

8. Sisteme de control în rețea

(Networked Control Systems - NCS)

8.1 Noțiuni, concepte, probleme și consecințe

8.2. Structuri NCS

8.3. Evaluarea controlerului

8.4. Metode NCS

8.5. Teleoperare

8.1 Noțiuni, concepte, probleme și consecințe

Wikipedia:

”A **Networked Control System** (NCS) is a [control system](#) wherein the control loops are closed through a communication [network](#). The defining feature of an NCS is that control and feedback signals are exchanged among the system's components in the form of information packages through a network.

They connect cyberspace to physical systems. They execute several tasks from long distance. They can be and are easily extended.”

Componente:

- *senzori*
- *controlere*
- *actuatori*
- *rețele de comunicație*

Aplicații:

- Explorări spațiale și terestre
- Automatizări industriale și fabrici
- Diagnostic la distanță și troubleshooting
- Roboți domestici
- Controlul avioanelor la distanță
- tele-operare
 - sisteme tele-operate cu operator uman
 - sisteme tele-operate fără intervenția omului (e.g.: NCS autonome)
- sisteme de monitorizare și control pentru rețele electrice
- sisteme de navigație pentru vehicule fără operator uman (unmanned vehicles)

Caracteristici:

- Partajare eficientă a datelor între controlere
- Informația globală este folosită pentru a lua decizii **inteligente** într-un spațiu fizic larg
- Buclele de feedback pentru control includ rețelele de comunicație
- Aplicații critice de timp sau sensibile la timp (time-critical/time-sensitive)
- Aplicații care nu sunt sensibile la întârzieri

Probleme principale:

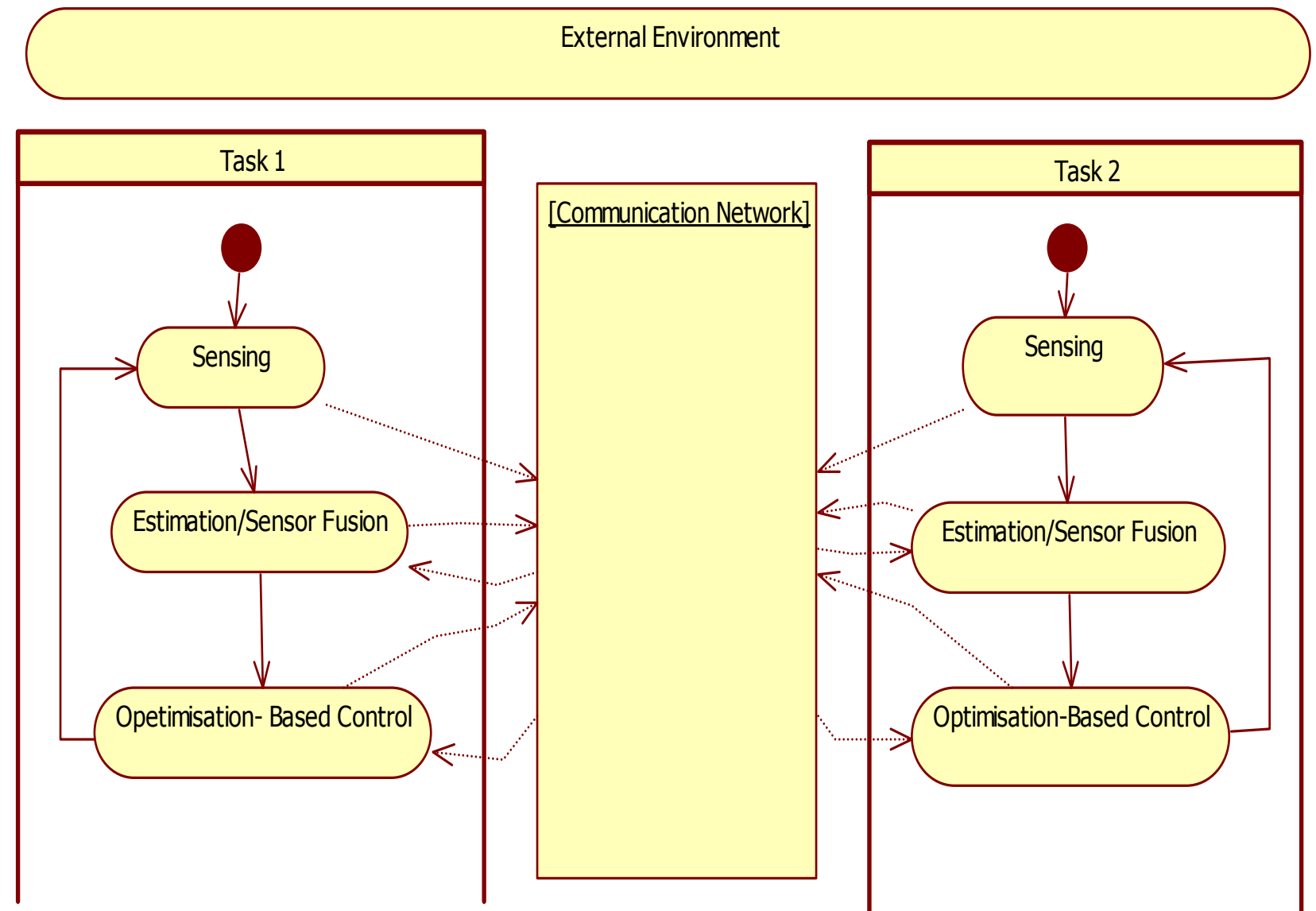
- Stabilitate la întârzieri de comunicație – un planificator median poate fi folosit pentru a ușura întârzierea rețelei – estimează traficul rețelei și efectul asupra întârzierii pentru NCS (= control predictiv)
- Alocarea și planificarea lățimii de bandă
- Securitatea rețelei (implementare fără fir ??)

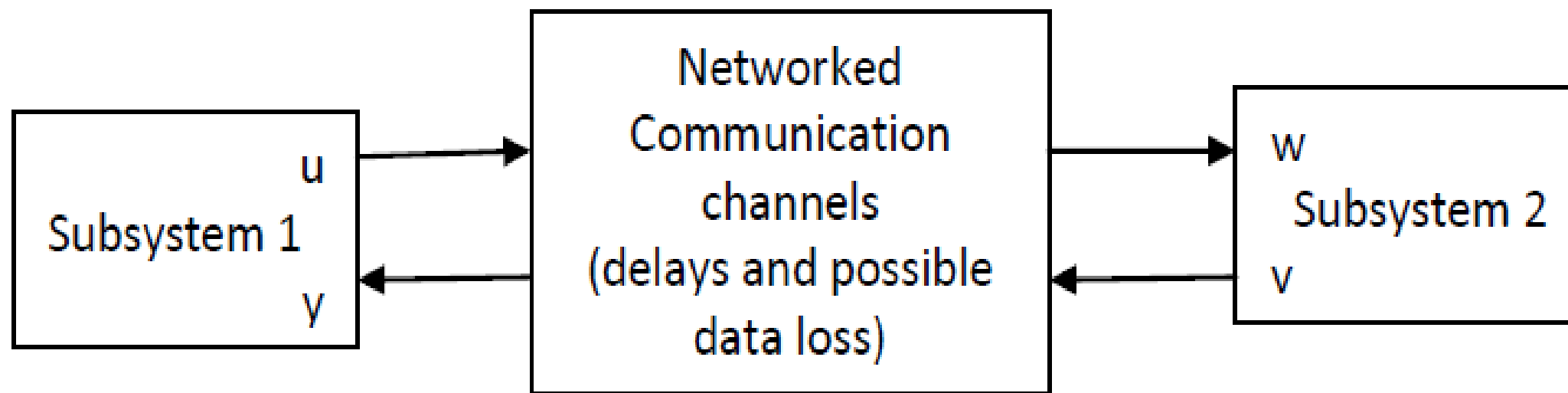
Discipline implicate:

- *Rețele de calculatoare*
- *Comunicație*
- *Procesarea semnalelor*
- *Robotică*
- *Tehnologia informației*
- *Teoria controlului*

8.2. Structuri NCS

E.g.: Arhitectură pentru rutarea vehiculelor





Întârzieri:

- Induse de rețea
- Ale sistemului (plant)
- Induse de calcule

SS1

$$\begin{aligned} \frac{dx(t)}{dt} &= \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t) \end{aligned}$$

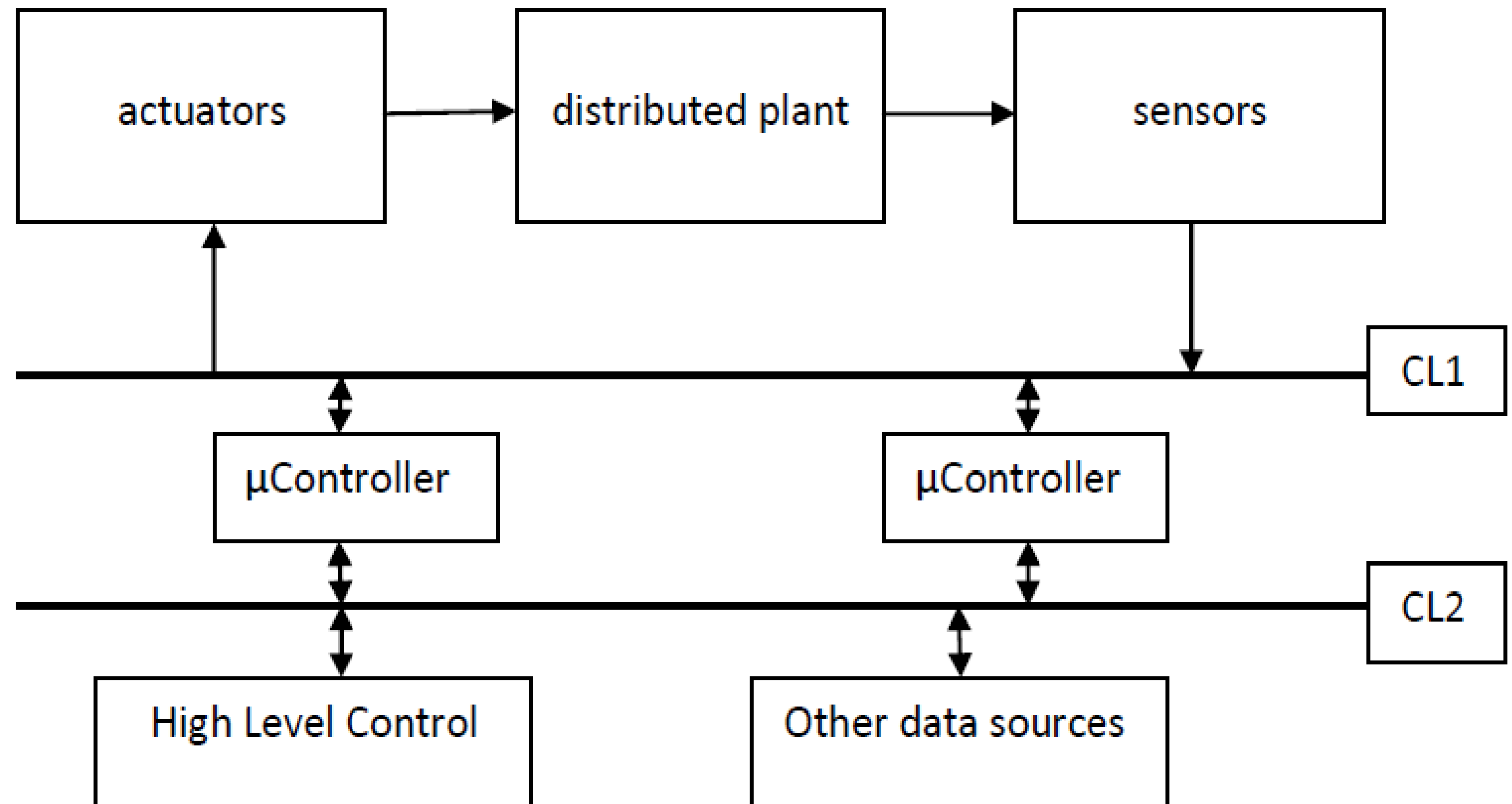
SS2

$$\begin{aligned} \mathbf{z}(k+1) &= \mathbf{E} \cdot \mathbf{z}(k) + \mathbf{F} \cdot \mathbf{w}(k) \\ \mathbf{v}(k) &= \mathbf{G} \cdot \mathbf{z}(k) + \mathbf{H} \cdot \mathbf{w}(k) \end{aligned}$$

Caracteristici presupuse în multe abordări:

- Ceasurile sunt sincronizate (perfect ??)
- Actuatori comandați de evenimente (event driven) sunt preferați față de actuatorii comandați de timp (time-driven)
- Doar cele mai actuale date sunt folosite
- Toți senzorii au aceeași perioadă de eșantionare, identică cu cea a controlerului
- Întârzierile controler-actuator pot fi foarte mici și deterministice
- Observatorul este parte a controlerului
- Comunicația fără fir este parte a sistemului de comunicație

CL1, CL2 –
Communication
Link 1, 2 (rețea
de comunicație)



Remote Direct Control = Controlul direct de la distanță

Mărimile măsurate din proces de către senzori sunt încapsulate de către o componentă de comunicație în cadre sau pachete și apoi sunt transmise prin rețea la controler.

Controlerul prelucrează informațiile recepționate, calculează mărimile de comandă și le pune în cadre sau pachete care sunt transmise spre instalație. Acolo sunt recepționate de o componentă de comunicație, care le despachetează și le aplică elementelor de execuție.

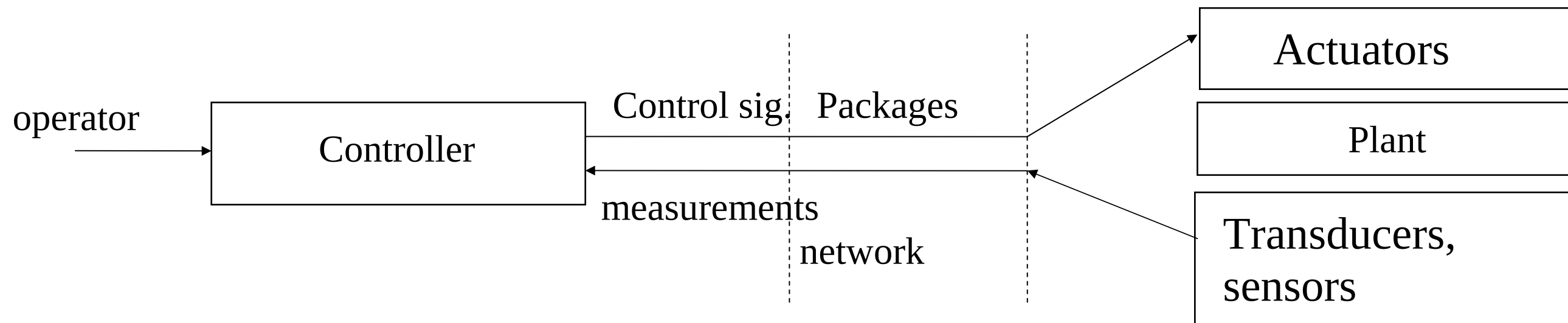


Fig. 10.1. Structura unui sistem de control direct la distanță.

Hierarchical Structure

Structura ierarhizată

Se consideră structură compusă dintr-un controler principal (aflat lângă operator, adică în centrul de comandă) și unul secundar.

Controlerul secundar aflat la distanță (engl. remote controler) realizează o buclă închisă de reglare.

Controlerul principal calculează periodic și transmite semnale de referință, în cadre sau pachete prin intermediul rețelei de comunicație, la sistemul de control îndepărtat.

Sistemul de control local în buclă închisă are, de regulă, o perioadă de eșantionare mai mică decât perioada de lucru a controlerului principal. Aceasta se face deoarece se presupune că, fiind mai aproape de instalație, controlerul îndepărtat satisface cerințele semnalului de referință înainte să sosească un nou semnal de referință.

Structura este utilizată deseori la controlul unui set de roboți mobili sau la tele-operații care implică să se comande mai multe motoare simultan. Pentru controlul brațului unui robot, controlul principal dă referințe pentru poziția lui, iar controlul local reglează vitezele pentru a realiza un răspuns mai rapid.

Alegerea utilizării uneia dintre variante (structura directă sau structura ierarhizată) depinde de cerințele aplicației și de preferințele proiectantului.

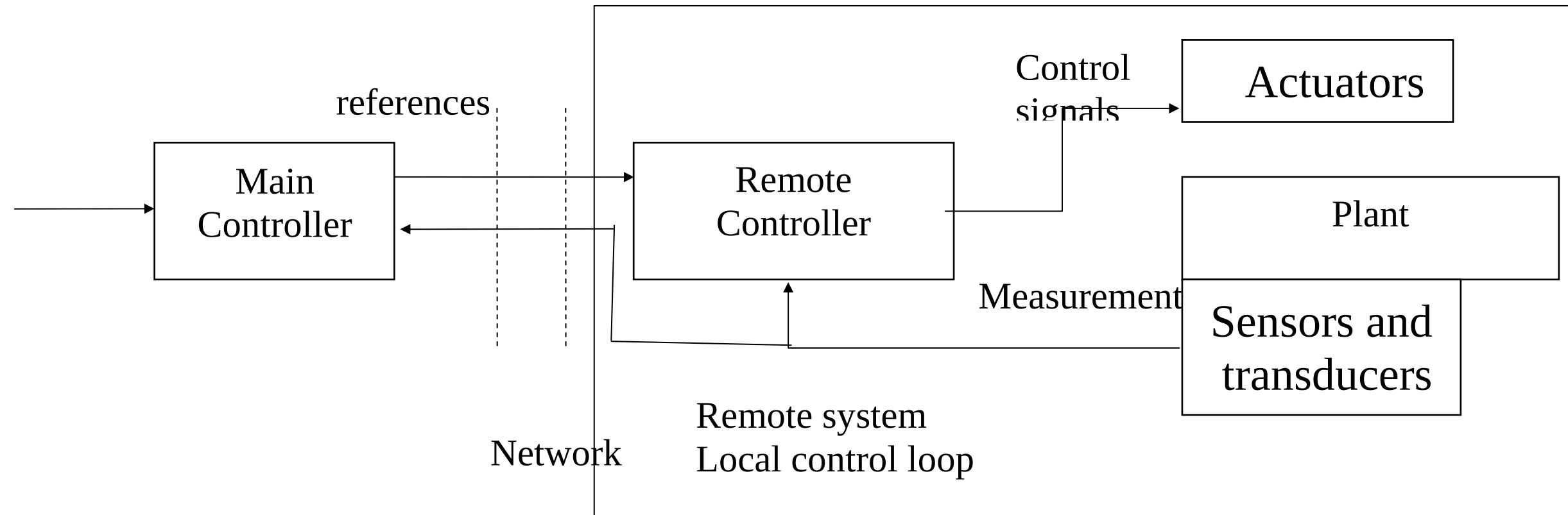


Fig. 10.2. Structura unui sistem de control ierarhic în rețea

Efectele întârzierilor la un sistem de control în rețea

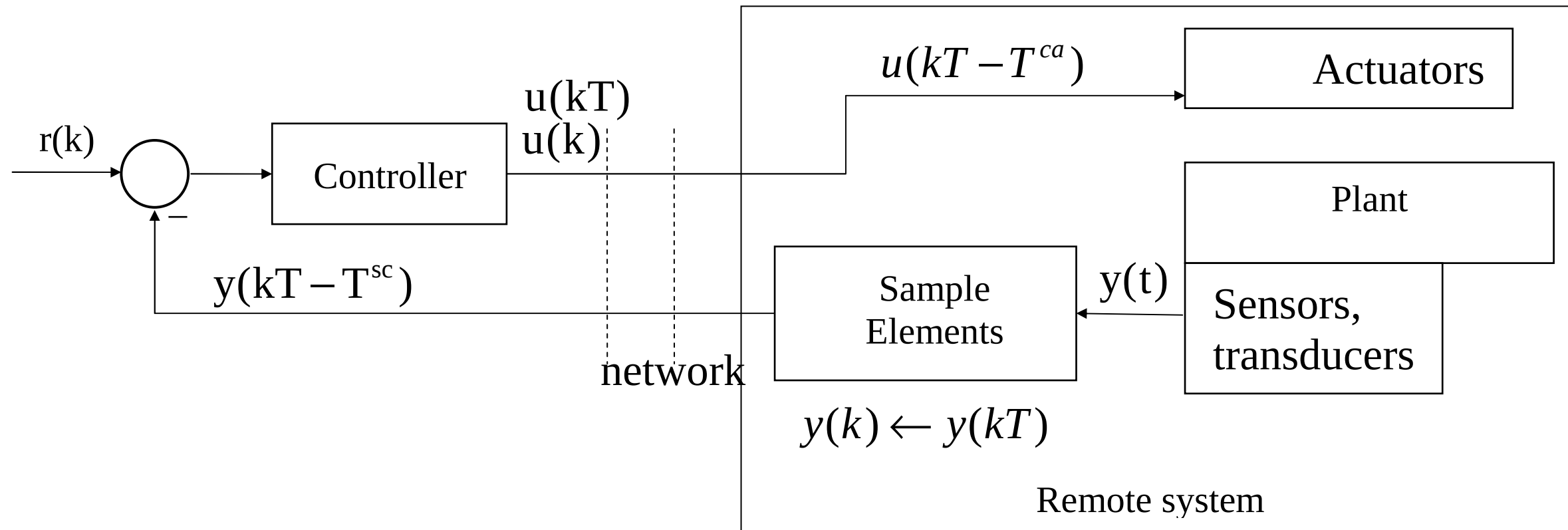


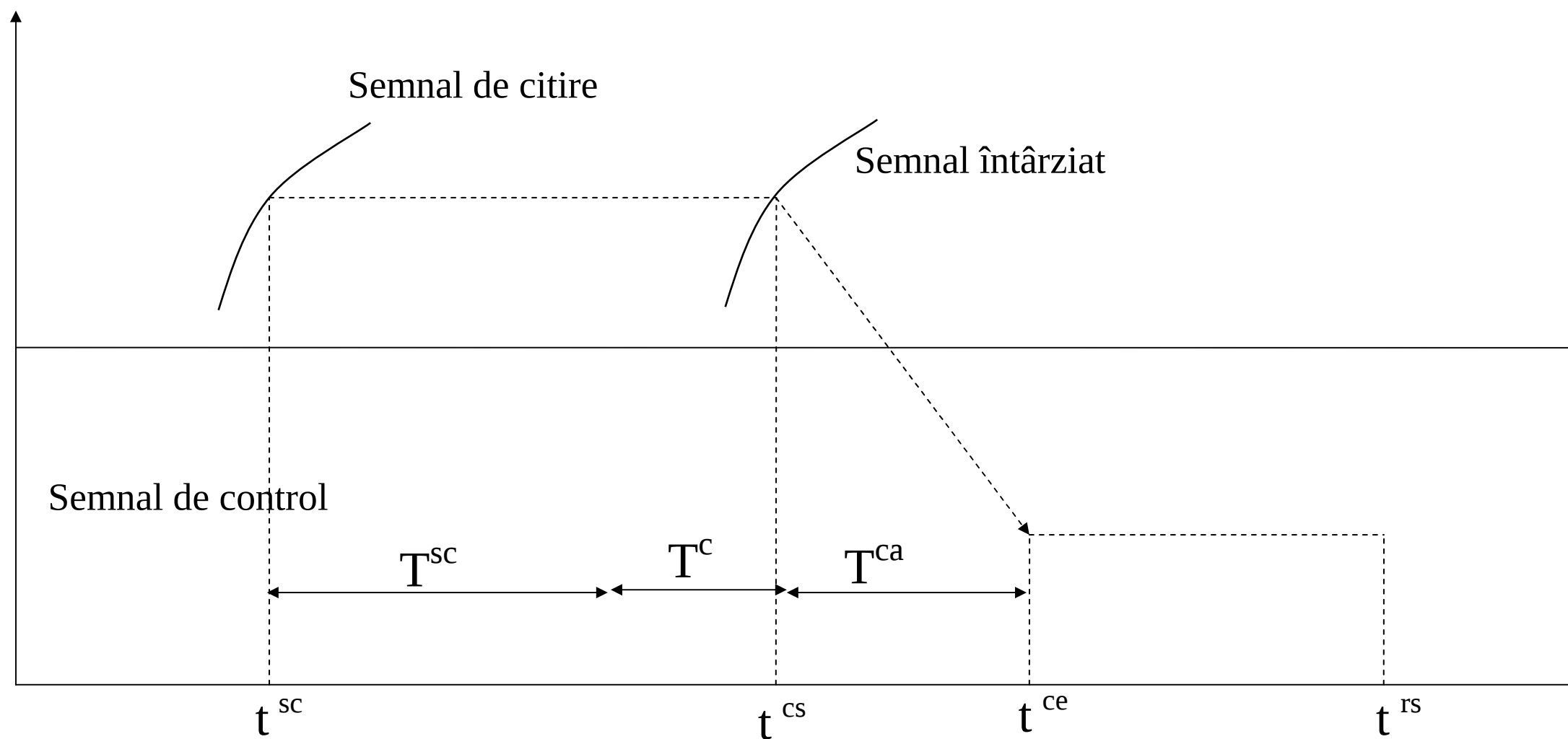
Fig. 10.3. Schema controlului direct la distanță.

T^{sc} : întârzierea sistemului de comunicație (de la senzor la controler) ; t^{sc} : momentul când senzorul împachetează datele

T^c : întârzierea de calcul a controlerului; t^{cs} : momentul când controlerul primește datele

T^{ca} : întârzierea controler - actuator ; t^{ce} : momentul când controlerul împachetează datele

Semnalul de ieşire $y(t)$ este eşantionat cu perioada T şi se obţine valoarea $y(k)$. Aceasta valoare se consideră constantă pe toată durata perioadei de eşantionare şi se transmite controlerului prin intermediul reţelei de comunicaţie sub forma mărimii $y(kT)$.



Notând cu:

- t^{sc} – momentul de timp când echipamentul îndepărtat încapsulează datele $y(kT)$ în cadre sau pachete pentru a fi transmise controlerului îndepărtat

- t^{cs} – momentul de timp când sosesc datele (cadrele sau pachetele) cu mărimile măsurate din proces la controler
- T^{sc} – durata transmiterii datelor (cadrelor sau pachetelor) de la senzori la controler,

se poate scrie relația

$$T^{sc} = t^{cs} - t^{sc}$$

Pentru exprimarea relației care descrie durata calculelor comenzilor controlerului către elementele de execuție se notează cu:

- t^{ce} - momentul de timp când controlerul încapsulează datele (comenzile în cadre sau pachete) și le transmite elementelor de execuție
- T_c - durata de procesare a mărimii de comandă obținându-se

$$T^c = t^{cs} - t^{ce}$$

Pentru exprimarea relației temporale care caracterizează transmiterea datelor prin rețea și aplicarea răspunsului controlerului se notează cu:

- t^{rs} – momentul de timp când sistemul îndepărtat startează aplicarea semnalelor de comandă (răspunsului controlerului)
- T^{ca} – întârzierea cu care se aplică semnalul de control elementului de execuție (procesului condus)

$$T^{ca} = t^{rs} - t^{ce}$$

Întârzierea cu care reacționează sistemul de control este:

$$\mathbf{T' = t^{rs} - t^{sc} = T^{sc} + T^{ca} + T^c}$$

Uzual durata de calcul T^c este mult mai mică decât T^{sc} și T^{ca} ceea ce permite ignorarea ei.

Se pot scrie următoarele relații:

$$T^{sc} = T^w + T^f + T^p$$

$$T^{ca} = T^w + T^f + T^p$$

unde:

- T^w – este durata de așteptare (waiting time) pentru punerea datelor în cadre sau pachete
- T^f – este durata de necesară pentru împachetarea datelor și startarea transmiterii cadrelor sau pachetelor
- T^p – este durata de propagare a cadrelor sau pachetelor prin rețea până la recepționarea lor de către destinatar

Duratele de mai sus nu sunt constante. Ele depind de factori ca:

- lărgimea de bandă alocată de protocol
- dimensiunea cadrelor sau pachetelor (depinzând de volumul datelor și informațiile auxiliare)
- încărcarea cu sarcini a sistemului de operare
- retransmiterea mesajelor în cazul detectării unor erori
- punerea la coadă de către comutatoare sau rutere etc.

Temă de casă:

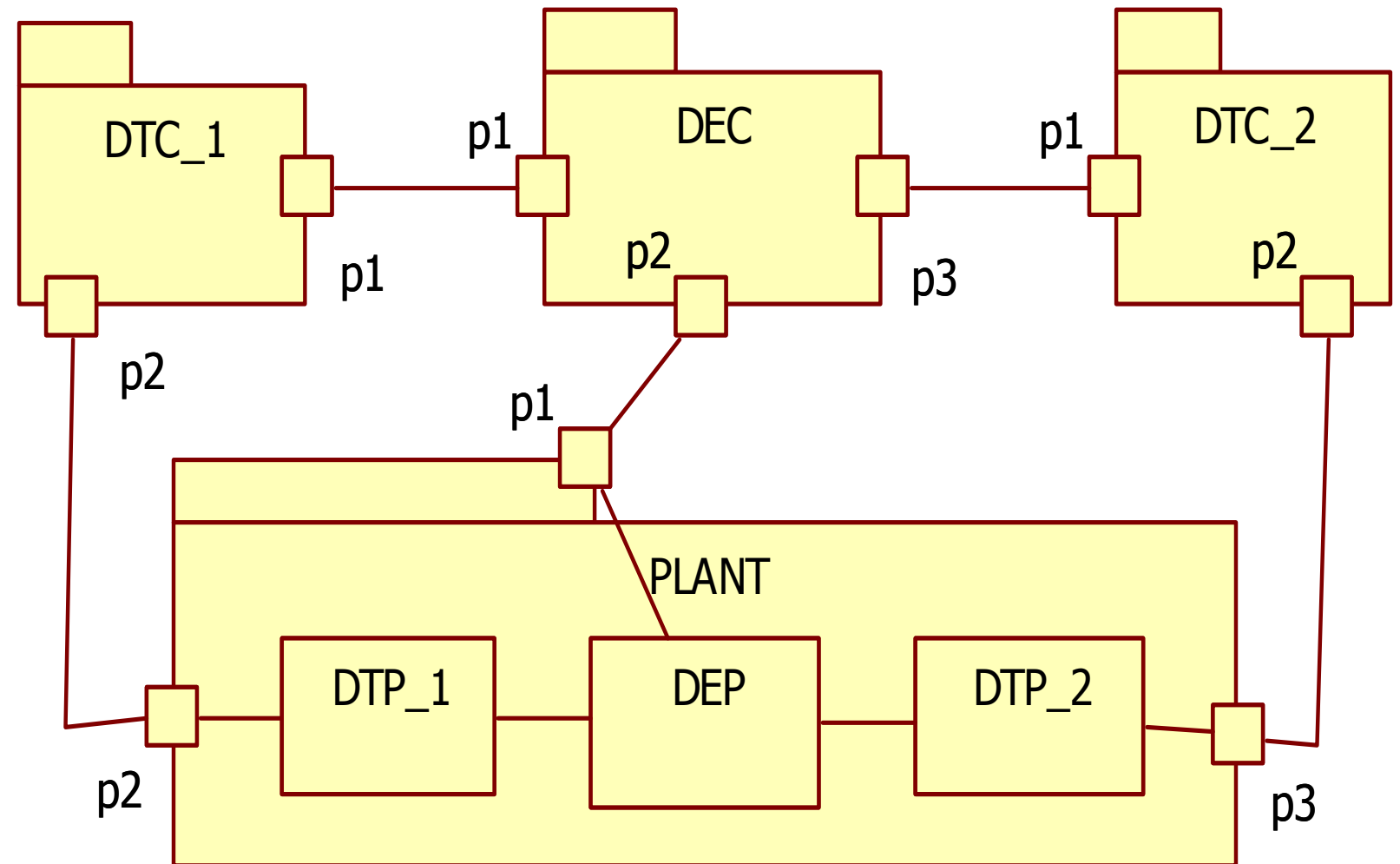
Controler simbiotic (DE-DT) pentru rețea hidro-electrică

Simbioza este o rețea mutuală a unor entități distincte și diferite.

Relația simbiotică a doi indivizi este una în care fiecare depinde de celălalt pentru supraviețuire sau îndeplinirea scopului.

Controlerele cooperative implică faptul că unele entități schimbă informații și se pun de acord asupra deciziilor de control pentru îndeplinirea scopului comun.

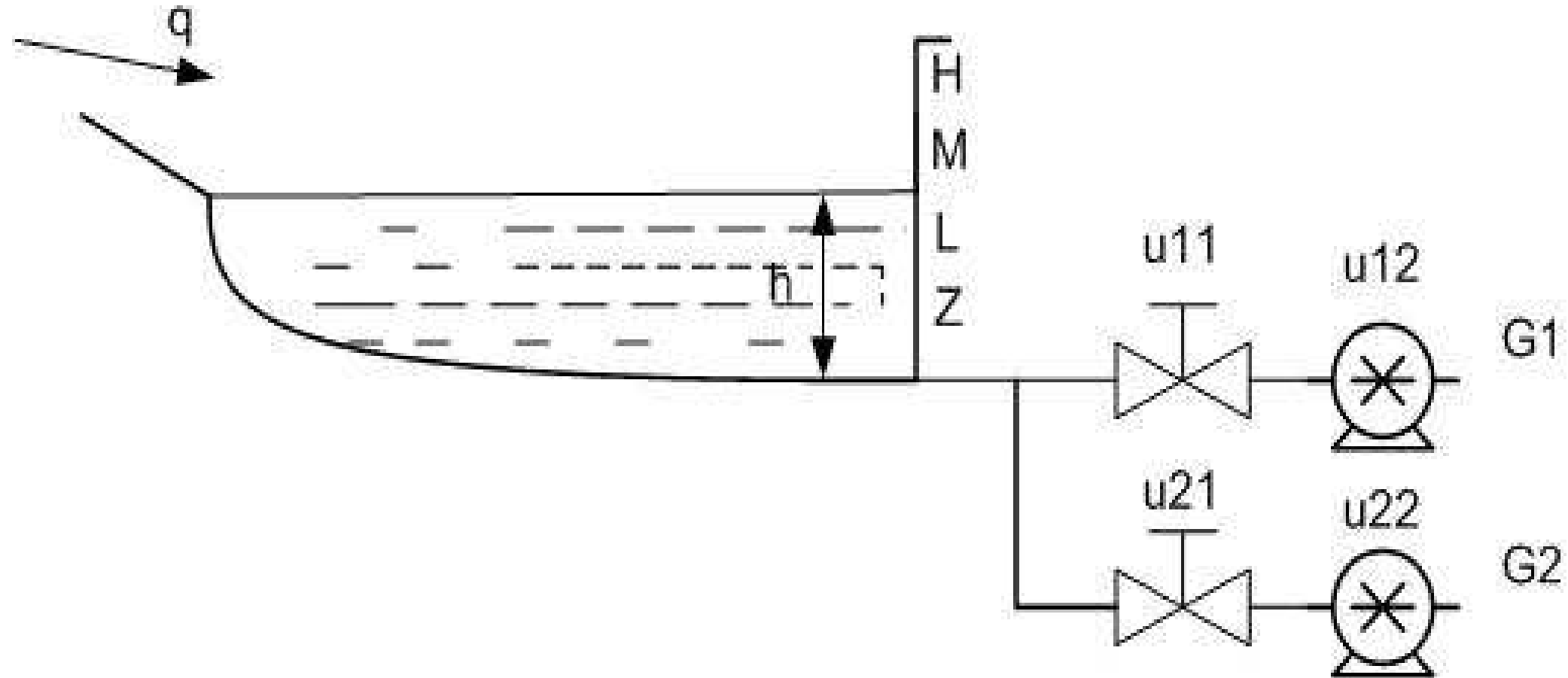
Controlul simbiotic este diferit de **controlul cooperative** pentru că în acest caz nu se schimbă informații și nu se cade de acord asupra unor decizii.



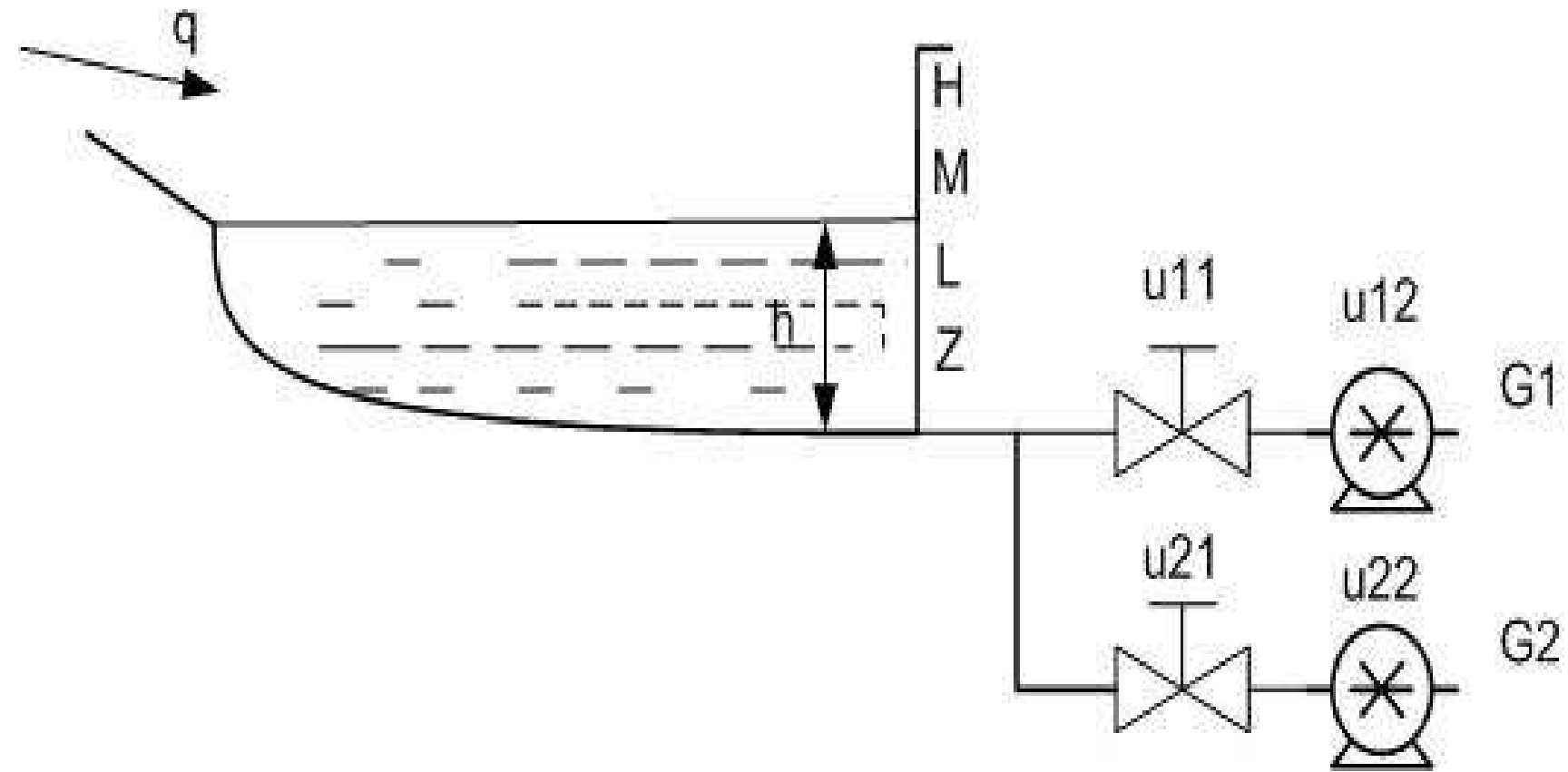
Specificațiile sistemului hidro-electric

Specificații:

Un system hidro-electric cu două generatoare electrice G1 și G2. Debitul de intrare q este variabil. Senzorii Z (zero), L (low), M (medium) și H (high) respectiv, semnalizează când nivelul apei din lac ajunge la poziția respectivă.



Sistemul de control primește informația de nivel a apei și trebuie să controleze ieșirea generatoarelor G1 și G2 folosind mărimile de control u_{11} , u_{12} , u_{21} și u_{22} , astfel încât parametrii de funcționare să îndeplinească cerințele. Ieșirea generatorului poate fi citită cu traductoare.



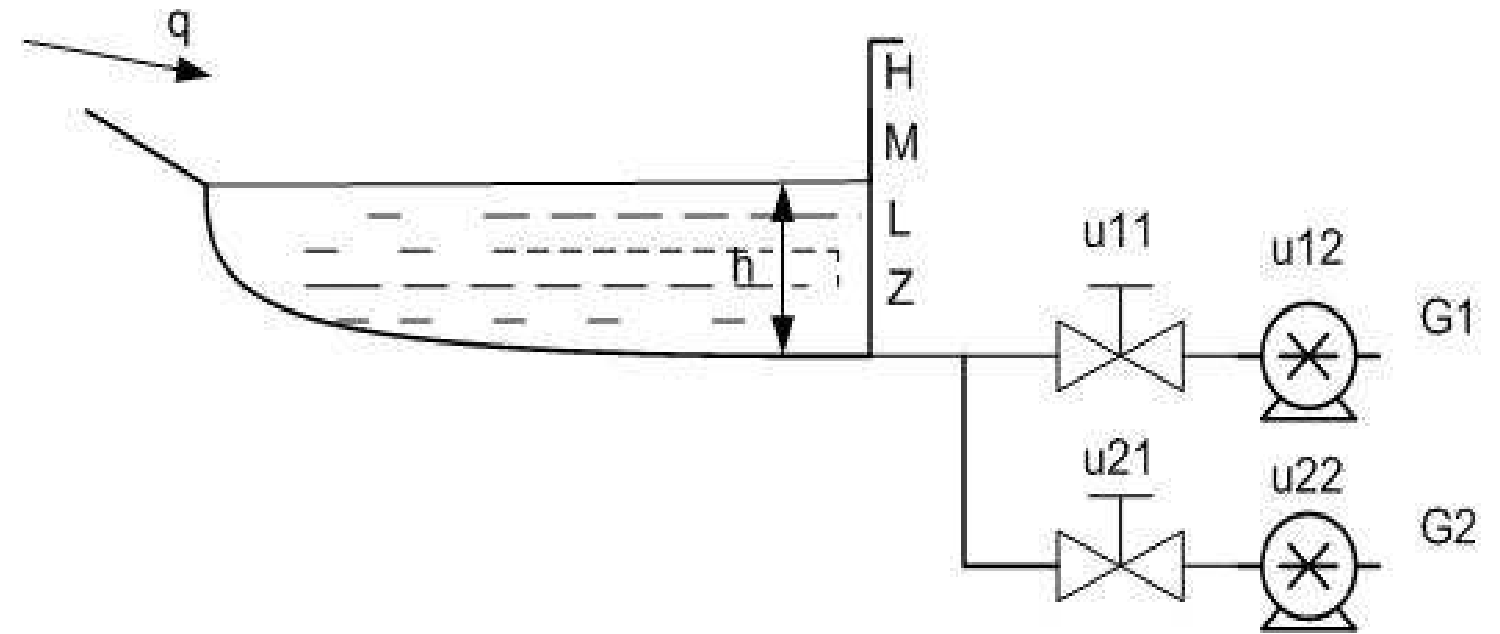
Fie e_Z , e_L , e_M , și e_H evenimentele semnalizate de senzorii Z, L, M și H.

Cerințe de control:

- când e_M start G1
- când e_H start G2
- când e_L stop G2
- când e_Z stop G1

Temă de casă – cerințe:

1. Modelul (TPN) al sistemului de control
2. Mașini de stare
3. Diagrama componentelor
4. Diagrama de comunicație
5. Diagrama secvențială
6. Diagrama spațiu-timp
7. Codul sursă relevant



Descrierea controlerului simbiotic

ETPN = Enhanced Time Petri Net

- are un set de locații de intrare $I = \{p_{i1}, \dots, p_{im}\}$ unde pot fi injectate jetoane de către o componentă exterioară – sunt folosite pentru a prinde evenimente exterioare de intrare
- are un set de locații de ieșire $O = \{p_{o1}, \dots, p_{on}\}$ care pot fi citite de către o componentă exterioară – sunt folosite pentru a semnaliza evenimente de control
- pot fi descrise folosind ETPNL (Enhanced Time Petri Nets based Language)

un sistem de control este compus dintr-un set de controlere $\mathbf{C} = \{C_1, C_2, \dots, C_c\}$
sistemul poate fi descris folosind un DTES (Discrete Time Equation System).

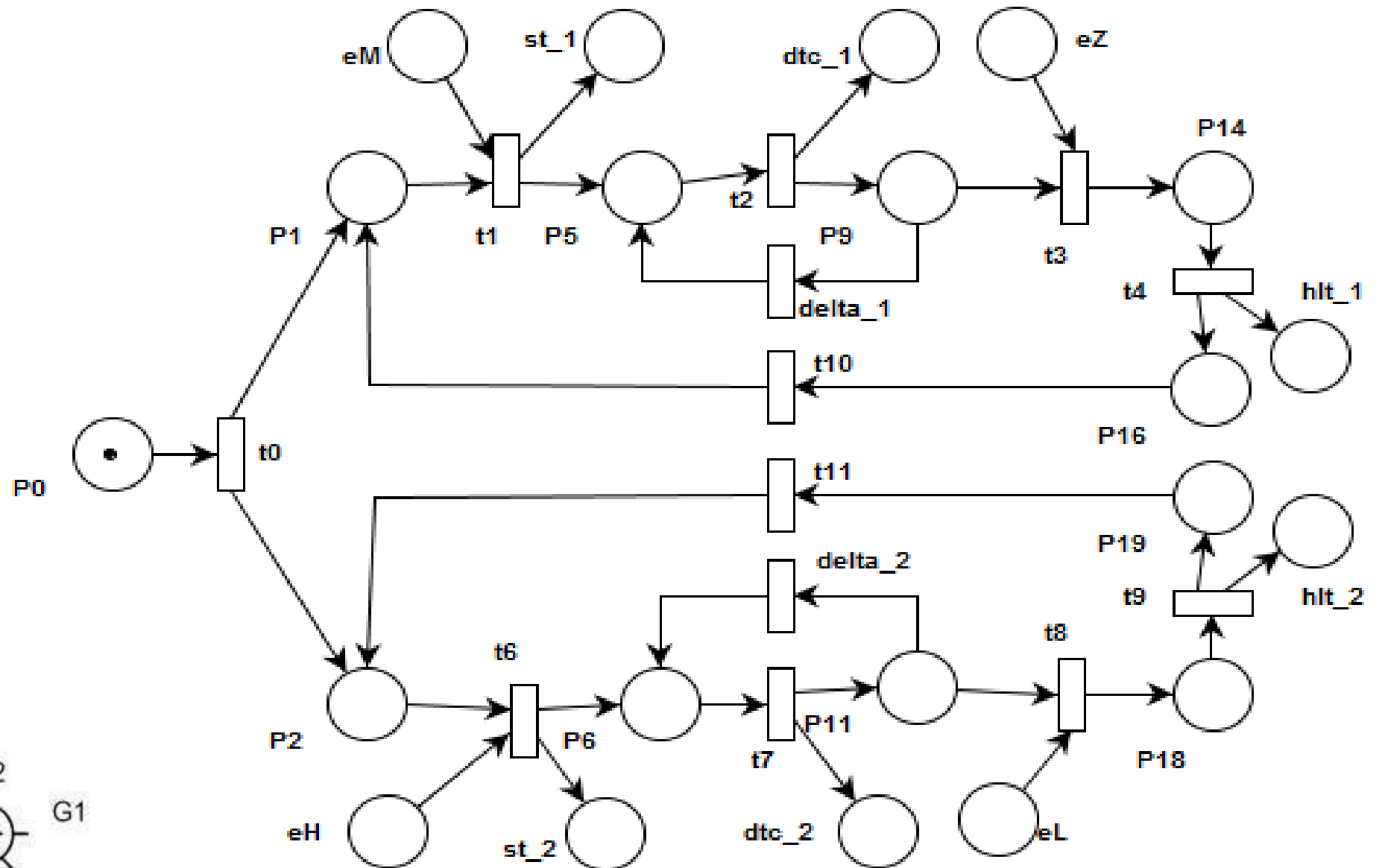
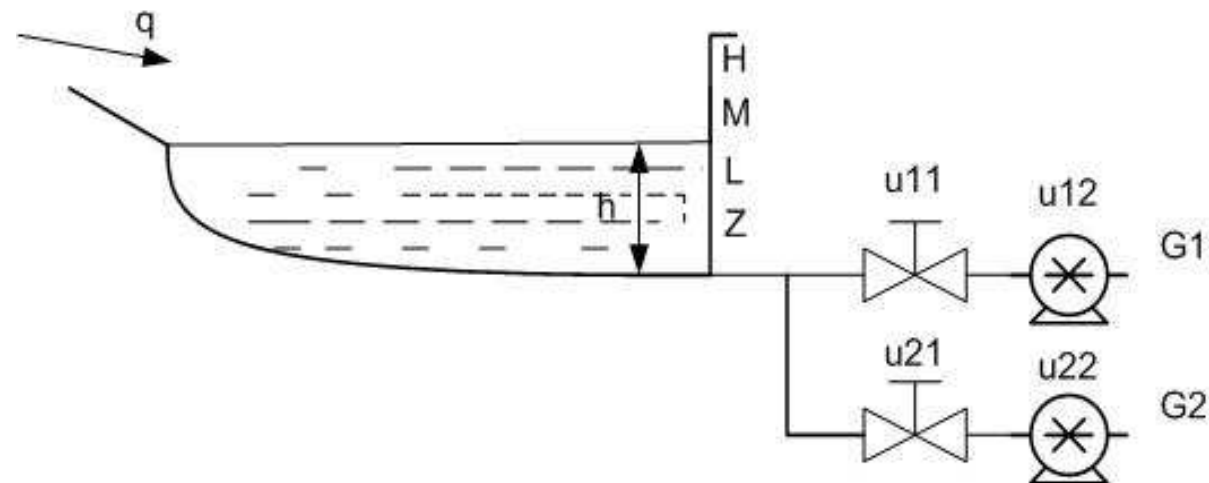
Fiecare controler C_k este descris printr-un DTES distinct, $DTES_k$. Ele pot fi executate cu perioadă diferită, acționate printr-o ramură a ETPN. Toate DTES sunt considerate componente distinct simbiotice.

Un DTES poate fi descris prin DTT (Discrete Time Tree)

Enhanced Time Petri Nets

DEC $\leftarrow$ TPN

DTC $\leftarrow$ Fuzzy Logic Controller



Cerințe de control:

- când e_M start G1
- când e_H start G2
- când e_L stop G2
- când e_Z stop G1



8.3. Evaluarea controlerului

Evaluarea DEC (discrete event controller)

Un DEC este descris prin TPNL.

Evaluare structurală

Obiective structurale

sob1. Dimensiunea (i. e. numărul de locații și tranziții):

$$n_d = |P| + |T|$$

sob2. Numărul de alternative: $n_a = |'+'|$

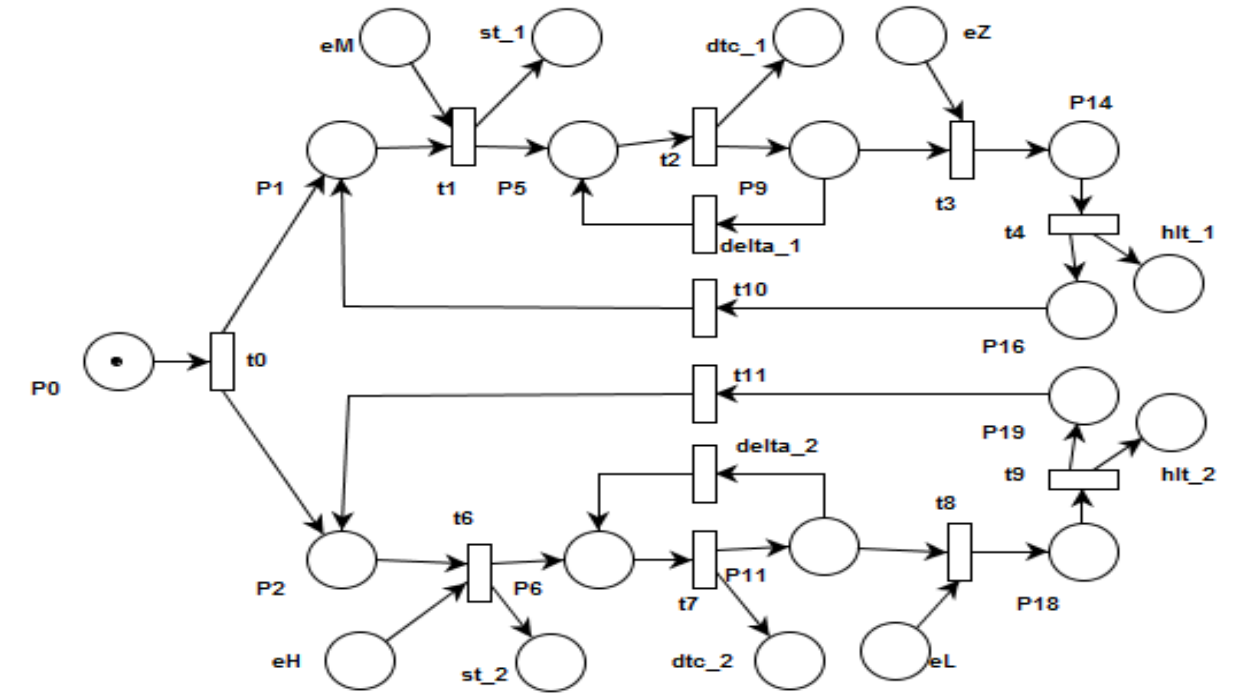
sob3. Numărul de bucle: $n_l = |' \# '|$

sob4. Numărul de secvențe concurente: $n_c = |' \& '|$

σ_1 = partea de sus ; σ_2 = partea de jos

$$\sigma_1 = (t_1(e_M, st_1) * (t_2(\varphi, dtc_1) \# t_5(\delta_1, \varphi)) * t_3(e_Z, \varphi) * t_4(\varphi, hlt_1)) \# t_{10}(\varphi, \varphi)$$

$$\sigma = \sigma_1 \& \sigma_2$$



Evaluarea DTC

Obiective structurale

Evaluarea topologiei

tob1. Dimensiunea (i. e. numărul de funcții): n_f

tob2. Numărul de operatori secvențiali: $n_s = |' * '|$

tob3. Numărul de bucle: $n_l = |' \# '|$

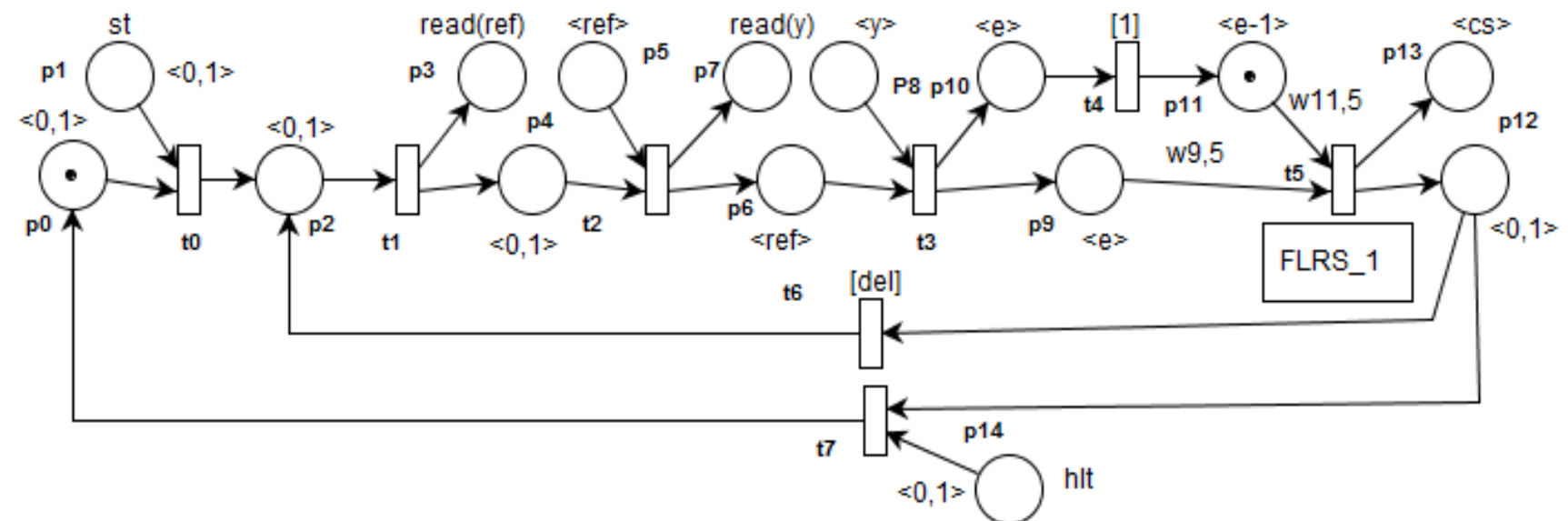
tob4. Numărul de secvențe concurente: $n_c = |' \& '|$

tob5. Numărul de alternative: $n_a = |' ? '|$

Complexitatea funcțională și încărcarea procesorului

cof1. Complexitatea este exprimată prin numărul de operatori folosiți de funcția curentă: o_f

Suma complexităților funcționale poate fi folosită pentru a evalua complexitatea DTC.



Obiectivele comportamentului de control (Control Behavioral Objectives)

Obiectivele comportamentului temporal (Temporal behavioral objective)

cbo1. Timpii de răspuns: $\delta_i = t_o - t_i$; $\delta_i \leq \Delta_i$ (pentru aplicații de timp real)
unde

- t_i momentul când apare evenimentul e_i
- t_o momentul când evenimentul de control (răspuns) este semnalizat
- Δ_i deadline (constrângere temporală) pentru răspunsul i

Evaluarea comportamentală a sistemului

Obiective comportamentale

Notatii:

- $\mathbf{X}(t)$, $t=0,1,2,\dots$ starea (vectorială) a sistemului
- $\mathbf{Y}(t)$, $t=0,1,2, \dots$ ieșirea (vectorială) a sistemului
- $\mathbf{u}(t)$, $t=0,1,2,\dots$ semnalul de control (ieșire din controler și intrare în sistemul controlat)
- $\mathbf{R}(t)$, $t=0,1,2,\dots$ referința (vectorială) cu dimensiunea egală cu cea a lui $\mathbf{X}(t)$
- $\mathbf{v}(t)$, $t=0,1,2,\dots$ perturbația (vectorială) sau mărimea de intrare necontrolată
- $\mathbf{C}_1$, $\mathbf{C}_2$ constrângerile (vectoriale). Sunt descrise prin $\mathbf{C}_1(\mathbf{v}(t))$ și $\mathbf{C}_2(\mathbf{v}(t))$ dependența vectorului perturbație trebuie specificată
- t_c și t_f - timp finit ($0 \leq t_c < t_f$); timpul poate fi discret sau continuu.

Considerând perturbația de intrare $\mathbf{v}(\mathbf{t})$, $t=0, 1, 2, \dots$, controlerul aplică mărimile de control $\mathbf{u}(\mathbf{t})$, $t=0,1,2,\dots$ astfel încât:

bob1. Constrângeri de stare: $\exists t_c, \forall t \geq t_c, C_1 \leq X(t) \leq C_2$ sau
 $\exists t_c, \forall t \geq t_c, C_1(v(t)) \leq X(t) \leq C_2(v(t))$

bob2. Deviația stării față de o referință: $J = \int_{t_c}^{t_f} \|X(t) - R(t)\| dt \leq \epsilon$ sau

forma discretă $J = \sum_{t=t_c}^{t_f} \|X(t) - R(t)\| \leq \epsilon$

cu ϵ o constant mică. Uneori ieșirea $\mathbf{Y}(\mathbf{t})$ este folosită în locul lui $\mathbf{X}(\mathbf{t})$ în funcțiile de mai sus.

Mulțimi temporale de test (test time sets)

Fie $T = \{t_1, t_2, \dots\}$ o mulțime de momente temporale și $\delta(t)$, $t \in T$ setul de întârzieri corespondente.

O mulțime de test *test set* este $(v(t), \mathbf{R}(t+\delta(t)))$, $t \in T$.

Notatii:

- $\mathbf{R}(t+\delta(t))$ vectorul de ieșire dorit (referință)
- $\delta(t)$ este întârzierea maximă cu care ieșirea ajunge la referința $\mathbf{R}(t+\delta(t))$ – ca valoare vectorială.

Observații:

- dacă sistemul este de timp discret, mulțimea momentelor de test îndeplinește relația $t_i = t_{i-1} + \tau$ cu τ constant.
- Un model poate fi folosit pentru construirea lui $\mathbf{R}(t + \delta(t))$
- Dacă sistemul este cu evenimente discrete, vectorul $\mathbf{v}(t)$ are elemente corespunzătoare la momentele când apare fiecare eveniment
- Pentru sisteme cu timp discret, vectorul $\mathbf{v}(t)$ are valori numerice.
- Mai multe mulțimi de test pot fi folosite pentru evaluarea comportamentului unui sistem.

Problema 1. Identificarea

Se dă un sistem cu intrarea controlată $\mathbf{u}$, necontrolată $\mathbf{v}$, ieșirea $\mathbf{Y}$ (**toate vectoriale**), $\mathbf{X}(0)$ starea inițială

- $T = \{t_1, t_2, \dots\}$ mulțimea de momente temporale de test (test time set)
- $\delta(t)$, $t \in T$ mulțimea întârzierilor
- $\Theta_i = (\mathbf{v}(t), \mathbf{R}(t + \delta(t)))$, $t \in T$, $i=1,2, \dots, m$

Se cere modelul sistemului care minimizează funcția de performanță
 $J = \sum_{t \in T} \|\mathbf{Y}(t) - \mathbf{R}(t)\|$

Problema 2. Sinteza controlerului

Se dă un sistem cu intrarea controlată $\mathbf{u}$, necontrolată $\mathbf{v}$, ieșirea $\mathbf{Y}$ (**toate vectoriale**), $\mathbf{X}(0)$ starea inițială

- $T = \{t_1, t_2, \dots\}$ mulțimea de momente temporale de test (test time set)
- $\delta(t)$, $t \in T$ mulțimea întârzierilor
- $\Theta_i = (\mathbf{v}(t), \mathbf{R}(t \delta(t)))$, $t \in T$, $i=1,2, \dots, m$

Se cere controlerul care minimizează funcția de performanță
 $J = \sum_{t \in T} \|\mathbf{Y}(t) - \mathbf{R}(t)\|$

Observație: nu s-a menționat dacă sistemul este de timp discret sau cu evenimente discrete. Dacă sistemul este hibrid, vectorii de intrare, stare și ieșire, precum și vectorul cu momente de test trebuie definite pentru ambele tipuri de subsisteme. Controlerele sintetizate trebuie să fie simbiotice.

Problema 3. Atingerea stării finale dorite (target state)

Sistemul trebuie să atingă starea finală $\mathbf{X}(t_f)$ la momentul t_f . Problema este calculul mărimilor de intrare $\mathbf{u}(t)$, $t=0,1,2,\dots$ care conduc sistemul la starea dorită. Testele sunt:

- $T = \{t_1, t_2, \dots\}$ mulțimea de momente temporale de test (test time set)
- $\delta(t)$, $t \in T$ mulțimea întârzierilor
- $\Theta_i = (\mathbf{v}(t), \mathbf{R}(t \delta(t)))$, $t \in T$, $i=1,2, \dots, m$

Problema poate fi definită cu momentul final t_f specificat sau liber.

Problema 4. Îndeplinirea unui task

Sistemul trebuie controlat în așa fel încât un set de task-uri (care au nevoie de resurse) trebuie îndeplinite. Problema constă în planificarea resurselor sistemului în așa fel încât task-urile să ajungă în starea finală specificată

De exemplu:

- Controlul traficului feroviar
- Controlul unui sistem de fabricație

8.4. Metode ale rețelelor de control

Networked Control Methods

Sisteme continue și liniare

Metoda punerii la coadă

Metoda de punere la coadă (*queueing methodology*, Luck și Ray, 1994) poate fi folosită pentru remodelarea aleatoare a întârzierilor rețelei dintr-un sistem de control determinist astfel încât acestea să devină invariante în timp. Metoda de control a sistemelor organizate în rețea prin punere la coadă se bazează pe utilizarea informațiilor deterministe și probabiliste în algoritmul de control.

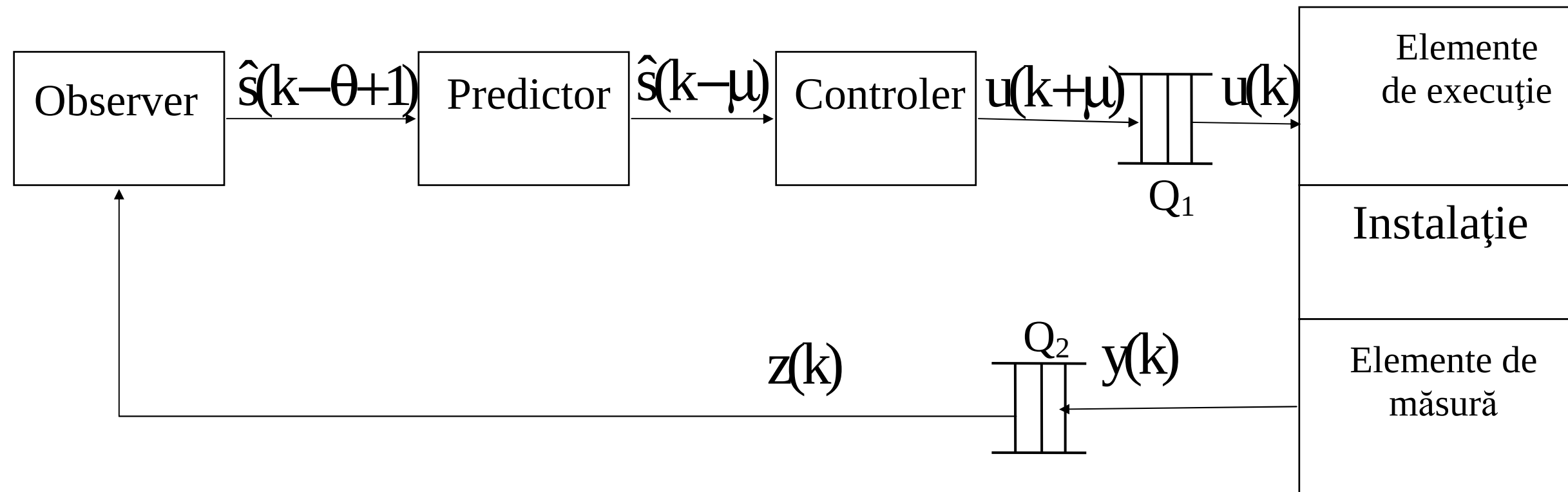


Fig. 10.6. Schema metodei deterministe de compensare a întârzierilor bazată pe predictor.

Discrete Time Systems – Linear robot control

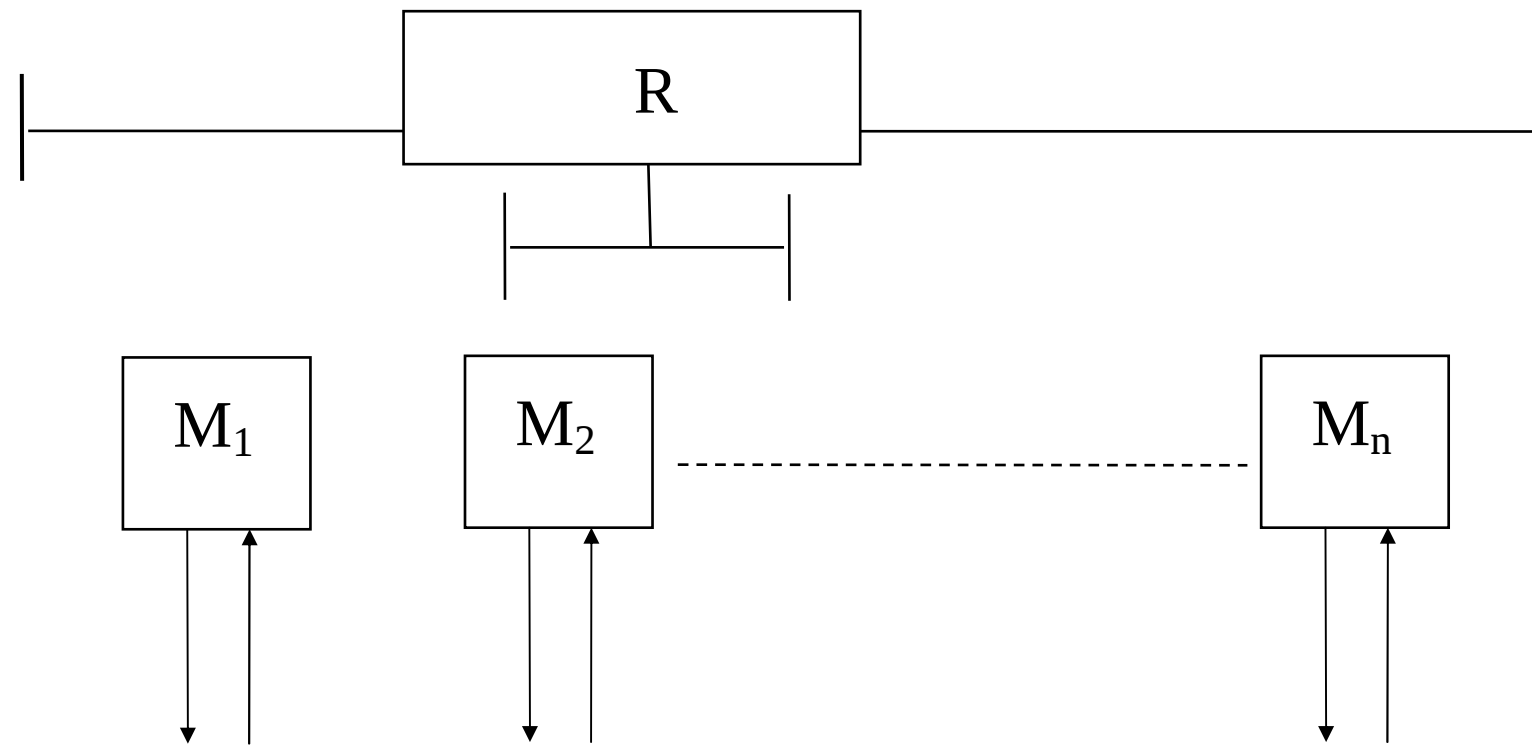


Fig. 10.8. Structura sistemului cu robotul liniar controlat

1) Prima variantă este mai simplu de controlat, dar la sarcini mari (containere sau piese grele), viteza de deplasare fiind constantă, pot apărea oscilații care scad performanțele de comportament ale sistemului. Se poate obține diminuarea oscilațiilor prin scăderea vitezei de deplasare, dar aceasta duce de asemenea la scăderea performanțelor sistemului, deoarece crește timpul de îndeplinire a comenzilor planificate.

2) Cea de a doua variantă de realizare se poate implementa cu tensiunea de alimentare a motorului constantă sau variabilă. În primul caz, robotul se deplasează cu viteză constantă și apare aceeași problemă ca în varianta anterioară. În cel de al doilea caz, tensiunea trebuie să varieze (să fie controlată pornirea și oprirea) astfel încât robotul să ajungă cât mai repede în poziția dorită și cu o abatere cât mai mică.

Senzorul sau traductorul poate fi de următoarele tipuri:

- traductor de turație – care furnizează o tensiune sau un curent proporțional (aproximativ liniar) cu turația momentană a motorului
- traductor de deplasare – care furnizează informații despre poziția (liniară a) robotului sau numărul de rotații (sau grade); uzual acestea sunt furnizate prin impulsuri.

Sistemul de control se compune dintr-un planificator care planifică secvențele de deplasări ale robotului și un controler care le implementează. Structura sistemului de control este reprezentată în figura 10.9. Planificatorul poate da soluții constând din direcția de deplasare a robotului și numărul de marcaje peste care trebuie să treacă, sau punctul de pe axă la care trebuie să ajungă.

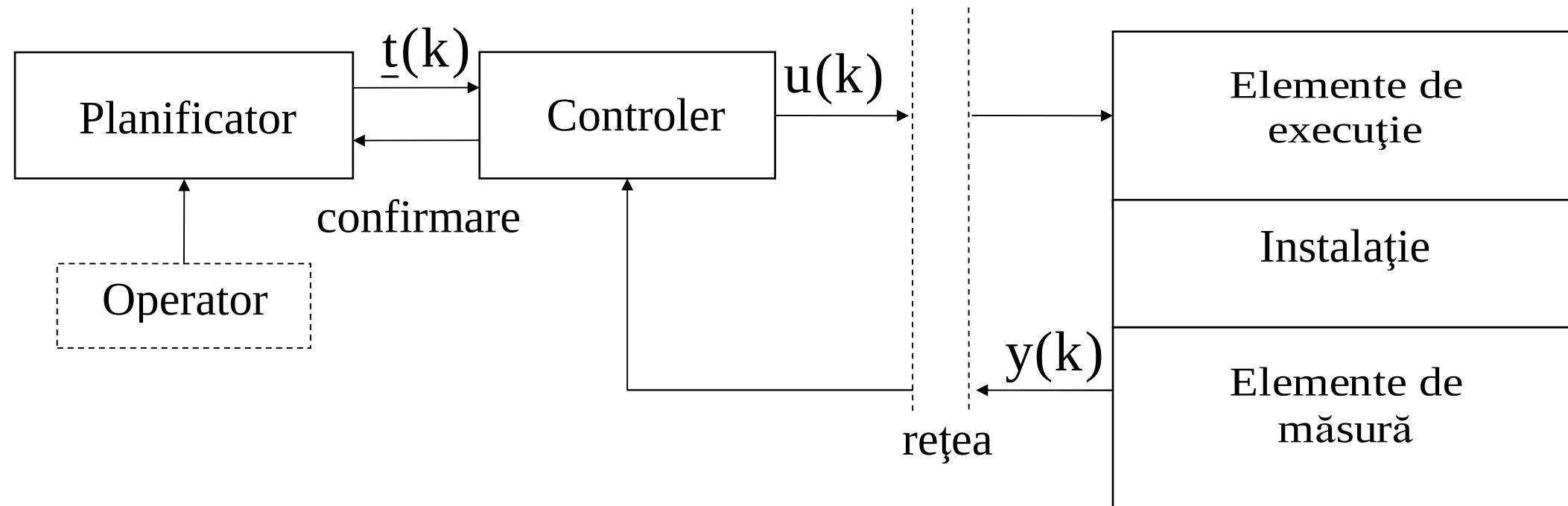


Fig. 10.9. Structura sistemului de control a robotului liniar.

În funcție de soluția planificatorului, controlul se poate implementa prin următoarele variante:

a) Locurile de oprire ale robotului sunt semnalizate de către un senzor (detector) montat pe robot. Există câte un marcaj în fiecare poziție în care trebuie să se oprească robotul, sau un set de marcaje pe un disc montat rigid pe rotorul motorului. Senzorul joacă rolul unui traductor de deplasare.

b) Controlerul calculează și memorează în permanență poziția în care este robotul. El primește de la planificator noua poziție de pe axă la care trebuie să se oprească robotul. Controlerul calculează direcția și distanța de deplasare, iar apoi trebuie să alimenteze motorul astfel încât robotul să se deplaseze în noua poziție. Datorită cumulării erorilor de poziționare, se recomandă instalarea unui senzor pentru recalibrarea robotului din când în când.

Se consideră că:

- se utilizează controlul ca reacție la detectarea unui marcaj
- robotul se deplasează tot timpul cu viteza constantă v
- întârzierile robotului la pornire sau oprire, precum și durata de aplicare a comenzilor de oprire sunt neglijabile.

Pentru o poziționare a robotului cu precizia d trebuie ca, din momentul în care senzorul semnalează marcajul de oprire și până în momentul în care se aplică impulsul de oprire, durata dintre cele două impulsuri să fie mai mică decât d/v unități de timp.

Prin urmare, în cazul în care controlerul se află la distanță față de robot, trebuie ca duratele de transmitere (prin rețea) a datelor care conțin comenzile să îndeplinească următoarea condiție:

$$T^{sc} + T^{ce} < d/v$$

unde T^{sc} și T^{ce} reprezintă duratele de transmitere a datelor de la senzor la controler și respectiv de la controler la elementul de execuție.

Dacă această condiție nu este îndeplinită, trebuie să se instaleze controlerul lângă robot, iar planificatorul poate fi lângă operator (adică la distanță).

Cazul al doilea este când:

- sistemul are un motor de curent continuu comandat prin tensiune
- traductorul furnizează informații despre turația motorului sub forma unei tensiuni proporționale cu aceasta
- controlerul este (numeric) de tip PI

Se cere să se determine constantele controlerului astfel încât să ajungă în cel mai scurt timp în poziția dorită. Prin urmare, planificatorul furnizează referința ca deplasare, dar mărimea măsurată este turației. Trebuie efectuată o conversie a turației în deplasare (realizată) și comparată cu referința (deplasarea) dată. Aceasta reprezintă eroarea căreia urmează să i se aplice algoritmul de control PI, care furnizează ca și comandă o valoare proporțională cu tensiunea ce urmează să se aplice motorului. Performanța sistemului se măsoară prin suma modulelor abaterilor de poziționare.

Soluții pentru evitarea efectului întârzierilor variabile

Există două tipuri de soluții pentru evitarea efectului întârzierilor variabile ale informațiilor de control:

- adaptarea vitezei schimbului de informații la posibilitățile rețelei de comunicație și
- evitarea apariției întârzierilor.

Primul tip de soluții se bazează pe acordarea (adaptarea) perioadei de transmitere a informațiilor cu posibilitățile curente ale rețelei.

Aceasta implică modificarea tactului de eșantionare a sistemului discret. Modificarea tactului de eșantionare a controlerelor implică modificarea parametrilor acestora. Efectuarea acestor operații on-line este dificilă, dar se poate rezolva prin calcularea off-line a parametrilor controlerului pentru diferite perioade de eșantionare. La modificarea tactului de eșantionare, datorită traficului, trebuie să se schimbe parametrii algoritmului de control cu cei corespunzători perioadei curente de control.

Pentru evitarea întârzierilor variabile în transmiterea pachetelor (sau cadrelor) se pot utiliza protocoale pentru alocarea lăţimii de bandă. Schema de alocare a lăţimii de bandă pentru sistemele de control distribuit se bazează pe o tehnică statică care garantează performanţele de control la preţul ocupării permanente a lăţimii de bandă disponibilă. Această tehnică statică de alocare poate să fie neeficientă în condiţiile în care se creşte volumul de date transmis de sistemul de control.

Problema 2:

Concepeți o soluție bazată pe OETPN pentru problema de control a robotului.

Specificații:

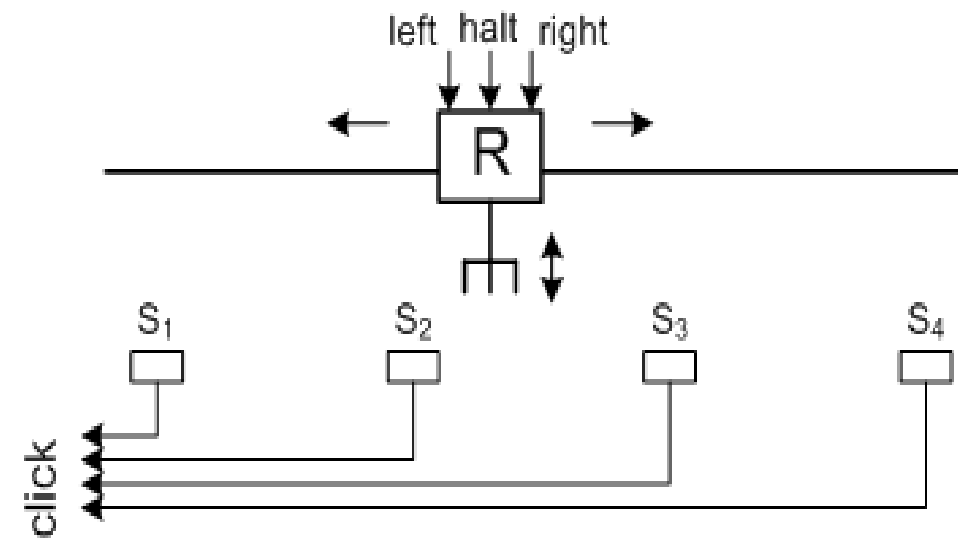
Operatorul dă o secvență de mișcări și timpi de așteptare.

C convertește secvența în semnale de control pentru robot.

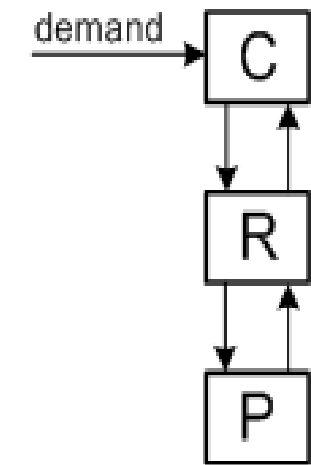
Întârzieri de transmisie C-R variable: 4- 20 minute.

Mișcarea robotului (viteză constantă) de la o poziție de la alta: 10-20 secunde.

Acuratețea poziționării: 0.1 secunde.

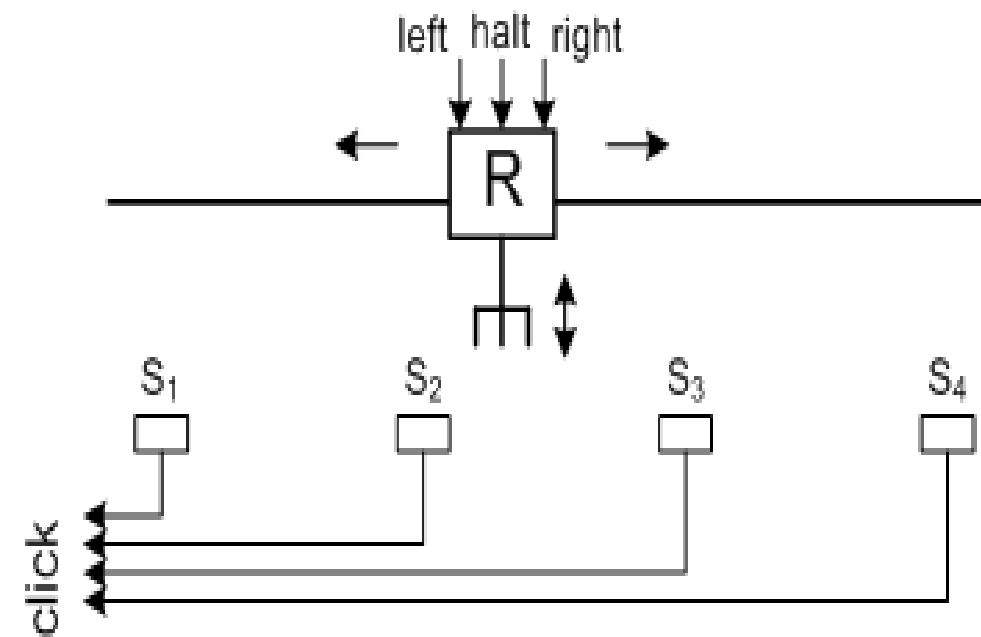
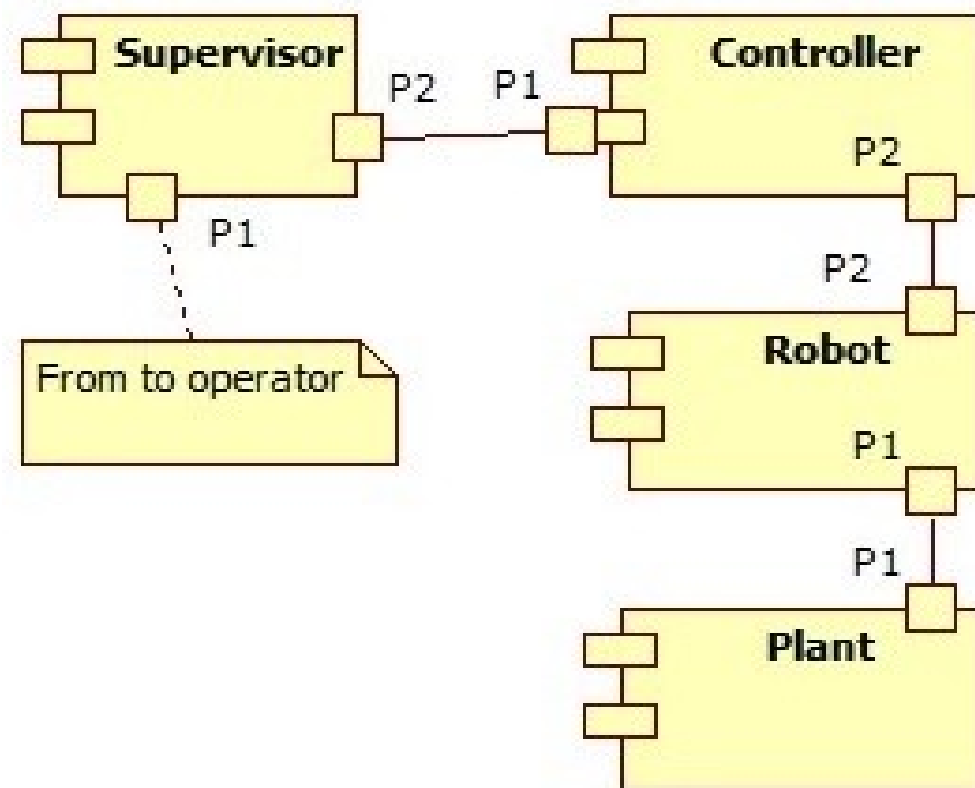


a) The linear robot and its signals

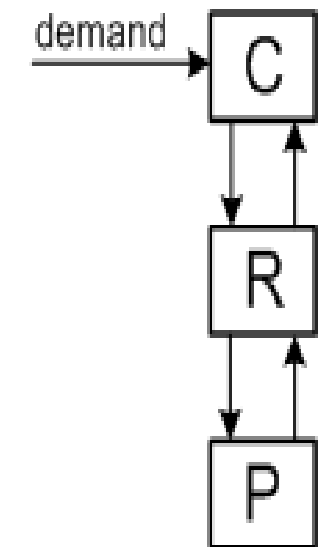


b) The robot environment

Teleoperare



a) The linear robot and its signals



b) The robot environment

Specificații

Robotul R trebuie să se miște de la poziția inițială la pozițiile p1, p2, p3 sau p4 și să rămână acolo niște durate stabilite.

Secvența de mișcări date și timpii de așteptare sunt dați de expresia:

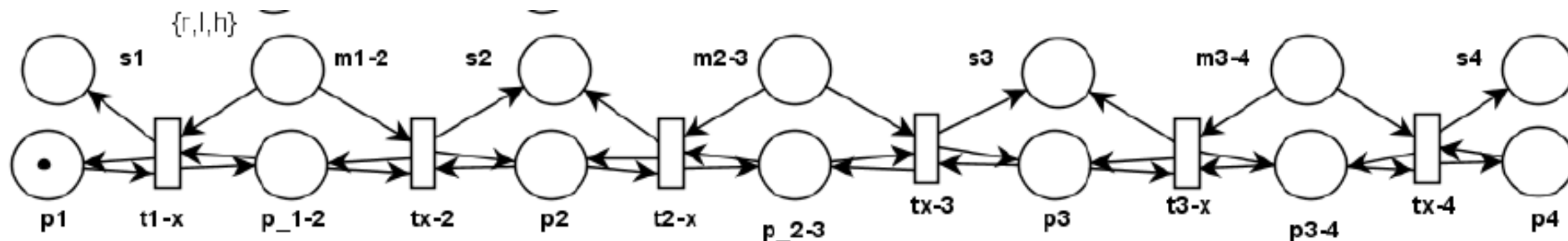
$$\sigma^D = p1[\tau1-1] * p3[\tau3-1] * p1[\tau1-2] * p4[\tau4-1] * p2[\tau2-1] * \dots$$

unde [a] reprezintă durata de așteptare în locația menționată.

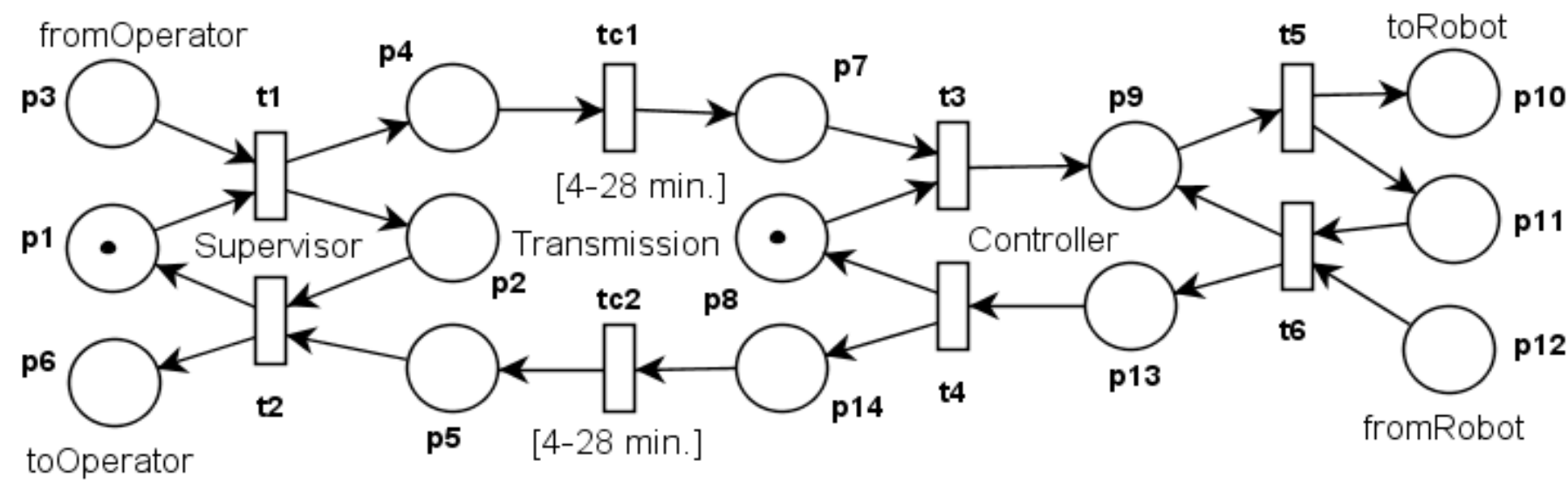
Activitățile executate de componente conform diagramei de mai sus sunt:

- Supervizorul calculează (stabilește) secvența de tranziții σ^T care trebuie executate de Plant pentru a atinge stările cerute
- $\sigma^D = t_{1-x}[\tau_{1-1}] * t_{x-2} * t_{2-x} * t_{x-3} * t_{x-3}[\tau_{3-1}] * t_{2-x} * t_{x-2} * t_{1-x} * t_{1-x}[\tau_{1-2}] * t_{x-2} * \dots * t_{x-4}[\tau_{4-1}] * \dots [\tau_{2-1}] * \dots$
- Controlul parsează secvența de tranziții primită și transmite robotului direcția următoarei mișcări $\{l$ pentru stânga (left), r pentru dreapta (right) și h pentru oprire (halt)} și timpul de așteptare atunci când acesta este necesar
- Robotul primește și execută fiecare pereche (direcție, durată) pentru a realiza mișcarea cerută
- Orice componentă care primește o cerere trimite înapoi informații despre îndeplinirea sarcinii

Soluții bazate pe modelul OETPN

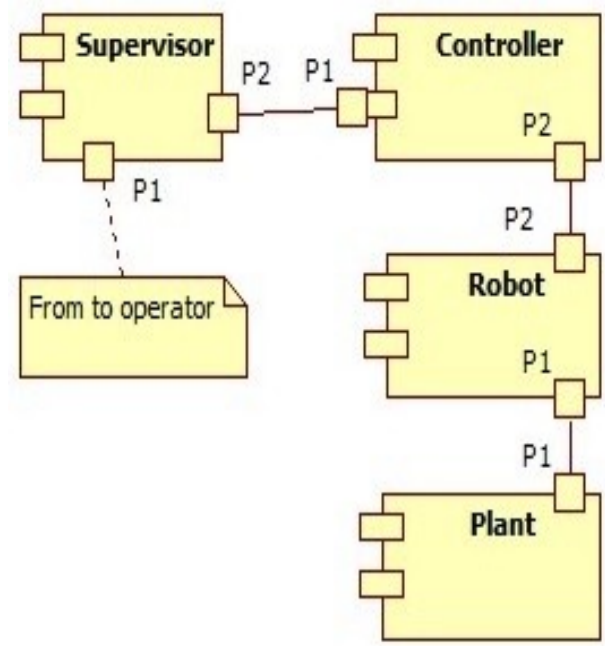
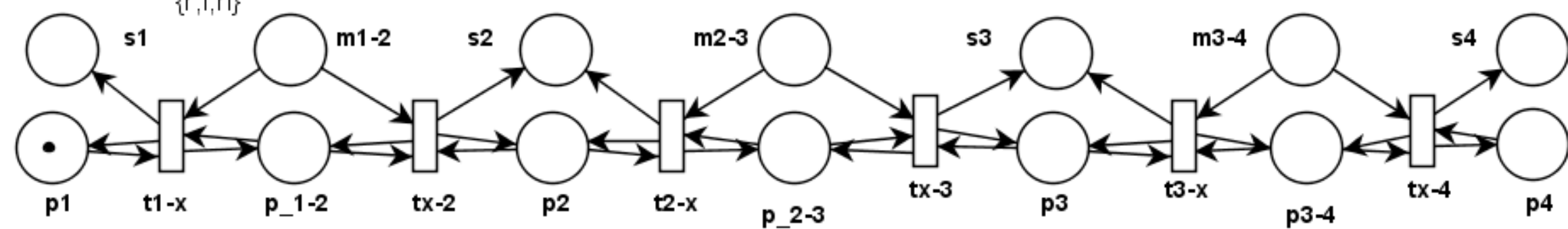
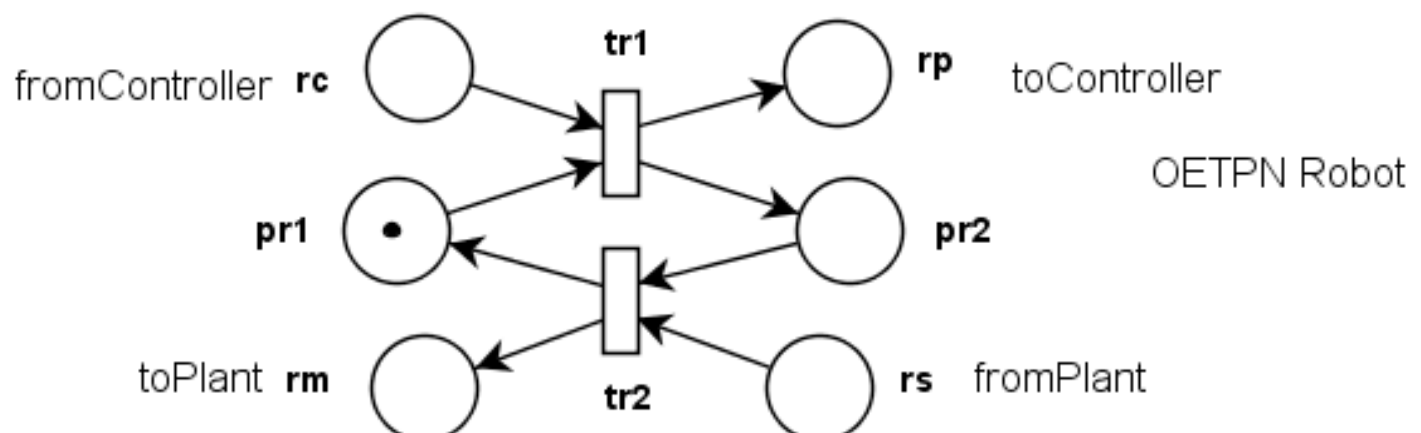


Prima soluție

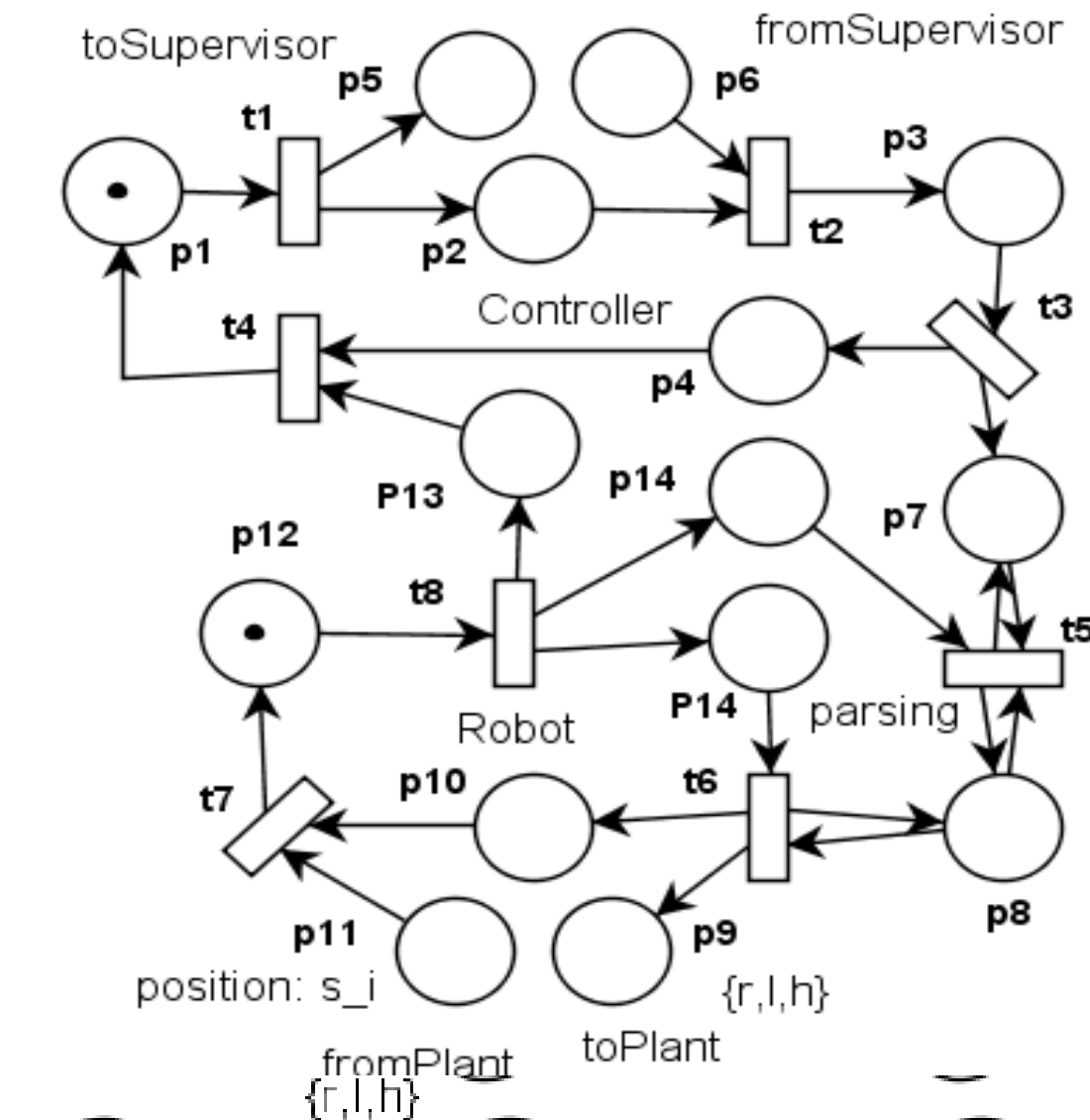


Tranziție	Semnificație
t1	compiling
t5	parsing
t4	send Robot state

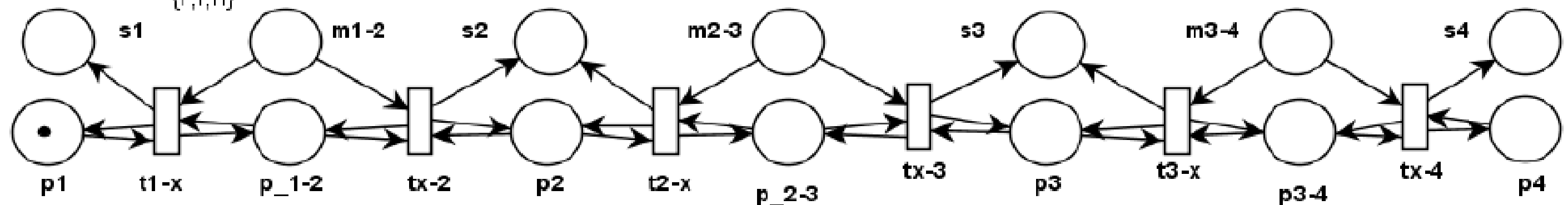
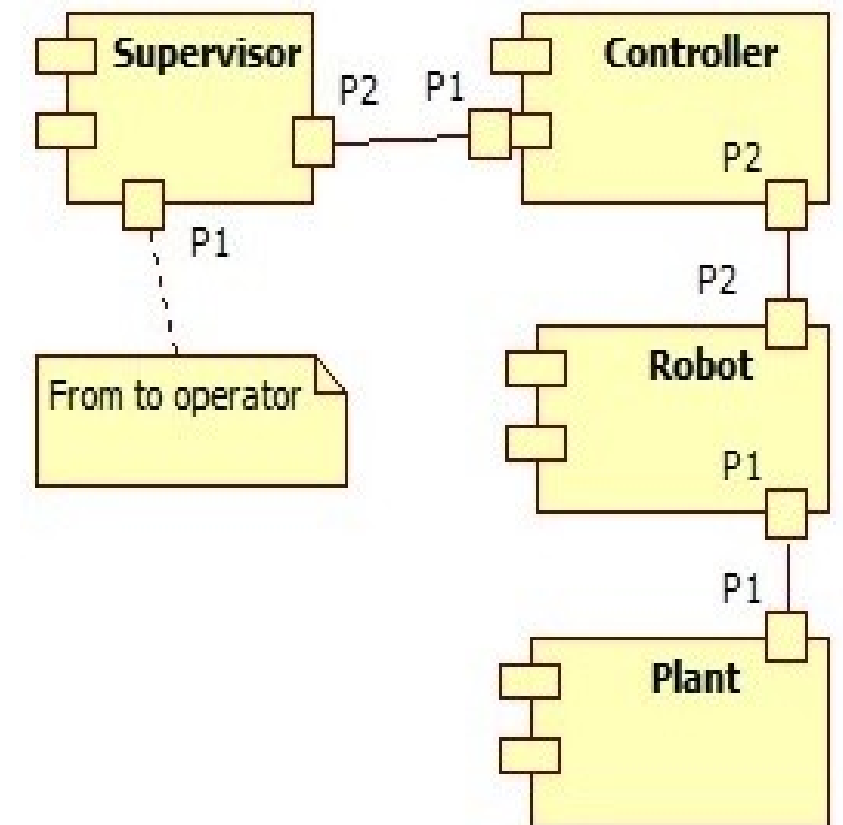
Temă de casă:
Adăugați tipurile locațiilor (i.e. clasele) și relațiile de evoluție



A doua soluție



Temă de casă:
Adăugați tipurile
locațiilor (i.e.
clasele) și relațiile
de evoluție



*

** END **

*