# Computer Controlled Systems II – Diagnosis

## Model analysis and diagnosis
## Dynamic analysis of Petri nets

Katalin Hangos

University of Pannonia
Faculty of Information Technology
Department of Electrical Engineering and Information Systems

hangos.katalin@virt.uni-pannon.hu

Apr 2018

# Lecture overview

1. Supporting methods for the diagnosis
   - Diagnosis: the problem statement
   - Dynamic analysis
   - Observer design for state estimation
2. Dynamic analysis of Petri nets
   - Solution of Petri net models
   - The reachability graph
   - Reachability analysis
3. Solution and analysis of CPN models
   - Qualitative models and CPNs
   - CPNs: solution - traces

# Supporting methods for the diagnosis

# Prediction-based diagnosis

**General problem statement**

*Given:*

- *The number of faulty modes* $N_F$ (0=normal)
- *Predictive dynamic model* for each faulty mode

$$y^{(Fi)}(k+1) = \mathcal{M}^{(Fi)}(\mathcal{D}[1,k];\ p^{(Fi)}) \quad , \quad k = 1, 2, \ldots$$

- *Measured data record*: $D[0,k] = \{\ (u(\tau), y(\tau) \mid \tau = 0, \cdots, k\}$
- *Loss function* $J^{(Fi)}$, $i = 0, \cdots, N_F$

$$J^{(Fi)}(y - y^{(Fi)}, u) \ = \ \sum_{\tau=1}^{k} [\ r^{(i)T}(\tau) Q r^{(i)}(\tau)\ ] \ , \ r^{(i)}(\tau) = y(\tau) - y^{(Fi)}(\tau) \ , \ \tau = 1$$

*Compute*: The actual faulty mode of the system, i.e. the fault index $i$ that minimizes the loss function.

***Fault isolation***

# Identification-based diagnosis

**General problem statement**

*Given:*

- *The number of faulty modes* $N_F$ (0=normal)
- Predictive *parametric dynamic model* for each faulty mode

$$y^{(Fi)}(k+1) = \mathcal{M}^{(Fi)}(\mathcal{D}[1,k];\ p^{(Fi)}) \quad , \quad k = 1, 2, \ldots$$

- *Measured data record*: $D[0,k] = \{\ (u(\tau), y(\tau) \mid \tau = 0, \cdots, k\}$
- *Loss function depending on the parameters* $J^{(Fi)},\ i = 0, \cdots, N_F$

$$J^{(Fi)}(p^{(estFi)} - p^{(Fi)}) \ = \ \rho^{(i)T} Q \rho^{(i)} \quad , \quad \rho^{(i)} = p^{(estFi)} - p^{(Fi)}$$

*Compute*: The actual faulty mode of the system, i.e. the fault index $i$ that minimizes the loss function.

***Fault isolation***

# CT-LTI state-space models

- General form - revisited

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}u(t) \quad , \quad \boldsymbol{x}(t_0) = \boldsymbol{x_0}$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$$

with

- signals: $\boldsymbol{x}(t) \in \mathbb{R}^n$ , $\boldsymbol{y}(t) \in \mathbb{R}^p$ , $\boldsymbol{u}(t) \in \mathbb{R}^r$
- system parameters: $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ , $\boldsymbol{B} \in \mathbb{R}^{n \times r}$ , $\boldsymbol{C} \in \mathbb{R}^{p \times n}$ $(D = 0)$

# Controllability of CT-LTI systems

- Problem statement
  - *Given*:
    - a state-space model with parameters $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$
    - an initial state $\boldsymbol{x}(t_1)$ and a final state $\boldsymbol{x}(t_2) \neq \boldsymbol{x}(t_1)$
  - *Compute*:
    an input signal $\boldsymbol{u}(t)$ which moves the system from $\boldsymbol{x}(t_1)$ to $\boldsymbol{x}(t_2)$ in finite time

# Controllability of CT-LTI systems

## Theorem (Controllability)

*Given* $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ *for*

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)$$

*This SSR with state space* $\mathcal{X}$ *is* state controllable ***iff*** *the* controllability matrix $\mathcal{C}_n$ *is of* ***full rank***

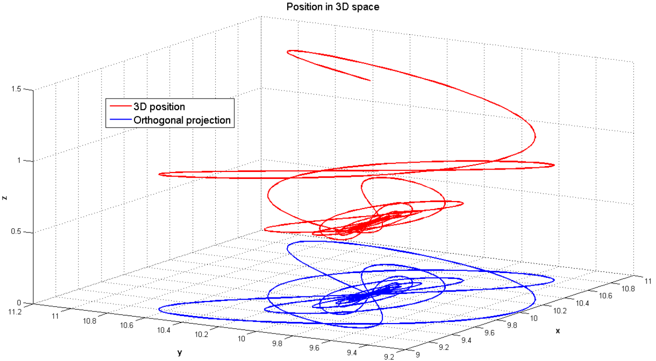
$$\mathcal{C}_n = \left[ \begin{array}{cccccc} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \boldsymbol{A}^2\boldsymbol{B} & . & . & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{array} \right]$$

*Kalman rank condition: If* $\dim \mathcal{X} = n$ *then rank* $\mathcal{C}_n = n$.

- Necessary and sufficient condition

# Observability of CT-LTI systems

- Problem statement
  - *Given*:
    - a state-space model with parameters $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$
    - a measurement record of $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ as over a finite time interval
  - *Compute*:
    - The state signal $\boldsymbol{x}(t)$ over the finite time interval
    - It is enough to compute $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$



# Observability of CT-LTI systems

## Theorem (Observability)

*Given* $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$. *This SSR with state space* $\mathcal{X}$ *is* state observable ***iff*** *the* observability matrix $\mathcal{O}_n$ *is of full rank*

$$\mathcal{O}_n = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{C}\boldsymbol{A}^{n-1} \end{bmatrix}$$

*Kalman rank condition: If* $\dim \mathcal{X} = n$ *then rank* $\mathcal{O}_n = n$.

- A necessary and sufficient condition

# Observer desing for CT-LTI systems

**Problem statement** *Given*:

- a SISO state-space model with parameters $(A, B, C)$
- a finite **measurement record** of $u$ and $y$ as signals
- an initial value $\hat{x}_0$

*Compute*:

An estimate of the state signal $x$ over the finite time interval such that $x(t) \to \hat{x}(t)$ as $t \to \infty$

# Observer equation

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

consider the **observer**

$$\frac{\hat{x}(t)}{dt} = A\hat{x}(t) + Bu(t) + L(y - C\hat{x}(t))$$

Introduce the **estimation error** signal: $\check{x} = x - \hat{x}$

$$\frac{\check{x}(t)}{dt} = (A - LC)\check{x}(t)$$

If the matrix $\check{A} = A - LC$ is a stability matrix then $\check{x} \to 0$ when $t \to \infty$ (asymptotic stability). **Task:** find $L$ such that $\check{A} = A - LC$ is a stability matrix

# Dynamic analysis of Petri net models

# Dynamics of Petri nets

**Marking function**: marking points (**tokens**)

$$\mu : \mathbf{P} \rightarrow \mathcal{N} \quad , \quad \mu(p_i) = \mu_i \geq 0$$
$$\underline{\mu}^T = [\mu_1, \mu_2, \ldots, \mu_n] \quad , \quad n = |\mathbf{P}|$$

Transition **fires** (operates): when its pre-conditions are "true" (there is a **token** on its input places)

$\underline{\mu}^{(i)}[t_j > \underline{\mu}^{(i+1)}$

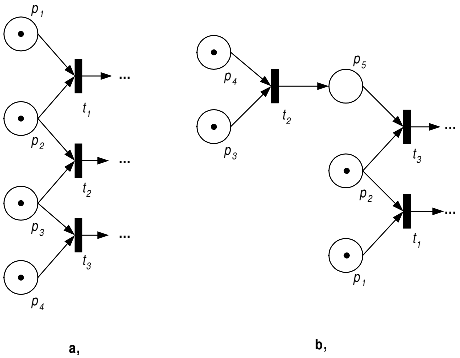
after firing the consequences become "true"

**Firing (operation) sequence**

$$\underline{\mu}^{(0)}[t_{j0} > \underline{\mu}^{(1)}[t_{j1} > ...[t_{jk} > \underline{\mu}^{(k+1)}$$

# Parallel events

**More than one enabled (fireable) transition**:
concurrency (independent conditions), conflict, confusion



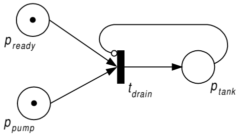
# Conflict resolution

Using **inhibitor edges**:

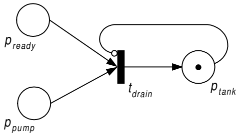
- priority given by the user
- test edges

**Other solutions**:

- capacity of the places

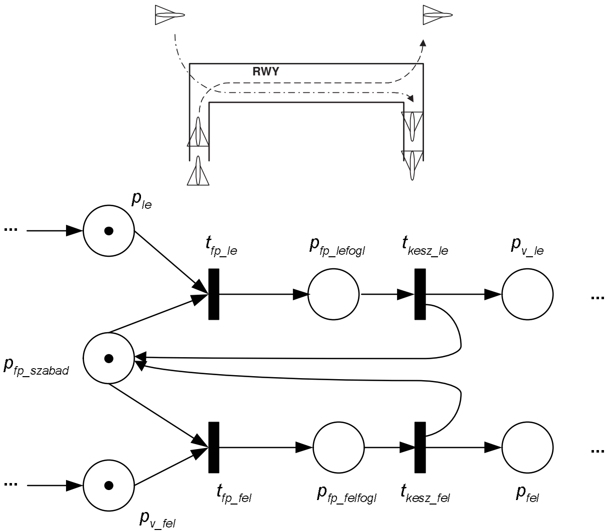



a,



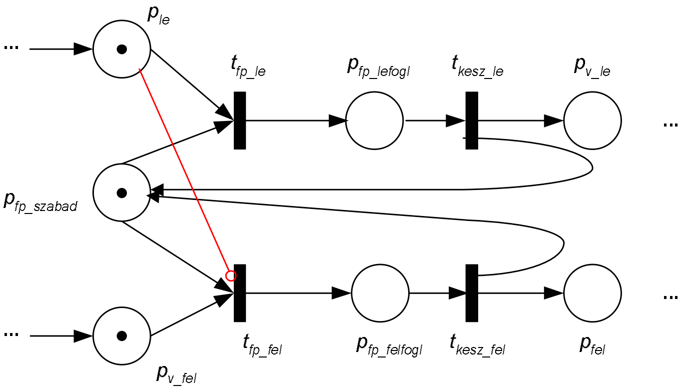

b,

# Petri net model of a runway – 1



# Petri net model of a runway – 2

**Conflict resolution**: landing aircraft has priority



# The solution problem

***Abstract problem statement***

**Given**:

- a *formal description* of a discrete event system model
- *initial state(s)*
- *external events*: system inputs

**Compute**:

- the sequence of *internal (state and output) events*

The solution is **algorithmic**! **The problem is NP-hard**!

# Petri net models – reachability graph

**Solution**: marking (systems state) sequences

**reachability graph (tree)** (weighted directed graph)

- *vertices*: markings
- *edges*: if exists transition the firing of which connects them
- *edge weights*: the transition and the external events

**Construction**:

1. *start*: at the given initial state (marking)
2. *adding a new vertex*: by firing an enabled transition (with the effect of inputs!)

May be NP-hard (in conflict situation or non-finite operation)

# The state space of Petri net models

**State vector**: marking in *internal* places
in- and out-degree is at least 1

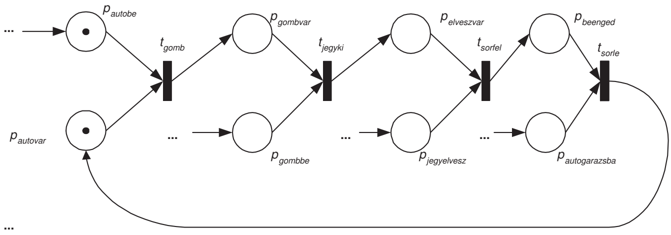
$$x(k) \quad \sim \quad \underline{\mu}_x^{(k)}$$

**Inputs**: marking in *input* places
in-degree is zero

$$u(k) \quad \sim \quad \underline{\mu}_u^{(k)}$$

# Example: garage gate

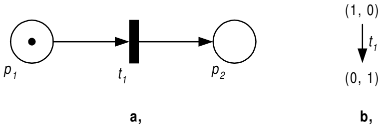
## Petri net model



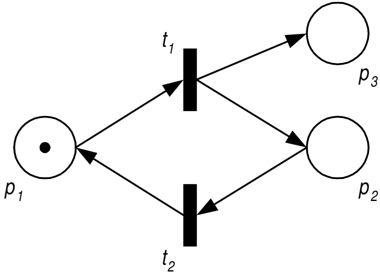

$$\underline{\mu}_x^T = [\mu_{autovar}, \mu_{gombvar}, \mu_{elveszvar}, \mu_{beenged}]$$
$$\underline{\mu}_u^T = [\mu_{autobe}, \mu_{gombbe}, \mu_{jegyelevesz}, \mu_{autogarazsba}]$$

# Reachability graphs

## Finite case

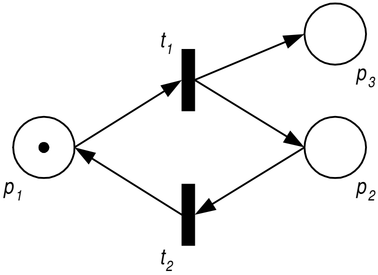


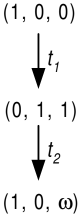

## Non-finite case



# Non-finite reachability graph

**Reduction**: using the $\omega$ symbol



# Analysis of Petri net models

## Dynamic properties

- *behavioural* (initial state dependent)
- *structural* (only depends on the structure graph)

## Behavioural properties

- *reachabiliy* (coverability, controllability)
- *deadlocks*, liveness
- *boundedness*, safeness
- (token) conservation

## Structural properties

- *state and transition invariant*: cyclic behaviour

# Reachability of Petri net models

The notion of **reachability**: whether there exists

- to a given *[initial state* ($\underline{\mu}^{(I)}$)*, final state* ($\underline{\mu}^{(F)}$)*]* pair
- *a firing sequence*, such that

$$\underline{\mu}^{(I)}[t_{j0} > \underline{\mu}^{(1)}[t_{j1} > ...[t_{jk} > \underline{\mu}^{(F)}$$

The notion of **coverability**:

$$\underline{\mu}'' \geq \underline{\mu}' \quad \Leftrightarrow \quad \forall i : \; \mu_i'' \geq \mu_i'$$

**The same as the usual controllability**

# Boundedness of Petri nets

Related properties to **boundedness**

- *finiteness (boundedness)*: Is the number of tokens finite for every initial state?
- *Safeness*: the bound is 1 for each place

Can be defined (examined) for the **whole net** or only for **a given set of places**

**Conservative Petri net**: the number of tokens is constant (resource-conservation)

# Liveness of Petri nets

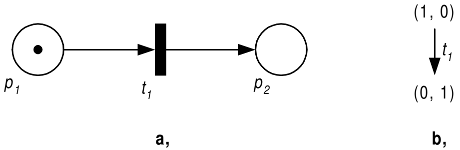
The notion of **liveness**: from a given initial state

- for a *transition*: is there a firing sequence when the transition is active?
- for a *set of transition*, for the whole net

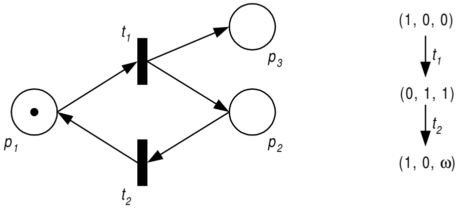
**Deadlock**: a non-final state from where there is no enabled (fireable) transition

# Simple Petri net examples

**Deadlock**: the marking $(0,1)$




**Non-bounded place**: $p_3$



# Dynamic analysis methods of Petri net models – 1

Analysis of **behavioural properties**

- by constructing the *reachability graph*
- and *searching* on the vertices of the graph
- may be *NP-hard*

Problems:

- cyclic behaviour
- non-bounded places

# Solution and analysis of CPN models

# The origin of qualitative models

Engineering dynamical models in **state-space form**:

$$\begin{aligned} \frac{dx}{dt} &= f(x,u) \quad \text{(state eq.)} \\ y &= h(x,u) \quad \text{(output eq.)} \end{aligned}$$

**Qualitative models** can be derived *systematically* from engineering models by using

- interval-valued variables and parameters
- simplified equations

# The derivation of discrete time qualitative DAEs

Dynamic models derived from first engineering principles: continuous time differential-algebraic equation models

- differential equations originate from conservation balances: ***to be transformed to difference equations*** (time discretization)
- selection of the ***qualitative range spaces*** of variables and parameters
- deriving the qualitative form

# Qualitative signals

**Qualitative range spaces**

$$\mathcal{Q} = \{H, N, L, 0\}, \quad \mathcal{B} = \{0, 1\}, \quad \mathcal{Q}_{\mathcal{E}} = \{H, N, L, 0, e+, e-\}$$

with *H*igh, *L*ow, *N*ormal, *e*rror.

**A qualitative signal** is a signal (input, output, state and ***disturbance (fault indicator)***) that takes its values from a finite qualitative range set

**An event** is generated when a qualitative signal changes its value. An event $e_X$ is formally described by a pair $e_X(t, q_X) = (t, [x](t) = q_X)$ where $t$ is the occurrence time when the qualitative signal $[x]$ takes the value $q_X$.

# Normalized intervals

**Qualitative range space**: for variables with "normal" $N$ value

$$\mathcal{Q} = \{H, N, L, 0\}, \quad \mathcal{B} = \{0, 1\}, \quad \mathcal{Q}_E = \{H, N, L, 0, e+, e-\}$$

***Intervals with non-fixed endpoints***

Operation table for interval addition

| $[a] + [b]$ | $0$ | $L$ | $N$ | $H$ |
|---|---|---|---|---|
| $0$ | $0$ | $L$ | $N$ | $H$ |
| $L$ | $L$ | $N$ | $H$ | $e+$ |
| $N$ | $N$ | $H$ | $e+$ | $e+$ |
| $H$ | $H$ | $e+$ | $e+$ | $e+$ |

This is only a possible definition!

# Solution of a qualitative DAE

In the form of a **solution table**
(interval operation table)

- collect all of the *right-hand side variables* (time-dependent values!)
- enumerate all of their signal traces
- systematically enumerate all of the **possible combinations**
  $\Longrightarrow$ exponentially growing size with the number of variables

# A static example: sensor with additive type fault

**Algebraic model equation**: $v^m = v + \chi \cdot E$
$[v] \in \mathcal{Q}$, $[v]^m \in \mathcal{Q}_e$, $\chi \in B_{-1} = \{-1, 0, 1\}$

| $[v^m]$ | $[\chi]$ | $[v]$ | mode |
|---|---|---|---|
| $N$ | 0 | $N$ | normal |
| $H$ | 0 | $H$ | normal |
| $L$ | 0 | $L$ | normal |
| 0 | 0 | 0 | normal |
| $e+$ | 1 | $H$ | faulty |
| $H$ | 1 | $N$ | faulty |
| $N$ | 1 | $L$ | faulty |
| $L$ | 1 | 0 | faulty |
| $N$ | -1 | $H$ | faulty |
| $L$ | -1 | $N$ | faulty |
| 0 | -1 | $L$ | faulty |
| $e-$ | -1 | 0 | faulty |

# A dynamic example: mass balance of the coffee machine

**Differential equation in discrete form**: $h^T = h + \chi_I \cdot v - \chi_O \cdot v$
$[h], [h]^T \in \mathcal{Q}_e$, $\chi_I, \chi_O \in \mathcal{B}$ and $[v] = L$
**Solution for constant inputs**

| $[h]^T$ | $[h](t_0)$ | $\chi_I$ | $\chi_O$ |
|---|---|---|---|
| $(N, N, N)$ | $N$ | (1,1,1) | (1,1,1) |
| $(L, L, L)$ | $L$ | (1,1,1) | (1,1,1) |
| ... | ... | ... | ... |
| $(N, N, N)$ | $N$ | (0,0,0) | (0,0,0) |
| ... | ... | ... | ... |
| $(e+, e+, H)$ | $N$ | (1,1,1) | (0,0,0) |
| $(e+, H, N)$ | $L$ | (1,1,1) | (0,0,0) |
| ... | ... | ... | ... |
| $(e-, 0, L)$ | $N$ | (0,0,0) | (1,1,1) |
| $(e-, e-, 0)$ | $L$ | (0,0,0) | (1,1,1) |
| ... | ... | ... | ... |

# CPN and qualitative models

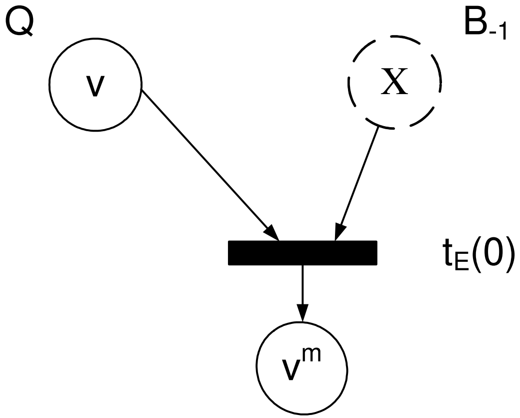
**Coloured Petri Net model** (CPN): can be obtained from a qualitative model

- colour sets: from the qualitative range space of the variables
- spaces: associated to variables
- transitions: associated to the equations (static [output] and dynamic [state] equations)

*Diagnostic applications: the **faults should be modelled***

# Static example: sensor with additive fault 2

CPN modell



# Qualitative signals

**Qualitative values** for variables with "normal" $N$ value

$$\mathcal{Q} = \{H, N, L, 0\}, \quad \mathcal{B} = \{0, 1\}, \quad \mathcal{Q}_{\mathcal{E}} = \{H, N, L, 0, e+, e-\}$$

where *H*igh, *L*ow, *N*ormal, *e*rror.

**Qualitative signal**: a signal (input, output, state or ***disturbance (fault indicator!)***) with a qualitative range space

**Event**: occurs when a qualitative signal changes its value.
Formal description of the event $e_X$:

$$e_X(t, q_X) = (t, [x](t) = q_X)$$

where $t$ is the discrete time instant when the qualitative signal $[x]$ takes the value $q_X$.

# Signal traces – event sequences

The **(signal trace)** of a qualitative signal $[x]$ is the event sequence

$$\mathcal{T}_{(x)}(t_0, t_F) = \{(t_0; [x](t_0) = q_{x0}), (t_1; [x](t_1)] = q_{x1}), ..., (t_F; [x](t_F) = q_{xF})\}$$

defined on the time interval $(t_0, t_F)$ with $q_* \in \mathcal{Q}_x$

A **vector-valued trace of multiple signals** is defined as $\mathcal{T}_{(u,d,y)}(t_0, t_F)$
Simplified notation: by omitting the time, e.g.

$$\mathcal{T}_{(h,T)}(1,3) = \{(N,N),(L,H),(L,e+)\}$$

*For diagnostic purposes* we define

- **nominal traces** (for describing normal behaviour)
- **characteristic traces** (for describing faulty behaviour)

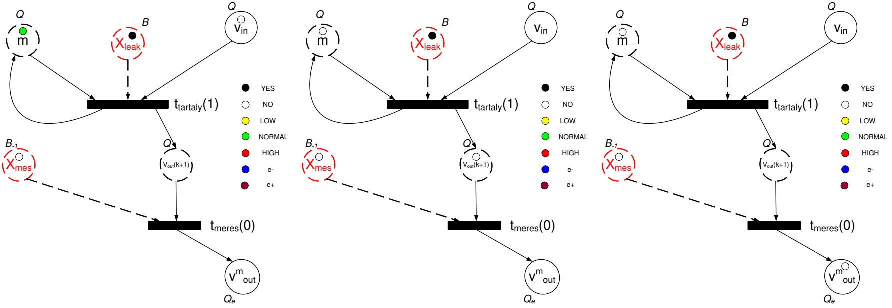
# Simple dynamic example – 1

Tank with free outflow: qualitative model equations

$$[m](k+\mathbf{1}) = [m](k) + [v_{in}](k) - K \cdot [m](k) - \chi_{leak} \cdot B$$

*"small" leakage - $[B] = L$*
+ sensor *with additive fault*

**CPN modell**:



# Simple dynamic example – 2

Solution: **qualitative input-output traces**

| | $[m]$ initial mass in tank | $[v_{in}]$ input flow sequence | $[\chi_{leak}]$ * tank leakage | $[\chi_{meas}]$ * sensor failure | $[v_T^m]$ measured flow sequence |
|---|---|---|---|---|---|
| | LOW | (NORMAL,NORMAL,NORMAL) | 0 | NEG | (LOW, LOW, LOW) |
| | LOW | (NORMAL,NORMAL,NORMAL) | 1 | POS | (LOW, LOW, LOW) |
| | HIGH | (LOW, LOW, LOW) | 1 | 0 | (LOW, NO, NO) |
| | HIGH | (LOW, LOW, LOW) | 0 | NEG | (LOW, NO, NO) |
| | ... | ... | ... | ... | ... |
| D | NORMAL | (NO, NO, NO) | 0 | 0 | (LOW, NO, NO) |
| D | NORMAL | (NO, NO, NO) | 1 | POS | (LOW, LOW, LOW) |
| | NORMAL | (NO, NO, NO) | 1 | NEG | (e-, e-, e-) |
| | NORMAL | (NO, NO, NO) | 0 | POS | (NORMAL, LOW, LOW) |
| D | NORMAL | (NO, NO, NO) | 0 | NEG | (NO, e-, e-) |
| D | NORMAL | (NO, NO, NO) | 1 | 0 | (NO, NO, NO) |
| | ... | ... | ... | ... | ... |