Modeling and Automation of Industrial Processes

Modelação e Automação de Processos Industriais / MAPI

Supervised Control of Discrete Event Systems - SCADA

http://www.isr.tecnico.ulisboa.pt/~jag/courses/mapi2223

Prof. Paulo Jorge Oliveira, original slides
Prof. José Gaspar, rev. 2022/2023

Syllabus:

...

Chap. 2 – Discrete Event Systems

Chap. 5a – Supervised Control of DESs

* SCADA
* Methodologies for the Synthesis of Supervision Controllers
* Failure detection

...

Some pointers on Supervised Control of DES

History: The SCADA Web, http://members.iinet.net.au/~ianw/
Monitoring and Control of Discrete Event Systems, Stéphane Lafortune, http://www.ece.northwestern.edu/~ahaddad/ifac96/introductory_workshops.html

Tutorial: http://vita.bu.edu/cgc/MIDEDS/
http://www.daimi.au.dk/PetriNets/

Analysers & Simulators: http://www.nd.edu/~isis/techreports/isis-2002-003.pdf (Users Manual)
http://www.nd.edu/~isis/techreports/spnbox/ (Software)

Bibliography:
* SCADA books http://www.sss-mag.com/scada.html
* K. Stouffer, J. Falco, K. Kent, "**Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security**", NIST Special Publication 800-82, 2006
* Moody J. e Antsaklis P., **"Supervisory Control of Discrete Event Systems using Petri Nets,"** Kluwer Academic Publishers, 1998.
* Cassandras, Christos G., **"Discrete Event Systems - Modeling and Performance Analysis,"** Aksen Associates, 1993.
* Yamalidou K., Moody J., Lemmon M. and Antsaklis P. **Feedback Control of Petri Nets Based on Place Invariants** http://www.nd.edu/~lemmon/isis-94-002.pdf

Supervision of DES: SCADA

***S**upervisory*

***C**ontrol*

***A**nd*

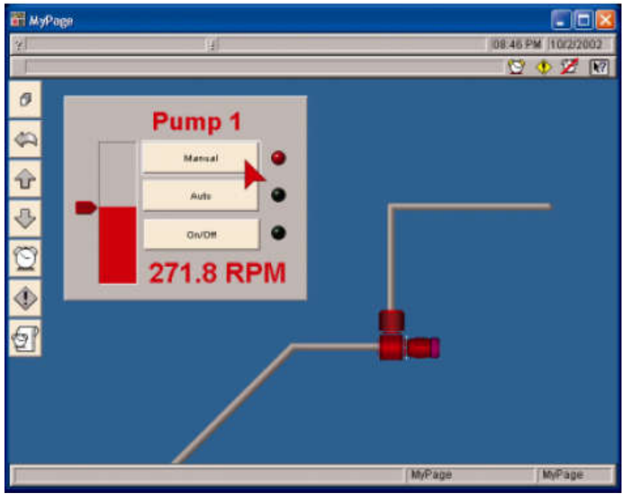
***D**ata*

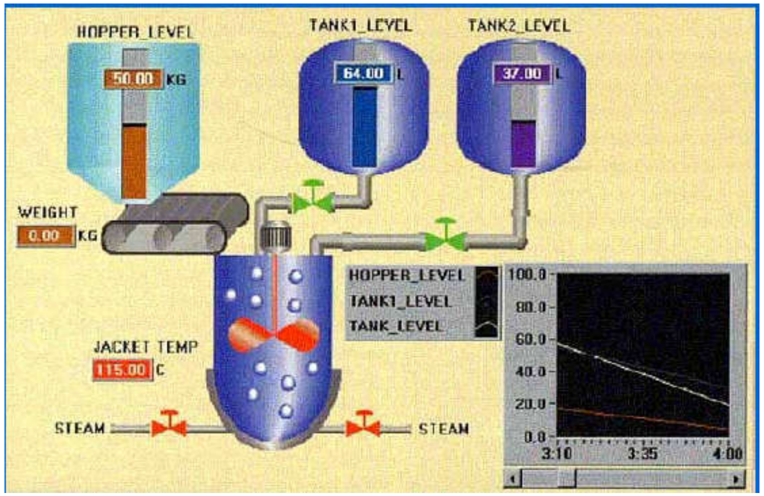
***A**cquisition*

Supervision of DES

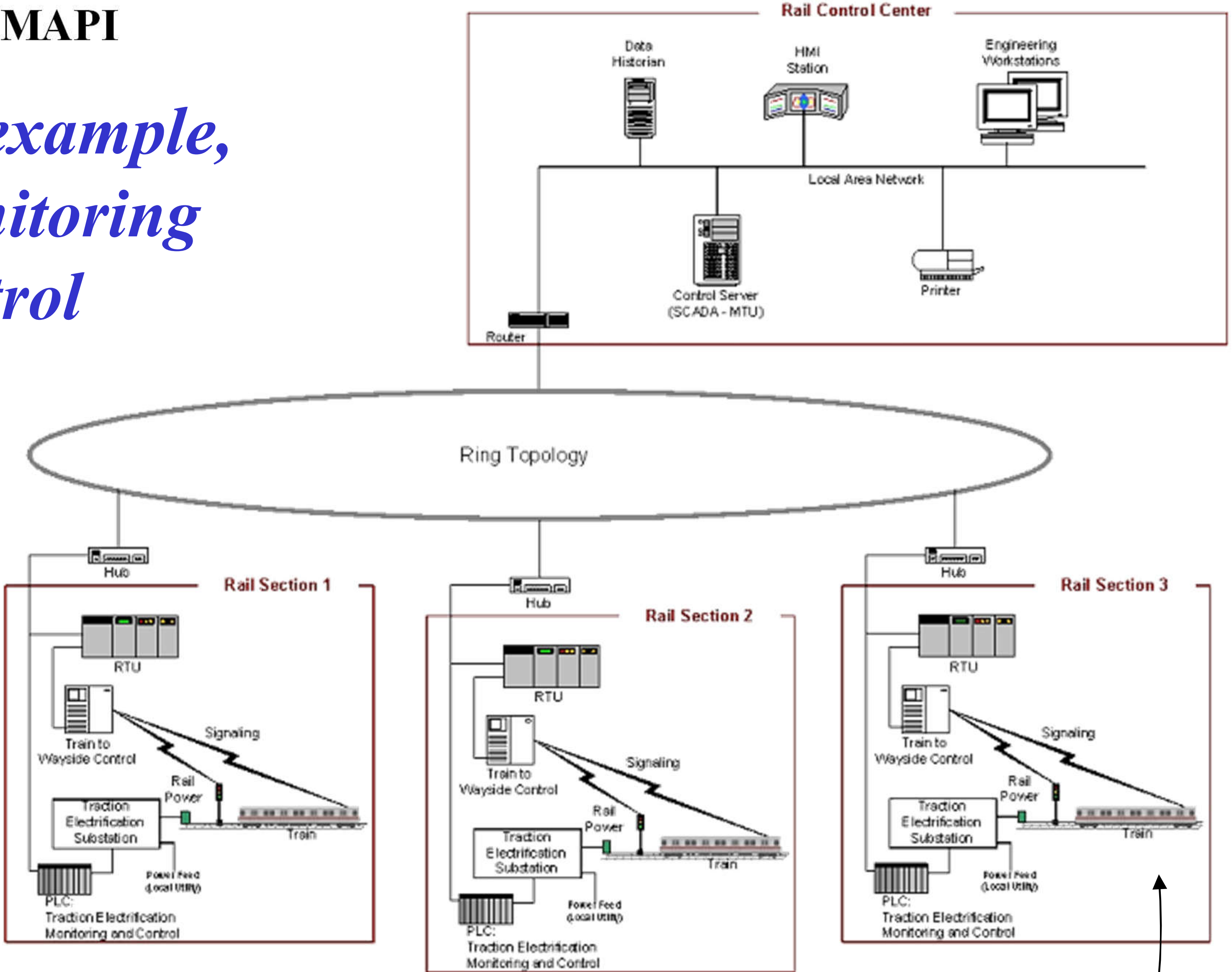
SCADA interface

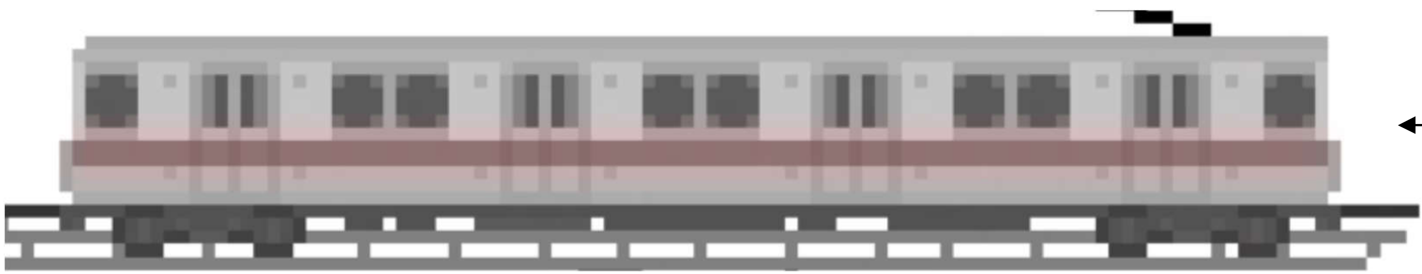
Control / Data
GUI / HMI





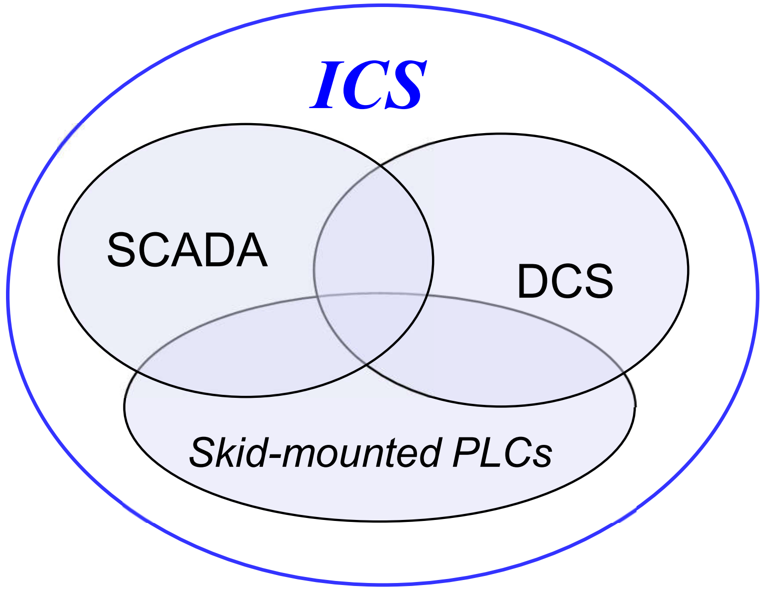
SCADA example, Rail Monitoring and Control





Supervision of DES

SCADA vs ICS



Industrial Control Systems (ICS):

- Supervisory Control and Data Acquisition (SCADA) systems,
- Distributed Control Systems (DCS), or
- smaller configurations such as skid-mounted PLCs

ICSs are typically used in industries such as electric, water, oil-and-gas, transportation, chemical, pharmaceutical, pulp-and-paper, food and beverage, and discrete-manufacturing (e.g. automotive, aerospace, and durable goods).

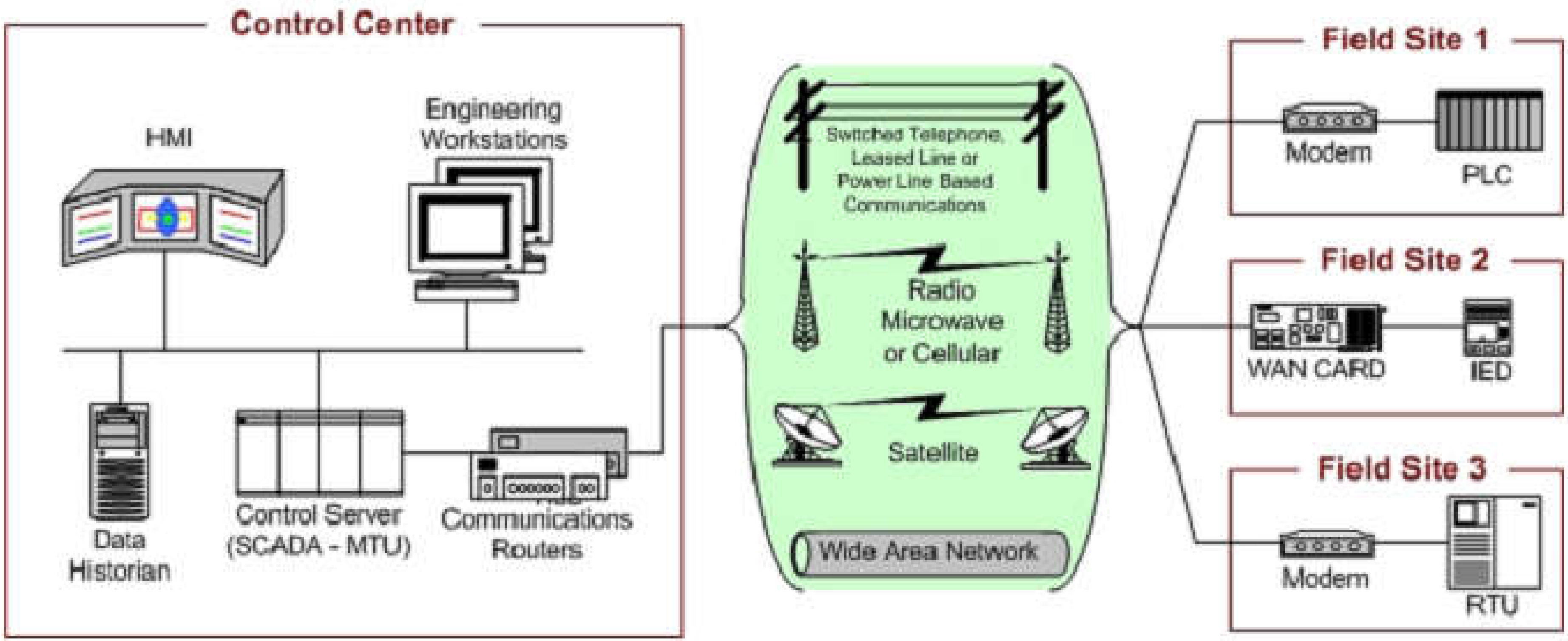
Supervision of DES

SCADA topics

- Remote monitoring of the state of automation systems
- Logging capacity (resorting to specialized Databases)
- Able to access to *historical* information (plots along time, with selectable periodicity)
- Advanced tools to design Human-Machine interfaces
- Failure Detection and Isolation capacity (*threshold* and/or logical functions) on supervised quantities
- Access control

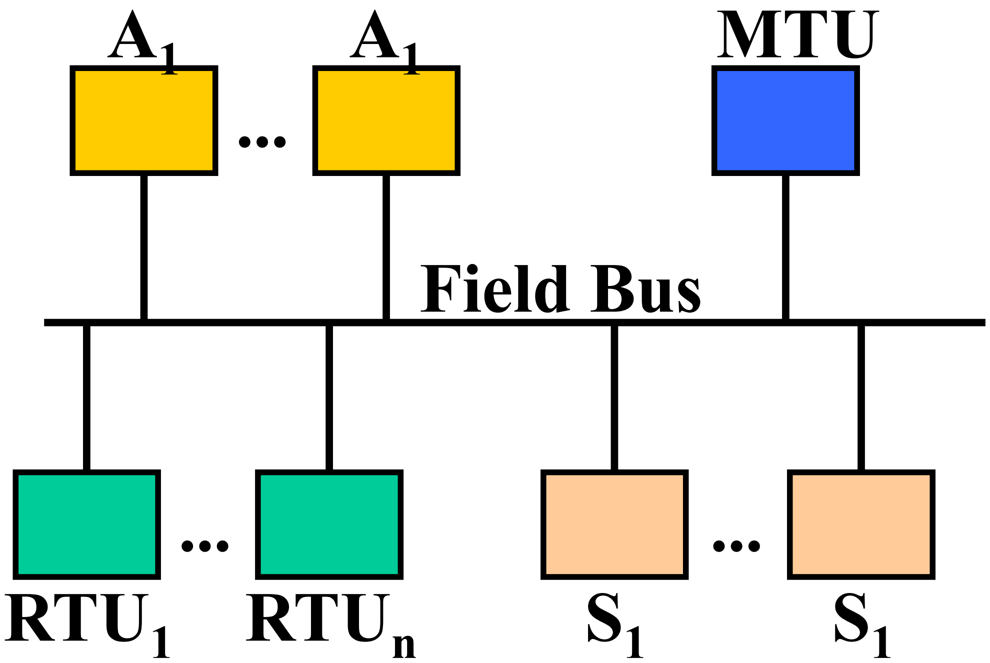
Supervision of DES

SCADA system general layout



Supervision of DES

Hardware Support Architecture of SCADA



Legend:

MTU - Main Terminal Unit

RTU - Remote Term. Unit

S – Sensor

A - Actuator

General term: *Fieldbus (IEC 61158).* ***Examples:*** *PROFIBUS (Fieldbus type, Siemens), MODBUS (Schneider), CAN bus (Bosch), …*

Supervision of DES

Examples of software packages including SCADA solutions

- **Aimax**, de Desin Instruments S.A.
- **CUBE**, Orsi España S.A.
- **FIX**, de Intellution.
- **Lookout,** National Instruments.
- **Monitor Pro,** de Schneider Electric.
- **SCADA InTouch**, de LOGITEK.
- **SYSMAC SCS**, de Omron.
- **Scatt Graph 5000,** de ABB.
- **WinCC**, de Siemens.

Fieldbus	Bus power	Cabling redundancy	Max devices	Synchronisation	Sub millisecond cycle
AFDX	No	Yes	Almost unlimited	No	Yes
AS-Interface	Yes	No	62	No	No
CANopen	No	No	127	Yes	No
CompoNet	Yes	No	384	No	Yes
ControlNet	No	Yes	99	No	No
CC-Link	No	No	64	No	No
DeviceNet	Yes	No	64	No	No
EtherCAT	Yes	Yes	65,536	Yes	Yes
Ethernet Powerlink	No	Optional	240	Yes	Yes
EtherNet/IP	No	Optional	Almost unlimited	Yes	Yes
Interbus	No	No	511	No	No
LonWorks	No	No	32,000	No	No
Modbus	No	No	246	No	No
PROFIBUS DP	No	Optional	126	Yes	No
PROFIBUS PA	Yes	No	126	No	No
PROFINET IO	No	Optional	Almost unlimited	No	No
PROFINET IRT	No	Optional	Almost unlimited	Yes	Yes
SERCOS III	No	Yes	511	Yes	Yes
SERCOS interface	No	No	254	Yes	Yes
Foundation Fieldbus H1	Yes	No	240	Yes	No
Foundation Fieldbus HSE	No	Yes	Almost unlimited	Yes	No
RAPIEnet	No	Yes	256	Under Development	Conditional

recent (2003, 2010)

used in our labs

An invitation for project 3:

Do a presentation about OpenSCADA

http://oscada.org/

Some links:

General characteristics of OpenSCADA

http://oscada.org/main/characteristics/

OpenSCADA on a Raspberry-Pi

http://oscada.org/wiki/Using/Raspberry_Pi

Modeling and Automation of Industrial Processes

Modelação e Automação de Processos Industriais / MAPI

Supervised Control of Discrete Event Systems
Supervision Controllers (Part 1/2)

http://www.isr.tecnico.ulisboa.pt/~jag/courses/mapi2223

Prof. Paulo Jorge Oliveira, original slides
Prof. José Gaspar, rev. 2022/2023

Syllabus:

...

Chap. 2 – Discrete Event Systems (DESs)

Chap. 5b – Supervised Control of DESs

* SCADA
* Methodologies for the Synthesis of Supervision Controllers
* Failure detection

...

Some pointers on Supervised Control of DES

History: The SCADA Web, http://members.iinet.net.au/~ianw/
Monitoring and Control of Discrete Event Systems, Stéphane Lafortune,
http://www.ece.northwestern.edu/~ahaddad/ifac96/introductory_workshops.html

Tutorial: http://vita.bu.edu/cgc/MIDEDS/
http://www.daimi.au.dk/PetriNets/

Analysers & Simulators: http://www.nd.edu/~isis/techreports/isis-2002-003.pdf (Users Manual)
http://www.nd.edu/~isis/techreports/spnbox/ (Software)

Bibliography:
* SCADA books http://www.sss-mag.com/scada.html
* K. Stouffer, J. Falco, K. Kent, "**Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security**", NIST Special Publication 800-82, 2006
* Moody J. e Antsaklis P., **"Supervisory Control of Discrete Event Systems using Petri Nets,"** Kluwer Academic Publishers, 1998.
* Cassandras, Christos G., **"Discrete Event Systems - Modeling and Performance Analysis,"** Aksen Associates, 1993.
* Yamalidou K., Moody J., Lemmon M. and Antsaklis P. **Feedback Control of Petri Nets Based on Place Invariants** http://www.nd.edu/~lemmon/isis-94-002.pdf

Supervision of DES

And

Now

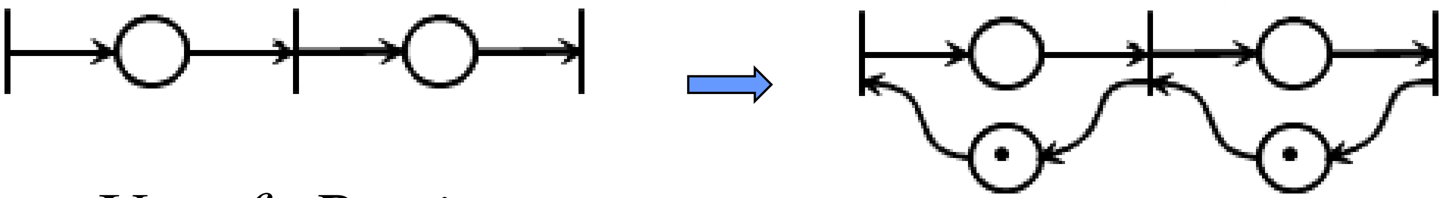
Something

Completely

Different

Supervision of DES: *motivation*

Given one Unsafe Petri net can one obtain a Safe Petri net?



Unsafe Petri net

Safe Petri net

In many cases yes: **Supervision** of DES may be achieved using linear algebra based methodologies. *See the next slides!*

Other possible goals:

- Supervise and bound the work of the supervised DES
- Reinforce that some properties are verified
- Assure that some states are not reached
- Performance criteria are verified
- Prevent deadlocks in DES
- Constrain on the use of resources (e.g. mutual exclusion)

Supervision of DES

Some history on Supervised Control

- Methods for finite automata [Ramadge et *al.*], 1989
  - some are based on brute-force search (!)
  - or may require simulation (!)

- Formal verification of *software* in Computer Science (since the 60s) and on *hardware* (90, ...)

- Supervisory Control Method of Petri Nets, method based on *monitors* [Giua et *al.*], 1992.

- Supervisory Control of Petri Nets based on **Place Invariants** [Moody, Antsaklis et *al.*], 1994 (shares some similarities with the previous one, but deduced independently!...).

Supervision of DES

Advantages of the Supervisory Control of Petri Nets

- Mathematical representation is clear (and easy)
- Resorts only to linear algebra (matrices)
- More compact then automata
- Straightforward the representation of infinity state spaces
- Intuitive graphical representation available

The representation of the controller as a Petri Net leads to simplified Analysis and Synthesis tasks

Supervision of DES

Place Invariants

Place invariants are sets of places whose token count remains always constant. Place invariants can be computed from integer solutions of $w^T D = 0$. Non-zero entries of w correspond to the places that belong to the particular invariant.

Supervisor Synthesis using Place Invariants [ISIS docs]:

What type of relations can be represented in the method of Place Invariants?

- Sets of linear constraints in the state space
- Representation of convex regions (there are extensions for non-convex regions)
- Constraints to guarantee liveness and to avoid deadlocks *(that can be expressed, in general, as linear constraints)*
- Constraints on the events and timings *(that can be expressed, in general, as linear constraints)*

Methods of Analysis/Synthesis

Method of the Matrix Equations *(just to remind)*

The dynamics of the Petri net state can be written in compact form as:

$$\mu(k+1) = \mu(k) + Dq(k)$$

where:

- $\mu(k+1)$ - marking to be reached
- $\mu(k)$ - initial marking
- $q(k)$ - firing vector (transitions)
- D - incidence matrix. Accounts the balance of tokens, giving the transitions fired.

Methods of Analysis/Synthesis

How to build the Incidence Matrix? *(just to remind)*

For a Petri net with *n* places and *m* transitions

$$\mu \in N_0^{\ n}$$

$$q \in N_0^{\ m}$$

$$\boxed{D = D^+ - D^-}\ , \quad D \in Z^{n\times m}, \quad D^+ \in N_0^{n\times m}, \quad D^- \in N_0^{n\times m}$$

The enabling firing rule is $\boxed{D^- q \leq \mu}$

Can also be written in compact form as the inequality $\mu + Dq \geq 0$, interpreted element-by-element.

Note: in this course all vector and matrix inequalities are read element-by-element unless otherwise stated.

Supervision of DES

$$w^T \mu(k+1) = w^T \mu(k) + \underbrace{w^T D}_{=0} \underbrace{q(k)}_{\forall k}$$

equal

Place Invariants

Place invariants are sets of places whose token count remains always constant. Place invariants can be computed from integer solutions of $w^T D = 0$. Non-zero entries of w correspond to the places that belong to the particular invariant.

Supervisor Synthesis using Place Invariants [ISIS docs]:

What type of relations can be represented in the method of Place Invariants?

- Sets of linear constraints in the state space
- Representation of convex regions (there are extensions for non-convex regions)
- Constraints to guarantee liveness and to avoid deadlocks *(that can be expressed, in general, as linear constraints)*
- Constraints on the events and timings *(that can be expressed, in general, as linear constraints)*

Methods of Synthesis

Some notation for the method

• The supervised system is modelled as a Petri net with n places and m transitions, and incidence matrix

$$\boxed{D_P \in Z^{n\times m}.}$$

$$\begin{bmatrix} \mu_p(k+1) \\ \mu_c(k+1) \end{bmatrix} = \begin{bmatrix} \mu_p(k) \\ \mu_c(k) \end{bmatrix} + \begin{bmatrix} D_p \\ D_c \end{bmatrix} q(k)$$

• The supervisor is modelled as a Petri net with n_C places and m transitions, and incidence matrix

$$\boxed{D_C \in Z^{n_C\times m}.}$$

• The resulting total system has an incidence matrix

$$\boxed{D \in Z^{(n+n_C)\times m}.}$$

Methods of Synthesis

Theorem: Synthesis of Controllers based on Place Invariants (T1)

Given the set of linear state constraints that the supervised system must follow, written as

$$\boxed{L\mu_P \le b,} \qquad \mu_P \in N_0^{\ n}, \qquad L \in Z^{n_C \times n} \qquad and \qquad b \in Z^{n_C}.$$

If $$\boxed{b - L\mu_{P_0} \ge 0,}$$

then the controller with incidence matrix and the initial marking, respectively

$$\boxed{D_C = -LD_P,} \qquad \text{and} \qquad \boxed{\mu_{C_0} = b - L\mu_{P_0},}$$

enforce the constraints to be verified for all markings obtained from the initial marking.

Methods of Synthesis

Theorem - proof outline :

The constraint $L\mu_P \leq b$ can be written as $L\mu_P + \mu_C = b$, using the slack variables μ_C. They represent the marking of the n_C places of the controller.

To have a place invariant, the relation $w^T D = 0$ must be verified and in particular, given the previous constraint:

$$w^T D = \begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} D_P \\ D_C \end{bmatrix} = 0, \text{ resulting } \boxed{D_C = -LD_P.}$$

From $L\mu_{P_0} + \mu_{C_0} = b$, follows that $\boxed{\mu_{C_0} = b - L\mu_{P_0}.}$ ∎

Some more details: $L\mu_p \leq b \longrightarrow D_c = -LD_p, \quad \mu_{c0} = b - L\mu_{p0}$

Using slack variables μ_c

allows $L\mu_p \leq b \longrightarrow L\mu_p + \mu_c = b$

which can be written in matrix form as $\begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} = b$

Given the dynamics

$$\begin{bmatrix} \mu_p(k+1) \\ \mu_c(k+1) \end{bmatrix} = \begin{bmatrix} \mu_p(k) \\ \mu_c(k) \end{bmatrix} + \begin{bmatrix} D_p \\ D_c \end{bmatrix} q(k)$$

one has $\begin{bmatrix} L & I \end{bmatrix} \left(\begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} + \begin{bmatrix} D_p \\ D_c \end{bmatrix} q(k) \right) = b$

$$\underbrace{\begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix}}_{=b} + \underbrace{\cdots}_{=0}$$

$\begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} D_p \\ D_c \end{bmatrix} q(k) = 0$ desired $\forall q(k)$

so $\begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} D_p \\ D_c \end{bmatrix} = 0$

$$LD_p + D_c = 0 \longrightarrow D_c = -LD_p$$

The initial state is direct:

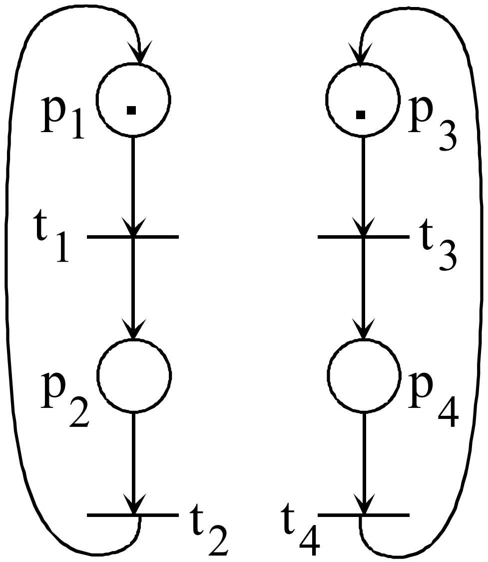
$$L\mu_p + \mu_c = b \longleftarrow \forall \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix}$$

$$L\mu_{p0} + \mu_{c0} = b$$

$$\mu_{c0} = b - L\mu_{p0}$$

Methods of Synthesis

Example 1 of controller synthesis: Mutual Exclusion



Linear constraint: $\mu_2 + \mu_4 \leq 1$

that can be written as:

$$L\mu_P \leq b \qquad \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} \leq 1.$$

Incidence Matrix $D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$ and initial marking $\mu_{P_0} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$

Methods of Synthesis

Example 1 of controller synthesis: Mutual Exclusion

1) Test

$$b - L\mu_{P_0} = 1 - \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = 1 \geq 0.$$

OK.

2) Compute

$$D_C = -LD_P = -\begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix},$$

and

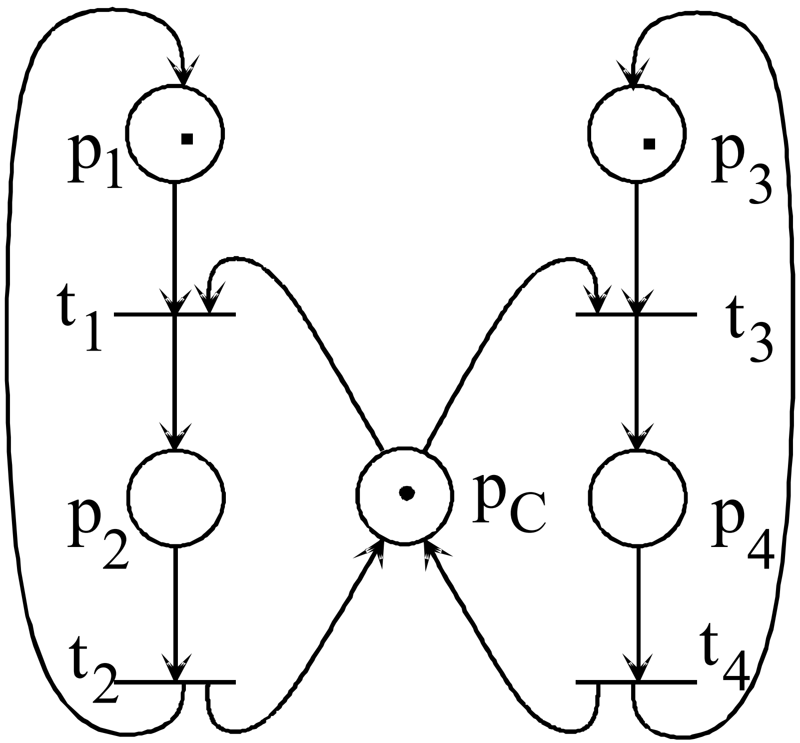
$$\mu_{C_0} = b - L\mu_{P_0} = 1.$$

OK.

Methods of Synthesis

Example 1 of controller synthesis: Mutual Exclusion

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ \boxed{-1 \quad 1 \quad -1 \quad 1} \end{bmatrix}$$

$$\mu_0 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \boxed{1} \end{bmatrix}$$

OK.
UAU!!!.

Methods of Synthesis

Example 1 of controller synthesis: Mutual Exclusion

```
% The Petri net D=Dp-Dm, and m0
%  (Dplus-Dminus= Post-Pre)

Dm= [1 0 0 0;
     0 1 0 0;
     0 0 1 0;
     0 0 0 1];

Dp= [0 1 0 0;
     1 0 0 0;
     0 0 0 1;
     0 0 1 0];

m0= [1 0 1 0]';

% Supervisor constraint
%
L= [0 1 0 1];
b= 1;

% Computing the supervisor
%
[Dfm, Dfp, ms0] = linenf(Dm, Dp, L, b, m0);
Df= Dfp-Dfm
ms0
```

Result using the function **linenf.m** of the toolbox SPNBOX:

```
Df =

    -1     1     0     0
     1    -1     0     0
     0     0    -1     1
     0     0     1    -1
    -1     1    -1     1


ms0 =

     1
     0
     1
     0
     1
```

Methods of Synthesis

Definition:

Maximal permissivity occurs when (i) all the linear constraints are verified and (ii) all legal markings can be reached.

Lemmas:

L1) **The controllers** obtained with T1 **have maximal permissivity.**

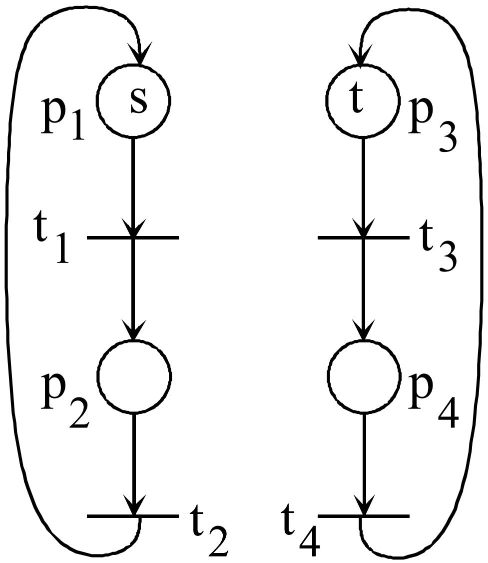
L2) Given the linear constraints used, **the place invariants** obtained with the controller synthesized with T1 **are the same** as the invariants associated with the initial system.

Methods of Synthesis

Example 2 of controller synthesis

$$\forall s \in N_0, \forall t \in N_0, \forall n \in N_0$$

n Readers / 1 Writer



Linear constraint $\mu_2 + n\mu_4 \leq n$

(max *n* readers or 1 writer)

That can be written as:

$$L\mu_P \leq b \qquad \begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} \leq n.$$

Incidence Matrix
$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$
and initial marking
$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

Methods of Synthesis

Example 2 of controller synthesis

n Readers / 1 Writer

1) Test

$$b - L\mu_{P_0} = n - \begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix} = n \geq 0.$$

OK.

2) Compute

$$D_C = -LD_P = -\begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & -n & n \end{bmatrix},$$

and

$$\mu_{C_0} = b - L\mu_{P_0} = n.$$

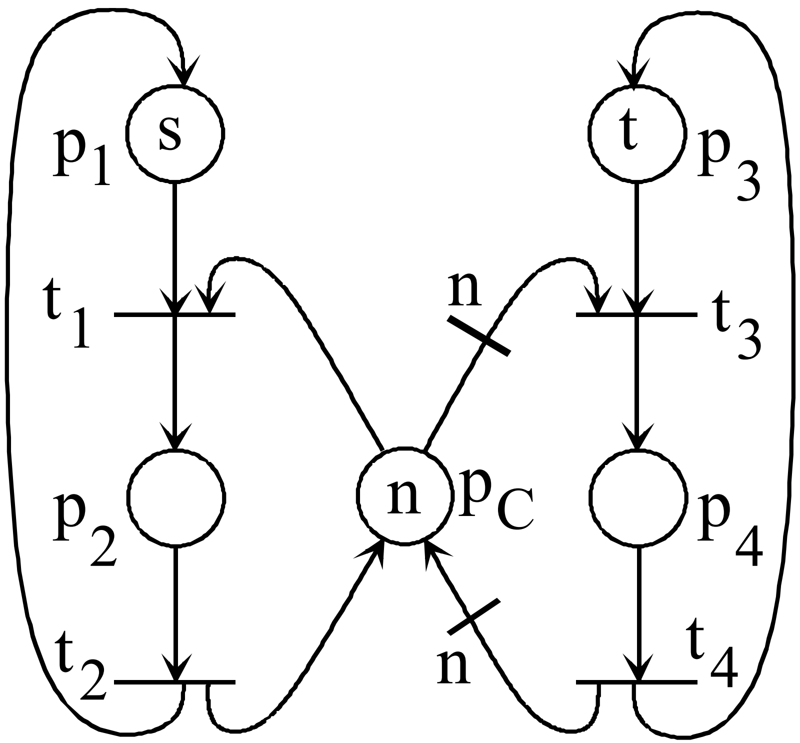
OK.

Methods of Synthesis

Example 2 of controller synthesis

n Readers / 1 Writer

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ \boxed{-1 \quad 1 \quad -n \quad n} \end{bmatrix}$$

$$\mu_0 = \begin{bmatrix} s \\ 0 \\ t \\ 0 \\ \boxed{n} \end{bmatrix}$$

OK.

UAU!!!.

Supervision of DES

Advantages of the Method of the Place Invariants [ISIS docs]:

Other characteristics that can impact on the solutions?

• Existence and uniqueness

• Optimality of the solutions (e.g. maximal permissivity)

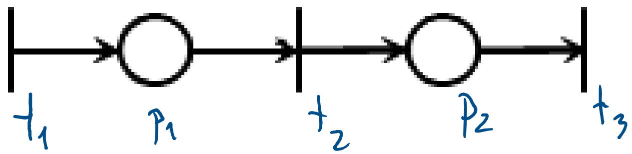
• Existence of transition non-controllable and/or not observable (remind definitions for time-driven systems)

In general the solutions can be found solving:

Linear Programming Problems, with Linear Constraints

Example 3 of controller synthesis

Given one Unsafe Petri net can one obtain a Safe Petri net?



Unsafe Petri net

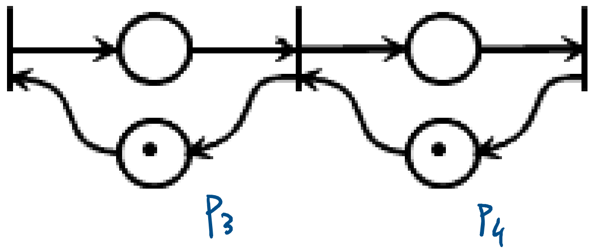
	t_1	t_2	t_3
μ_1	$+1$	-1	
μ_2		$+1$	-1

$\underbrace{}_{D_p}$

$$\begin{cases} \mu_1 \leq 1 \\ \mu_2 \leq 1 \end{cases} \longleftrightarrow L\mu \leq b \quad \Rightarrow \quad L = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, \; b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$D_c = -L\,D_p = -I D_p = -D_p \;, \quad \mu_{c_0} = b - L\mu_{p_0} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \;//$$

$$D_c = -D_p = \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_3 \end{array} \\ \begin{bmatrix} -1 & +1 & \\ & -1 & +1 \end{bmatrix} \end{array} \begin{array}{c} \\ p_3 \\ p_4 \end{array} \Longrightarrow$$



Safe Petri net

Modeling and Automation of Industrial Processes

Modelação e Automação de Processos Industriais / MAPI

Supervised Control of Discrete Event Systems
Supervision Controllers (Part 2/2)

http://www.isr.tecnico.ulisboa.pt/~jag/courses/mapi2223

Prof. Paulo Jorge Oliveira, original slides
Prof. José Gaspar, rev. 2022/2023

Some pointers on Supervised Control of DES

Analysers & simulators

http://www.nd.edu/~isis/techreports/isis-2002-003.pdf (Users Manual)
http://www.nd.edu/~isis/techreports/spnbox/ (Software)

Bibliography:

Supervisory Control of Discrete Event Systems using Petri Nets, J. Moody J. and P. Antsaklis, Kluwer Academic Publishers, 1998.

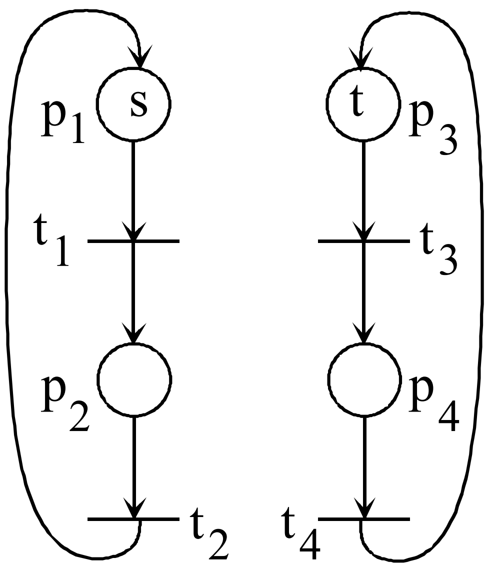
Supervised Control of Concurrent Systems: A Petri Net Structural Approach, M. Iordache and P. Antsaklis, Birkhauser 2006.

Discrete Event Systems - Modeling and Performance Analysis, Christos G. Cassandras, Aksen Associates, 1993.

Feedback Control of Petri Nets Based on Place Invariants, K. Yamalidou, J. Moody, M. Lemmon and P. Antsaklis, http://www.nd.edu/~lemmon/isis-94-002.pdf

Methods of Synthesis

Example of controller synthesis: s Producers / t Consumers



Incidence matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Initial marking

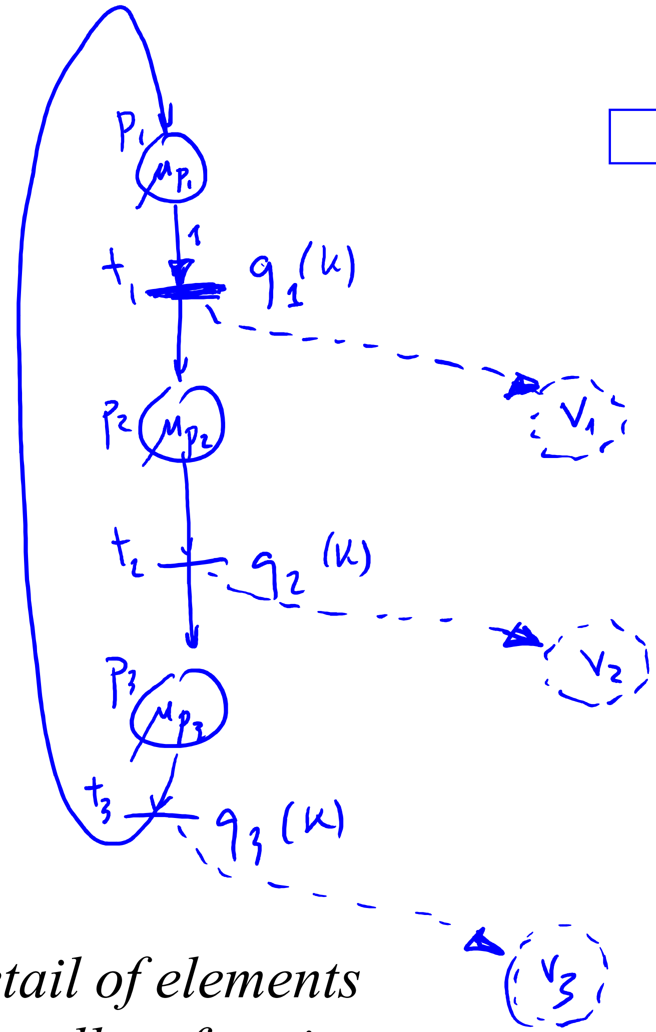
$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

Let p2= #machines working, t2= product produced
p3= #consumers, t3= request to consume (e.g. transport product)

Q: How to write *consume only when produced* ? What is the linear constraint?

Not possible to write it as a linear constraint on places $L\mu_p \leq b$.
Is it impossible to solve this problem with the supervised control ?

Methods of Synthesis Generalized linear constraint



State: $\mu_p(k) = \begin{bmatrix} \mu_{p1} \\ \mu_{p2} \\ \mu_{p3} \end{bmatrix}_k$

Firing vector: $q_p(k) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}_k$

Parikh vector: $v_p(k) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_k$

$$L\mu_P + Fq_P + Cv_P \leq b$$

$$\mu_P \in N_0{}^n,\ v_P \in N_0{}^m,\ q_P \in N_0{}^m,$$
$$L \in Z^{nc\times n},\ F \in Z^{nc\times m},\ C \in Z^{nc\times m},$$
$$b \in Z^{nc}$$

Detail of elements that allow forming generalized linear constraints

Methods of Synthesis Generalized linear constraint

Let the generalized linear constraint be

$$\boxed{L\mu_P + Fq_P + Cv_P \leq b}$$

n = #places
m= #transitions
n_C= #constraints

$\mu_P \in {N_0}^n, \quad q_P \in {N_0}^m, \quad v_P \in {N_0}^m$

$L \in Z^{nc \times n}, \; F \in Z^{nc \times m}, \; C \in Z^{nc \times m} \; and \quad b \in Z^{nc}$

where

- μ_P is the marking vector for system P
- q_P is the firing vector since t_0
- v_P is the number of transitions (firing) that can occur, also designated as ***Parikh vector***

Methods of Synthesis

Function LINENF of SPNBOX

Theorem*: Synthesis of Controllers based on Place Invariants, for **Generalized Linear Constraints**

Given the generalized linear constraint $\boxed{L\mu_P + Fq_P + Cv_P \leq b,}$

if $b - L\mu_{P_0} \geq 0,$ then the controller with incidence matrix and initial marking, respectively

$$\boxed{\begin{aligned} D_C^- &= \max\left(0,\, LD_P + C,\, F\right) \\ D_C^+ &= \max\left(0,\, F - \max\left(0, LD_P + C\right)\right) - \min\left(0,\, LD_P + C\right), \end{aligned}}$$

$$\boxed{\mu_{C_0} = b - L\mu_{P_0} - Cv_{P0},}$$

guarantees that constraints are verified for the states resulting from the initial marking.

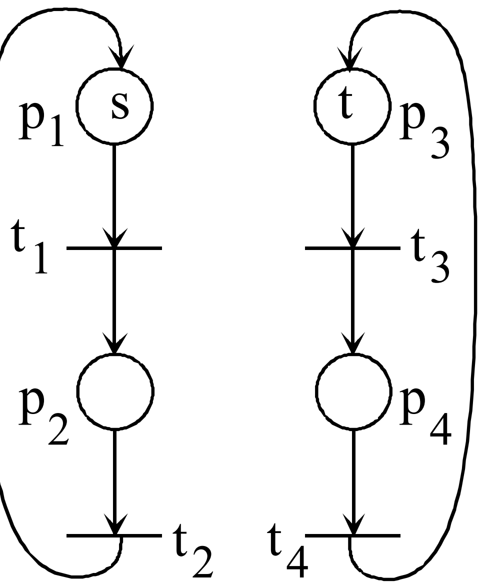
* *In the next slides this will be called the* ***LINENF theorem.***

Methods of Synthesis

Example 1 of controller synthesis

$$\forall s \in N_0, \forall t \in N_0, \forall n \in N_0$$

Producer / Consumer



Linear constraint: $v_3 \leq v_2$

that can be written as:

$$Cv_P \leq b$$

$$L = 0, F = 0$$

$$\begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \leq 0.$$

Incidence matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Initial marking

$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

Methods of Synthesis

Example of controller synthesis

Producer / Consumer

1) Test $$b - L\mu_{P_0} = 0 - 0 \geq 0.$$

OK.

2) Compute

$$\boxed{\begin{aligned} D_C^- &= \max\left(0,\ LD_P + C,\ F\right) \\ D_C^+ &= \max\left(0,\ F - \max\left(0, LD_P + C\right)\right) - \min\left(0,\ LD_P + C\right), \end{aligned}}$$

$$D_C^- = \max\left(0,\ [0\ \ -1\ \ 1\ \ 0],\ 0\right) = [0\ \ 0\ \ 1\ \ 0]$$

$$D_C^+ = \max\left(0, -[0\ \ 0\ \ 1\ \ 0]\right) - \min\left(0, [0\ \ -1\ \ 1\ \ 0]\right)$$

$$= [0\ \ 0\ \ 0\ \ 0] - [0\ \ -1\ \ 0\ \ 0] = [0\ \ 1\ \ 0\ \ 0]$$

$$D_C = D_C^+ - D_C^- = \underline{[0\ \ 1\ \ -1\ \ 0]}$$

and

$$\boxed{\mu_{C_0} = b - L\mu_{P_0} - Cv_{P0},}$$

$$\mu_{C_0} = b - L\mu_{P_0} = 0 - 0 = 0.$$

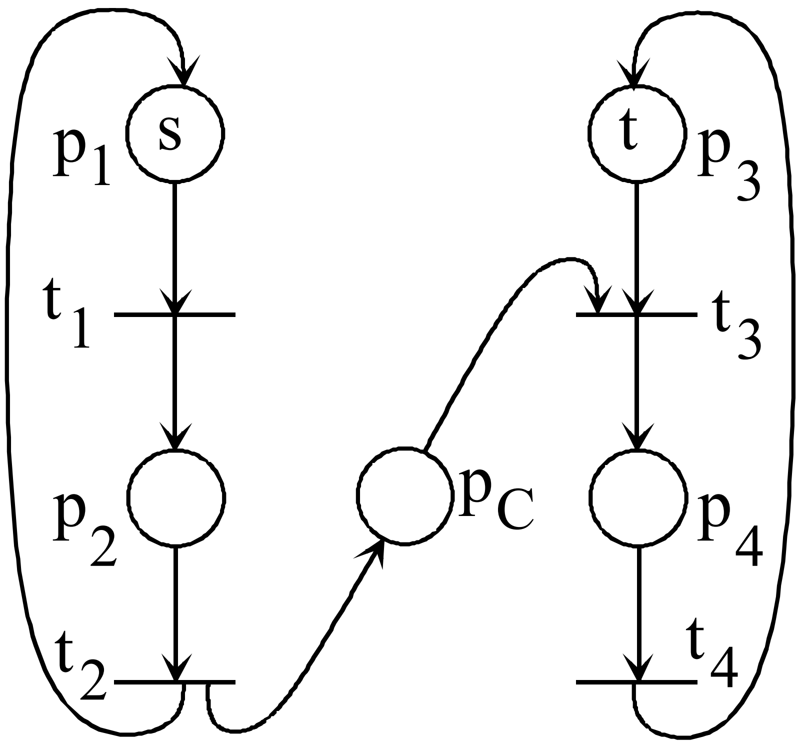
OK.

Methods of Synthesis

Example of controller synthesis

Producer / Consumer

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ \boxed{0} & \boxed{1} & \boxed{-1} & \boxed{0} \end{bmatrix}$$

$$\mu_0 = \begin{bmatrix} s \\ 0 \\ t \\ 0 \\ \boxed{0} \end{bmatrix}$$

OK.
UAU!!!.

Methods of Synthesis

Example of controller synthesis: Producer Consumer

```
% The Petri net D=Dp-Dm, and m0
%  (Dplus-Dminus= Post-Pre)

Dm= [1 0 0 0;
     0 1 0 0;
     0 0 1 0;
     0 0 0 1];

 Dp= [0 1 0 0;
     1 0 0 0;
     0 0 0 1;
     0 0 1 0];

m0= [1 0 1 0]';

% Supervisor constraint
%
L= []; F= [];  C= [0 -1 1 0];
b= 0;

% Computing the supervisor
%
[Dfm, Dfp, ms0] = linenf(Dm, Dp, L, b, m0, F, C)
Df= Dfp-Dfm
ms0
```

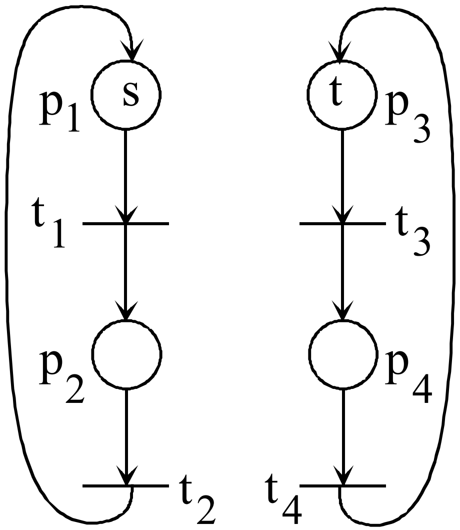
Result using the function LINENF.m of the toolbox SPNBOX:

```
Df =

    -1     1     0     0
     1    -1     0     0
     0     0    -1     1
     0     0     1    -1
     0     1    -1     0


ms0 =

     1
     0
     1
     0
     0
```

Methods of Synthesis

Example 2 of controller synthesis

Bounded Producer / Consumer



Incidence matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Initial marking

$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

TWO linear constraints:

$$\begin{cases} v_3 \le v_2 \\ v_2 \le v_3 + n \end{cases} \Leftrightarrow \begin{cases} v_3 - v_2 \le 0 \\ v_2 - v_3 \le n \end{cases}$$

$$\forall s \in N_0, \forall t \in N_0, \forall n \in N_0$$

The two linear constraints can be written as:

$$Cv_P \le b$$
$$i.e.\ L = 0, F = 0$$

$$\Leftrightarrow \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \le \begin{bmatrix} 0 \\ n \end{bmatrix}$$

Methods of Synthesis

Example of controller synthesis

Bounded Producer / Consumer

1) Test

$$b - L\mu_{P_0} = b = \begin{bmatrix} 0 \\ n \end{bmatrix} \geq 0.$$

OK.

2) Compute

$$D_C^- = \max\left(0, \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}, 0\right) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$D_C^+ = \max\left(0, 0 - \max\left(0, \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}\right)\right) - \min\left(0, \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

and

$$\mu_{C_0} = b - L\mu_{P_0} = \begin{bmatrix} 0 \\ n \end{bmatrix}.$$

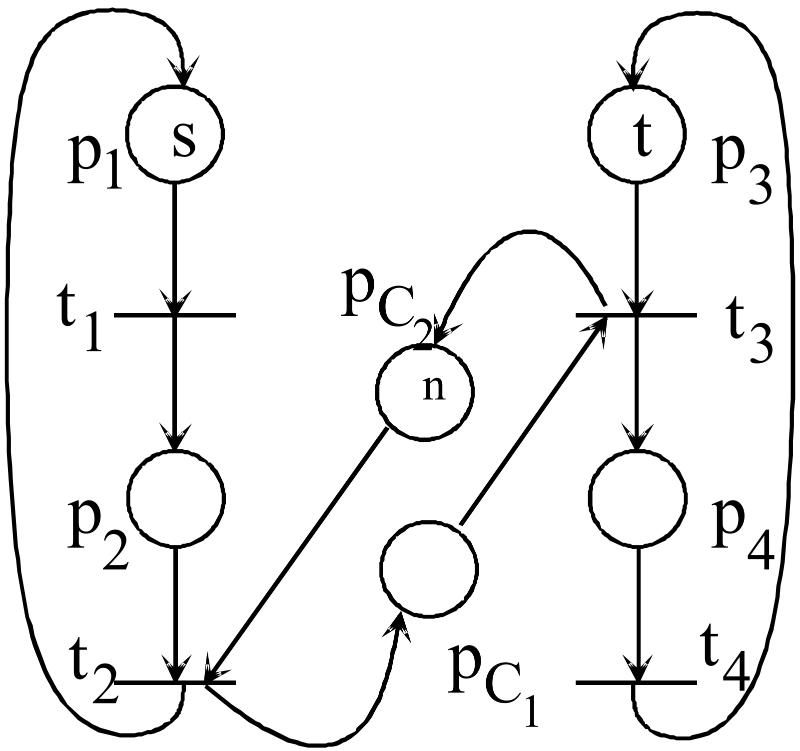
OK.

Methods of Synthesis

Example of controller synthesis

Bounded Producer / Consumer

3) Resulting in



$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \end{bmatrix}$$

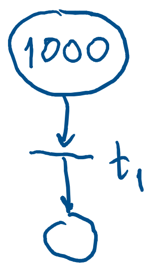
$$\mu_0 = \begin{bmatrix} s \\ 0 \\ t \\ 0 \\ 0 \\ n \end{bmatrix}$$

OK.
UAU!!!.

Methods of Synthesis

Example 3 of controller synthesis – *Flow regulation*

Consider a Petri net with a large initial marking



Objective: do ***NOT allow consuming too many tokens*** in a single step.

For example, one wants to enforce max q_1 to be 2, i.e. *accepting only* $q_1 = 0$ *or* $q_1 = 1$ *or* $q_1 = 2$.

Constraint:

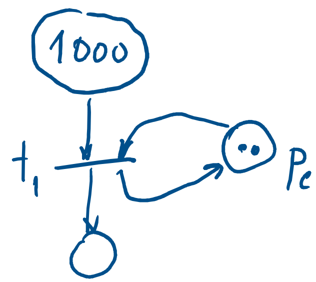
$$\underset{\substack{\uparrow \\ F}}{1} \cdot q_1 \leq \underset{\substack{\uparrow \\ b}}{2}$$

Solution:

$$D_c^+ = 1$$

$$D_c^- = 1$$

$$\mu_{c_0} = 2$$



Methods of Synthesis

Function LINENF of SPNBOX

LINENF Lemma 1: From General Constraints to Theorem T1

Given the generalized linear constraint $L\mu_P + Fq_P + Cv_P \leq b$ and the conditions of the LINENF theorem:

If $\boxed{L \neq 0,}$ $F = 0,$ $C = 0$

then $D_C^+ = (LD_P)^-,$ $D_C^- = (LD_P)^+$ and $\boxed{D_C = -LD_P}$

$$\boxed{\mu_{C0} = b - L\mu_{P0}}$$

Notation:

$$\begin{aligned} D^+ &= \max(0, D) \\ D^- &= -\min(0, D) \\ D &= D^+ - D^- \end{aligned}$$

$D^+, D^- \in N_0^{n \times m}$ and $D \in Z_0^{n \times m}$

(see proof in the next page)

$$D_C^- = \max\left(0,\, LD_P + C,\, F\right)$$
$$D_C^+ = \max\left(0,\, F - \max\left(0, LD_P + C\right)\right) - \min\left(0,\, LD_P + C\right),$$

$$\mu_{C_0} = b - L\mu_{P_0} - Cv_{P0},$$

$$\boxed{L \neq 0, \quad F = 0, \quad C = 0 \quad \Rightarrow L\mu_P \leq b}$$

$$D_c^- = \max(0,\, LD_p + \overset{=0}{\cancel{C}},\, \overset{=0}{\cancel{F}})$$
$$= \max(0,\, LD_p)$$
$$= (LD_p)^+$$

$$D_c^+ = \max(0,\, \overset{=0}{\cancel{F}} - \max(0, LD_p + \overset{=0}{\cancel{C}})) - \min(0,\, LD_p + \overset{=0}{\cancel{C}})$$
$$= \underbrace{\max(0,\, \underbrace{-(LD_p)^+}_{\leq 0})}_{=0} + (LD_p)^-$$
$$= +(LD_p)^-$$

$D^- = -\min(0, D)$

$D^- \in \mathbb{N}_0^{m \times m}$

$$D_c = D_c^+ - D_c^- = (LD_p)^- - (LD_p)^+ = -\left((LD_p)^+ - (LD_p)^-\right) = -LD_p$$

$$\mu_{c_0} = b - L\mu_{p_0} - \overset{=0}{\cancel{C}}\, v_{p_0} = b - L\mu_{p_0}$$

Methods of Synthesis

Function LINENF of SPNBOX

LINENF Lemma 2: Firing Regulation

Given the generalized linear constraint $L\mu_P + Fq_P + Cv_P \leq b$ and the conditions of the LINENF theorem:

If $\quad L = 0, \quad \boxed{F \neq 0,} \quad C = 0$

then $\quad D_C^+ = F^+, \quad D_C^- = F^+ \quad$ and $\quad \boxed{D_C = 0}$

$$\boxed{\mu_{C0} = b}$$

(homework, prove this lemma)

Methods of Synthesis

Function LINENF of SPNBOX

LINENF Lemma 3: Constraints on Counters

Given the generalized linear constraint $L\mu_P + Fq_P + Cv_P \leq b$ and the conditions of the LINENF theorem:

If $L = 0, \quad F = 0, \quad \boxed{C \neq 0}$

then $D_C^+ = C^-, \qquad D_C^- = C^+$ and $\boxed{D_C = -C}$

$$\boxed{\mu_{C0} = b - Cv_{P0}}$$

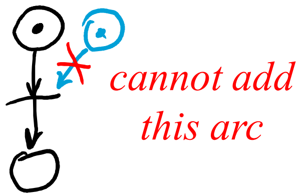
(homework, prove this lemma)

(empty page, do yourself the proof of the last two lemmas)

Methods of Synthesis: intro to **Uncontrollable** and **Unobservable** transitions

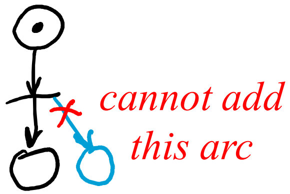
Definition of **Uncontrollable Transition:**

A transition is uncontrollable if its firing **cannot be inhibited** by an external action (e.g. a supervisory controller).



Definition of **Unobservable Transition:**

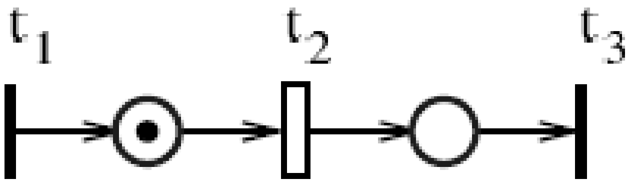
A transition is unobservable if its firing **cannot be detected or measured** (therefore the study of any supervisory controller can not depend from that firing).



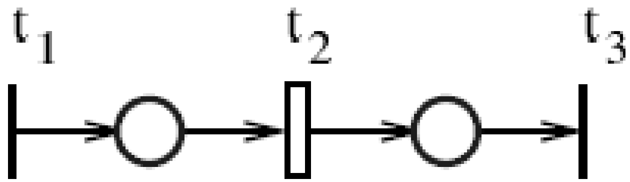
Proposition:

In a Petri net based controller, both input and output arcs to/from plant transitions are used to trigger state changes in the controller. *Since a controller cannot have arcs connecting to unobservable transitions, then all **unobservable transitions are also implicitly uncontrollable**.*

Methods of Synthesis: intro to **Uncontrollable** and **Unobservable** transitions



(a)



(b)

If **t1 is controllable** and **t2 is uncontrollable**:
- case (a), then t2 cannot be directly inhibited; it will eventually fire
- case (b), then **t2 can be indirectly prevented** from firing by inhibiting t1.

i.e. may exist indirect solution despite t2 being uncontrollable.

If **t2 is unobservable** and **t3 is observable**, then we cannot detect when t2 fires. The state of a supervisor is not changed by firing t2. However we can **indirectly detect that t2 has fired**, by detecting the firing of t3.

i.e. may exist indirect solution despite t2 being unobservable.

∴ *may exist indirect solution despite t2 uncontrollable and/or unobservable.*

Methods of Synthesis

Definition: A **marking** $\boldsymbol{\mu_P}$ is **admissible** if

i) $L\mu_P \leq b$ and **ii)** $\forall \mu' \in R(C, \mu_P)$ verifies $L\mu' \leq b$

Definition: A **Linear Constraint (*L*, *b*)** is **admissible** if

i) $L\mu_{P_o} \leq b$ and

ii) $\forall \mu' \in R(C, \mu_{P_o})$ such that $L\mu' \leq b$

μ' is an admissible marking.

Note: ii) indicates that the firing of uncontrollable transitions can never lead from a state that satisfies the constraint to a new state that does not satisfy the constraint.

Methods of Synthesis

Proposition: Admissibility of a constraint

A linear constraint is admissible ***iff***

- The initial markings satisfy the constraint.
- There **exists a controller** with maximal permissivity that forces the constraint and **does not inhibit any uncontrollable transition**.

Two sufficient (not necessary) conditions:

Corollary: given a system with uncontrollable transitions,

$\boxed{l^T D_{uc} \leq 0}$ implies admissibility.

Corollary: given a system with unobservable transitions,

$\boxed{l^T D_{uo} = 0}$ implies admissibility.

Methods of Synthesis

Function MRO_ADM of SPNBOX

Lemma *: Structure of Constraint transformation

If $\boxed{L'\mu_p \leq b'}$ is verified by supervision and was created from $\boxed{L\mu_p \leq b \text{ and } (R_1, R_2)}$

where

$L' = R_1 + R_2 L$ and $b' = R_2(b+1) - 1$

$\boxed{R_1} \in Z^{nc \times n}$ and $R_1\mu_p \geq 0$

$\boxed{R_2} \in Z^{nc \times nc}$ is a matrix with positive elements in the diagonal

then $\boxed{L\mu_p \leq b}$ is also verified by the same supervisor.

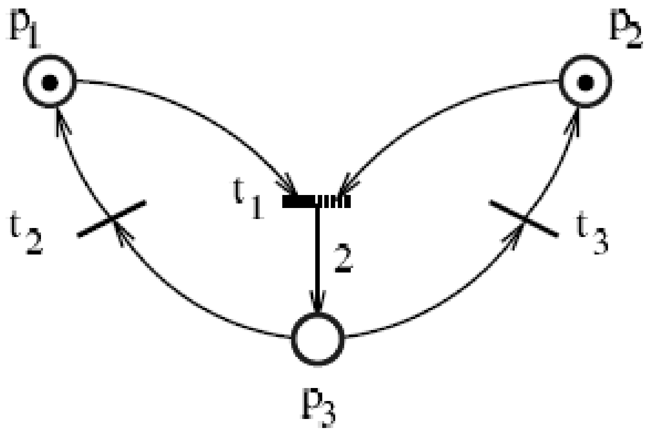
Typical usage:

- list **extra constraints** as unobservable (uo) and/or uncontrollable (oc) transitions
- constraints (L, b) + **extra constraints** $\Rightarrow$ $(R_1, R_2) \rightarrow (L', b')$
- compute the supervisor $(L', b') \rightarrow (D_C^+, D_C^-, \mu_{C0})$ *(example in the next slides)*

* Lemma 4.10 in [Moody98] pg46

Methods of Synthesis

Example 4: design controller with t1 unobservable (1/4)



$$D = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 2 & -1 & -1 \end{bmatrix}, \quad D_{uo} = \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix}$$

Objectives: $\mu_1 + \mu_3 \geq 1$ and $\mu_2 + \mu_3 \geq 1$ which can be written in matrix form as

$$L\mu \leq b, \qquad L = \begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Example extracted from "Supervised Control of Concurrent Systems: A Petri Net Structural Approach", M. Iordache and P. Antsaklis, Birkhauser 2006.

Methods of Synthesis

Example: design controller with t1 unobservable (2/4)

```
% System and constraints

D= [-1  1  0;
    -1  0  1;
    +2 -1 -1];

Dm= -D.*(D<0);
Dp=  D.*(D>0);

m0= [1 1 0]';

L= [-1 0 -1; 0 -1 -1];
b= [-1; -1];


% Supervisor computation

[Dfp, Dfm, mf0] =
    linenf( Dp, Dm, L, b, m0 );
```

Dfp =

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

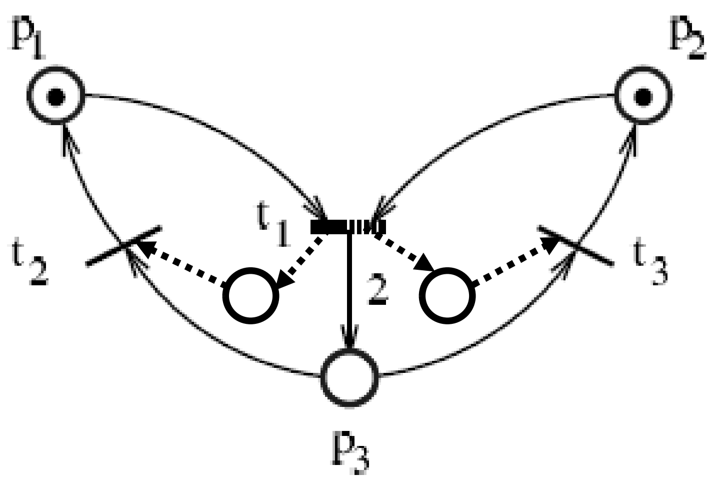
Dfm =

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

mf0 =

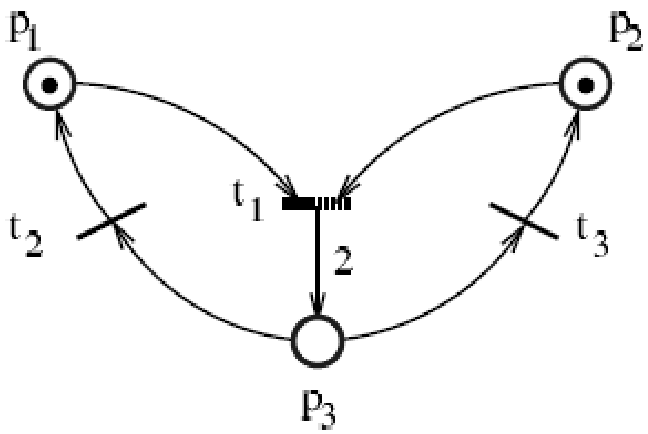
$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

^ *Bad news, supervisor touches t_1.*



Methods of Synthesis

Example: design controller with t1 unobservable (3/4)




```
D= [-1  1  0;
     -1  0  1;
      2 -1 -1];

Tuo= [1]; Tuc= [];

L= [-1 0 -1; 0 -1 -1];
b= [-1 -1]';

[La, ba, R1, R2] = mro_adm( L, b, D, Tuc, Tuo );
```

Solution obtained with the function MRO_ADM.m of the SPNBOX toolbox:

$$R1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{La} = \begin{bmatrix} -2 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix} \qquad \mathbf{ba} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$R2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Note: verify that $L_a \mu \leq b_a$ implies $L \mu \leq b$

Methods of Synthesis

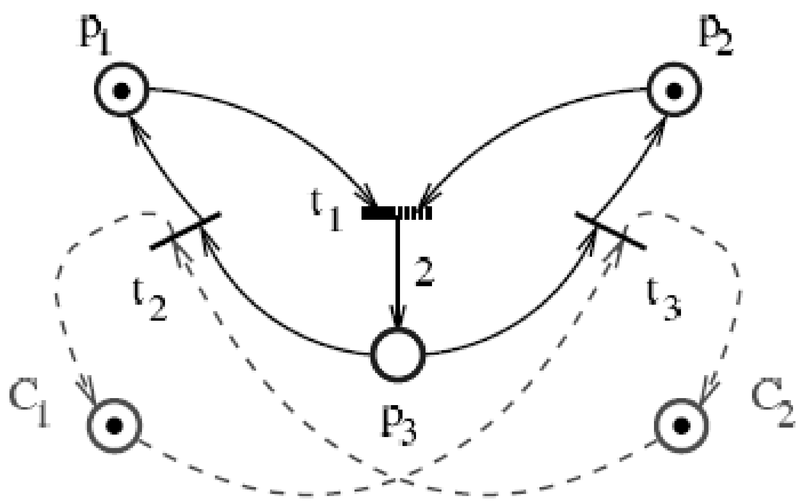
Example: design controller with t1 unobservable (4/4)

Finally the supervised controller is simply obtained from L_a and b_a:

$$D_c = -L_a D_p$$

$$= \begin{bmatrix} -2 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 2 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$



$$\mu_{c0} = b_a - L_a \mu_{p0}$$

$$= \begin{bmatrix} -1 \\ -1 \end{bmatrix} - \begin{bmatrix} -2 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Obtained the desired result: supervisor does not touch t_1.

End of chapter on supervision control.

What is next? *

** The starting point for studying and supervising Discrete Event Systems (DES), base elements of the Theory of Computation, have been introduced in early years of the ECE MSc*
https://fenix.tecnico.ulisboa.pt/disciplinas/ETC/2021-2022/1-semestre/programa
Many evolutions are expected to the teaching of supervision control!