

# Lekcija 10:

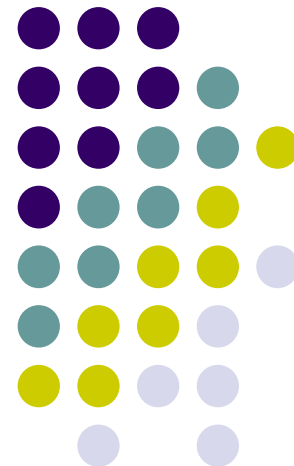
## *Upravljanje u zatvorenoj petlji preko kom. mreže*

---

Prof.dr.sc. Jasmin Velagić  
Elektrotehnički fakultet Sarajevo

Kolegij: Distribuirani sistemi

2012/2013



## Sadržaj poglavlja:

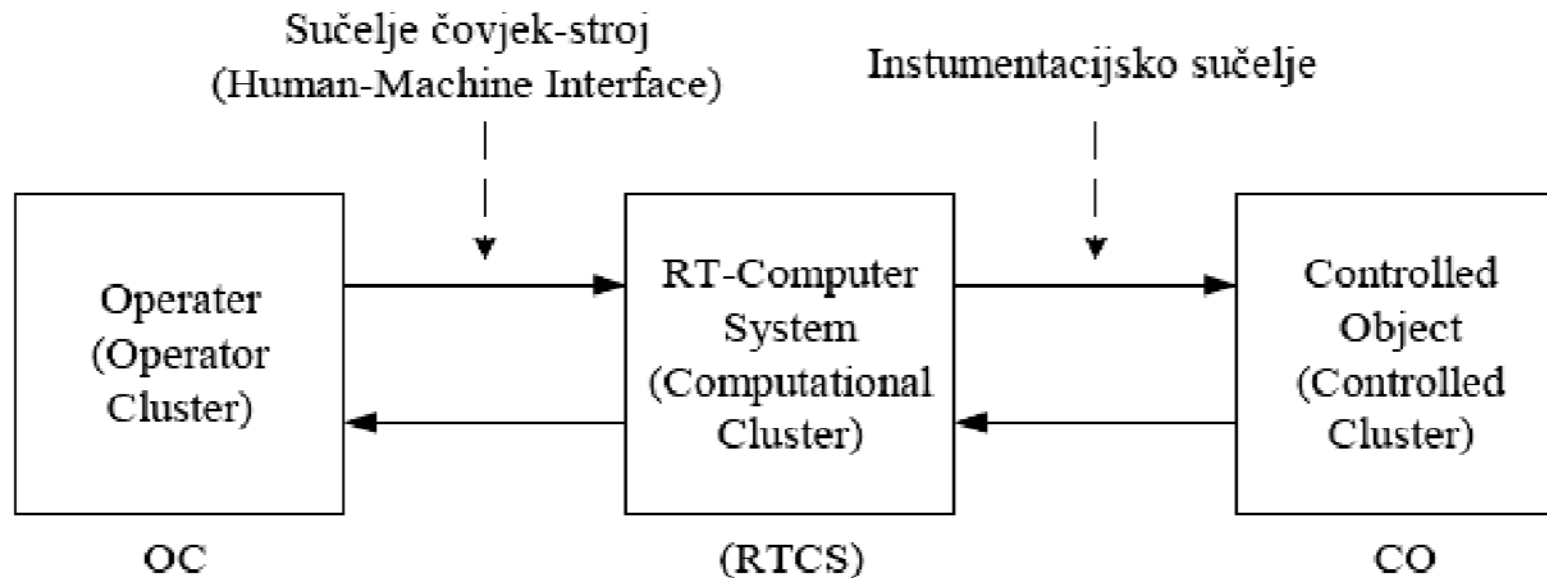
- ✚ Upravljanje u zatvorenoj petlji preko komunikacijske mreže
  - 🌐 RT distribuirani komunikacijski sistemi
  - 🌐 RT veličine, RT slike i RT objekti
  - 🌐 Zatvoreni sistem upravljanja preko komunikacijske mreže
  - 🌐 Vremena uzorkovanja više regulacijskih petlji
  - 🌐 Uzorkovanje na mreži sa prosljeđivanjem zaloga (TP)
  - 🌐 Uzorkovanje u CAN mreži
  - 🌐 Izvedbe sistema upravljanja preko komunikacijske mreže



## 10.1. RT distribuirani komunikacijski sistem

### RT komunikacijski sistem (RTCS)

- ❑ Ispravnost rada sistema ne ovisi samo o logičkom rezultatu proračuna već i o trenutku dobivanja rezultata.



- ❑ RTCS mora reagirati na pobudu od CO unutar definiranog vremenskog intervala  $\Rightarrow$  deadline:  $\blacklozenj$  hard real-time systems;  $\blacklozenj$  soft real-time systems.

# RT distribuirani komunikacijski sistem

## Funkcionalni zahtjevi na RTCS

### □ Prikupljanje podataka (Data Collection)

- značajne procesne varijable  $\Rightarrow$  RT veličine (RT entity);
- svaka RT veličina pripada samo jednom podsistemu (clusteru)  $\Rightarrow$  SOC (Sphere Of Control);
- izvan SOC-a RT veličina može se mjeriti (obzervirati), ali se ne može mijenjati;
- izmjerena vrijednosti RT veličine u RTCS-u naziva se njenom **RT slikom** (RT image);
- budući da se RT veličina mijenja, RT slika predstavlja njenu **ispravnu vrijednost** u ograničenom vremenskom intervalu;
- skup svih RT slika naziva se **RT baza** (RT Database). Baza se može osvježavati **periodički** (TT, Time-Triggered) i **dogodajno** (ET, Event-Triggered);
- kondicioniranje signala;
- nadzor alarma: primarni i sekundarni, pri čemu je važno utvrđivanje redoslijeda.

### □ Upravljanje u zatvorenoj petlji

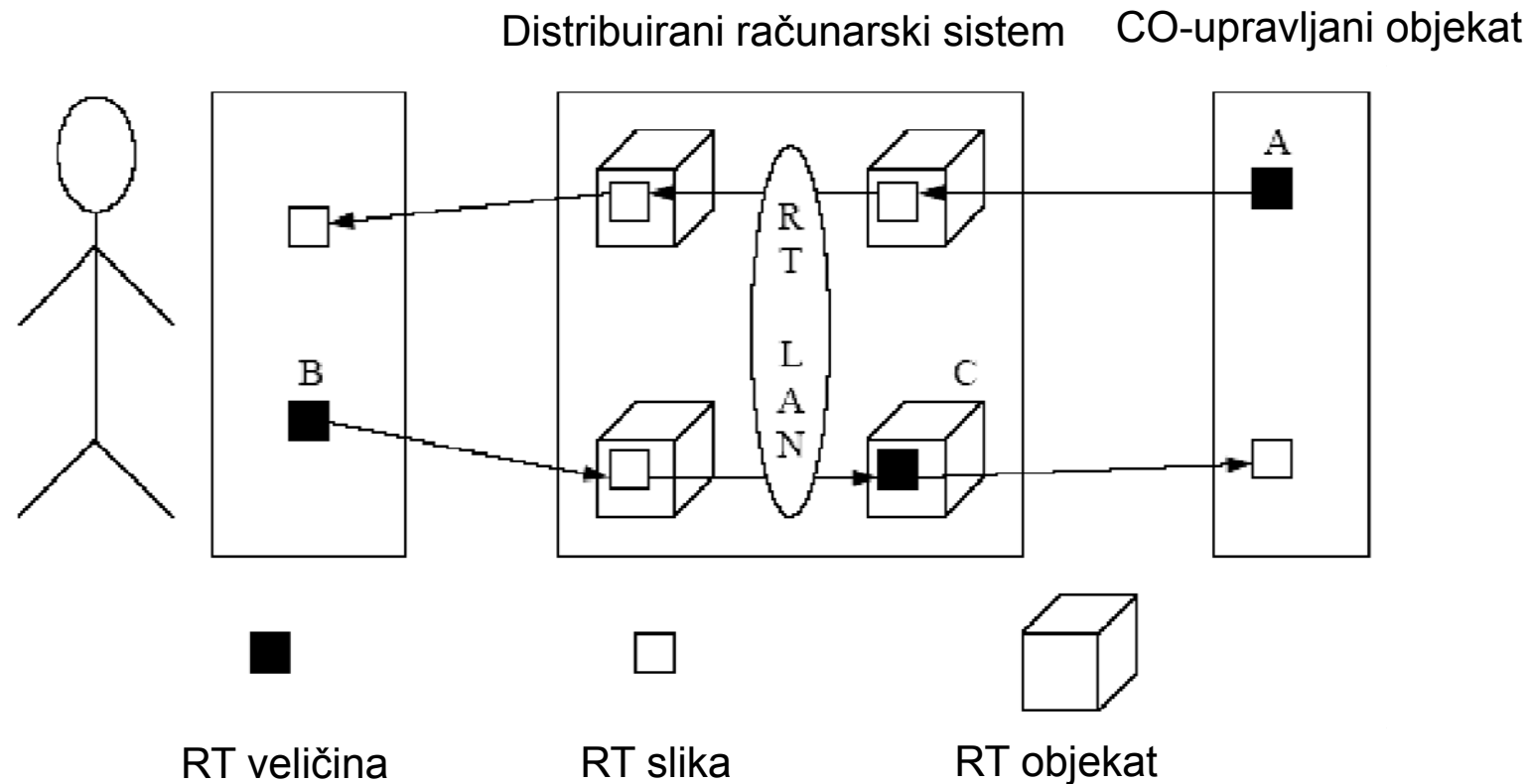
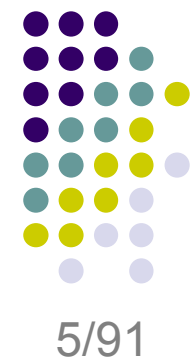
- fiksno vrijeme uzorkovanja  $\rightarrow$  strogi vremenski zahtjevi ( $100 \mu\text{s} \rightarrow 10 \text{ ms}$ ).

### □ Sučelje čovjek-stroj (HMI)

- $50 \rightarrow 100 \text{ ms} \rightarrow$  zbog spore percepcije čovjeka.



## 10.2. RT veličine, slike i objekti



A → mjerna vrijednost protoka.

B → referentna vrijednost protoka.

C → upravljačka veličina (npr. željeni otvor ventila).

# RT veličine, slike i objekti

## RT veličine

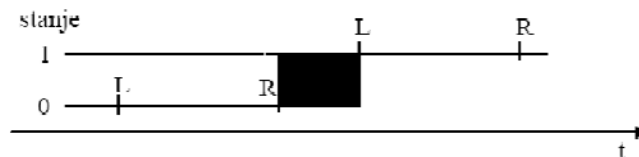
### Podjela

- **statički atributi**: ime, tip, područje vrijednosti, max d/dt,...
  - **dinamički atributi**: trenutna vrijednost, brzina promjene,...
- SOC:  $A \rightarrow CO$ ;  $B \rightarrow \text{Operater}$ ;  $C \rightarrow \text{RTCS (CC)}$ .

### Diskretne i kontinuirane RT veličine

*Diskretna RT veličina:*

$L \rightarrow R$  definirano stanje  
 $R \rightarrow L$  nedefinirano stanje  
 npr. vrata garaže u srednjem položaju – ni otvorena ni zatvorena

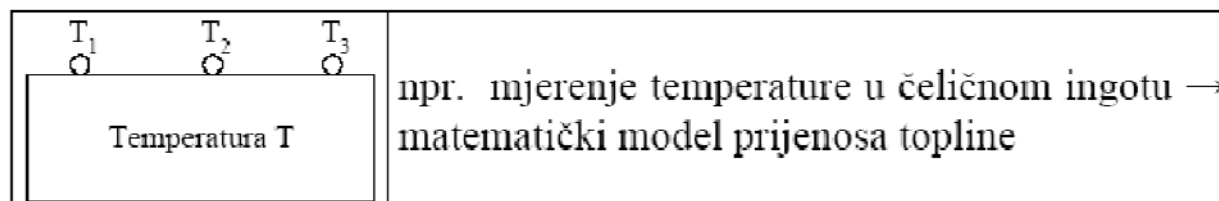


### Observacija RT veličine

- Informacija o stanju RT veličine u određenom vremenskom trenutku.
- OBSERVACIJA** je jedinična podatkovna struktura:

**OBSERVACIJA** =  $\langle \text{Ime}, t_0, \text{Vrijednost} \rangle$ ,  $t_0$ -trenutak observacije RT veličine

### Posredna observacija



# RT veličine, slike i objekti

## RT slike i RT objekti

### □ RT slike

- Opisuju RT veličinu po vrijednosti i po vremenu → privremeno valjane

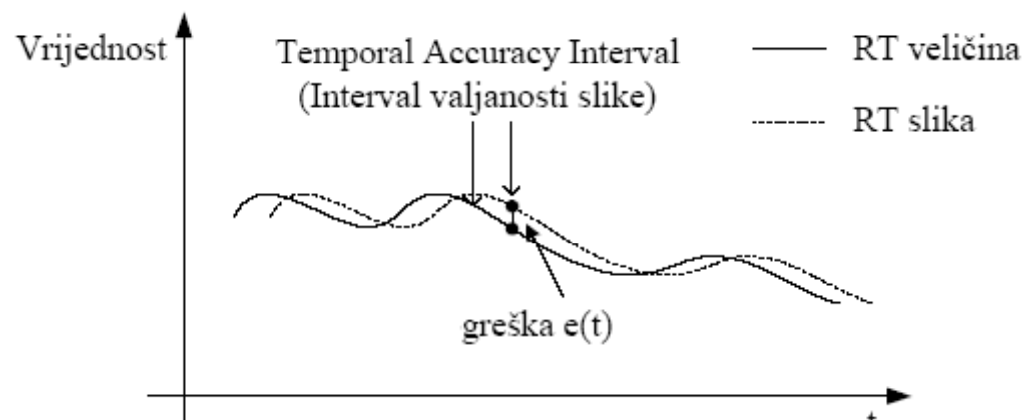
### □ RT objekti → spremnik RT veličina

- sinhroni RT objekti → aktiviraju se pomoću kloka
- distribuirani RTO → kopije na čvorovima → problem konzistentnosti

### □ Vremenska valjanost RT slike

- Definirana je u odnosu na određeni vremenski interval u neposrednoj prošlosti (recent history - RH). U trenutku  $t_i$   $RH_i$  je poredani skup vremenskih tačaka  $\{t_i, t_{i-1}, t_{i-2}, \dots, t_{i-k}\}$ .
- U trenutku  $t_i$  RT slika je ispravna (valjana) ako:

$$\exists t_j \in RH_i : \text{Vrijednost(RT slike } t_i) = \text{Vrijednost(RT slike } t_j)$$



# RT veličine, slike i objekti

## RT slike i RT objekti



### Interval valjanosti slike

- Trajanje intervala valjanosti RT slike ( $d_{acc} = z_{t_i} - z_{t_i-k}$ ) ovisi o dinamičkim svojstvima RT veličine upravljano procesa.
- Greška  $e(t)$  koja nastaje zbog kašnjenja između očitavanja (observacija) RT mjerne veličine  $v$  i korištenja njene RT slike može se približno odrediti prema izrazu:

$$e(t) = \frac{dv}{dt} [z(t_u) - z(t_o)]$$

$t_u$  – trenutak korištenja RT slike

$t_o$  – trenutak očitavanja RT veličine

- Ako se koristi valjanja RT slika, najveća greška iznosi:

$$e_{\max}(t) = \left( \frac{dv}{dt} \right)_{\max} \cdot d_{acc}$$

$e(t)_{\max}$  – treba biti u redu veličine mjerne greške mjernog člana za RT veličine koje se brzo mijenjaju, interval  $d_{acc}$  mora biti kratak.

- Da bi rezultat bio tačan mora biti:

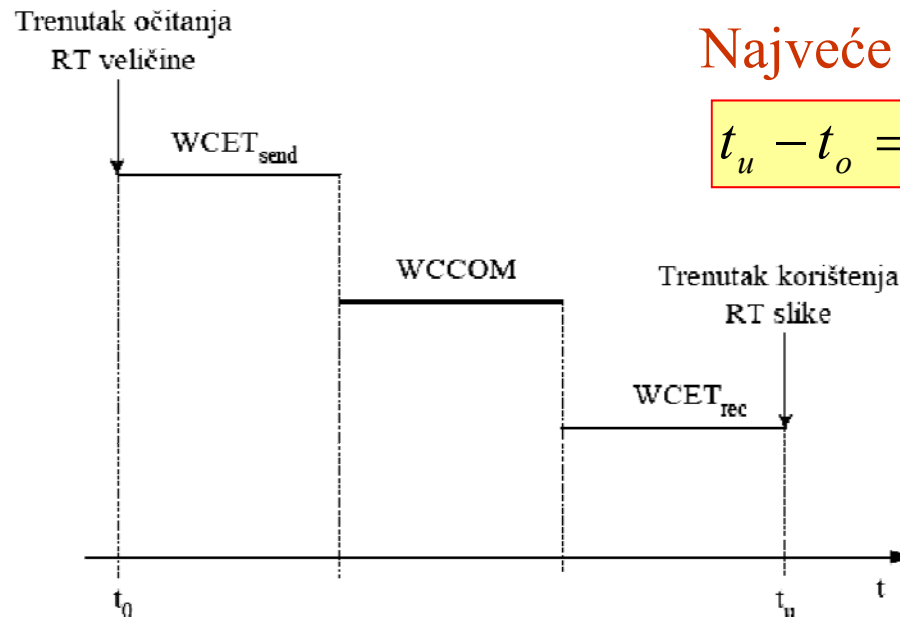
$$z(t_o) \leq z(t_u) \leq [z(t_o + d_{acc})], \text{ odnosno } z(t_u) - z(t_o) \leq d_{acc}$$



# RT veličine, slike i objekti

## RT slike i RT objekti

- ❑ **Sinhronizirane faze prijenosa (Phase-Aligned Transaction)**
- ❑ Razmatra se slučaj RT prijenosa koji se sastoji od tri sinhronizacijske faze:
  1. proračun na čvoru koji očitava RT veličinu sa maksimalnim trajanjem  $WCET_{send}$  (Worst Case Execution Time sender);
  2. prijenos očitane vrijednosti do čvora koji koristi RT sliku, sa maksimalnim trajanjem prijenosa  $WCCOM$  (Worst Case Communication delay);
  3. proračun na prijemnom čvoru sa maksimalnim trajanjem  $WCET_{rec}$  (Worst Case Execution Time receiver).



**Najveće kašnjenje (najgori slučaj):**

$$t_u - t_o = WCET_{send} + WCCOM + WCET_{rec}$$

# RT veličine, slike i objekti

## RT slike i RT objekti

### □ Klasifikacija RT slika

#### ➤ **Fazno neosjetljive RT slike** (*FNORT* slike)

Neka se RT slika osvježava sa periodom  $d_{update}$ . Ako je

$$d_{acc} > d_{update} + WCET_{send} + WCCOM + WCET_{rec}$$

tada je RT slika fazno neosjetljiva, tj. RT slika se može koristiti u bilo kojem trenutku, bez potrebe razmatranja fazne ovisnosti između dolazećeg očitavanja RT veličine i tačke njenog korištenja.

#### ➤ **Fazno osjetljive RT slike** (*FORT* slike)

$$d_{acc} \leq d_{update} + WCET_{send} + WCCOM + WCET_{rec}$$
$$\& \quad d_{acc} > WCET_{send} + WCCOM + WCET_{rec}$$

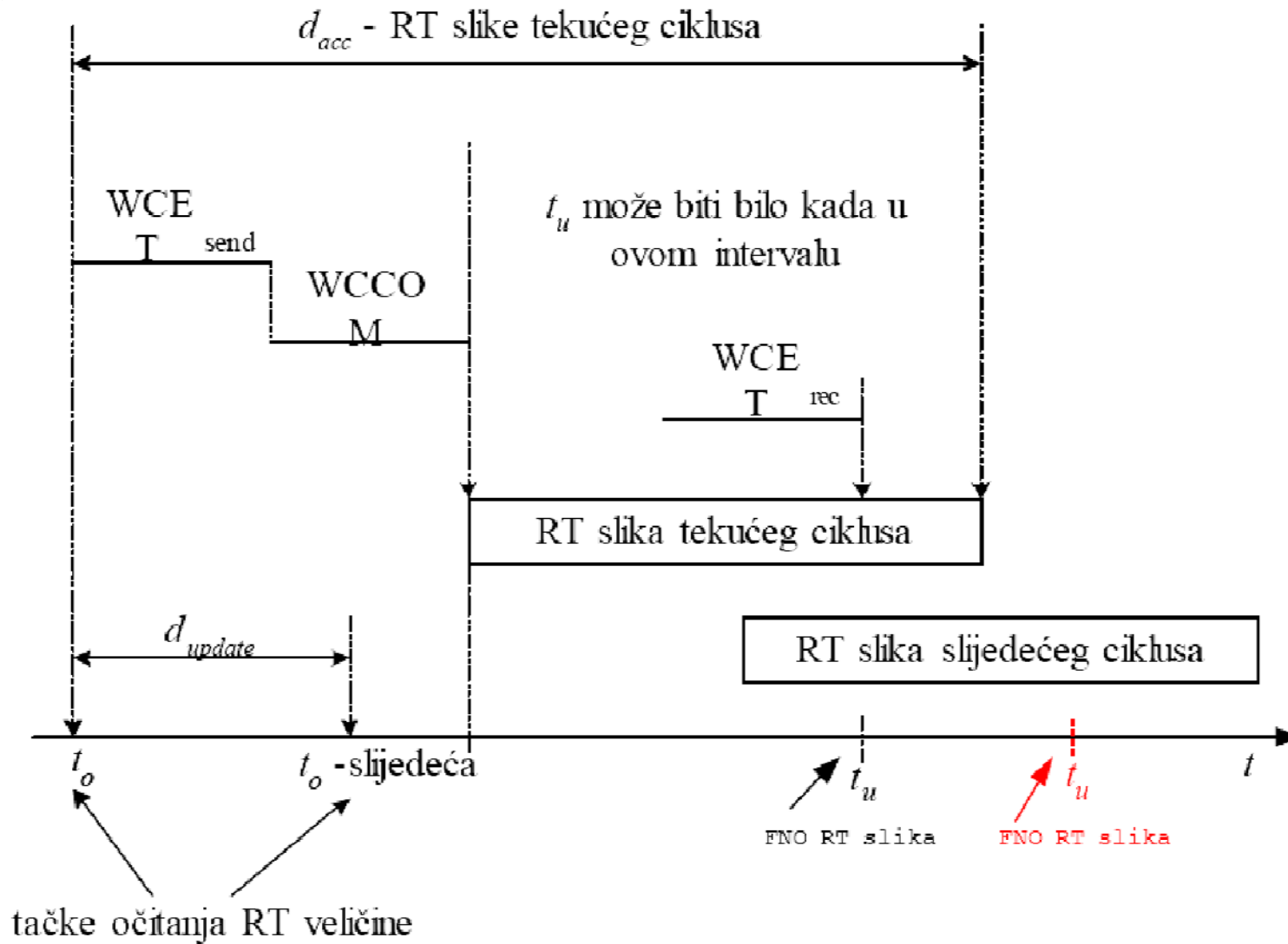
- Mora se voditi računa o faznom pomaku između trenutaka  $t_o$  i  $t_u$ .
- Znatno složenije raspoređivanje taskova da se ne poremeti redoslijed korištenja RT slika.
- Da se izbjegnu FORT slike treba: povećati frekvenciju očitavanja RT veličina (dodatno opterećenje komunikacijskog sistema) ili primijeniti SE (State Estimation) tehniku (dodatno opterećenje procesora).



# RT veličine, slike i objekti

## RT slike i RT objekti

### □ Klasifikacija RT slika



# RT veličine, slike i objekti

## RT slike i RT objekti

### □ Kašnjenje akcije (Action Delay)

- Interval između trenutka početka prijenosa poruke i trenutka kada ona postane postojana u odredišnom čvoru. Odredišni čvor mora odgoditi svaku akciju utemeljenu na primljenoj poruci dok ova ne postane postojana.
- Poruka postaje postojana u čvoru u vremenskom trenutku kada čvor zna da su sve njemu ranije poslana poruke stigle.

#### ➤ Iznos kašnjenja akcije

- Sistemi sa globalnom vremenskom bazom (sinhronizirani klokovci):

$$t_p = t_{send} + d_{max} + 2g; \quad g\text{-granularnost globalnog vremena}$$

- Sistemi bez globalnog vremena (nesinhronizirani klokovci):

$$t_p = t_{send} + 2d_{max} - d_{min} + g_l; \quad g_l\text{- granularnost lokalnog vremena}$$

- prijemnik ne zna kada je poruka poslana pa mora čekati  $(d_{max} - d_{min})$  nakon što je poruka stigla.

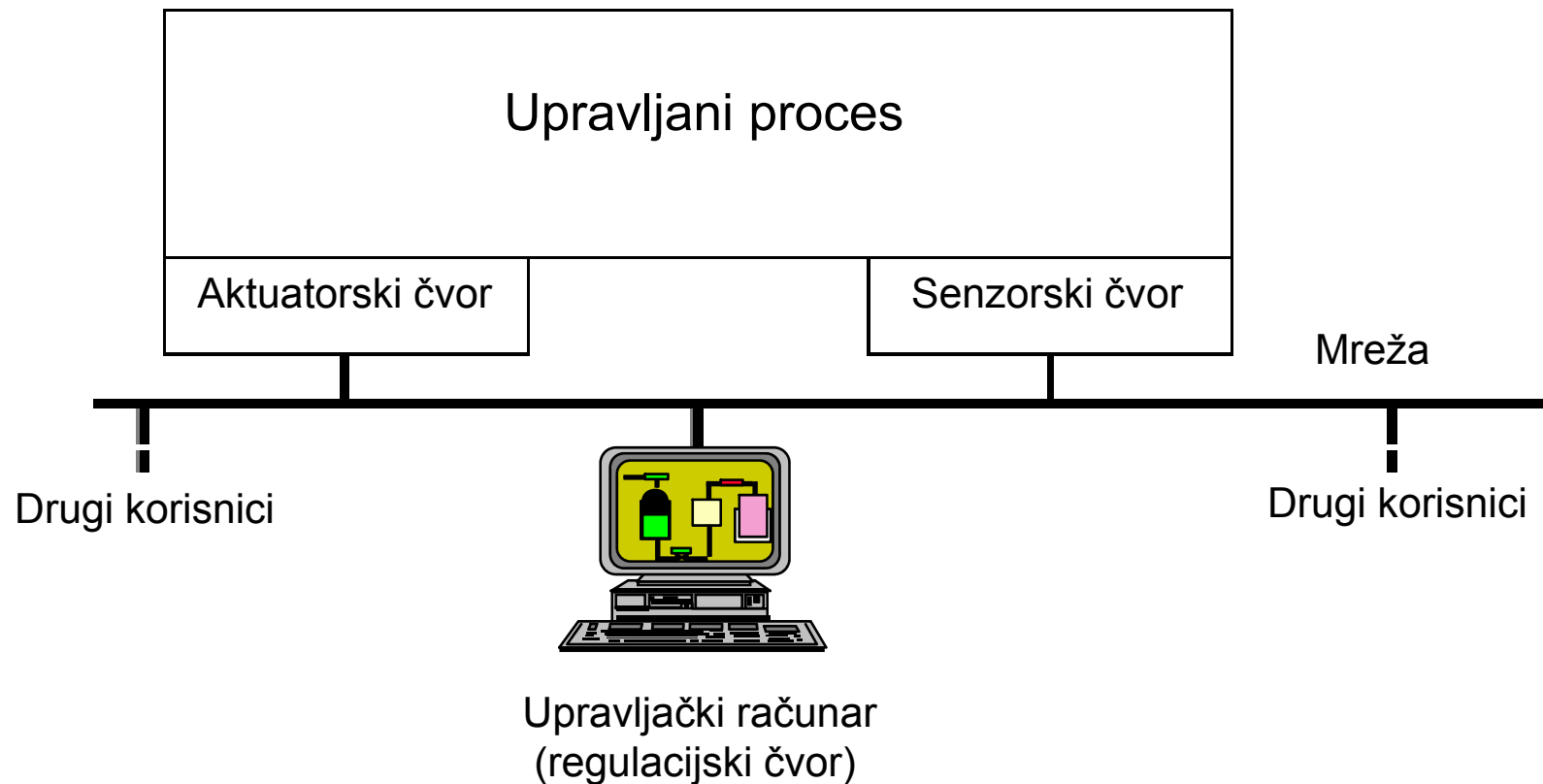
$d_{max} - d_{min} + g_l > 2g \Rightarrow$  sistemi bez globalnog vremena sporiji.



## 10.3. ZSU preko komunikacijske mreže

### Struktura ZSU preko kom. mreže

- ❑ U literaturi se za ove sisteme upravljanja koriste i sljedeći nazivi:
  - Networked Control Systems (NCS)
  - Integrated Communication and Control Systems (ICCS).



## ZSU preko komunikacijske mreže

### Zašto upravljati preko komunikacijske mreže?

#### □ Prednosti:

- **Modularnost:** decentralizirano upravljanje, distribuirano procesiranje;
- **Integrirana dijagnostika:** brzo i jednostavno održavanje, otpornost na kvarna stanja;
- **Niska cijena:** smanjenje ožičenja, razvoj sistema korak-po-korak.

#### □ Nedostaci:

- **Komunikacijska kašnjenja** zbog:
  - ograničene propusne moći komunikacijske mreže,
  - preopterećenja čvorova i mreže;
- **Ova su kašnjenja obično stohastički promjenjiva** → sistem upravljanja postaje vremenski promjenjiv;
- **Standardna teorija LTI sistema postaje neprimjenjiva.**



# ZSU preko komunikacijske mreže

## Upravljačke komunikacijske mreže

### □ Zahtjevi na upravljačke mreže:

- prijenos malih paketa između relativno velikog broja čvorova, ali sa visokom frekvencijom prijenosa kojom se zadovoljavaju zahtjevi za rad u striktno stvarnom vremenu;
- ograničeno vremensko kašnjenje;
- zagarantirana isporuka paketa (nema praznih intervala niti intervala sa više istih poruka);
- visoko iskorišćenje mreže.

Razvijeno je više algoritama pristupu mediju sa ciljem ispunjenja gornjih zahtjeva.

### □ Algoritmi pristupa mediju (MAC):

- **Stohastički pristup:**
  - Ethernet (IEEE 802.3: **CSMA/CD**)
  - CAN (**CSMA/BA**) ⇒ DeviceNet, SDS.



# ZSU preko komunikacijske mreže

## Upravljačke komunikacijske mreže

### □ Algoritmi pristupa mediju (MAC):

#### ▪ Deterministički pristup:

- Prosljeđivanje zaloga (Token bus - IEEE 802.4, Token ring – IEEE 802.5) ⇒ PROFIBUS FMS, MAP, ControlNet,...
- Centralizirana arbitraža (polling) ⇒ PROFIBUS DP, FIP, TTPA,...
- TDMA ⇒ TTP/C, FlexRay.

### □ Zadatak:

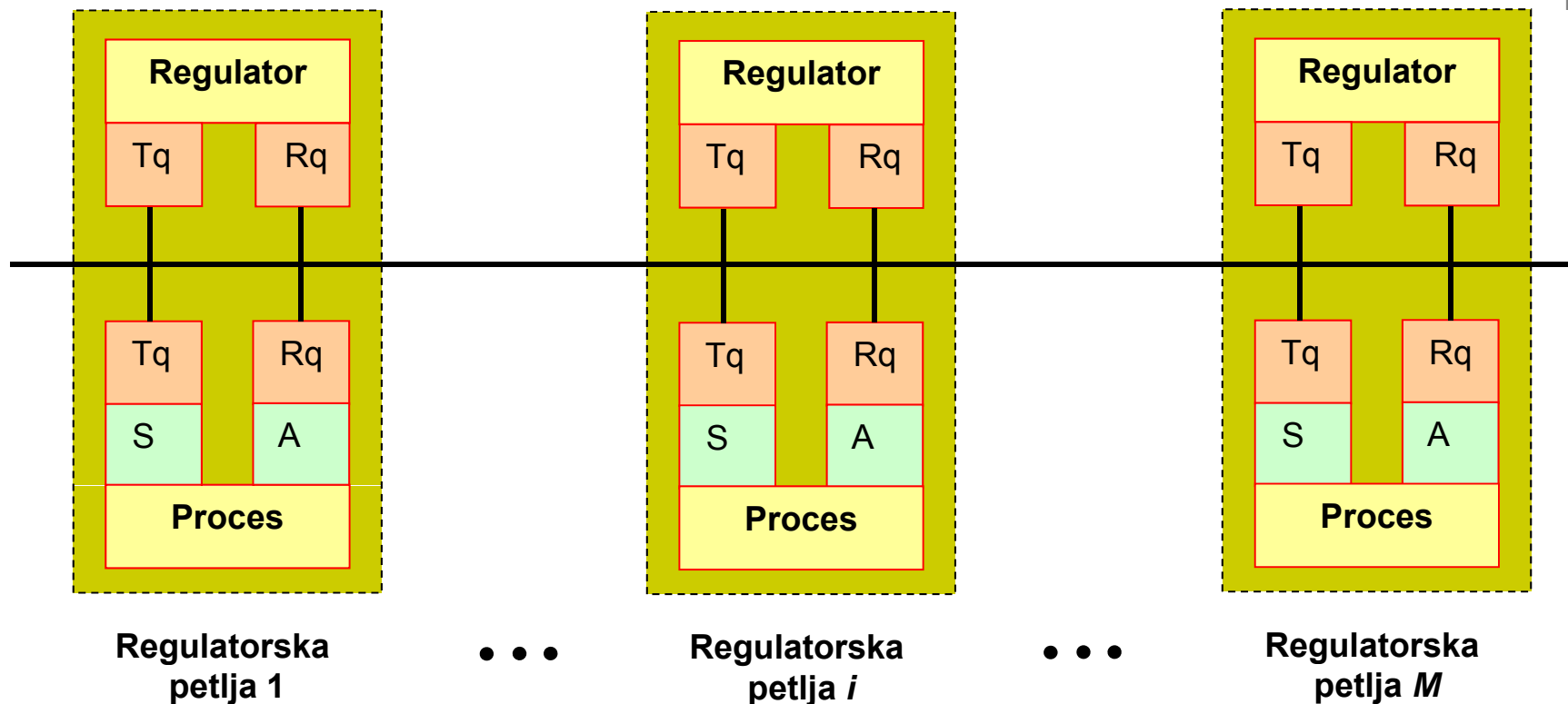
- Pronaći algoritam za određivanje trenutaka slanja poruka (pristupa mediju) čvorova na mreži, koji su ujedno i elementi zatvorenih regulacijskih krugova.
- Pri tome se moraju zadovoljiti unaprijed navedeni zahtjevi na upravljačke komunikacijske mreže.
- U nastavku se izvodi algoritam za određivanje trenutaka slanja poruka u mrežama sa prosljeđivanjem zaloga, a zatim se prilagođava i za CAN mreže.





# 10.4. Vrijeme uzorkovanja više regulacijskih petlji

## Opis problema



Legenda:

Tq - čvor predajnika, Rq - čvor prijemnika, S - senzor, A - aktuator

# Vrijeme uzorkovanja više regulacijskih petlji

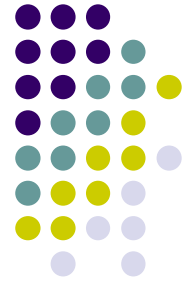
## Opis problema

### □ Pretpostavke

- Ukupan broj mrežnih čvorova koji šalju poruke  $N=2M$ .
- Senzorski i regulatorski čvor  $i$ -te regulatorske petlje imaju jednaka vremena uzorkovanja iznosa  $T_i$ .
- Svi čvorovi generiraju poruke (pakete) jednakih i fiksnih duljina  $\bar{L}$ .
- Brzina prijenosa podataka mrežom iznosi  $B$  [bit/s], pa je vrijeme prijenosa paketa  $L=\bar{L}/B$ .

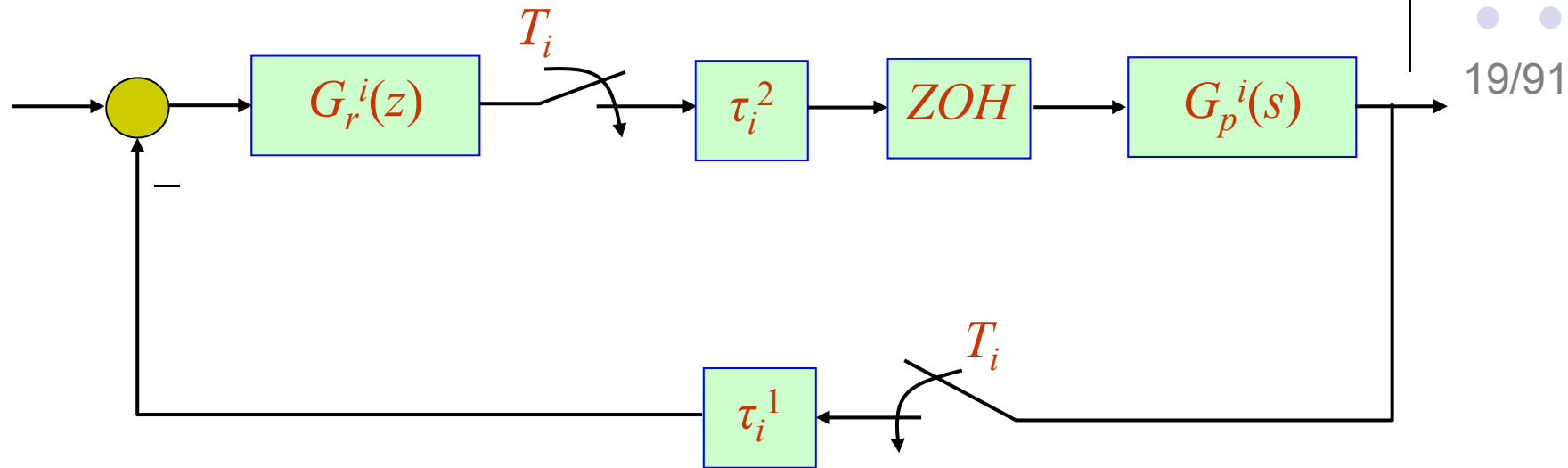
### Napomena:

- Čvorovi mogu imati različite duljine poruka. U tom se slučaju poruke razlažu (segmentiraju) na pakete duljine  $\bar{L}$ .
- Na primjer poruka duljine  $L'(>\bar{L})$  koja se uzorkuje sa periodom  $T$  razlaže se u  $p=[L'/\bar{L}]$ , a zatim se svaki paket uključuje u red čekanja za prijenos sa periodom  $T/p$ . Čvor prijemnik slaže primljene pakete u poruku.



# Vrijeme uzorkovanja više regulacijskih petlji

## Komunikacijska kašnjenja u $i$ -toj regulacijskoj petlji



$G_r^i(z)$  – Prijenosna funkcija regulatora  
 $ZOH$  – Ekstrapolator nultog reda

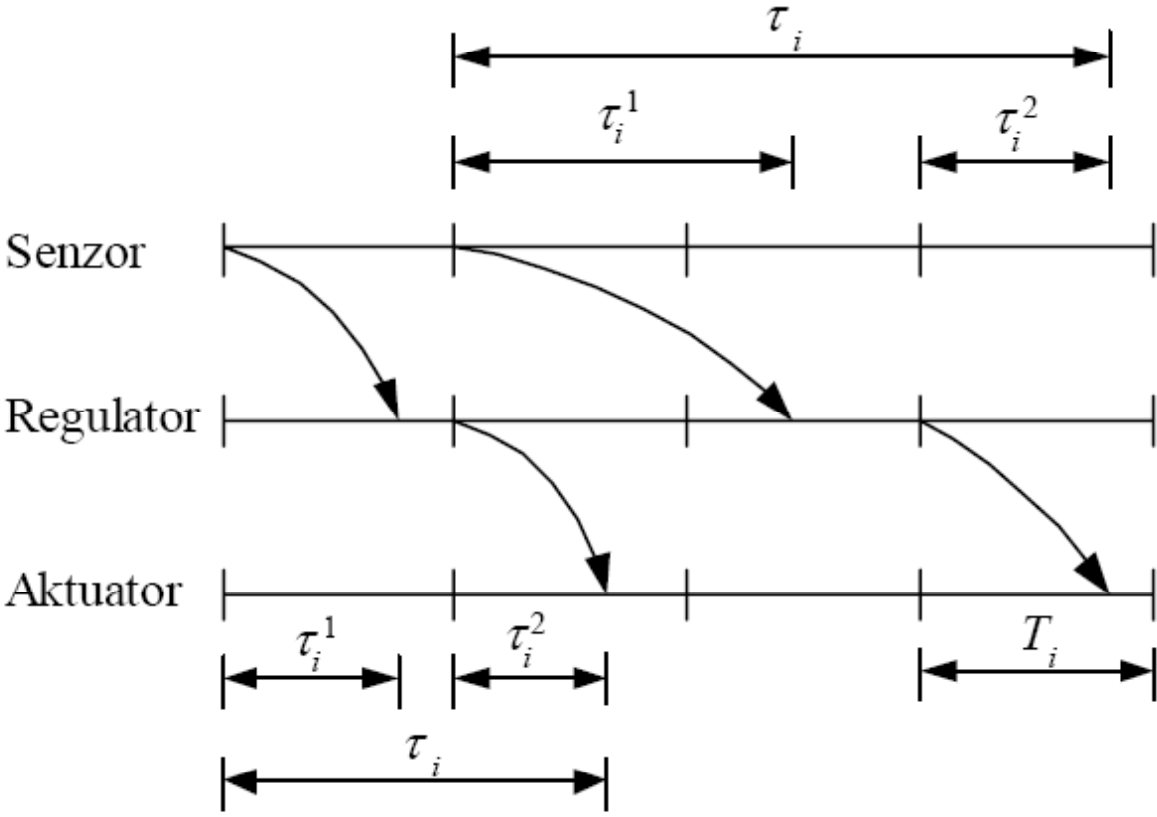
$G_p^i(s)$  – Prijenosna funkcija procesa  
 $T_i$  – Vrijeme uzorkovanja  $i$ -te regulacijske petlje.

- Uzorkovani podaci u čvorovima senzora i aktuatora moraju čekati u redu za slanje dok dobiju pravo pristupa mediju tj. dok stanica dobije zalog kod mreža za prosljeđivanje zaloga ili dok bude prozvana kod “master-slave” mreža. Ovo uzrokuje kašnjenje u regulacijskom krugu:

- $\tau_i^1$  - kašnjenje između senzora i regulatora.
- $\tau_i^2$  - kašnjenje između regulatora i aktuatora.

# Vrijeme uzorkovanja više regulacijskih petlji

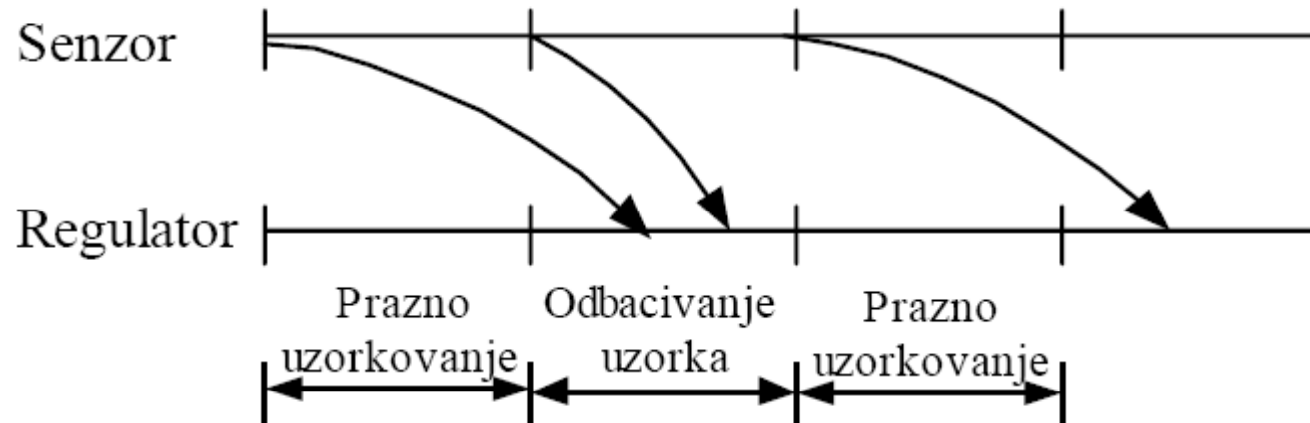
## Ilustracija kašnjenja u regulacijskoj petlji $\tau_i$



$$\tau_i = \left[ \frac{\tau_i^1}{T_i} \right] T_i + \tau_i^2$$

# Vrijeme uzorkovanja više regulacijskih petlji

## Odbacivanje uzoraka i praznih intervala uzorkovanja



21/91

- ❑ Odbacivanje uzoraka i prazni intervali uzorkovanja narušavaju kvalitet ponašanja sistema upravljanja i dovode do distorzije upravljačkog signala.
- ❑ Distorzija upravljačkog signala uzrokuje visokofrekvencijski šum u aktuatoru što može dovesti do njegovog trošenja i oštećenja.
- ❑ Dakle, svaki mjerni podatak mora stići od senzorskog do regulatorskog čvora prije sljedećeg uzorkovanja mjernog signala.



## Vrijeme uzorkovanja više regulacijskih petlji

### Dopušteno kašnjenje u regulacijskoj petlji

- ❑ Neka su  $\Phi_i, i=1$  do  $M$ , unaprijed zadana najveća dopuštena kašnjenja u  $i$ -toj regulacijskoj petlji. Tada  $i$ -ta petlja mora zadovoljiti sljedeća dva uvjeta:

- a) Spriječiti odbacivanja podataka i prazna uzorkovanja:  $\tau_i^m < T_i, i=1,2$
- b) Ograničiti kašnjenja u regulacijskim petljama:  $\tau_i < \Phi_i$ . (1)

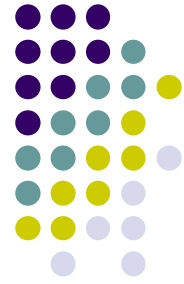
- ❑ Cilj je odrediti periode uzorkovanja  $T_i, i=1$  do  $M$ , koji zadovoljavaju zahtjeve na kvalitet sistema upravljanja, ali i osigurati visoku iskoristivost komunikacijske mreže.
- ❑ Ako podaci ne stignu do čvora prijemnika unutar vremenskog intervala  $T_i$ , pojavio se “**preljev**”, tj. čvor nije uspio poslati, jer nije došao na red.
- ❑ Algoritam određivanja trenutaka slanja poruka je ispravan ako:
  - **nema “preljeva” niti na jednom čvoru u mreži te,**
  - **ako su kašnjenja u svim regulacijskim petljama manja od dopuštenih.**

tj. ako su zadovoljeni uvjeti a) i b).

## Vrijeme uzorkovanja više regulacijskih petlji

### Modificirano konstantno kašnjenje u regulacijskoj petlji

- ❑ Kašnjenje u petlji (1) je vremenski promjenjivo jer je  $\tau_i^2$  vremenski promjenjivo.
- ❑ Zamjena  $\tau_i^2$  njegovom maksimalnom vrijednošću nije dobro rješenje.
- ❑ Neka je  $t_k$   $k$ -ti trenutak kada upravljački signal stiže do procesa. Budući da je  $\tau_i$  vremenski promjenjiv, to je i interval  $[t_k, t_{k+1})$  promjenjiv.
- ❑ Dakle, na ponašanje sistema upravljanja utječe vremenska promjenjivost intervala  $[t_k, t_{k+1})$ , a ne samo promjenjivo komunikacijsko kašnjenje.
- ❑ Interval  $[t_k, t_{k+1})$  postaje najdulji kada se  $\min \tau_i^2$  pojavi u  $k$ -om dolasku, a  $\max \tau_i^2$  u  $(k+1)$ -vom dolasku.
- ❑ Dakle, ako se vremenski promjenjivo kašnjenje  $\tau_i^2$  modificira u konstantno kašnjenje preko  $\sup[t_k, t_{k+1}) = T_i + (\sup \tau_i^2 - \min \tau_i^2)$ , zahtjevi na kvalitet ponašanja sistema upravljanja sa promjenjivim komunikacijskim kašnjenjem bit će zadovoljeni.





## Vrijeme uzorkovanja više regulacijskih petlji

### Modificirano konstantno kašnjenje u regulacijskoj petlji

- ❑ Uz kašnjenje od regulatora do aktuatora, treba uzeti u obzir i kašnjenje od senzora do regulatora.
- ❑ Ova dva kašnjenja mogu se promatrati zajednički, ako se ne pojavljuju prazni intervali uzorkovanja i ako ne dolazi do odbacivanja uzoraka, tj. ako je ispunjen uvjet a).
- ❑ Konačno modificirano kašnjenje u u petlji  $\tau_i'$  postaje:

$$\tau_i' = 2T_i + (\sup \tau_i^2 - \min \tau_i^2) \quad (2)$$

- ❑ Opravdanost primjene modificiranog konstantnog kašnjenja provjerena je simulacijom na sistemu prvog reda.

$$\dot{x}(t) + \alpha x(t) + \beta x(t - \tau(t)) = 0 \quad (3)$$

gdje je  $\tau(t)$  promjenjivo vremensko kašnjenje.

- ❑ Područja stabilnosti za modificirano konstantno kašnjenje i za promjenjivo kašnjenje u ovisnosti o promjeni parametara  $\alpha$  i  $\beta$  ( $T_i=0.5$  s,  $\tau(t)=\text{rand}[0\text{s}, 0.5\text{s}]$ ) prikazana su na sljedećoj slici:

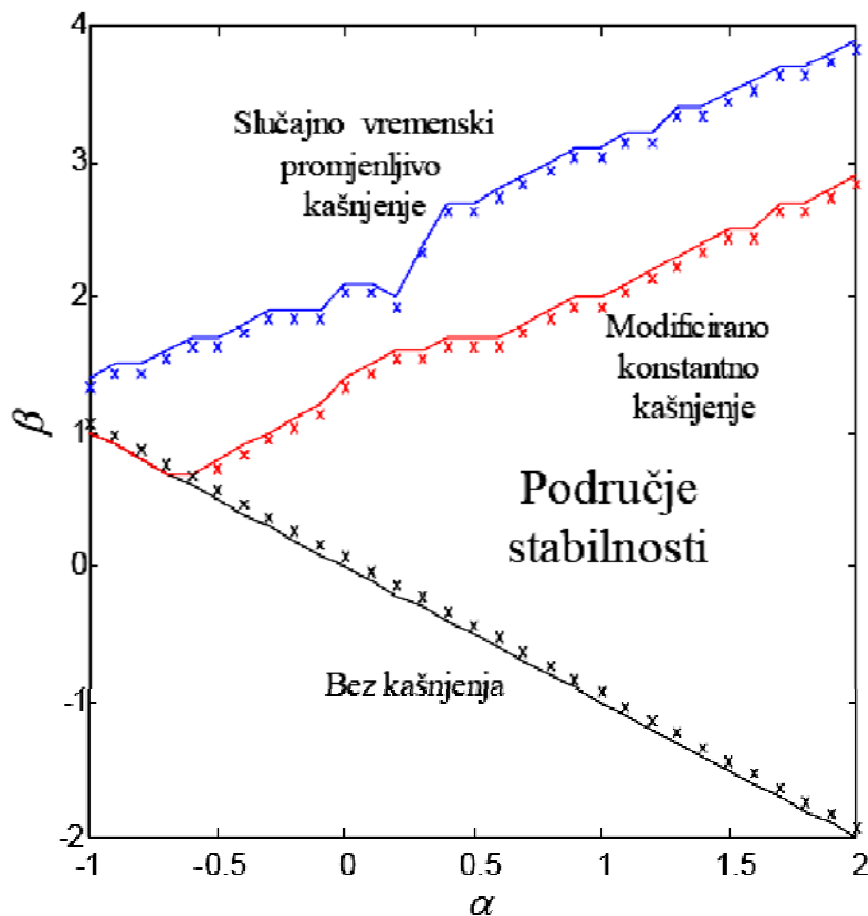


# Vrijeme uzorkovanja više regulacijskih petlji

## Modificirano konstantno kašnjenje u regulacijskoj petlji



25/91



◆ Područje stabilnosti za modificirano konstantno kašnjenje obuhvaćeno je područjem stabilnosti za promjenjivo kašnjenje, što potvrđuje da je moguće projektirati ICCS na temelju modificiranog konstantnog kašnjenja.

◆ Na temelju modificiranog konstantnog kašnjenja, uvjet b)  $\tau_i < \Phi_i$  može se izraziti preko vremena uzorkovanja  $T_i$ , tj.

$$\tau'_i = 2T_i + (\sup \tau_i^2 - \min \tau_i^2) \leq \Phi_i \quad (4)$$

◆ Budući da je  $\tau'_i$  konstantno, maksimalno dopušteno kašnjenje u  $i$ -toj regulacijskoj petlji  $\Phi_i$  može se jednostavno odrediti, npr. klasičnom analizom u frekvencijskom području (Nyquistov ili Bodeovi dijagrami).



## 10.5. Uzorkovanje na mreži sa prosljeđiv. zaloga

### Algoritam za određivanje trenutaka uzorkovanja

#### - koncept prozora

- ❑ Neka je  $\mathbf{T}$  vektor vremena uzorkovanja  $M$  petlji, tj.

$$\mathbf{T} = [T_1, T_2, \dots, T_M] \quad (5)$$

gdje su  $T_i$  poredani od najmanjeg do najvećeg, tj.  $T_i \leq T_{i+1}, \forall i$ .

- ❑ Neka je  $r_i$  broj podataka koji se mogu prenijeti medijem pri najvećem kašnjenju u petlji  $i$ .
- ❑ Najveći  $\tau_i^m$  se pojavljuje kad čvor  $m$  ( $m=1$  i  $2$  označavaju senzorski, odnosno regulatorski čvor) u petlji  $i$  izgubi zalog neposredno prije nego uzorkuje (pošalje) podatak.
- ❑ Zalog će ciklički biti pridružen svim ostalim čvorovima prije nego ponovo dođe do čvora  $m$ . Prema tome, u najgorem slučaju  $\tau_i^m$  postaje:

$$\max \tau_i^m = r_i L + N\sigma, \quad m = 1, 2 \quad (6)$$

## Uzorkovanje na mreži sa prosljeđivanjem zaloga

### Algoritam za određivanje trenutaka uzorkovanja

gdje je  $\sigma$  dodatno maksimalno vrijeme potrebno za obradu i prostiranje zaloga.

- Iz a)  $\tau_i^m < T_i$  slijedi da se  $r_i$  mora ograničiti tako da vrijedi:

$$r_i = \left\lfloor \frac{T_i - N\sigma}{L} \right\rfloor \quad (7)$$

- Neka je  $r$  jednako  $r_i$  za odgovarajuće odabrano  $T_1$  (najkraće vrijeme uzorkovanja), tj.

$$r_i = \left\lfloor \frac{T_i - N\sigma}{L} \right\rfloor \Rightarrow T_1 \geq rL + N\sigma \quad (8)$$

- Dakle  $T_1$  se sastoji od  $r$  prozora duljine jednake vremenu prijenosa paketa  $L$ .

⇒ **KONCEPT PROZORA** (vidi sljedeću sliku).

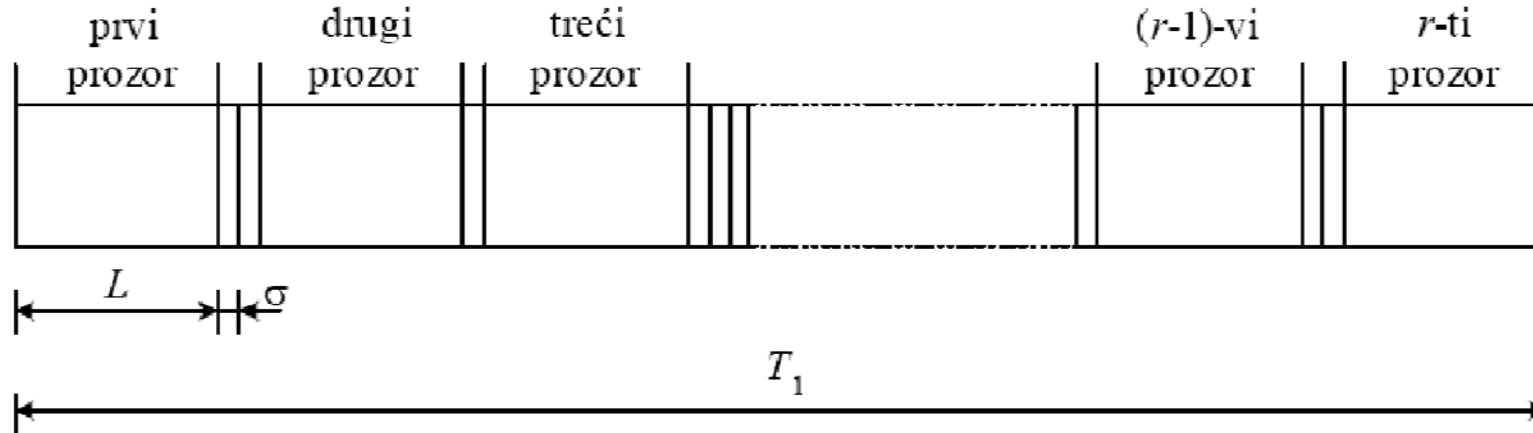


# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Algoritam za određivanje trenutaka uzorkovanja



28/91



- ❑ Algoritam za određivanje vremena uzorkovanja zasniva se na **konceptu prozora** gdje u svakom intervalu  $T_1$ ,  $N$  čvorova u mreži dijeli  $r$  prozora ( $r < N$ ), tako da broj paketa koji se prenose u bilo kojem intervalu  $T_1$  ne prijeđe iznos  $r$ .
- ❑ Razmotrimo proizvoljan čvor  $j$ , gdje je  $T_j = k_j T_1$  ( $k_j \geq 1$ ). Ako se ne šalje više od  $r$  paketa unutar bilo kojeg intervala  $T_1$ , tada broj paketa koji će biti preneseni u intervalu  $T_j$  neće prijeći iznos  $r_j = k_j r$ . Dakle, čvor  $j$  nikad neće doživjeti “preljev”.

# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Algoritam za određivanje trenutaka uzorkovanja

- ❑ Najkraći  $\tau_1^2$  se pojavljuje kada čvor 2 (regulator) u petlji 1 uzorkuje (pošalje) njegov podataka neposredno prije nego izgubi zalog, tj.  $\min \tau_1^2 = L$ .
- ❑ Iz a) ( $\tau_i^m < T_i, m=1,2$ ) i b) ( $\tau_i < \Phi_i$ ) proizlazi  $\sup \tau_1^2$  jednako  $T_1$ , odnosno  $\tau_i' = \Phi_1 (= \min[\Phi_i, i=1 \text{ do } M])$ .
- ❑ Odakle se uz zadano  $\Phi_1, T_1$  izračuna prema izrazu (4) ( $\tau_i' = 2T_i + \sup \tau_1^2 + \min \tau_1^2 < \Phi_i$ ) kao

$$T_1 = \frac{\Phi_1 + L}{3} \quad (9)$$

- ❑ Broj prozora  $r$  u  $T_1$  prema (7) ( $r = \left\lceil \frac{T_1 - N\sigma}{L} \right\rceil$ ) iznosi:

$$r = \left\lceil \frac{(\Phi_1 + L)/3 - N\sigma}{L} \right\rceil \quad (10)$$

## Uzorkovanje na mreži sa prosljeđivanjem zaloga

### Algoritam za određivanje trenutaka uzorkovanja

- Uvedimo vektor  $\mathbf{K}$  koji sadrži omjere vremena uzorkovanja u odnosu na najkraće vrijeme uzorkovanja  $T_1$ :

$$\mathbf{K} = [k_1, k_2, \dots, k_M], \quad k_i = \frac{T_i}{T_1}, \quad k_i \leq k_{i+1}, \forall i \quad (11)$$

- Razmotrimo proizvoljni čvor  $j$  sa vremenom uzorkovanja  $T_j$ , gdje je  $T_j < T_M$ . Ako je  $\text{Rem}[k_j, k_i] = 0, j > i$ , broj paketa koje čvor  $j$  šalje za vrijeme intervala  $T_M$  (najveće vrijeme uzorkovanja koje se sastoji od  $k_M$  intervala  $T_1$ ) je fiksni cjelobrojnik iznosa  $k_M/k_j$ .
- Ponavljajući ovo za  $j=1$  do  $N$ , dobije se ukupni (cjelobrojni) broj paketa koje svi čvorovi pošalju unutar intervala  $T_M$ :

$$k_M \sum_{j=1}^N (1/k_j) = 2k_M \sum_{i=1}^M (1/k_i) = k_M \alpha_K \quad (12)$$

gdje je  $\alpha_K$  prosječan broj paketa koji se pošalju u intervalu  $T_1$ :

$$\alpha_K = 2 \sum_{i=1}^M (1/k_i) \quad (13)$$



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Algoritam za određivanje trenutaka uzorkovanja

- ❑ Ako je  $\alpha_K \leq r$ ,  $\text{Rem}[k_j, k_i] = 0$  za  $\forall i$  i  $j, j \geq i$ , tada će svi podaci uzorkovani u intervalu  $T_M$  moći biti raspoređeni u raspoložive prozore unutar istog intervala  $T_M$ .



### Ovo je osnovni kriterij ispravnosti rada algoritma.

- ❑ Ako je  $\alpha_K > r$ , kapacitet mreže nije dovoljan za postavljene zahtjeve, tj. komunikacijski sistem je preopterećen.
- ❑ U tom slučaju, projektant ICCS-a mora:
  - primijeniti mrežu većeg prijenosnog kapaciteta ( $B$ ) i manjeg dodatnog opterećenja ( $\sigma$ ), ili
  - smanjiti broj čvorova ( $N$ ).



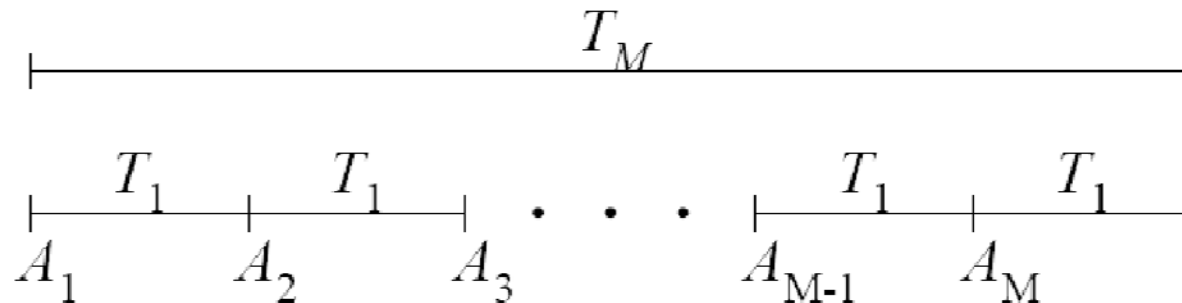
# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Određivanje trenutka prvog uzorkovanja za svaki čvor



32/91

- ❑ Neka su  $t_j, j=1$  do  $N$ , trenuci uzorkovanja za čvor  $j$  u  $T_M$ .
- ❑ Neka je  $u^n(A_l)$  broj uzorkovanih (poslanih) podataka od čvora  $1$  do čvora  $N$  u  $A_l$ , i  $A_l$  je početni trenutak  $l$ -tog intervala  $T_1$  unutar intervala  $T_M$ .



- ❑ Za čvor  $1$  (koji ima najmanje vrijeme uzorkovanja), pridruži  $t_1=A_1$ .
- ❑ Budući da čvor  $1$  uzorkuje podatke sa periodom  $T_1$ :

$$u^1(A_l) = 1, \quad l = 1 \text{ do } k_M$$

- ❑ Za ostale čvorove pridruži:

$$t_j = \inf[A_l \geq A_{l-1} : u^j(A_l) \leq r], \quad j = 2 \text{ do } N$$



## Uzorkovanje na mreži sa prosljeđivanjem zaloga

### Određivanje trenutka prvog uzorkovanja za svaki čvor

- Kako je  $\text{Rem}[k_j, k_i]=0$ , vrijeme uzorkovanja čvora  $j$  je cjelobrojni umnožak čvora  $j-1$ , pa vrijedi:

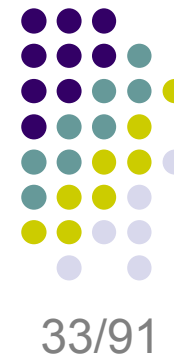
$$u^j(A_l + nT_j) \leq r, \quad \forall n = 1, 2, \dots \quad (14)$$

- Nijedan čvor neće doživjeti “preljev” jer je broj podataka koji se uzorkuju u bilo kojem trenutku  $A_l$  ograničen na  $r$ , tj.

$$u^N(A_l) \leq r, \quad \forall l = 1 \text{ do } k_M \quad (15)$$

### Napomena:

- Budući da je vrijeme prijenosa paketa svakog čvora  $L$ , striktni uvjet istovremenog uzorkovanja više distribuiranih čvorova u trenutku  $A_l$  u (15) može se ostvariti uz uvjet da je maksimalna razlika trenutaka uzorkovanja ovih čvorova manja od  $L$ .
- Ovo se može jednostavno obaviti periodičkim slanjem sinhronizacijske poruke od čvora zaduženog za provođenje sinhronizacije.



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Određivanje $k_i$ za $i=2$ do $M$

- Najviše  $r$  paketa može biti poslano za vrijeme bilo kojeg intervala  $T_1$ . Dakle,  $\sup \tau_j^m = T_1$  i  $\min \tau_j^m = L$ ,  $\forall i$  i  $m$ .
- Iz (4) ( $\tau_i' = 2Ti + (\sup \tau_i^2 - \min \tau_i^2) \leq \Phi_i$ ), za zadani  $\Phi_i$ , uvjet b) ( $\tau_i \leq \Phi_i$ ) može se pisati kao

$$2T_i + (T_1 - L) \leq \Phi_i \quad (16)$$

- Iz (16) i uvjeta 1)  $T_i = k_i T_1$ ,  $i=1$  do  $M$  i 2)  $\text{Rem}[k_j, k_i] = 0$  za  $\forall i$  i  $j, j \geq i$ , dobije se

$$k_i = \left\langle \frac{\Phi_i - (T_1 - L)}{2T_1} \right\rangle \quad (17)$$

gdje  $y = \langle x \rangle$  znači  $y$  je potencija broja 2,  $2^{n_i}$ ,  $n_i \in \{0, 1, 2, \dots\}$  koja je najbliža broju  $x$ , ali je manja od njega.  $T_1$  je dan izrazom (7).



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Mali promet na mreži

- ❑ Ako je promet na mreži mali, tj. ako je  $r > N$ , algoritam određivanja trenutaka uzorkovanja postaje znatno jednostavniji.
- ❑ Za vrijeme bilo kojeg intervala  $T_1$ , najviše jedan podatak se šalje sa čvora  $j$ , jer je  $T_j > T_i, j=2$  do  $N$ .
- ❑ Budući da je maksimalni broj podataka koji se mogu uzorkovati u  $T_1$  jednak  $N$  nema opasnosti od pojave “preljeva” niti u jednom  $T_j$ .
- ❑ Algoritam će raditi ispravno ako se vremena uzorkovanja  $T_i, i=1$  do  $M$  izaberu tako da se zadovolji uvjet kvalitete ponašanja sistema regulacije dan uvjetom 2) ( $\tau_i \leq \Phi_i$ ), tj.

$$T_i = \frac{\Phi_i - (T_1 - L)}{2}, \quad \forall i = 1 \text{ do } M$$

(18)



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Iskorištenost mreže

- ❑ Iskorištenost mreže  $U$  definira se kao dio ukupnog vremena unutar kojeg mrežom putuju paketi:

$$U = \sum_{j=1}^N \frac{L}{T_j} = \frac{2L}{T_1} \sum_{i=1}^M \frac{1}{k_i}$$

(19)

- ❑  $U$  se može povećati smanjenjem  $k_i$  (ili  $T_i$ ). Ovim se smanjenjem ne narušava uvjet b) ( $\tau_i \leq \Phi_i$ ) ali je smanjenje ograničeno iznosom  $\tau_i^m$  danim uvjetom a) ( $\tau_i < \Phi_i, m=1,2$ ).
- ❑ Povećanje broja petlji  $M$  (tj. broja čvorova  $N$ ) također povećava  $U$ . Ovo se povećanje mora ograničiti tako da  $\alpha_K$  ne postane veće od  $r$ , inače će mreža postati preopterećena.
- ❑ Dakle, optimalna je situacija kada je  $U$  maksimalno, a da se pri tome ne narušavaju uvjeti a) i b), što se postiže ako  $\alpha_K \rightarrow r$ .



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Ilustrativni primjer – 5 regulacijskih petlji zatvorenih preko TP mreže

Zadano je:

- ❑  $M=5; N=10$ .
- ❑ Čvorovi pridruženi petljama: čvorovi (1,2); (3,4); (5,6); (7,8); (9,10) pridruženi *petljama* 1,2,3,4,5, respektivno.
- ❑  $L=2$  ms;  $\sigma=0.1$  ms.
- ❑ Prijenosne funkcije petlje 1:

Proces: 
$$G_p(s) = \frac{1}{(1 + 0.3s)(1 + 0.03s)}$$

Regulator: 
$$G_r(s) = \frac{35(1 + 0.2s)}{0.2s}$$

- ❑ Dopuštena kašnjenja u regulacijskim petljama:

$$\Phi = [\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5] = [25 \text{ ms}, 60 \text{ ms}, 100 \text{ ms}, 200 \text{ ms}, 400 \text{ ms}]$$

# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Primjena algoritma

**Korak 1:** Odredi  $\Phi_1$ ,  $T_1$  i  $r$

$$\Phi_1 = \min[\Phi_i, i=1 \text{ do } M]$$

$$\Rightarrow \Phi_1 = 25 \text{ ms}$$

$$T_1 = \frac{\Phi_1 + L}{3}$$

$$\Rightarrow T_1 = \frac{25 + 2}{3} = 9 \text{ ms}$$

$$r = \left\lceil \frac{T_1 - N\sigma}{L} \right\rceil$$

$$\Rightarrow r = \left\lceil \frac{9 - 10 \cdot 0.1}{2} \right\rceil = 4$$

**Korak 2:** Odredi  $T_i$

Ako je  $r \geq N$  (lagani promet),

$$\Rightarrow r = 4 < N = 10$$

tada

$$T_i = \frac{\Phi_i - (T_1 - L)}{2}, \quad \forall i = 1 \text{ do } 10$$

inače



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Primjena algoritma

$$k_i = \left\langle \frac{\Phi_i - (T_1 - L)}{2T_1} \right\rangle, \quad \forall i = 2 \text{ do } M$$

$$k_2 = \left\langle \frac{60 - (9 - 2)}{2 \cdot 9} \right\rangle = \langle 2.94 \rangle \Rightarrow, k_2 = 2^1 = 2$$

$$k_3 = \left\langle \frac{100 - (9 - 2)}{2 \cdot 9} \right\rangle = \langle 5.17 \rangle \Rightarrow, k_3 = 2^2 = 4$$

$$k_4 = \left\langle \frac{200 - (9 - 2)}{2 \cdot 9} \right\rangle = \langle 10.72 \rangle \Rightarrow, k_4 = 2^3 = 8$$

$$k_5 = \left\langle \frac{400 - (9 - 2)}{2 \cdot 9} \right\rangle = \langle 21.83 \rangle \Rightarrow, k_5 = 2^4 = 16$$



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Primjena algoritma

Provjeri je li sistem preopterećen

$$\alpha_K = 2 \sum_{i=1}^M \frac{1}{k_i} \Rightarrow \alpha_K = 2 \left( \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \right) = 3.875$$

$$\text{ako } \alpha_K > r, \Rightarrow \alpha_K = 3.875 < r = 4$$

tada (mreža je preopterećena  $\Rightarrow$  smanji  $N$  i idi na korak 1)

$$\text{inače } T_i = k_i T_1 \Rightarrow$$

$$T_2 = 18 \text{ ms}; T_3 = 36 \text{ ms}; T_4 = 72 \text{ ms}; T_5 = 144 \text{ ms}.$$

**Korak 3:** Odredi  $t_j$  (prvi trenutak uzorkovanja za čvor  $j$ )

$$t_1 = A_1 = 0$$

Za ( $j=2, l=1; j \leq N, l \leq k_M; j=j+1, l=l+1$ )

$$t_j = \inf[A_l \geq A_{l-1} : w(A_l) \leq r]$$

$$w(A_l + nT_j) \leq r, \forall n=1, 2, \dots$$





# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Primjena algoritma



Petlja 1 $T_1=9\text{ms}$	$A_l=A_1=0\text{ms}$	$u^1(A_1)=1 < r=4 \Rightarrow t_1=A_1$ $u^2(A_1)=2 < r=4 \Rightarrow t_2=A_1$	
Petlja 2 $T_2=18\text{ms}$	$A_l=A_1=0\text{ms}$	$u^3(A_1)=3 < r=4 \Rightarrow t_3=A_1$ $u^4(A_1)=4 = r=4 \Rightarrow t_4=A_1$	Ne može više nijedan čvor započeti u $A_1=0\text{ms}$
Petlja 3 $T_3=36\text{ms}$	$A_l=A_2=9\text{ms}$	$u^5(A_2)=3 < r=4 \Rightarrow t_5=A_2$ $u^6(A_2)=4 = r=4 \Rightarrow t_6=A_2$	$u^2(A_1+nT_1), \forall n=1,2,..$ $\Rightarrow u^2(A_2=A_1+T_1)=2$
Petlja 4 $T_4=72\text{ms}$	$A_l=A_3=18\text{ms}$	Ne može više nijedan čvor započeti u $A_3=18\text{ms}$	$u^2(A_1+nT_1), \forall n=1,2,..$ $\Rightarrow u^2(A_3=A_1+2T_1)=2$ $u^2(A_1+nT_2), \forall n=1,2,..$ $\Rightarrow u^2(A_3=A_1+T_2)=4$
	$A_l=A_4=27\text{ms}$	$u^7(A_4)=3 < r=4 \Rightarrow t_7=A_4$ $u^8(A_4)=4 = r=4 \Rightarrow t_8=A_4$	$u^2(A_1+nT_1), \forall n=1,2,..$ $\Rightarrow u^2(A_4=A_1+3T_1)=2$

# Uzorkovanje na mreži sa prosljeđivanjem zaloga

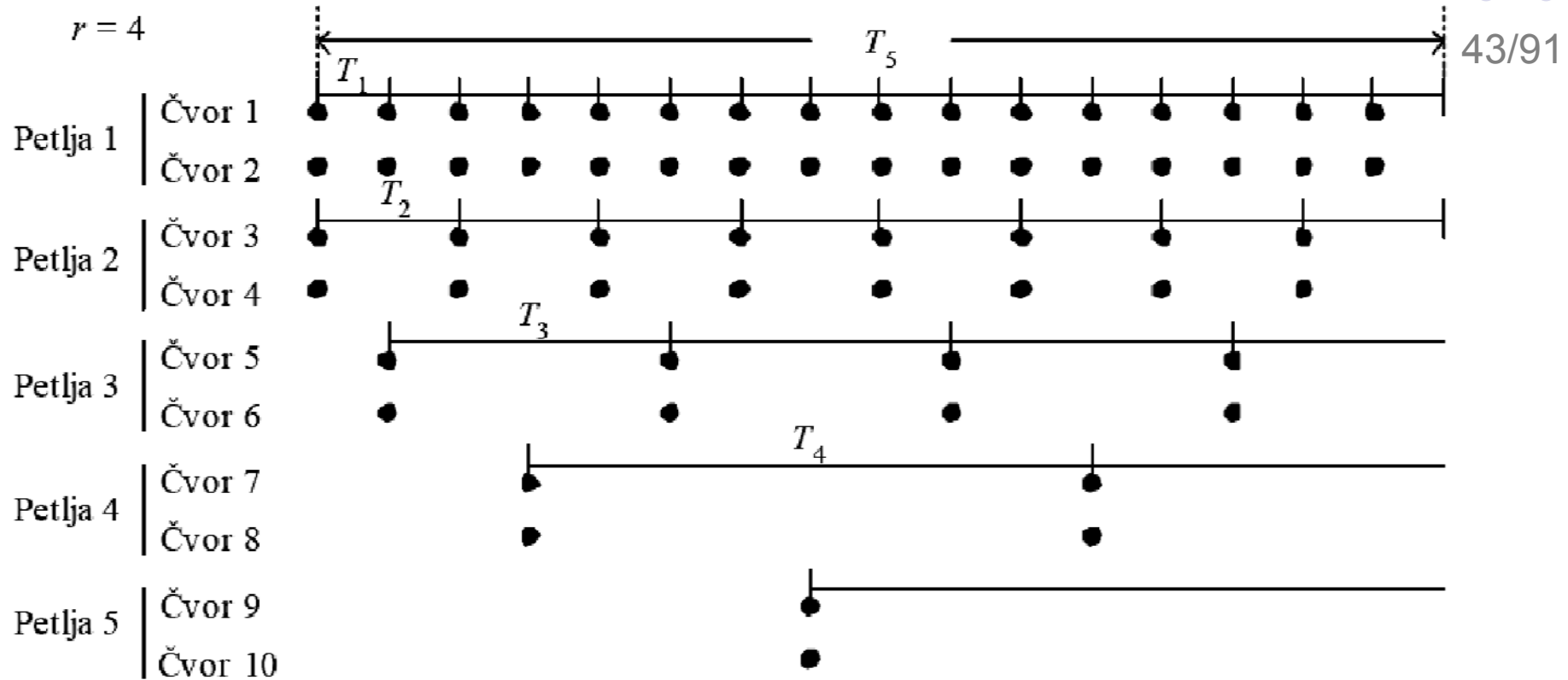
## Primjena algoritma



Petlja 5 $T_4=144\text{ms}$	$A_l=A_5=36\text{ms}$	Ne može više nijedan čvor započeti u $A_5=36\text{ms}$	$u^2(A_1+nT_1), \forall n=1,2,..$ $\Rightarrow u^2(A_5=A_1+4T_1)=2$ $u^2(A_1+nT_2), \forall n=1,2,..$ $\Rightarrow u^2(A_5=A_1+2T_2)=4$
	$A_l=A_6=45\text{ms}$	Ne može više nijedan čvor započeti u $A_6=45\text{ms}$	$u^2(A_1+nT_1), \forall n=1,2,..$ $\Rightarrow u^2(A_6=A_1+5T_1)=2$ $u^2(A_2+nT_2), \forall n=1,2,..$ $\Rightarrow u^2(A_6=A_2+T_3)=4$
	$A_l=A_7=54\text{ms}$	Ne može više nijedan čvor započeti u $A_7=54\text{ms}$	$u^2(A_1+nT_1), \forall n=1,2,..$ $\Rightarrow u^2(A_7=A_1+6T_1)=2$ $u^2(A_1+nT_2), \forall n=1,2,..$ $\Rightarrow u^2(A_7=A_1+3T_2)=4$
	$A_l=A_8=63\text{ms}$	$u^9(A_8)=3 < r=4 \Rightarrow t_9=A_8$ $u^{10}(A_8)=4 = r=4 \Rightarrow t_{10}=A_8$	$u^2(A_1+nT_1), \forall n=1,2,..$ $\Rightarrow u^2(A_8=A_1+7T_1)=2$

# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Primjena algoritma



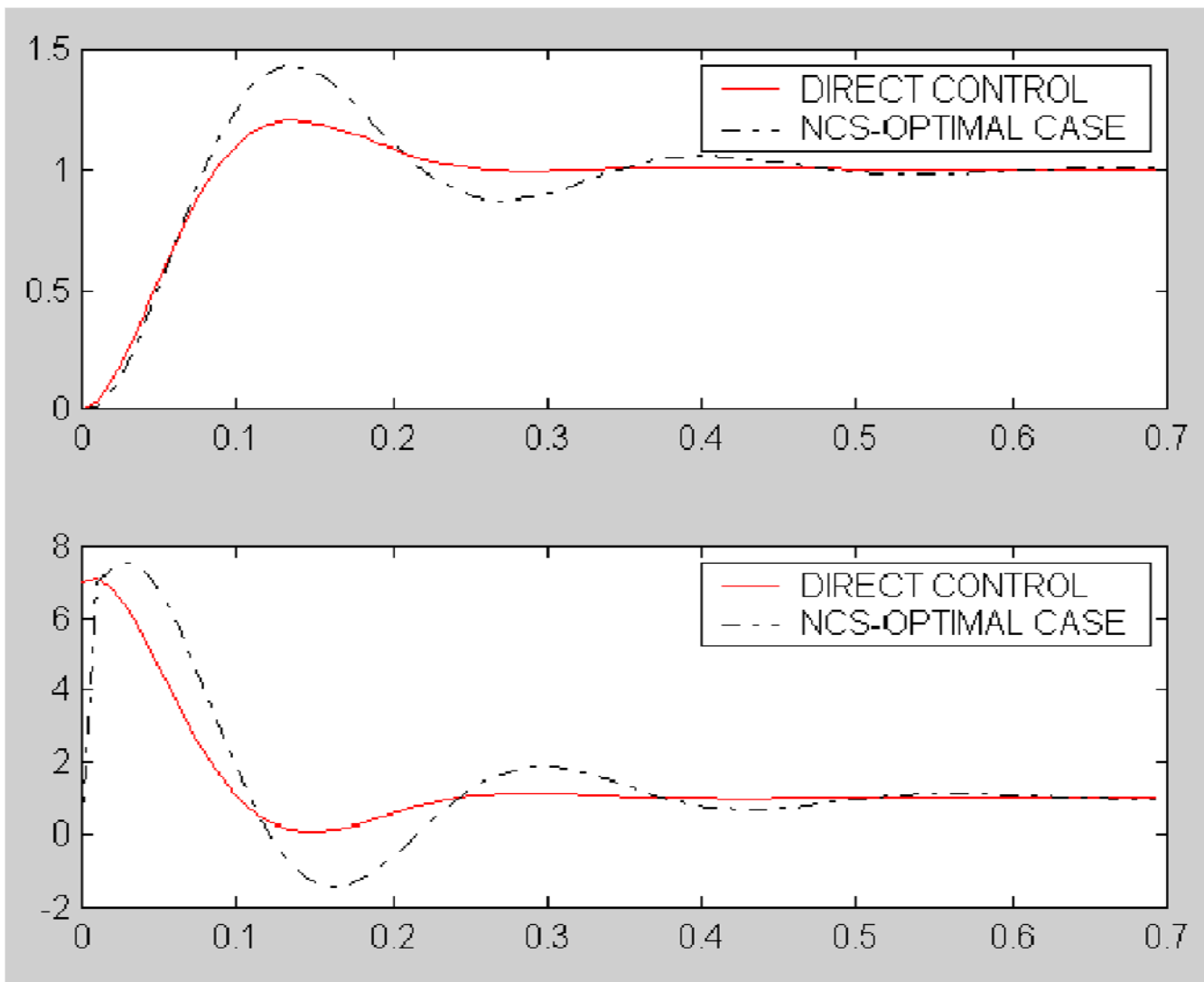
Iskorištenost mreže:

$$U = \sum_{j=1}^N \frac{L}{T_j} = \frac{2L}{T_1} \sum_{i=1}^M \frac{1}{k_i}, \quad U = 86.1\%$$

# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Kvalitet ponašanja regulacijske petlje 1

Optimalni slučaj:  $T_1=9\text{ms}$ ;  $T_2=18\text{ms}$ ;  $T_3=36\text{ms}$ ;  $T_4=72\text{ms}$ ;  $T_5=144\text{ms}$ ;



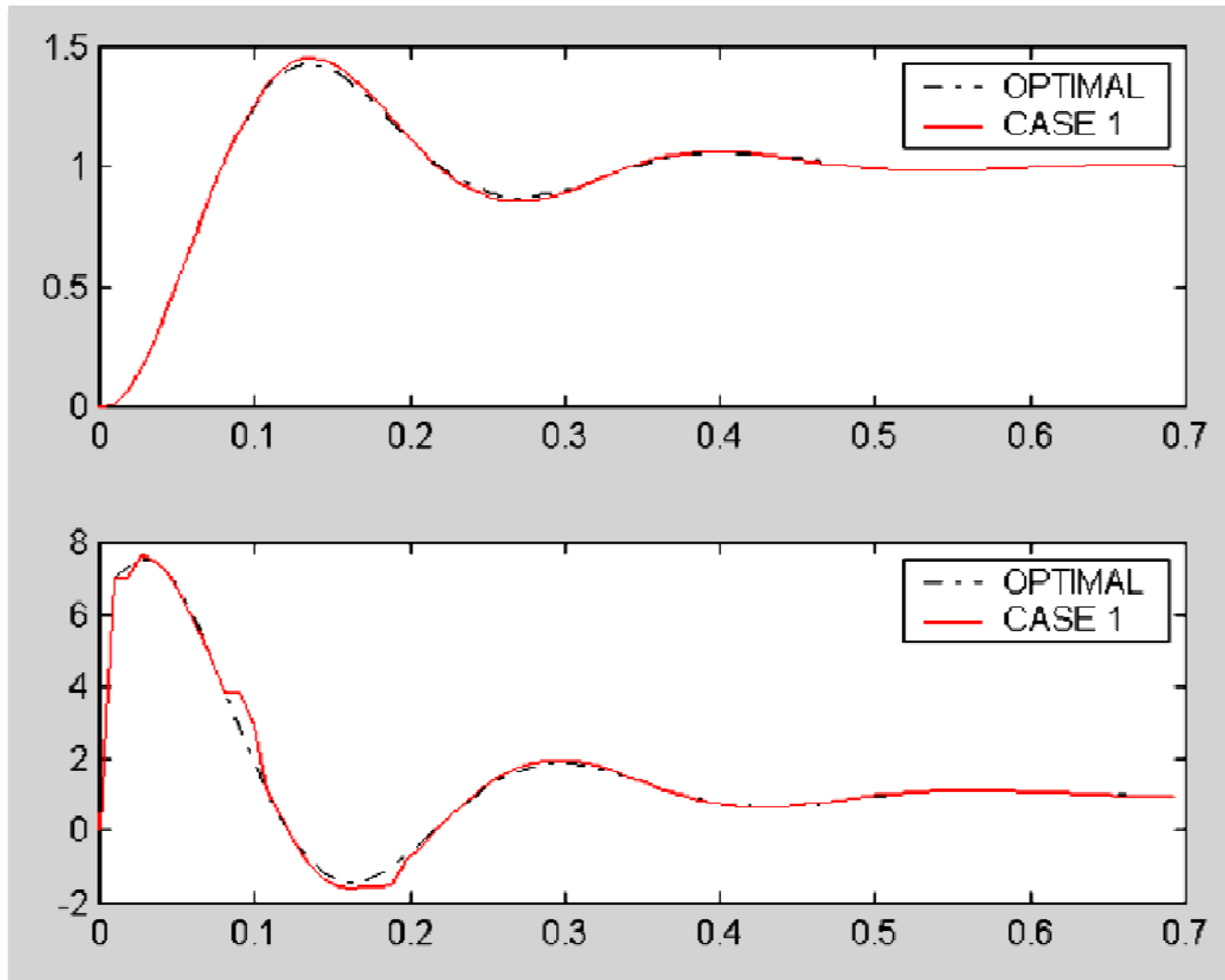
# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Kvalitet ponašanja regulacijske petlje 1

**Slučaj 1:**  $T_1=9\text{ms}$ ;  $T_2=16\text{ms}$ ;  $T_3=40\text{ms}$ ;  $T_4=85\text{ms}$ ;  $T_5=170\text{ms}$ ;  
 $U=86.5\%$  - iskorištenost mreže je gotovo jednaka kao i u optimalnom slučaju.



45/91



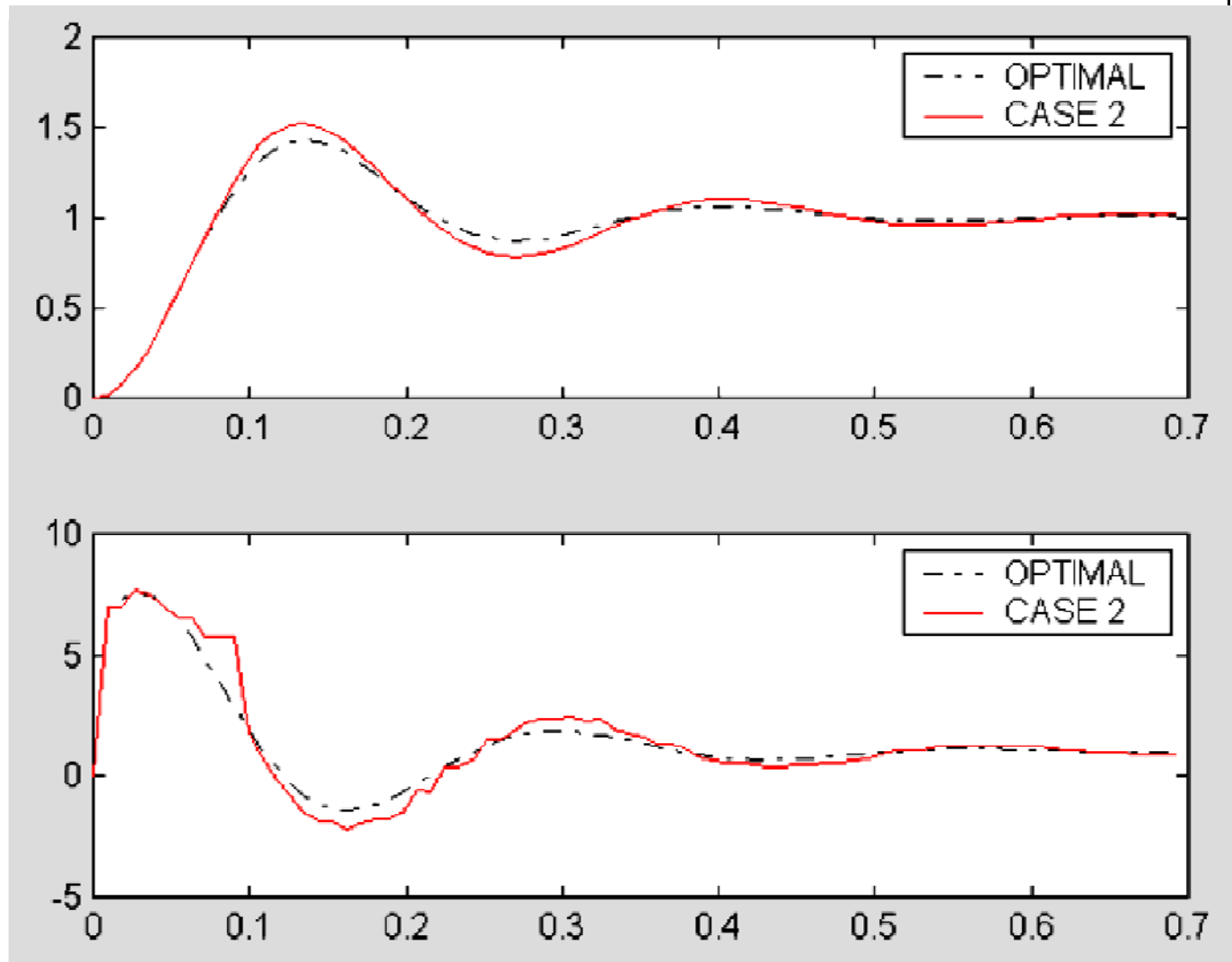
# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Kvalitet ponašanja regulacijske petlje 1

**Slučaj 2:**  $T_1=9\text{ms}$ ;  $T_2=9\text{ms}$ ;  $T_3=18\text{ms}$ ;  $T_4=36\text{ms}$ ;  $T_5=72\text{ms}$ ;  
 $U=127.7\%$  - mreža je preopterećena.



46/91



# Uzorkovanje na mreži sa prosljeđivanjem zaloga

## Zaključak

- ❑ Svojstva zatvorene regulacijske petlje direktno ovise o komunikacijskim kašnjenjima.
- ❑ Određivanje trenutaka uzorkovanja čvorova koji sudjeluju u formiranju zatvorenih regulacijskih petlji je važno pri projektiranju ICCS-a.
- ❑ Opisani algoritam zadovoljava temeljne zahtjeve na ICCS (za mreže sa prosljeđivanjem zaloga) kao što su:
  - ograničenje kašnjenja u regulacijskim petljama unutar najviše dopuštene vrijednosti,
  - sprječavanje odbacivanja uzoraka i praznih uzorkovanja, te
  - povećanje iskorištenosti komunikacijskih resursa.
- ❑ Može li se ovaj algoritam iskoristiti u drugim komunikacijskim mrežama?
  - U TDMA i “Polling” mrežama se može direktno primijeniti, a u CAN mrežama ga je potrebno modificirati (obrađuje se u nastavku).



## 10.6. Uzorkovanje u CAN mreži

### Algoritam određivanja trenutaka uzorkovanja

- ❑ Kod izvođenja algoritma za mreže sa prosljeđivanjem zaloga pretpostavljeno je da se mrežom prenose samo periodičke upravljačke informacije, tj. mjerne veličine od senzora do regulatora i upravljački signali od regulatora prema aktuatorima.
- ❑ Podaci koji se prenose u CAN mrežama mogu se grupirati kao:
  - **RT (real-time) podaci**, kod kojih pravovremenost prijenosa igra veliku ulogu
    - Upravljački RT podaci (Control Data, URT)
    - Događajni RT podaci (Event Data, DRT)
  - **NRT (non-real-time) podaci**, kod kojih vrijeme stizanja podataka na odredište nije od ključne važnosti.
- ❑ DRT i NRT podaci se ne prenose periodički kao URT podaci, već nasumice ⇒ **potrebno je proširiti algoritam određivanja trenutaka uzorkovanja kod TP mreža.**





# Uzorkovanje u CAN mreži

## Zahtjevi za algoritam

### □ URT podaci:

- prenose se periodički, svaki čvor sa svojim vremenom uzorkovanja, čvorovi istog regulacijskog kruga imaju ista vremena uzorkovanja
- svaki podatak mora biti prenesen do prijemnika unutar vremena uzorkovanja te regulacijske petlje.

### □ DRT podaci:

- to su nadzorni, dijagnostički i alarmni podaci, tj. moraju biti preneseni sa minimalnim vremenom kašnjenja ⇒ **najviši prioritet.**
- generiraju se nasumice, neperiodički.

### □ NRT podaci:

- to su arhivski podaci, bitni za naknadno korištenje unutar analize rada sistema
- obično veća količina podataka koji nisu bitni za trenutno odvijanje rada procesa ⇒ **najniži prioritet.**

- Uz ove zahtjeve algoritam treba osigurati i maksimalno moguće iskorištenje mrežnih resursa.



# Uzorkovanje u CAN mreži

## Popis oznaka koje se koriste

- $M$  - broj regulacijski petlji spojenih na zajedničku CAN sabirnicu
- $N_c$  – broj čvorova koji generiraju URT podatke
- $N_e$  – broj čvorova koji generiraju DRT podatke
- $N_n$  – broj čvorova koji generiraju NRT podatke
- $L_c$  – vrijeme prijenosa paketa URT podataka
- $L_e$  – vrijeme prijenosa paketa DRT podataka
- $L_n$  – vrijeme prijenosa paketa NRT podataka
- $[\Phi_i, i=1-M]$  – maksimalno dopušteno kašnjenje  $i$ -te regulacijske petlje
- $[\lambda_e^i, i=1-N_e]$  – frekvencija generiranja DRT paketa  $i$ -tog čvora
- $[\lambda_n^i, i=1-N_n]$  – frekvencija generiranja NRT paketa  $i$ -tog čvora
- $[T_i, i=1-M]$  – vrijeme uzorkovanja  $i$ -te regulacijske petlje.



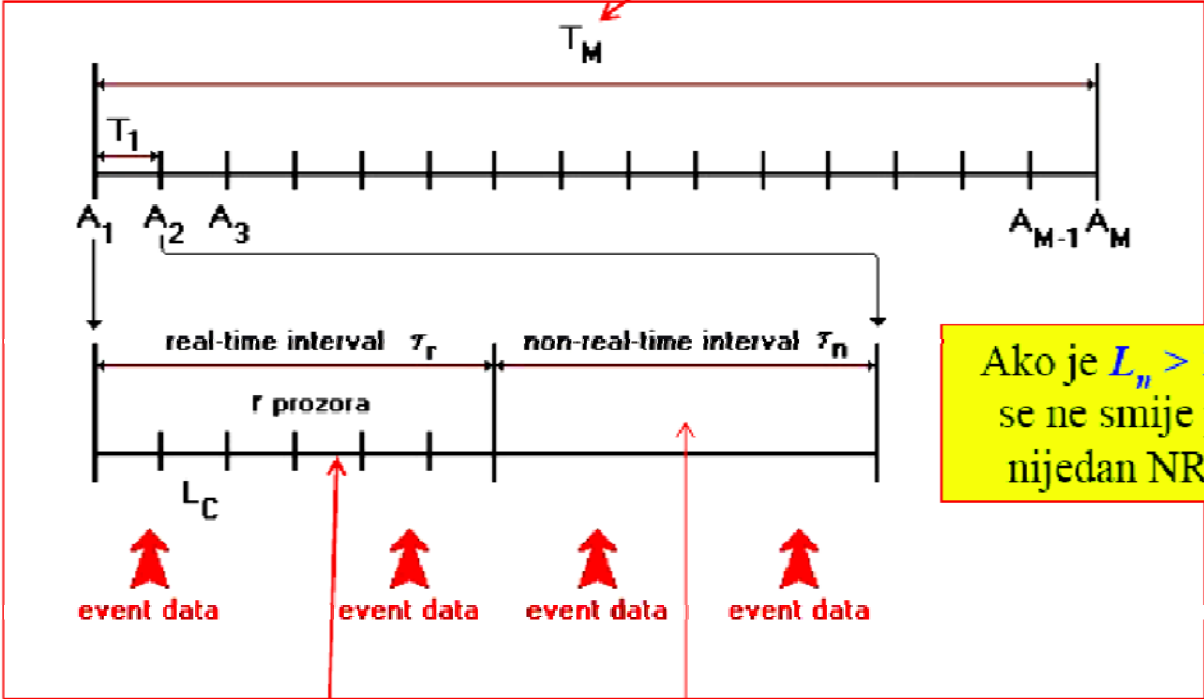
# Uzorkovanje u CAN mreži

## Koncept dodjeljivanja medija



Neophodna sinkronizacija lokalnih satova stvarnog vremena

Vrijeme uzorkovanja najsporije regulacijske petlje



Ako je  $L_n > T_1 - \tau_r$  tada se ne smije prenijeti nijedan NRT paket

Prenose se URT podaci prema algoritmu izvedenom za TP mreže

Ako se u ovom intervalu pojave URT podaci imat će veći prioritet od NRT podataka

# Uzorkovanje u CAN mreži

## Koncept dodjeljivanja kanala

- **Pretpostavke:**

- Generiranje DRT i NRT podataka može se modelirati Poissonovom razdiobom:

$$P_{\lambda}(n) = \frac{e^{-\lambda} \lambda^n}{n!}$$

$$\lambda = Np$$

$p$  – vjerojatnost događaja u jednom mjerenju ( $\ll 1$ )

$N$  – broj ponovljenih mjerenja

$\lambda$  – očekivani broj događaja

$P$  – vjerojatnost  $n$  događaja uz zadano očekivanje

- Redovi čekanja za URT podatke duljine **1 paket**, jer paket na jednom čvoru mora biti prenesen prije generiranja novog paketa na njemu.
- Redovi čekanja za DRT i NRT podatke dovoljno velike duljine tako da nikad ne dolazi do odbacivanja podataka zbog preljeva na čvoru ukoliko je promet na mreži stabilan.
- Da bi promet na mreži bio stabilan, frekvencija kojom čvorovi postavljaju zahtjeve za prijenos mora biti ograničena na mjeru koja ne prekoračuje mrežni kapacitet.



# Uzorkovanje u CAN mreži

## Stabilnost prijenosa DRT signala

- Pretpostavimo da u nekom vremenskom intervalu  $t$  na čvor  $i$  stiže neki prosječan broj DRT podataka.
- Red čekanja na čvoru  $i$  bit će stabilan ako on u intervalu  $t$  pošalje u prosjeku veći broj paketa od broja paketa pristiglih u red čekanja za prijenos. Drugim riječima, vrijeme potrebno da se svi podaci koji čekaju na prijenos pošalju mora biti manje od  $t$ .
- Prosječan broj DRT paketa generiranih u vremenu  $t$  je  $\lambda_e^i t \Rightarrow$  vrijeme potrebno za prijenos svih tih paketa je  $(\lambda_e^i t)L_e$ .
- Za čvorove  $j \neq i$ , prosječan broj je  $\lambda_e^j t$ , ako je  $\lambda_e^j t < \lambda_e^i t$ , odnosno  $\lambda_e^i t$ , ako je  $\lambda_e^j t > \lambda_e^i t$ .  $\leftarrow$  svi događajni čvorovi dobivaju pravo na prijenos jednakog broja paketa podataka.
- Red čekanja na prijenos u čvoru  $i$  bit će stabilan, ako se svi ovi paketi prenesu unutar vremena  $t \Rightarrow$  uvjet stabilnosti reda čekanja na čvoru  $i$ .



$$(\lambda_e^i t)L_e + L_e \sum_{j=1, j \neq i}^{N_e} \min[\lambda_e^i t, \lambda_e^j t] < t$$

## Uzorkovanje u CAN mreži

### Stabilnost prijenosa DRT signala

- Da bi mrežni sistem bio stabilan, potrebno je da ovakvo razmatranje vrijedi za svaki pojedini događajni čvor.
- Ako je red čekanja čvora koji generira podatke sa najvećom frekvencijom stabilan, tada slijedi da će i svi ostali redovi čekanja događajnih čvorova biti stabilni.
- Tako uvjet stabilnosti za prijenos događajnih podataka postaje:

$$L_e \sum_{j=1}^{N_e} \lambda_e^j < 1$$

(21)





## Uzorkovanje u CAN mreži

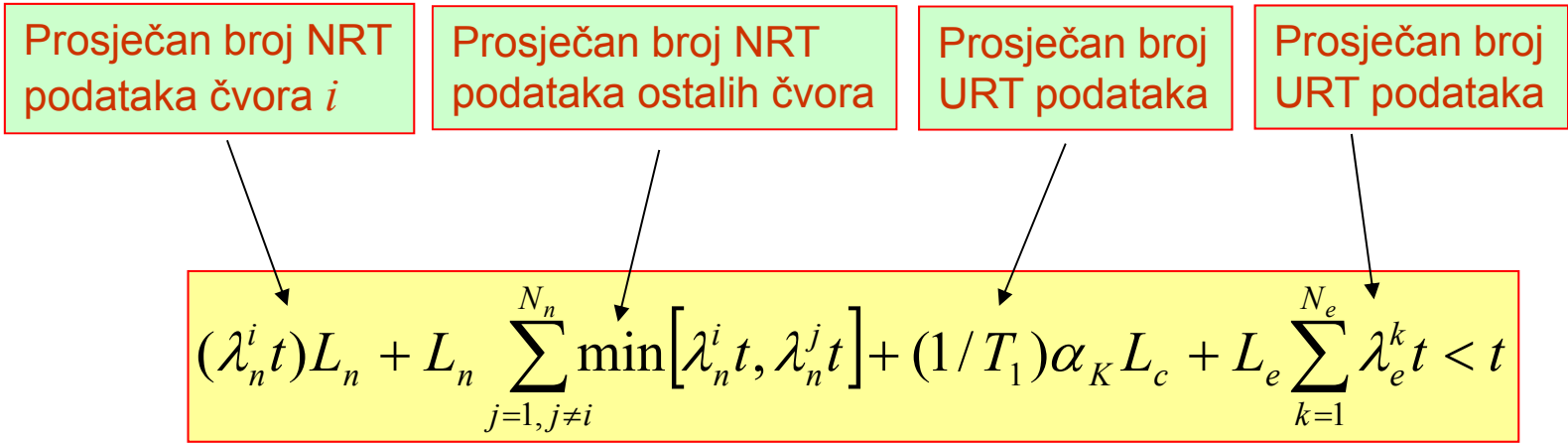
### Stabilnost prijenosa NRT signala

- NRT podaci obično su veći, pa se zato dijele u više paketa i kao takvi postavljaju u redove čekanja za prijenos. Paketi se šalju jedan za drugim te se na prijemnom čvoru ponovo sastavljaju u izvornu poruku.
- Neka je  $\lambda_n^i$  frekvencija generiranja novih NRT podataka na čvoru  $i$ , podaci se segmentiraju u  $p_i$  paketa  $\Rightarrow$  prosječan broj NRT paketa generiranih i postavljenih u red čekanja na čvoru  $i$  u jedinici vremena iznosi  $\lambda_n^i = p_i \lambda_n^i$ .
- NRT paketi na mreži imaju najmanji prioritet, pa je zato prijenos ovih paketa ovisan o pojavama zahtjeva za prijenos DRT i URT paketa  $\Rightarrow$  U analizi stabilnosti redova čekanja na NRT čvorovima moraju se uračunati i paketi koje u toj jedinici vremena generiraju RT čvorovi  $\Rightarrow$  uvjet stabilnosti reda čekanja na NRT čvoru  $i$ :

$$(\lambda_n^i t) L_n + L_n \sum_{j=1, j \neq i}^{N_n} \min[\lambda_n^i t, \lambda_n^j t] + (1/T_1) \alpha_K L_c + L_e \sum_{k=1}^{N_e} \lambda_e^k t < t \quad (22)$$

# Uzorkovanje u CAN mreži

## Stabilnost prijenosa NRT signala



- Ako je red čekanja čvora sa najvećom frekvencijom generiranja NRT podataka stabilan, tada su stabilni redovi čekanja i ostalih NRT čvorova.
- **Dakle, uvjet stabilnosti prijenosa NRT podataka je:**

$$L_n \sum_{j=1}^{N_n} \lambda_n^j + \alpha_K L_c / T_1 + L_e \sum_{k=1}^{N_e} \lambda_e^k < 1 \tag{23}$$





## Uzorkovanje u CAN mreži

### Stabilnost prijenosa URT signala

- DRT paketi mogu biti prenošeni i za vrijeme RT intervala, koji je izvorno namijenjen prijenosu URT paketa, a RT interval ne smije postati veći od  $T_1$ .  $\Rightarrow$  prevelik broj DRT paketa najvećeg prioriteta mogao bi poremetiti koncept prozora.
- U stvarnosti je broj DRT paketa na mreži znatno manji od broja URT paketa. Problem je stohastičke naravi i može se izraziti vjerojatnošću generiranja  $n$  DRT podataka tokom intervala  $T_1$ :

$$\frac{(\lambda_e^k T_1)^n}{n!} e^{-\lambda_e^k T_1}$$

(24)

- Na primjer za  $T_1=10\text{ms}$  i  $\lambda_e^i=0.01\text{ms}$ , vjerojatnost da će biti generirano više od 10 DRT paketa u vremenu  $T_1$ , iznosi svega  $7.89 \cdot 10^{-5}$ .

## Uzorkovanje u CAN mreži

### Stabilnost prijenosa URT signala

- Pretpostavimo da je dopušteno  $n_i$  DRT paketa, što se odredi prema uvjetu (24) tako da vjerojatnost generiranja većeg broja DRT paketa bude zanemarljivo mala, te da promet upravljačkih podataka na mreži prema algoritmu ne bude ugrožen.
- Ako je to ispunjeno, uvjet stabilnosti za upravljačke čvorove glasi:

$$L_c r + L_e n_i \leq T_1$$

(25)

- Kada ovaj uvjet ne bi bio zadovoljen, neki upravljački podaci bili bi odbačeni za vrijeme ciklusa  $T_1$ .



# Uzorkovanje u CAN mreži

## Stabilnost ukupne komunikacije

- Prijenos svih podataka na CAN-u (DRT, URT i NRT podataka) bit će stabilan ako su ispunjeni uvjeti (26), (27) i (28), tj. ako je:

$$L_e \sum_{j=1}^{N_e} \lambda_e^j < 1$$

(26)

$$L_n \sum_{j=1}^{N_n} \lambda_n^j + \alpha_K L_c / T_1 + L_e \sum_{k=1}^{N_e} \lambda_e^k < 1$$

(27)

$$L_c r + L_e n_i \leq T_1$$

(28)

- Uvjet (27) obuhvaća uvjet (26) pa se uvjet (26) ne treba promatrati.



# Uzorkovanje u CAN mreži

## Primjer – 1/5

- ❑ Promatrana petlja je najmanje prioriteta, a podatke uzorkuje vremenom  $T_1$ .
- ❑ Broj regulacijskih petlji:  $M=5$
- ❑ Broj čvorova:  $N=N_e+N_c+N_n=5+10+5=20$  .
- ❑ Vremena prijenosa:  $L_e=0.584$  ms,  $L_c=1.04$  ms
- ❑ Frekvencija generiranja DRT paketa:  $\lambda_e^i=0.1$  [ms]<sup>-1</sup>, za  $i=1-N_e$
- ❑ Frekvencija generiranja NRT paketa:  $\lambda_n^i=0.0584$  [ms]<sup>-1</sup>, za  $i=1-N_n$
- ❑ Dopuštena kašnjenja regulacijskih petlji:  
 $\theta=[25$  ms, 60 ms, 100 ms, 200 ms, 400 ms]
- ❑ Prijenosne funkcije procesa i kontinuiranog ekvivalenta regulatora najbrže regulacijske petlje:

$$G_p(s) = \frac{1}{(1 + 0.3s)(1 + 0.03s)}$$

$$G_r(s) = \frac{35(1 + 0.2s)}{0.2s}$$

# Uzorkovanje u CAN mreži

## Primjer – 2/5

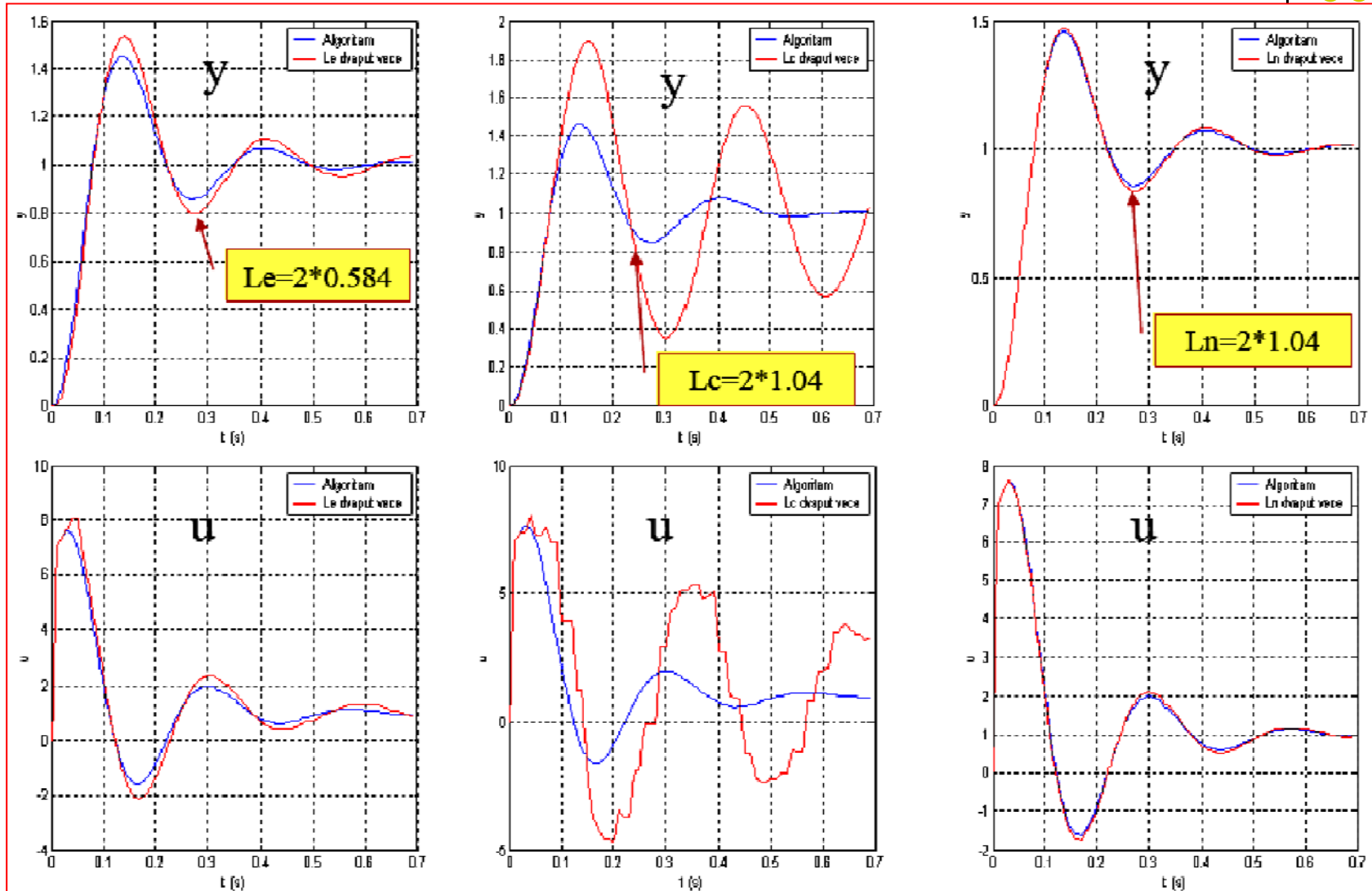
- ❑ Algoritam određivanja trenutaka uzorkovanja daje sljedeće vrijednosti koje zadovoljavaju uvjete stabilnosti i daju skoro u potpunosti iskorištenu mrežu:
- ❑  $L_n=1.04$ ,
- ❑  $\alpha_K=3.875$ ,
- ❑  $r=4$ ,
- ❑  $K=[1, 2, 4, 8, 16]$ ,
- ❑  $T=[10 \text{ ms}, 20 \text{ ms}, 40 \text{ ms}, 80 \text{ ms}, 160 \text{ ms}]$ ,
- ❑  $t=[0 \text{ ms}, 0 \text{ ms}, 0 \text{ ms}, 0 \text{ ms}, 10 \text{ ms}, 10 \text{ ms}, 30 \text{ ms}, 30 \text{ ms}, 70 \text{ ms}, 70 \text{ ms}]$ ,

$$L_n \sum_{j=1}^{N_n} \lambda_n^j + \alpha_K L_c / T_1 + L_e \sum_{k=1}^{N_e} \lambda_e^k = 0.9992 < 1$$



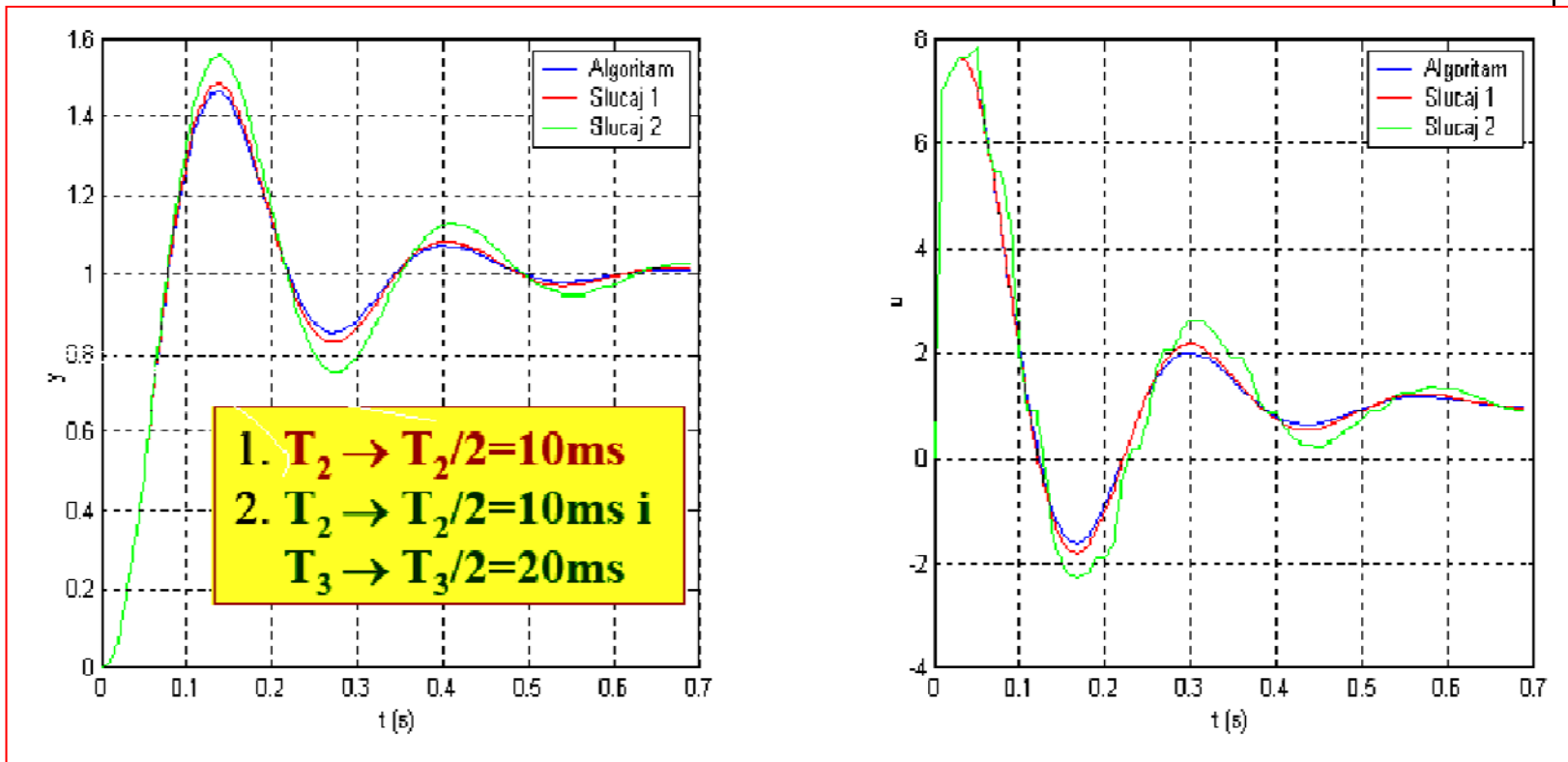
# Uzorkovanje u CAN mreži

## Primjer – 3/5



# Uzorkovanje u CAN mreži

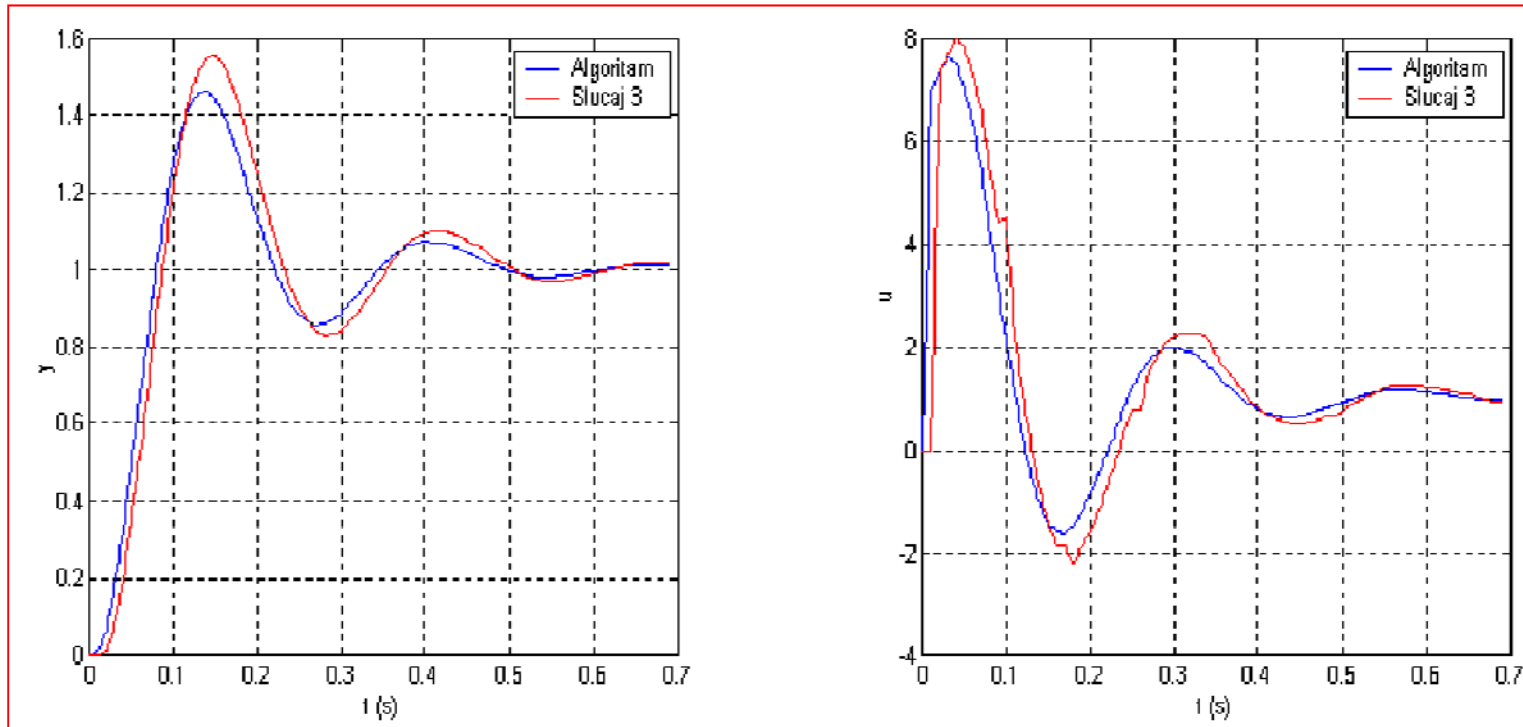
## Primjer – 4/5



- ❑ Povećan broj URT podataka na mreži, a trajanje RT intervala ostalo nepromijenjeno.
- ❑ Nužna je posljedica narušavanja kvalitete ponašanja regulacijskog kruga.

# Uzorkovanje u CAN mreži

## Primjer – 5/5



- ❑ Svi parametri postavljeni po algoritmu, osim članova vektora  $t$  koji su početno postavljeni na  $0$ .
- ❑ Vidi se velika distorzija upravljačkog signala.
- ❑ Do sličnog efekta može doći kada je sve podešeno prema algoritmu, ali se poremeti sinhronizacija lokalnih satova stvarnog vremena u čvorovima ⇒ važna dovoljno česta resinhronizacija.



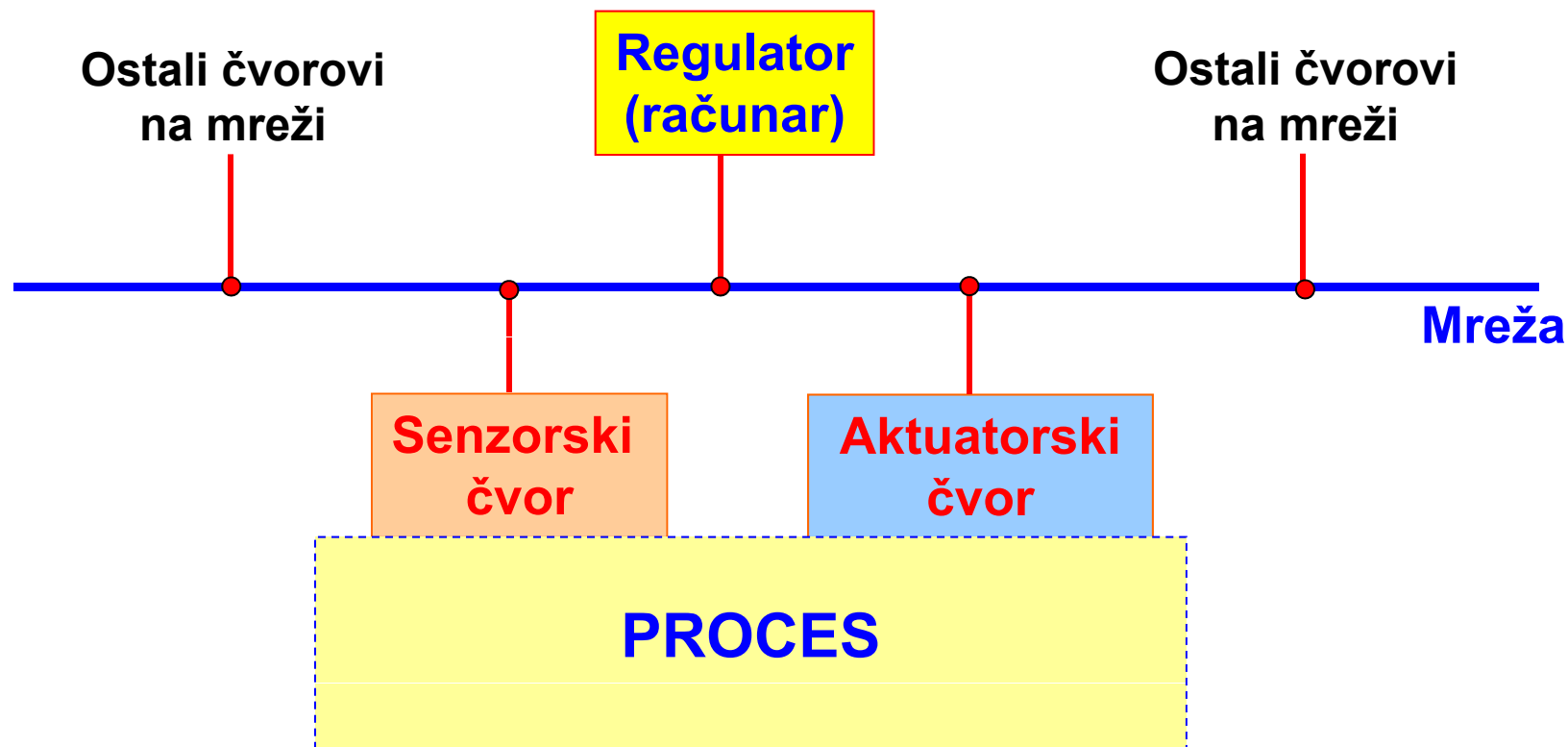
## 10.7. Upravljanje u ZP preko komunikacijske mreže

### Uvod

- Upravljanje u zatvorenoj petlji preko komunikacijske mreže postaje sve češće jer se sve više primjenjuju periferni uređaji (mjerni i izvršni članovi) sa direktnim priključkom na mrežu.



65/91



# Upravljanje u ZP preko komunikacijske mreže

## Uvod

- ❑ Izvedbom sistema upravljanja preko zajedničke komunikacijske mreže smanjuju se troškovi ožičenja, a povećava modularnost i fleksibilnost sistema.
- ❑ **Problemi :**
  - Komunikacijska mreža unosi kašnjenje:
    - zbog ograničene brzine prijenosa,
    - zbog preopterećenja prometom na mreži.
- ❑ Kašnjenje je u pravilu promjenjivo, i to stohastički, pa sistem postaje **vremenski promjenjiv**.

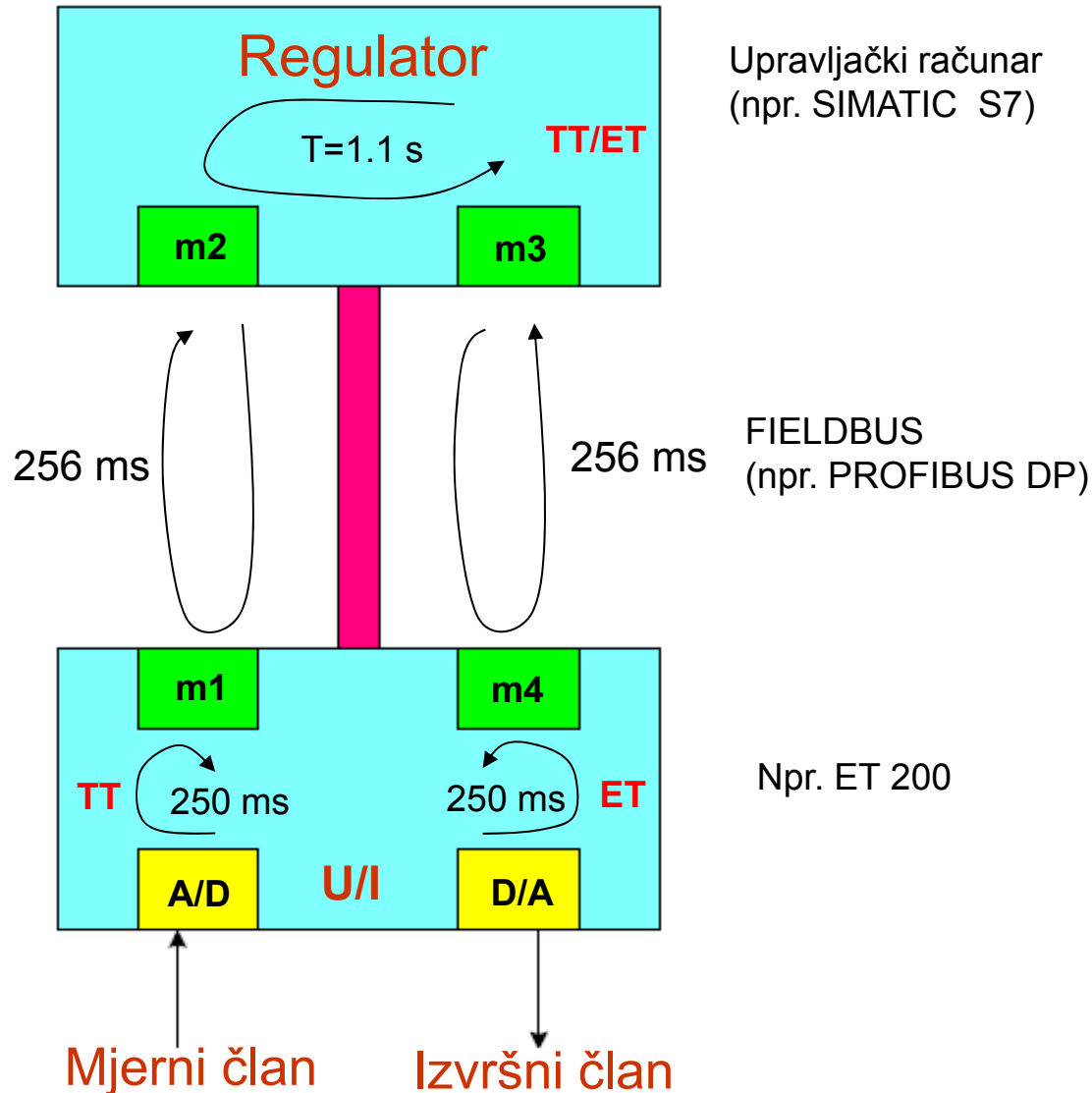


# Upravljanje u ZP preko komunikacijske mreže

## Analiza kašnjenja u regulacijskom krugu



67/91



- Mjerna veličina svakih 250ms se sprema u memorijsku lokaciju **m1**.
- Izlazna veličina (upravljачka) se svakih 250ms čita iz memorijske lokacije **m4** i šalje preko D/A pretvornika na izvršni član.
- Iznos mjerne veličine se preko komunikacijske mreže prebaci iz **m1** u **m2** u svakih 256 ms.
- Regulacijski algoritam u upravljačkom računaru izvodi se vremenom diskretizacije  $T=1.1\text{ s}$ .

# Upravljanje u ZP preko komunikacijske mreže

## Koliko iznosi kašnjenje u sistemu?



### □ Nesinhronizirani klokov

- Budući da nema globalnog vremena u sistemu, opisane periodičke aktivnosti međusobno će se “driftati”.
- Kašnjenje od trenutka očitavanja mjerne veličine  $t_o$  do trenutka slanja upravljačkog signala, koji koristi tu mjernu veličinu, na izvršni član  $t_u$  može maksimalno iznositi:

$$\tau_k = (t_u - t_o) = \tau_k^s + \tau_k^{SR} + \tau_k^R + \tau_k^{RA} + \tau_k^A = 250 + 256 + 1100 + 256 + 250 = 2.112s$$



# Upravljanje u ZP preko komunikacijske mreže

## Koliko iznosi kašnjenje u sistemu?

### □ Sinhronizirani klokov

- Sa stajališta teorije upravljanja prirodno je uzorkovati mjernu veličinu sa fiksnim vremenom uzorkovanja  $T$  (TT – time triggered).
- Isto tako prirodno je držati kašnjenje u sistemu što je moguće kraćim (vremensko kašnjenje unosi i fazno kašnjenje).
- Regulatorski i aktuatorski čvor trebali bi se izvoditi na događaj (ET – event triggered), tj. čim dobiju novu informaciju od senzorskog čvora, odnosno od upravljačkog čvora.
- Ovo je moguće kod CAN mreža, dok kod mreža sa prosljeđivanjem zaloga i regulatorski čvor mora biti TT, pa se kašnjenja povećavaju za još jedno vrijeme uzorkovanja.

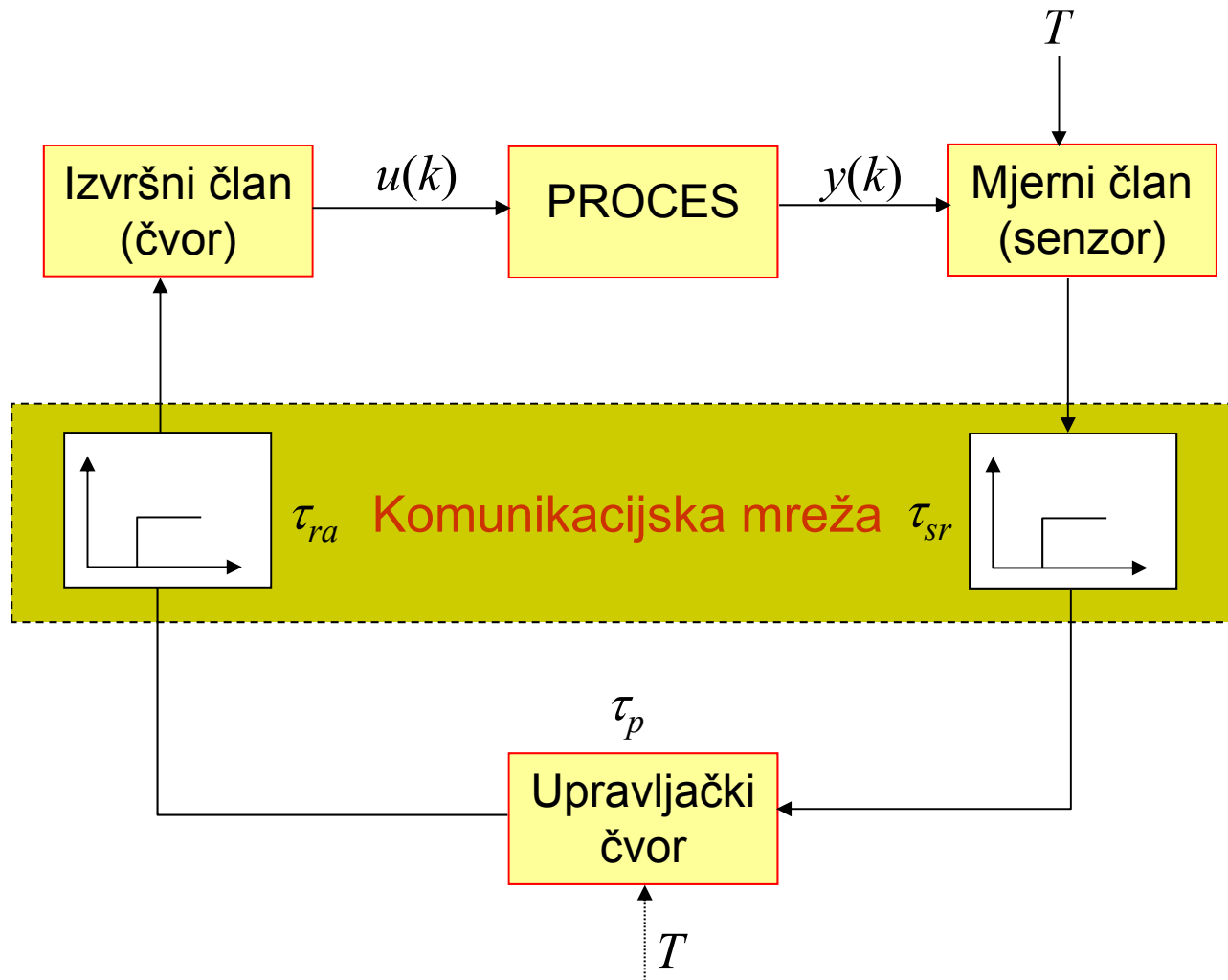


# Upravljanje u ZP preko komunikacijske mreže

## Modeliranje sistema upravljanja preko mreže



70/91



Ukupno dodatno mrtvo vrijeme:

$$\tau_k = \tau_{sr} + \tau_p + \tau_{ra}$$

- $\tau_{sr}$  – vrijeme prijenosa poruke od senzora do regulatora,
- $\tau_p$  – vrijeme proračuna upravljačkog algoritma,
- $\tau_{ra}$  – vrijeme prijenosa poruke od regulatora prema aktuatoru.

## Upravljanje u ZP preko komunikacijske mreže

### Modeliranje sistema upravljanja preko mreže

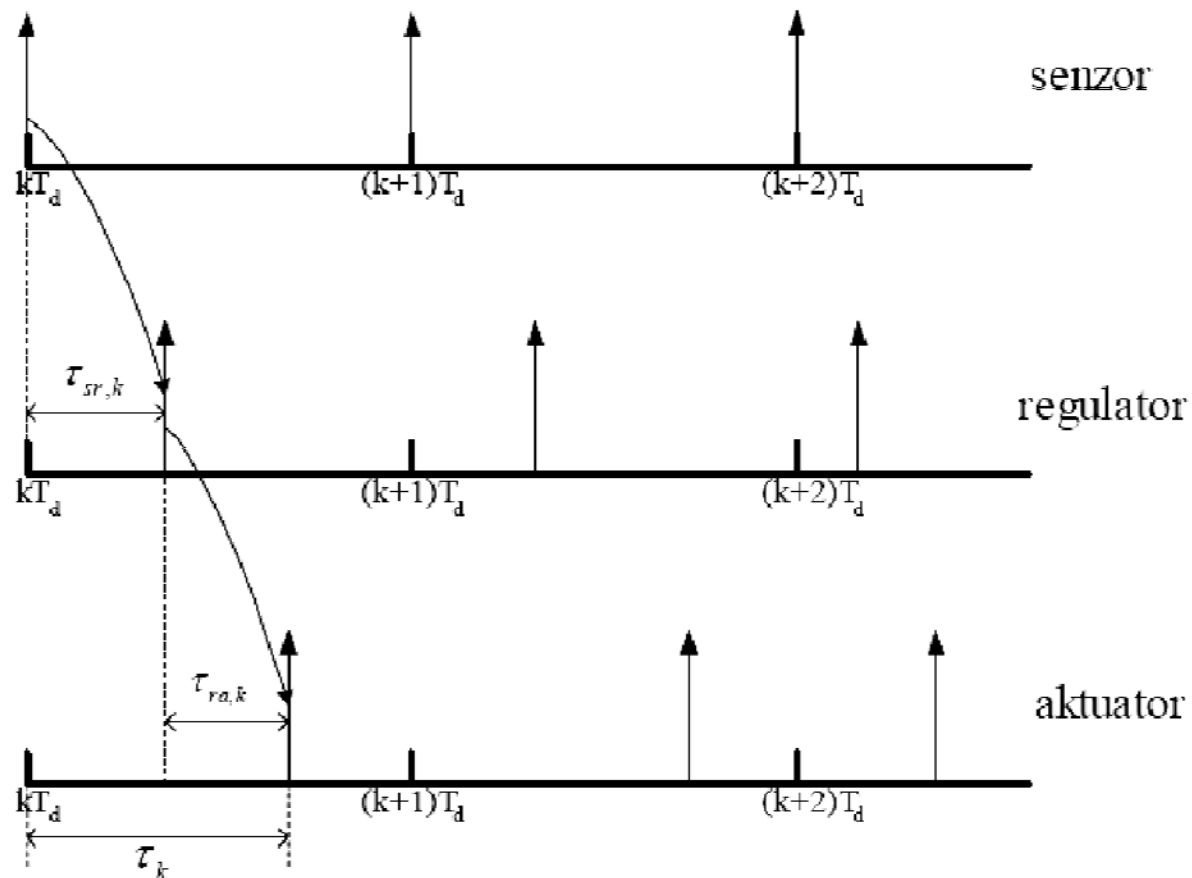
- ❑ U literaturi se mogu naći različite varijante zatvorenog sistema upravljanja preko mreže koje se razlikuju uglavnom u načinu kako se čvorovi na mreži aktiviraju.
- ❑ Čvor se može aktivirati kao posljedica nekog događaja, npr. prekida usljed dolaska poruke na čvor (eng. event-driven, ED) ili se može periodički aktivirati u skladu sa lokalnim satom na čvoru (eng. time-driven, TD).
- ❑ Da bi vrijeme diskretizacije bilo konstantno, čvor senzora mora biti TD.
- ❑ Ovisno o tome jesu li čvorovi regulatora i aktuatora TD ili ED dobivamo različite modele sisteme upravljanja, što je prikazano u nastavku.



# Upravljanje u ZP preko komunikacijske mreže

## 1. Regulator i aktuator ED

- Ovaj slučaj daje najmanje ukupno vrijeme kašnjenja, ali je kašnjenje promjenjivo.





# Upravljanje u ZP preko komunikacijske mreže

## Regulator i aktuator ED

- Matematički model procesa u prostoru stanja (bez kašnjenja):

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (29)$$

- Diskretni regulator po varijablama stanja opisan je jednačbom:

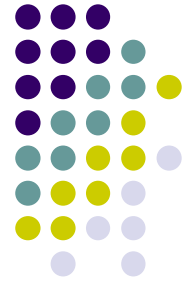
$$u(kT_d) = -Kx(kT_d) \quad (30)$$

- Uz pretpostavku da je kašnjenje u mreži  $\tau_k$  manje od vremena diskretizacije  $T_d$ , diskretizacijom procesa dobiva se:

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma_0(\tau_k)u(k) + \Gamma_1(\tau_k)u(k-1) \\ y(k) &= Cx(k) \end{aligned} \quad (31)$$

gdje je:

$$\Phi = e^{AT_d}; \Gamma_0(\tau_k) = \int_0^{T_d - \tau_k} e^{As} B ds; \Gamma_1(\tau_k) = \int_{T_d - \tau_k}^{T_d} e^{As} B ds .$$



# Upravljanje u ZP preko komunikacijske mreže

## Regulator i aktuator ED

- Proširenjem vektora stanja zatvoreni krug upravljanja može se opisati sljedećom jednačinom:

$$z(k+1) = \tilde{\Phi}(k)z(k), \quad (32)$$

gdje je:

$$z(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}; \quad \tilde{\Phi} = \begin{bmatrix} \Phi - \Gamma_0(\tau_k) & \Gamma_1(\tau_k) \\ -K & 0 \end{bmatrix}$$

- Ako je  $\tau_k > T_d$  i ako za svaki  $k$  vrijedi:  $(d-1)T_d < \tau_k < dT_d$ , dobije se sljedeći oblik matrice:

$$\tilde{\Phi} = \begin{bmatrix} \Phi & \Gamma_1(\tau_k) & \Gamma_0(\tau_k) & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ -K & 0 & 0 & \cdots & 0 \end{bmatrix} \quad (33)$$

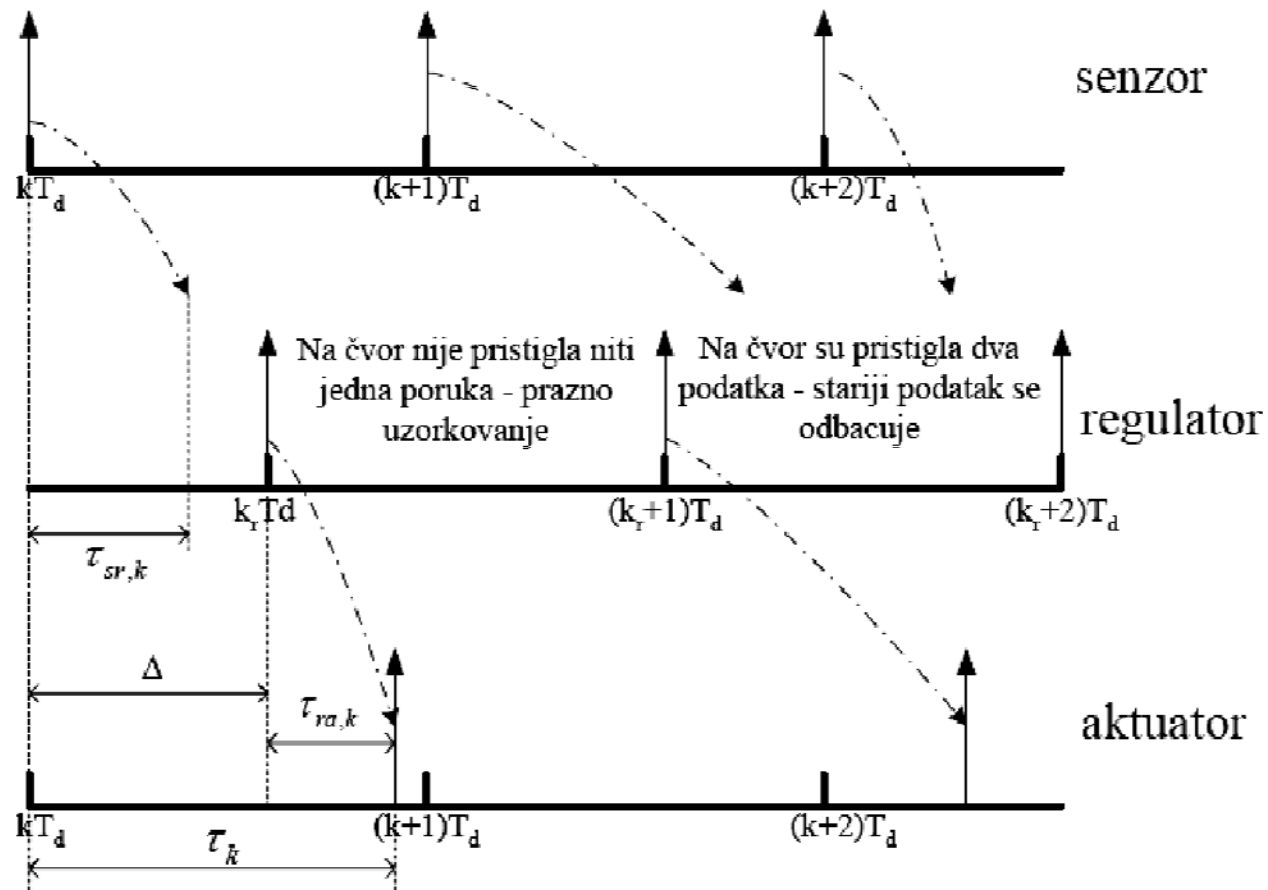
Vremenski promjenjiv sistem



# Upravljanje u ZP preko komunikacijske mreže

## 2. Regulator TD, aktuator ED

- Ako se regulator periodički aktivira dobiva se različito ponašanje sistema nego u prethodnom slučaju.



# Upravljanje u ZP preko komunikacijske mreže

## Regulator TD, aktuator ED

- ❑ Senzor i aktuator imaju vrijeme diskretizacije  $T_d$ , a trenuci diskretizacije su pomaknuti za vremenski pomak  $\Delta$ .
- ❑  $\Delta$  je sporo promjenjiva veličina zbog posmaka (drifta) satova na čvorovima.
- ❑ Vrijeme kašnjenja se povećalo za vrijeme između dolaska poruke na čvor regulatora i proračunavanja upravljačkog algoritma.
- ❑ Na vremenskom dijagramu prikazan je slučaj kada su vremena kašnjenja manja od vremena diskretizacije. Čak i u tom slučaju može se dogoditi da na čvor regulatora do trenutka proračunavanja upravljačke veličine ne stigne rezultat posljednjeg mjerenja.
- ❑ Usljed toga može doći do praznog uzorkovanja (vacant sampling), kada u jednom intervalu na čvor regulatora ne stigne ni jedan rezultat mjerenja, ili odbacivanja podataka (data rejection) kada na čvor regulatora stignu dva rezultata mjerenja, pa se stariji rezultat odbacuje. Ovaj slučaj očito je kompliciraniji od prethodnog.
- ❑ Diskretizirani proces može se u ovom slučaju opisati sljedećim jednadžbama:

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma_0 u(k) + \Gamma_1 u(k-1) + \Gamma_2 u(k-2) \\y(k) &= Cx(k)\end{aligned}\tag{34}$$



# Upravljanje u ZP preko komunikacijske mreže

## Regulator TD, aktuator ED

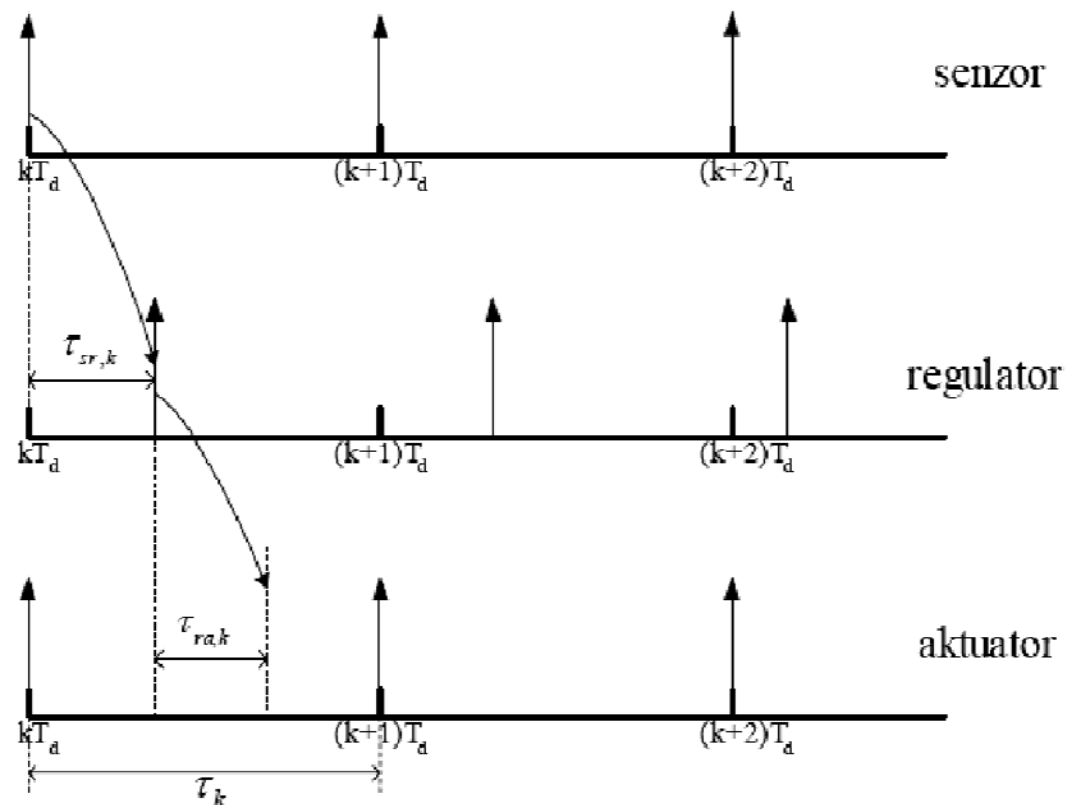
- ❑ Izrazi  $\Gamma_0$ ,  $\Gamma_1$  i  $\Gamma_2$  slični su izrazima dobivenim u prethodnom slučaju, ali se ovisno o tome je li  $\tau_{ra,k}$  veći ili manji od  $T_d - \Delta$  dobivaju različiti izrazi.
- ❑ Parametri sistema u ovom slučaju ovise, osim o vremenu kašnjenja poruka u mreži, i o vremenskom pomaku  $\Delta$  između trenutaka diskretizacije.
- ❑ Zbog toga je važno sinhronizirati satove na čvorovima regulatora i senzora.
- ❑ Sinhronizacija je važna i uslučaju kada je regulator ED, budući da se uz sinhronizirane satove može proračunati kašnjenje poruke u mreži, i iz toga odrediti dodatno mrtvo vrijeme koje treba kompenzirati.



# Upravljanje u ZP preko komunikacijske mreže

## 3. Aktuator TD

- Aktuator i senzor spojeni na isti čvor na mreži  $\Rightarrow$  djelovanje aktuatora sinhronizirano sa trenucima uzorkovanja mjernog signala. Ako su aktuator i senzor udaljeni ovo je moguće sinhronizacijom satova na čvoru aktuatora i senzora.



# Upravljanje u ZP preko komunikacijske mreže

## Aktuator TD

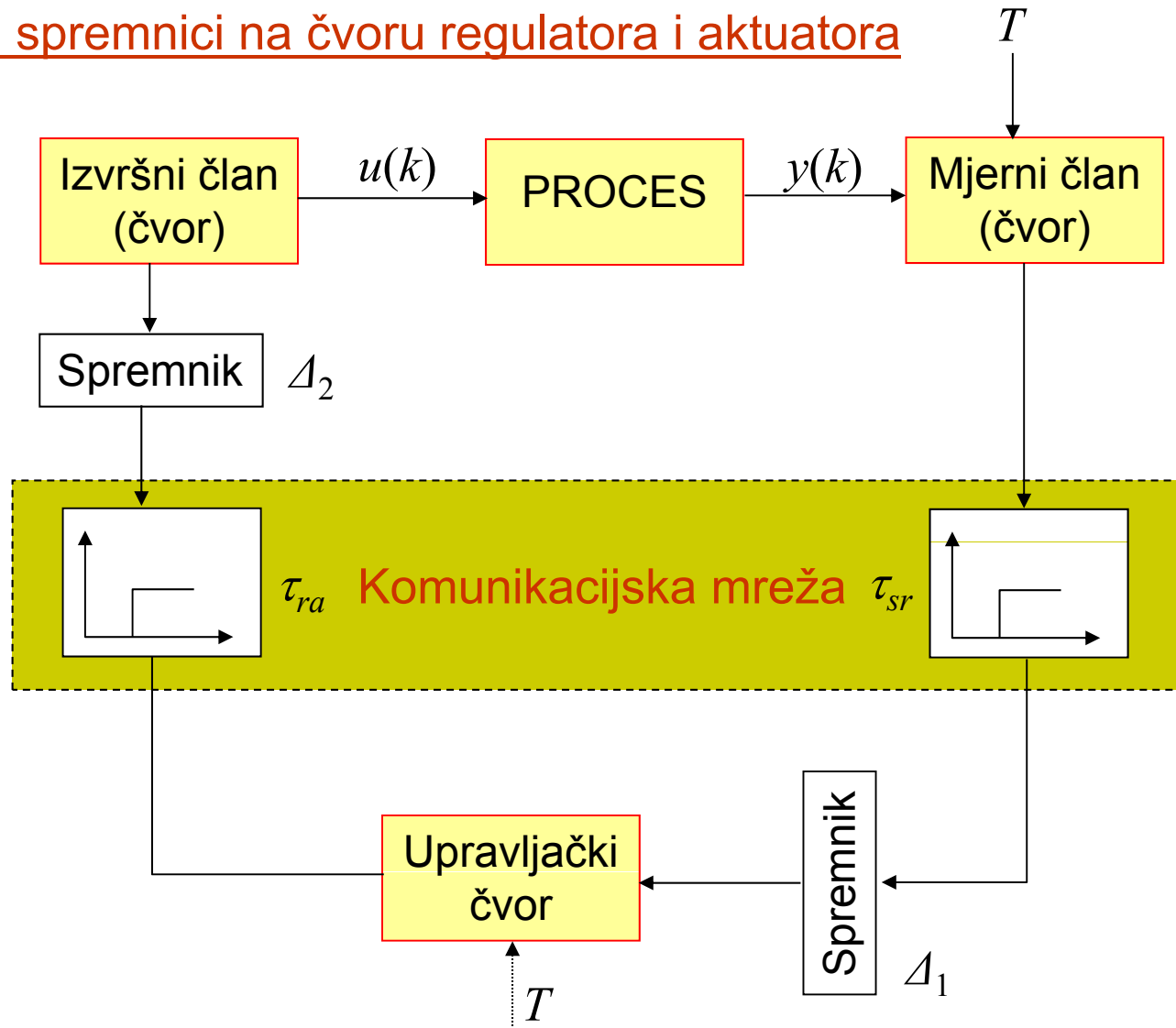
- ❑ Uz sinhronizirano djelovanje aktuatora i senzora ukupno mrtvo vrijeme zbog upravljanja preko mreže bit će cjelobrojni koeficijent vremena diskretizacije:  $\tau_k = nT_d$ ,  $n=1,2,3,\dots$
- ❑ Ako je ukupno vrijeme diskretizacije u mreži manje od vremena diskretizacije (uz regulator ED), mrtvo vrijeme bit će konstantno i iznositi će  $T_d$ .
- ❑ Ako je kašnjenje veće od vremena diskretizacije moguće je dodavanjem spremnika (buffers) na čvor regulatora i čvor aktuatora učiniti sistem vremenski nepromjenjivim.
- ❑ Uz aktuator sinhroniziran sa senzorom može se postići da ukupno dodatno mrtvo vrijeme usljed zatvaranja kruga upravljanja preko mreže bude konstantno, bez obzira je li kašnjenje poruka promjenjivo ili nije.
- ❑ To olakšava posao projektiranja sistema regulacije, ali daje ukupno mrtvo vrijeme veće nego što je potrebno. Ipak, u slučajevima kad povećanje mrtvog vremena ne narušava značajno ponašanje sistema, preporučljivo je koristiti ovako projektiran sistem regulacije.



# 10.8. Pregled pristupa upravljanja preko mreže

## 1. Učiniti sistem vremenski nepromjenjivim (determinističko ponašanje)

Dodatni spremnici na čvoru regulatora i aktuatora





# Upravljanje u ZP preko komunikacijske mreže

## Učiniti sistem vremenski nepromjenjivim

- ❑ Čvorovi su periodički i sinhronizirani. U spremnike se spremaju poruke do njihovog preuzimanja od čvora.
- ❑ Spremnici se dimenzioniraju tako da budu veći od najvećeg kašnjenja na mreži, tako da u trenutku diskretizacije čvor uvijek može preuzeti podatak iz spremnika, bez obzira na kašnjenja u mreži.
- ❑ Takav sistem može se opisati jednadžbama:

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k - \Delta_1), \\y(k) &= Cx(k), \\w(k) &= y(k - \Delta_2),\end{aligned}$$

(35)

gdje su sa  $w(k)$  označeni podaci na čvoru regulatora, a sa  $\Delta_1$  i  $\Delta_2$  veličine spremnika u broju koraka diskretizacije na čvoru aktuatora, odnosno regulatora.

- ❑ Budući da su duljine spremnika konstantne, komunikacijsko kašnjenje  $\tau_k = nT_d$  moguće je uključiti u model procesa.



# Upravljanje u ZP preko komunikacijske mreže

## Učiniti sistem vremenski nepromjenjivim

- ❑ Dakle, sistem se može promatrati kao sistem sa konstantnim mrtvim vremenom i na njega je moguće primijeniti standardne regulatora, na primjer:
  - **PID regulator.** U slučaju da je komunikacijsko kašnjenje malo, sinteza PID regulatora može se obaviti aproksimacijom kašnjenja PT1 članom ili primjenom Padéove aproksimacije. Druga je mogućnost primjena Zigler-Nicholsovog postupka.
  - **FPPI regulator (Filtered Predictive PI).** Primjenjuje se kada je komunikacijsko kašnjenje veliko. Mijenjanjem iznosa koeficijenta filtera može se postići željeno ponašanje sistema i robusnost. Za manji iznos ovog koeficijenta sistem upravljanja je robusniji, dok je za veći iznos brža kompenzacija poremećaja.
  - **Regulator u prostoru stanja sa estimacijom kašnjenja.** Cilj je postići zakon upravljanja kojim se ublažava utjecaj kašnjenja, što se postiže primjenom “prediktor-regulator” sistem upravljanja (regulator + estimator) . Regulator je sa linearnim zakonom upravljanja.



# Upravljanje u ZP preko komunikacijske mreže

## 2. Upravljanje procesom uz promjenjivo vremensko kašnjenje

- ❑ Uvođenjem vremenskih spremnika na čvorovima regulatora i aktuatora unosi se u sistem kašnjenje veće nego što stvarno postoji, što u zahtjevanim primjenama može biti nedopustivo, a u nekim manje zahtjevnim nepotrebno.
- ❑ U manje zahtjevnim primjenama promjenjivo komunikacijsko kašnjenje se može nadomjestiti srednjom vrijednošću:

$$\bar{\tau} = \frac{1}{N} \sum_{k=1}^N \tau_k$$

(36)

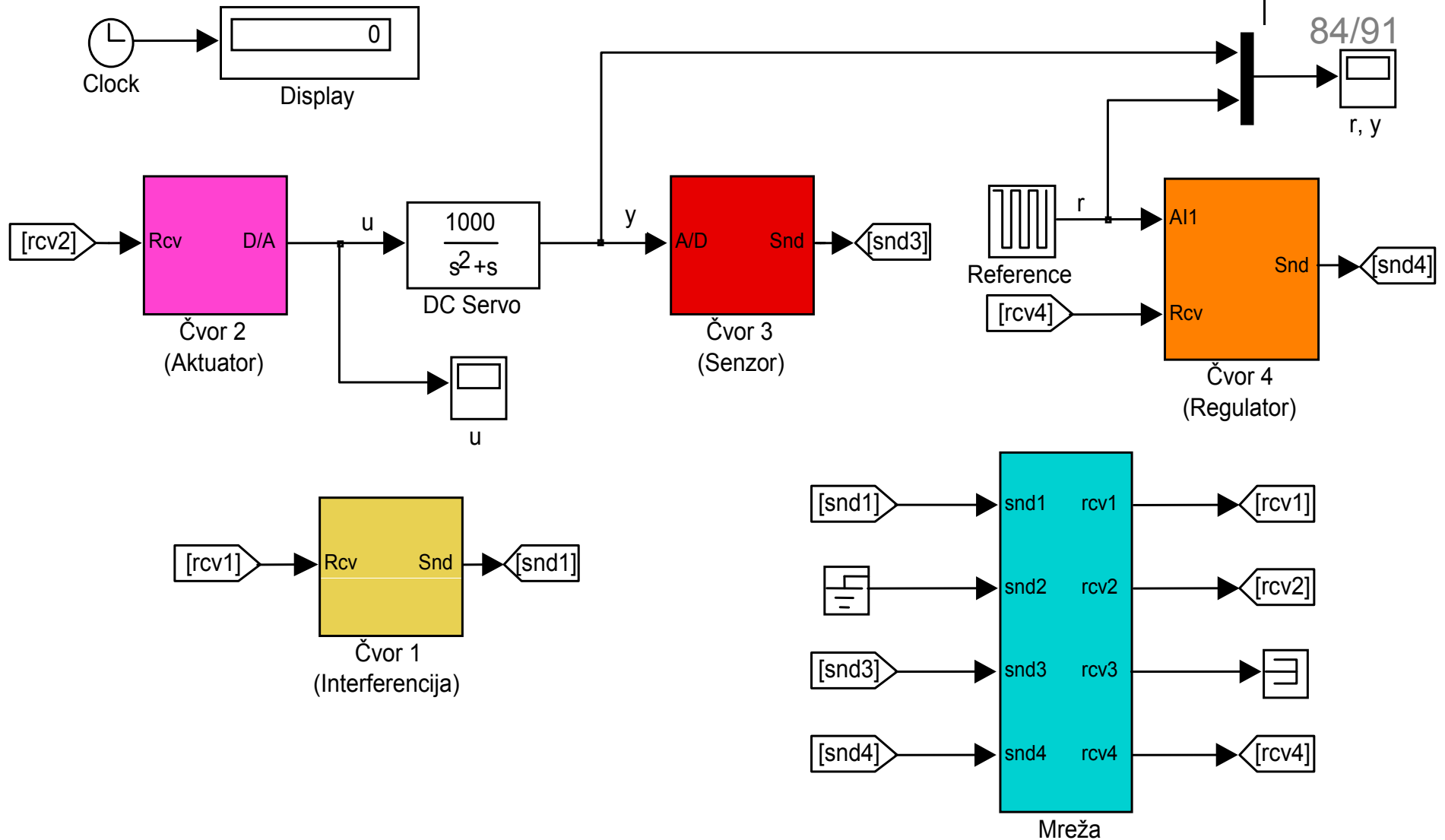
i provesti sintezu na isti način kao i kod uvođenja vremenskih spremnika.

- ❑ U zahtjevnim primjenama komunikacijsko kašnjenje se modelira kao slučajni proces poznatog očekivanja i poznate varijance pa se projektira optimalni stohastički regulator (npr. LQG regulator).



# Primjer upravljanja u zatvorenoj petlji preko mreže

Primjer: Sistem upravljanja istosmjernim motorom (digitalni PID regulator) zatvoren preko CAN mreže sa 4 čvora



# Primjer upravljanja u zatvorenoj petlji preko mreže

## Opis procesa i regulatora



- DC servo motor opisan je sljedećom vremenski kontinuiranom prijenosnom funkcijom:

$$G(s) = \frac{1000}{s(s+1)}$$

- PID regulator je implementiran u skladu sa sljedećim jednažbama:

$$P(k) = K \cdot (r(k) - y(k))$$

$$I(k+1) = I(k) + \frac{Kh}{T_i} (r(k) - y(k))$$

$$D(k) = a_d D(k-1) + b_d (y(k-1) - y(k))$$

$$u(k) = P(k) + I(k) + D(k)$$

(37)

gdje su  $a_d = \frac{T_d}{Nh + T_d}$ ,  $b_d = \frac{NKT_d}{Nh + T_d}$ . Parametri regulatora su odabrani tako da se osiguraju željene performanse:  $\omega_c = 20$  rad/s i  $\zeta = 0.7$ .

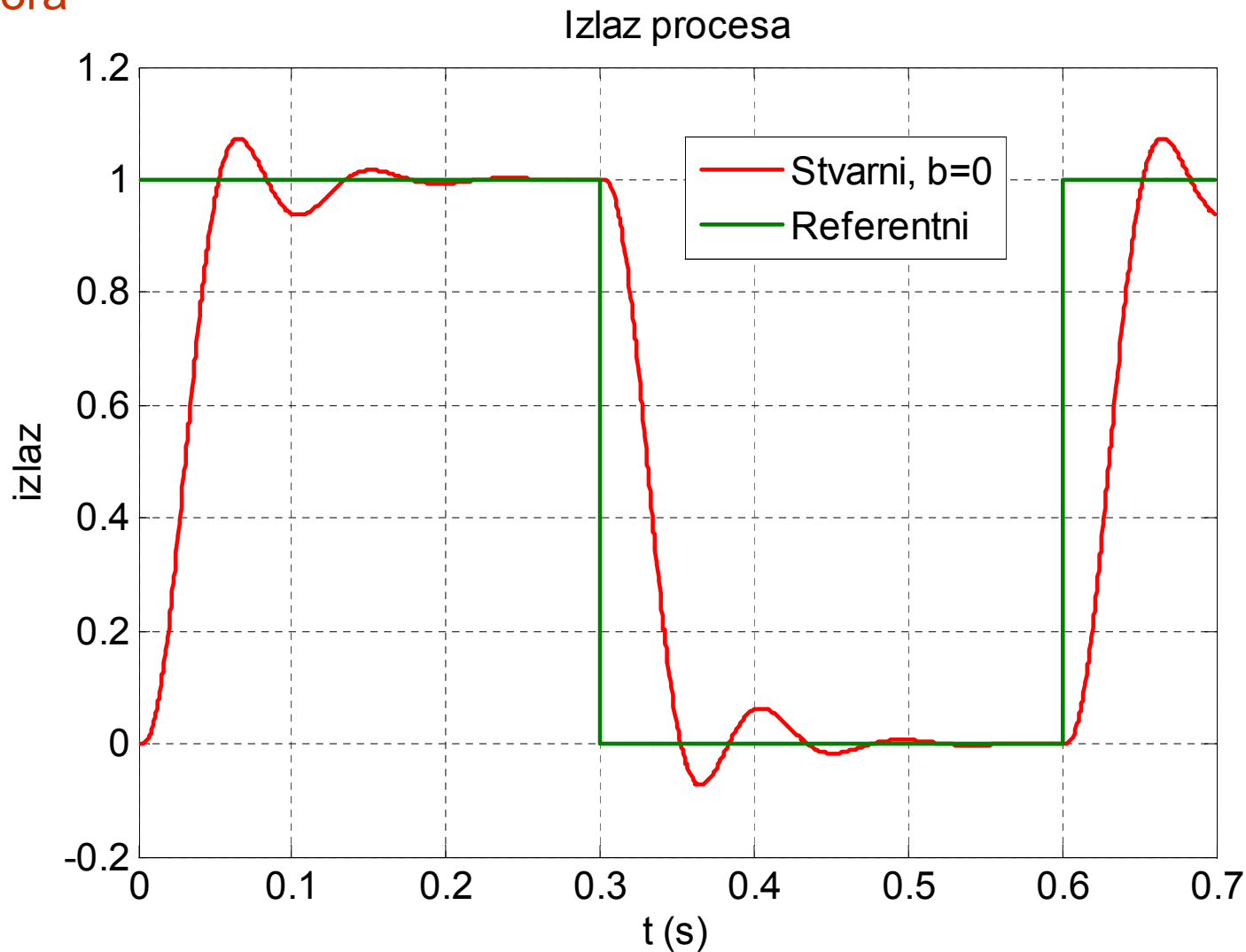
# Primjer upravljanja u zatvorenoj petlji preko mreže

## Odziv procesa bez smetnji i kašnjenja u čvorovima

kašnjenja u mreži zanemarljiva i sinhronizirani čvorovi regulatora i senzora



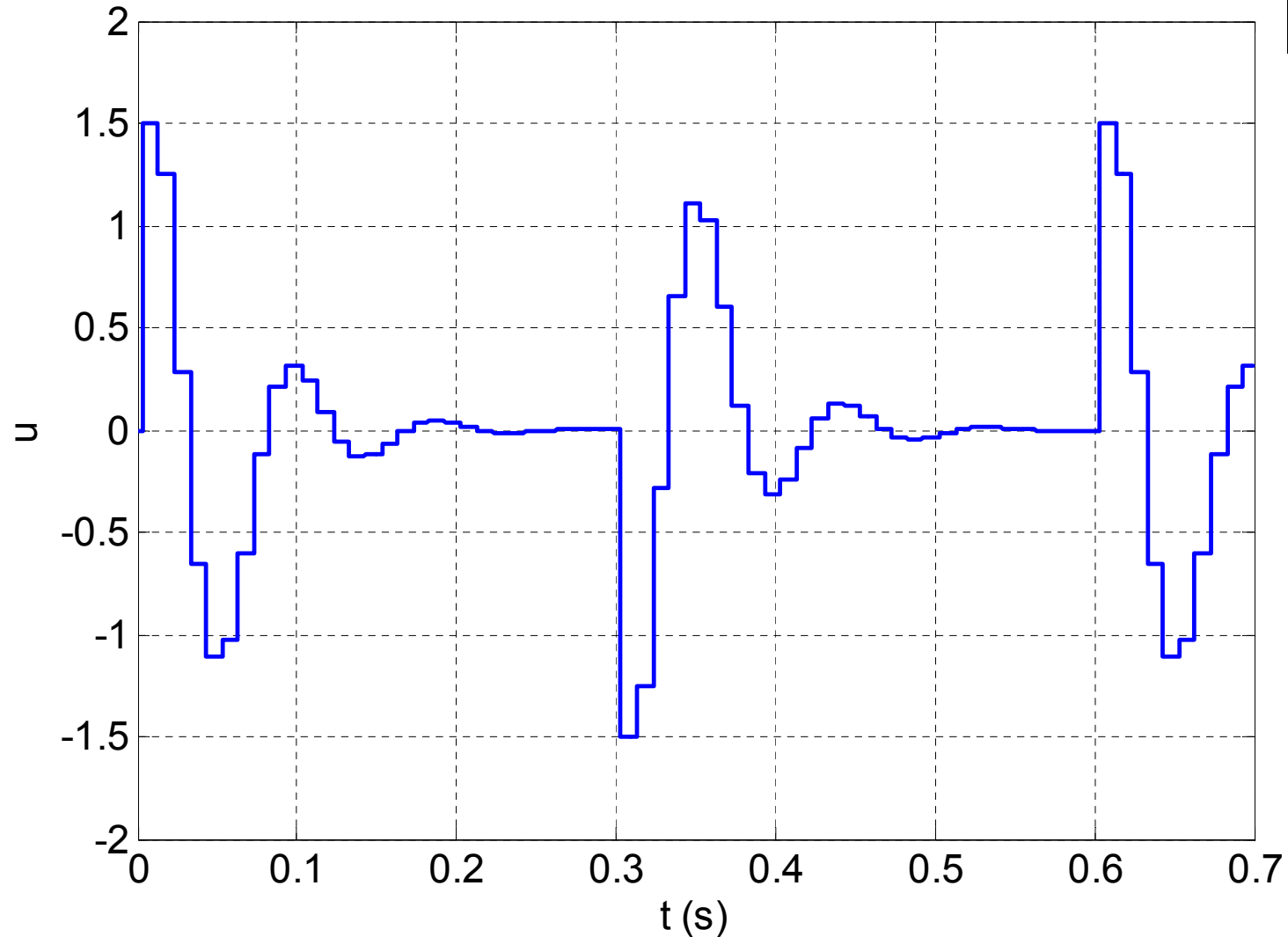
86/91



# Primjer upravljanja u zatvorenoj petlji preko mreže

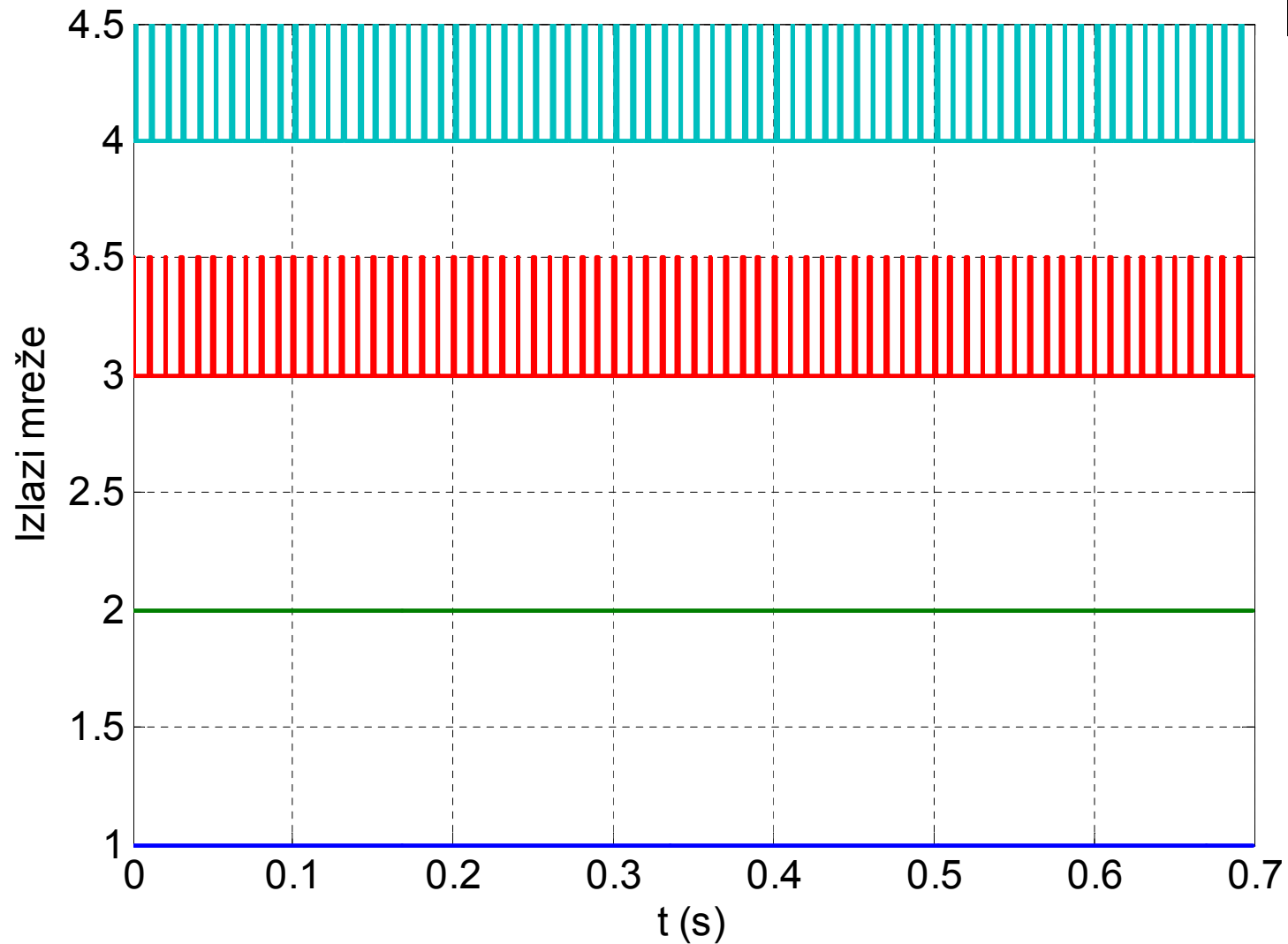
## Upravljački signal (izlaz iz regulatora)

Signal upravljanja (izlaz regulatora)



# Primjer upravljanja u zatvorenoj petlji preko mreže

## Izlazi mreže



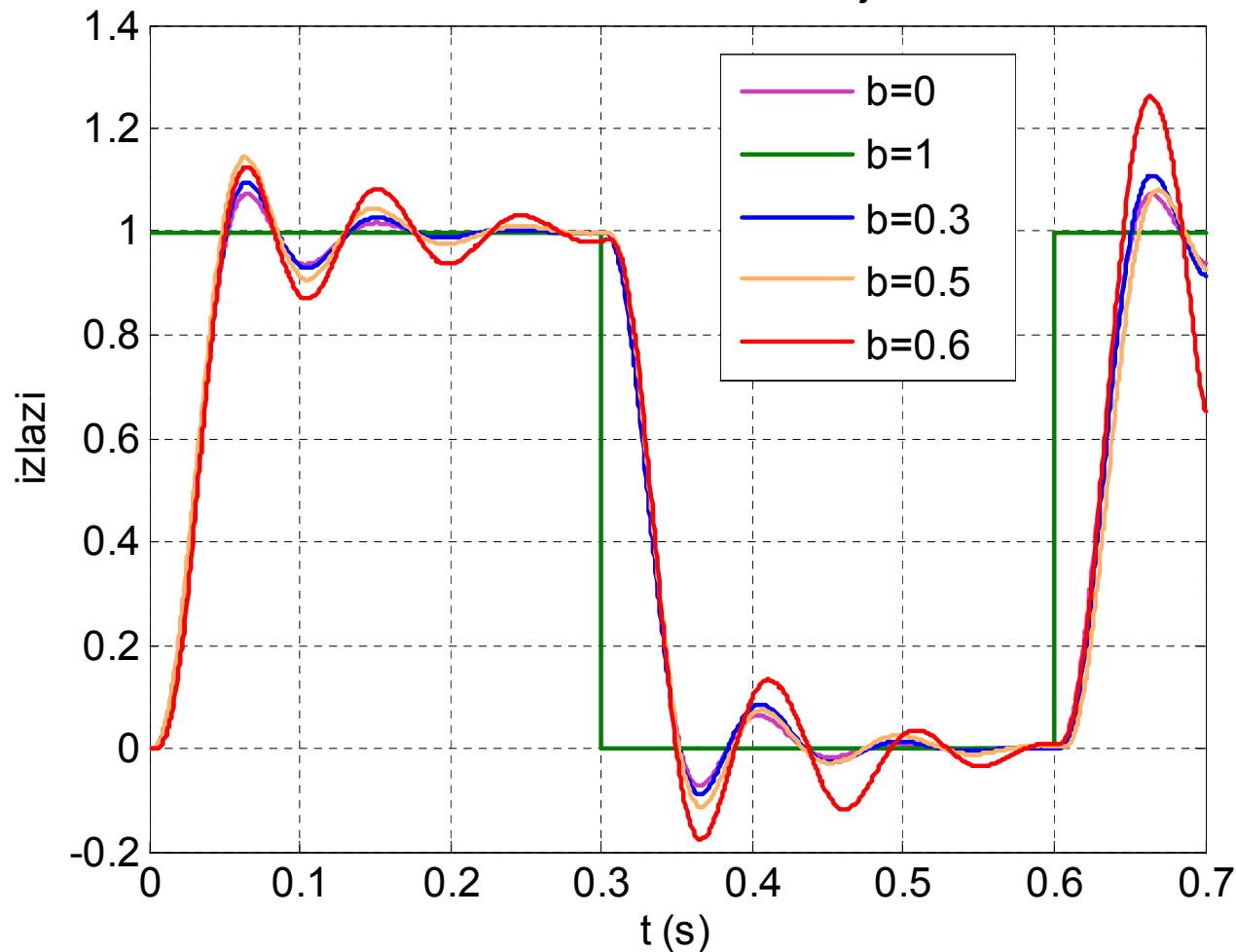


# Primjer upravljanja u zatvorenoj petlji preko mreže

## Usporedba odziva za različite intenzitete smetnji u mreži – utjecaj kašnjenja mreže na performanse sistema upravljanja – parametar $b$ povezan sa kašnjenjem.



Odzivi za razne iznose smetnji u mreži

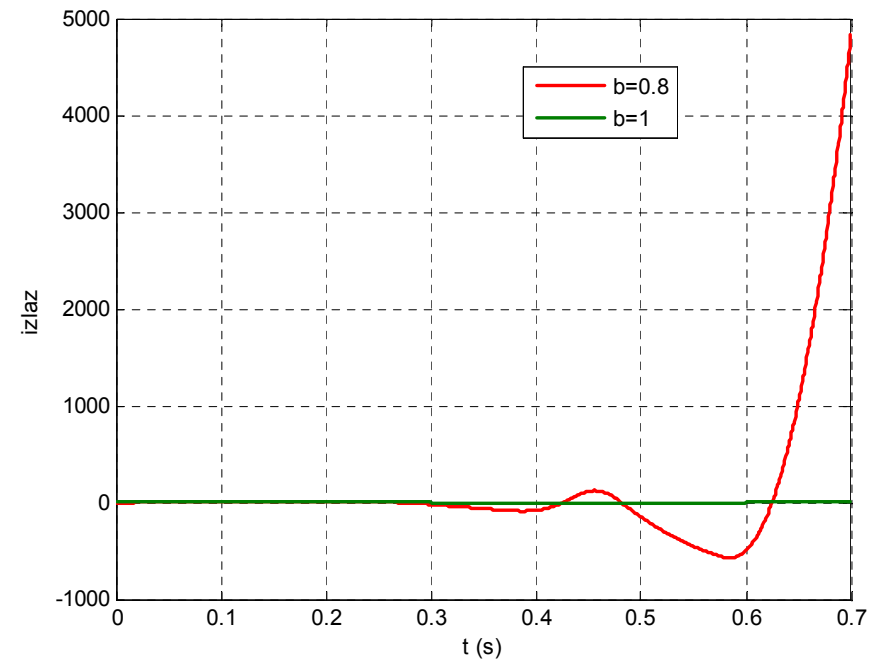
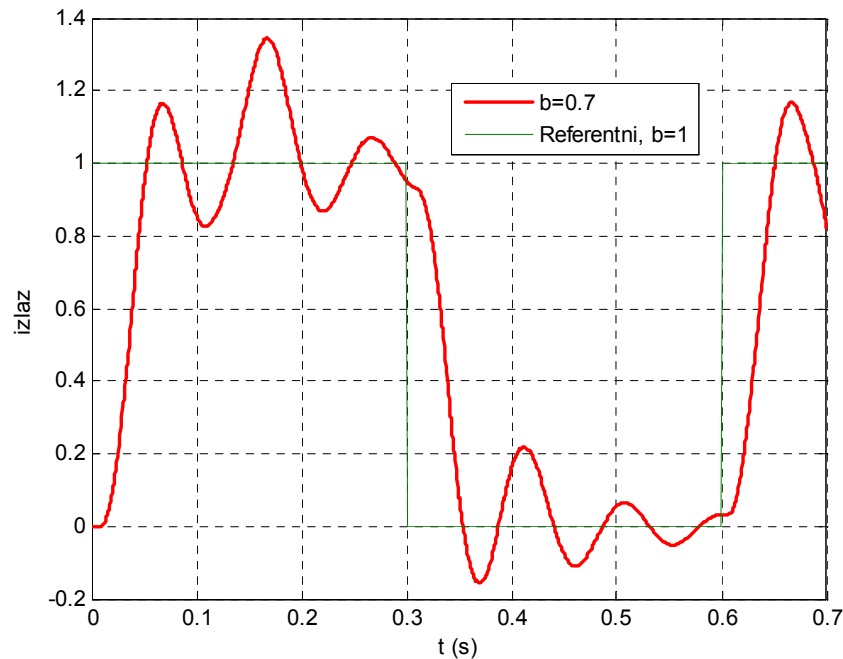


# Primjer upravljanja u zatvorenoj petlji preko mreže

Usporedba odziva za različite intenzitete smetnji u mreži – kašnjenja sa parametrom  $b$  između 0.7 i 1.



90/91



**Za  $b=0.7$  znatno se povećavaju oscilacije.**

**Za  $b$  između 0.8 i 1 oscilacije se raspršavaju  $\Rightarrow$  sistem nestabilan.**

Malo kašnjenje znači: vrijeme kašnjenja u mreži manje od vremena diskretizacije.  
Velika vremena kašnjenja: promjenjiva i veća od vremena diskretizacije.

# Primjer upravljanja u zatvorenoj petlji preko mreže

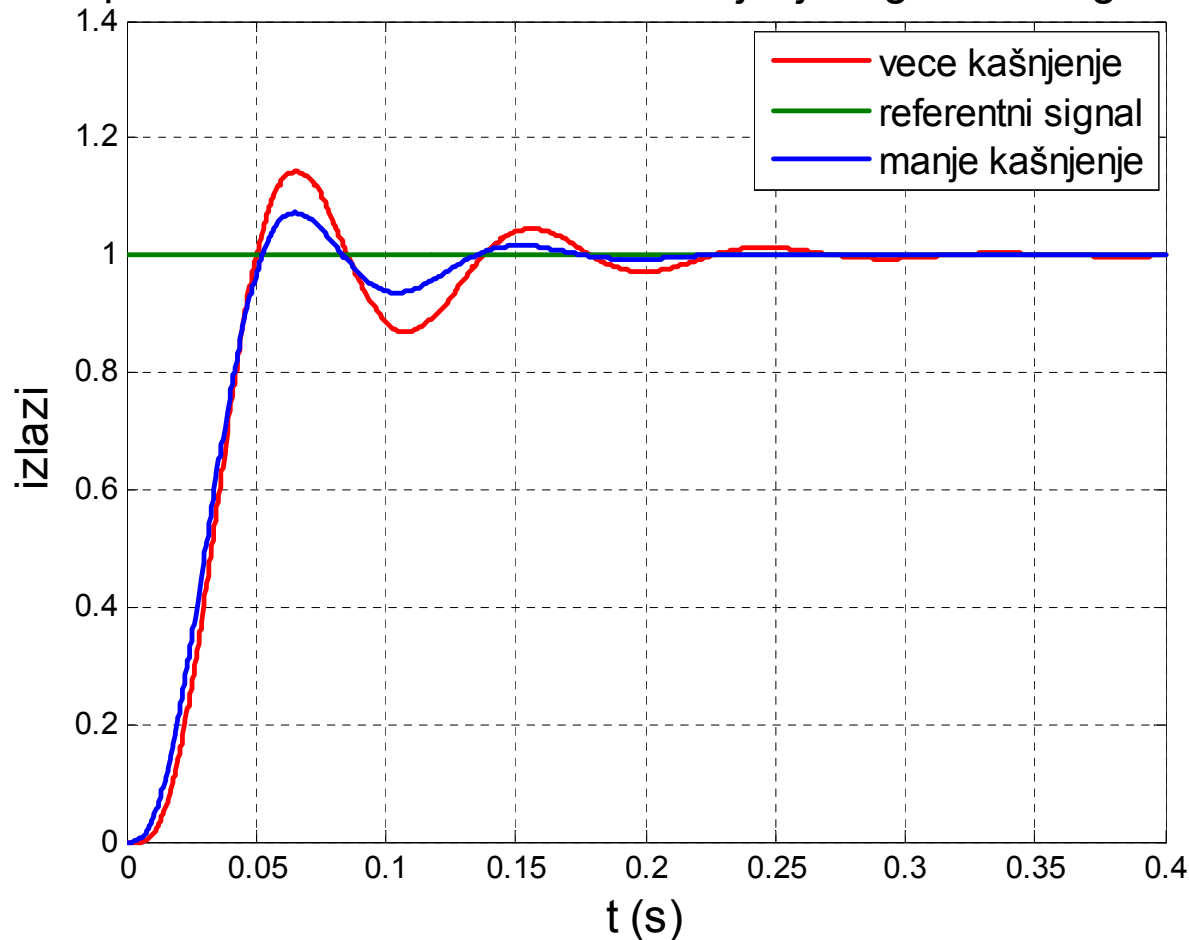
## Utjecaj nesinhroniziranosti čvorova regulatora i senzora

Smetnja regulatora usljed loše sinhronizacije čvorova regulatora i senzora – utjecaj na nadvišenje (oscilatornost) i brzinu odziva.



91/91

Usporedba odziva za razlicita kašnjenja regulatorskog cvora



- Veće kašnjenje – lošija sinhronizacija između čvora regulatora i čvora senzora (čvor regulatora kasnije se aktivira – povećanje pomaka između trenutaka aktiviranja čvorova senzora i regulatora) ⇒ brži odziv ali veće nadvišenje i oscilatornost.

- Sistemi upravljani preko mreže mogu usljed loše sinhroniziranosti čvora senzora i regulatora imati odziv različit od željenog čak i ako su kašnjenja u mreži zanemarivo mala.