathbf{x}}_{k|k}
\end{aligned}
$$

-
$$
\begin{aligned}
P_0 &= \Sigma \\
F_k &= P_k C'(CP_kC' + R_v)^{-1}, \\
P_{k+1} &= AP_kA' + R_w - AF_k(CP_kC' + R_v)F_k'A' \\
P_k &= E[(\mathbf{x} - \hat{\mathbf{x}}_{k|k+1})], \ \forall k \in \mathbb{N}
\end{aligned}
$$

Optimal estimation for Bernoulli dropouts

- Critical value p_c determines convergence: large enough p_c guarantees convergence

- Value p_c satisfies $\underline{p} \leq p_c \leq \overline{p}$, where
  $\underline{p} = \max\{p \geq 0 : \Psi_p(Y, Z) > 0, 0 \leq Y \leq I \text{ for some } Y, Z\}$

- In some special cases, $p_c = \overline{p}$

Multisensor plants (Matveev and Savkin)

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k \qquad \mathbf{y}_{v,k} = C_v\mathbf{x}_k + \mathbf{v}_{v,k},\ v \in \{1, \ldots, N\}$$

- Recursive Kalman filter is used
  - condition under which estimation error process is a.s. stable is derived

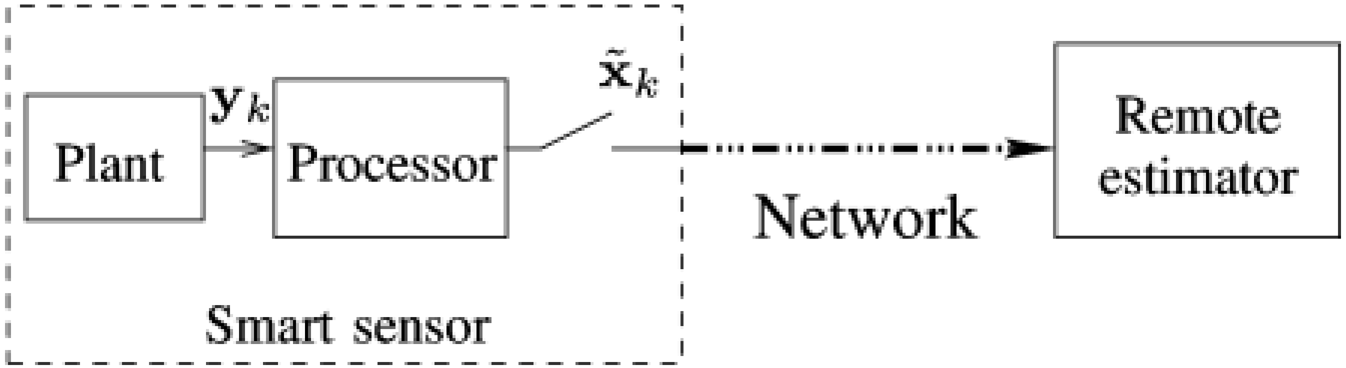
  - Optimal estimate:

$$\hat{\mathbf{x}}_{k|k-1} = E[\hat{\mathbf{x}}_k | \theta_{v,l}, \forall l \leq k - \tau_v(l); \mathbf{y}_{v,l}, \forall l \leq k - \tau_v(l)\ s.t.\ \theta_{v,l} = 1].$$

Reduced-computation estimation

- F_k and P_k cannot be computed offline
- Pre-computing a finite set of gains to be selected according to the dropout history in the last time steps
- Finite Loss History Estimator: FLHE

Estimation with local computation



➢Benefits of preprocessing the measurements before transmission to the network

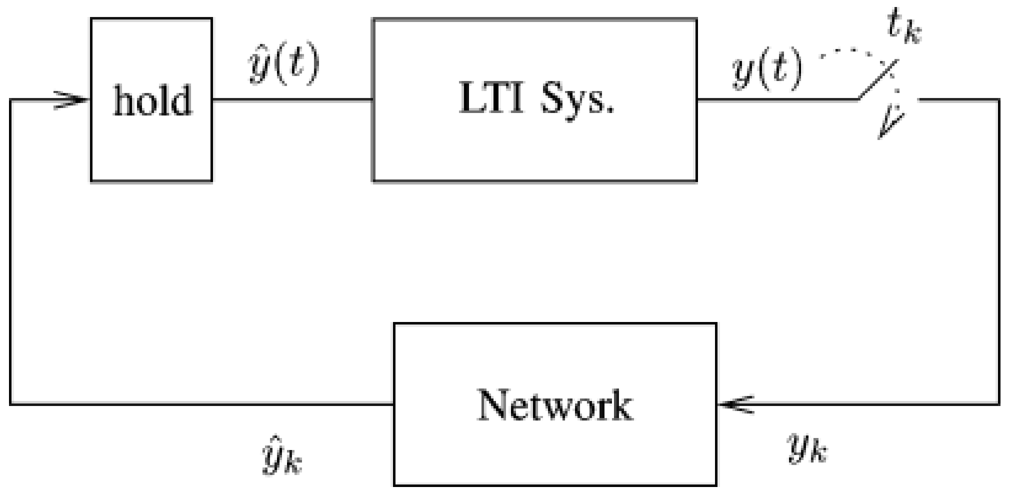
Estimation with controlled communication

- Sensor measurements not sent to the remote estimator at every step to reduce traffic
- Tradeoff between **communications** and **estimation performance**

3. Stability of NCS

- How system stability affected by sampling, delay, and packet dropouts
- Assumption: sensors, controllers, actuators, use a shared network to communicate

Sampling and delay



- One channel continuous: $\dot{x} = Ax + B\hat{y}, \quad y = Cx$
- Lossless with delay: $\hat{y}_k = y_k, \;\; \hat{y}(t) = \begin{cases} \hat{y}_{k-1}, & t \in [t_k, t_k + \tau_k) \\ \hat{y}_k, & t \in [t_k + \tau_k, t_{k+1}) \end{cases}$

Sampling and delay

- Delays longer than one sampling interval may result in more than one $\hat{y}_k$ (or none) arriving during a single sampling interval, making the derivation of recursive formulas difficult
- For simplicity, assume delays smaller than one sampling interval

- **Periodic** sampling and **constant** delay: time invariant system
- **Periodic** sampling and **variable** delay: DT switched system
- **Variable** sampling and delay: needs a Lyapunov-based analysis

General nonlinear case

- Nonlinear plant + exogenous disturbance:

$$\dot{x}_P = f_P(x_P, \hat{u}, w), \quad y = g_P(x_P)$$
$$\dot{x}_C = f_C(x_C, \hat{y}, w), \quad u = g_C(x_C)$$

- Impulsive system:

$$\dot{x} = f(x, e, w), \forall t \geq 0$$
$$\dot{e}_C = g(x, e, w), \forall t \in (t_k, t_{k+1}],$$
$$\dot{e}(t_k^+) = h(k, e(t_k)), \ \forall k \in \mathbb{N},$$

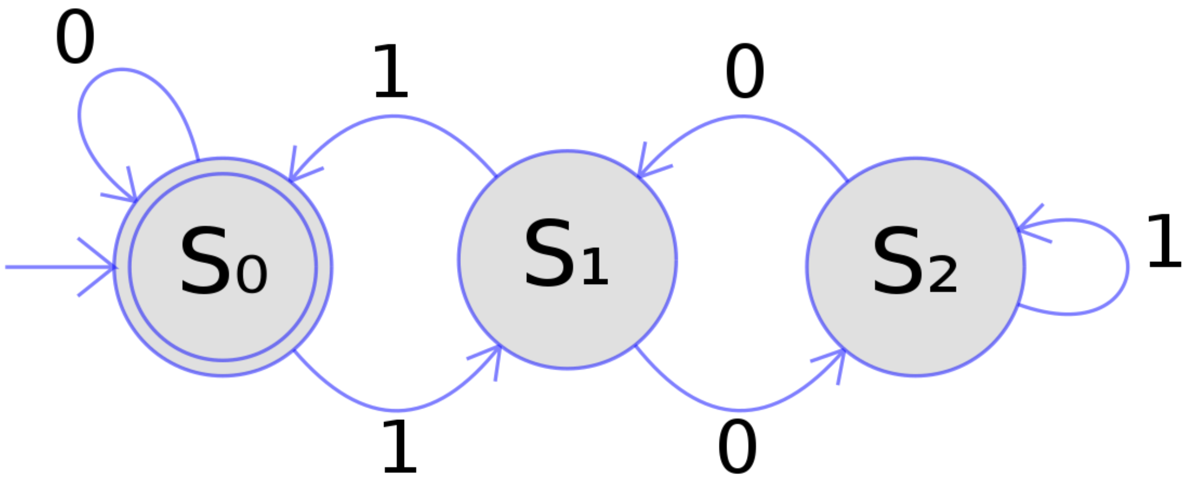
Packet dropouts

- Can be modeled either as stochastic or deterministic phenomena
- Simplest: realizations of a Bernoulli process
- Common assumption:

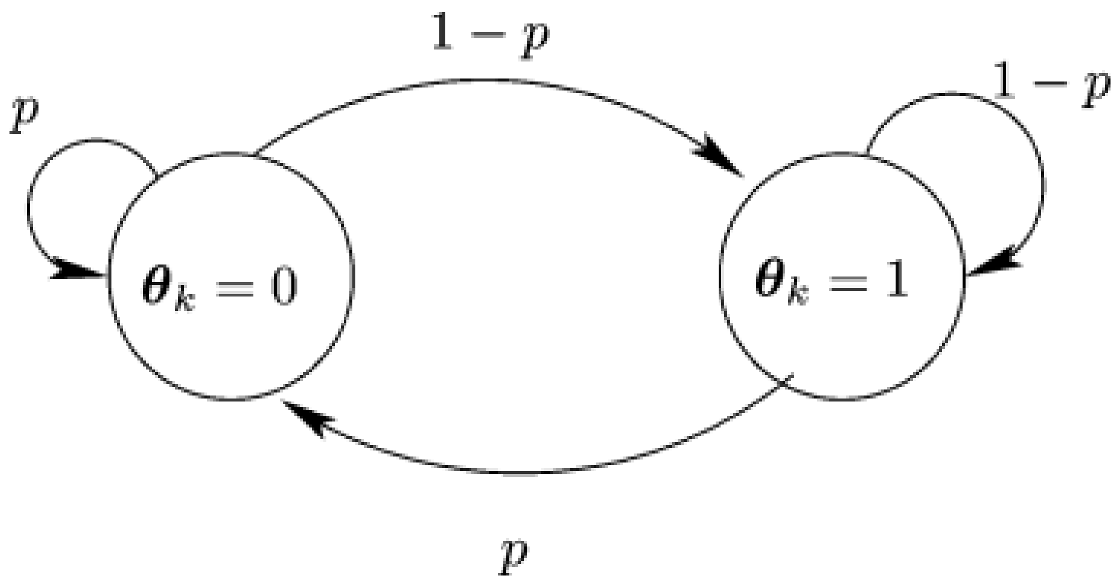
$$\hat{y}_k = \theta_k y_k + (1-\theta_k)\hat{y}_{k-1}, \quad \theta_k = 1 \text{ (no packet dropout)}$$

Deterministic dropouts

- Asynchronous dynamical system: hybrid systems whose
  - **continuous dynamics** are governed by **differential or difference equations**
  - **discrete dynamics** are governed by **finite automata**



Stochastic dropouts



NCS as delayed differential equations (sampling, delays, dropouts)

- Delay $\tau_k \geq 0,$
- Model lossless case $\hat{y}(t) = Cx(t_k),$

- Also can be represented as $\hat{y}(t) = Cx(t - \tau(t)),$
- $\tau(t)$ can be time varying

NCS as delayed differential equations (sampling, delays, dropouts)

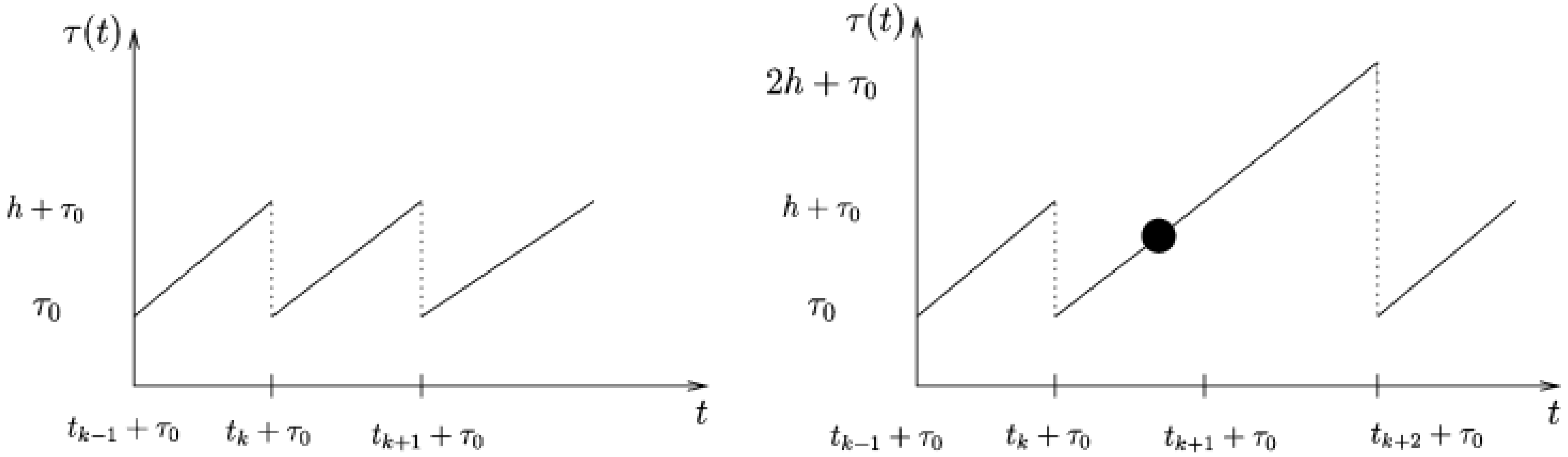
- System equation becomes: $\hat{x}(t) = Ax(t) + BCx(t - \tau(t)),$

- Delay: $\tau(t) \in [\tau_{\min}, \tau_{\max}),\ \tau_{\min} := \min_{k\in\mathbb{N}}\{\tau_k\}\ \ \tau_{\max} := \max_{k\in\mathbb{N}}\{\tau_k\}$

- These equations are valid **even when the delay exceeds the sampling interval**

NCS as delayed differential equations (sampling, delays, dropouts)

Dropouts as Delays?

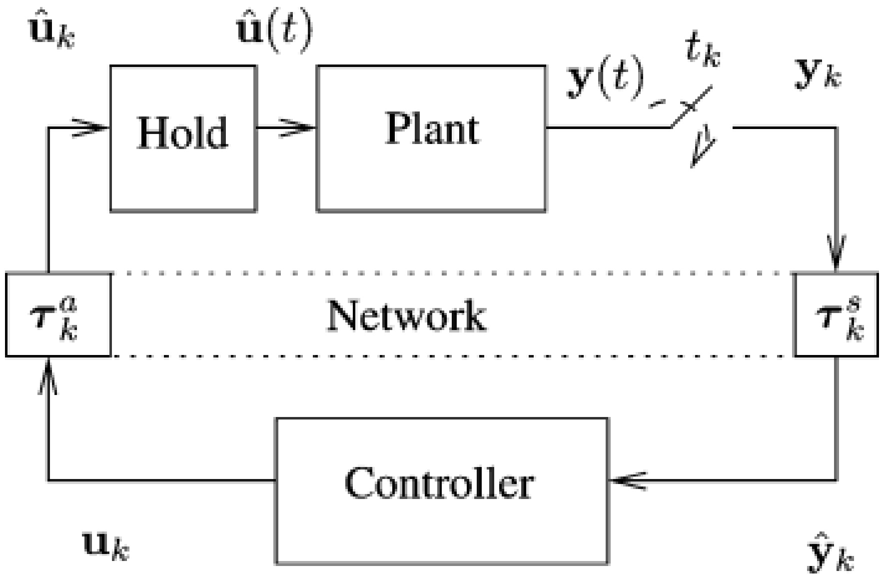


Controller synthesis

- Sampling and Delay
- Packet Dropout
- NCS as DDE

Some results are direct extension from stability analysis above

Controller synthesis under sampling and delay



Assuming no packet dropout:

$$\hat{\mathbf{u}} = \begin{cases} \hat{\mathbf{u}}_{k-1}, & t \in [t_k, t_k + \tau_k^s + \tau_k^a) \\ \hat{\mathbf{u}}_k, & t \in [t_k + \tau_k^s + \tau_k^a, t_{k+1}) \end{cases}$$

delays $\tau_k^s, \tau_k^a > 0$

Controller synthesis under packet dropouts

- Deterministic dropout rates: static output-feedback controller
  $$u_k = K_{k-\kappa_j}\hat{y}_k, \quad \forall k \in \{\kappa_j, \kappa_j + 1, \ldots, \kappa_{j+1} - 1\}$$
  - Stability can be established with quadratic Lyapunov functions

- Stochastic (Markovian) dropout

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k$$

$$\bar{x}_{k+1} = A_{\theta_k}\bar{x}_k + B_{\theta_k}\hat{y}_k, \quad u_k = C_{\theta_k}\hat{x}_k$$

Controller synthesis under DDE

- The stability of the NCS can be verified by studying the feasibility of a (convex) LMI $P_2 > 0$

- Controller gain may be conservative: may sacrifice performance

- To reduce conservativeness, *cone complementarity algorithm* is used

Conclusion and future challenges (2007)

- Focus on limitations on packet-rate, sampling, network delay, and packet dropout
- Several models have been proposed to tackle problems caused by NCS
- System performances under network?