University of Ljubljana

Faculty of Electrical Engineering

Model formulations of complex systems for predictive control design

Asst. Prof. Dr. Gorazd Karer

Sofia, October 2016

Outline

- Modelling of a hybrid fuzzy model:
  - Hybrid system hierarchy
  - Generalization of the TS formulation for a nonlinear hybrid system
- Identification of a hybrid fuzzy model:
  - Fuzzy clustering
  - Projections of the fuzzy clusters into the input space
  - Estimation of the parameters
- Batch reactor:
  - Validation
- Predictive control implementations
- Conclusions

Complex systems: modelling and control

- Many systems met in practice: hybrid, nonlinear.
  - Hybrid systems:
    - Interlacing of continuous and discrete dynamics
      - Continuous dynamics
      - Autonomous switching
      - Autonomous jumps
      - Controlled switching
      - Controlled jumps
  - Nonlinearity:
    - Fuzzy logic

- Predictive control:
  - Well established in practice (simple, useful, good performance, comprehensibale algorithms)
  - Predictive control of complex systems … complex (no general model)

- Classic modelling and identification methods from linear-system theory -> inadequate.

- Special methods and formulations needed.

PWA or equivalent models

- Models for predictive control:

PWA or equivalent models

- Models for predictive control:
  - piecewise affine – PWA

$$x(k+1) = A_i x(k) + B_i u(k) + f_i$$

$$y(k) = C_i x(k) + D_i u(k) + g_i$$

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i$$

$$\Omega_i \equiv \left\{ \begin{bmatrix} x(\cdot) \\ u(\cdot) \end{bmatrix} \; ; \; \begin{array}{l} H_i x(\cdot) + J_i u(\cdot) \leq K_i, \\ \tilde{H}_i x(\cdot) + \tilde{J}_i u(\cdot) < \tilde{K}_i \end{array} \right\}$$

PWA or equivalent models

- Models for predictive control:
  - piecewise affine – PWA
  - linear complementarity – LC

$$x(k+1) = Ax(k) + B_1u(k) + B_2w(k)$$

$$y(k) = Cx(k) + D_1u(k) + D_2w(k)$$

$$v(k) = E_1x(k) + E_2u(k) + E_3w(k) + g_4$$

$$0 \leq v(k) \perp w(k) \geq 0$$

PWA or equivalent models

- Models for predictive control:
  - piecewise affine – PWA
  - linear complementarity – LC
  - extended linear complementarity – ELC

$$x(k+1) = Ax(k) + B_1u(k) + B_2d(k)$$

$$y(k) = Cx(k) + D_1u(k) + D_2d(k)$$

$$E_1x(k) + E_2u(k) + E_3d(k) \leq g_4$$

$$\sum_{i=1}^{p} \prod_{j\in\phi_i} (g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j = 0$$

PWA or equivalent models

- Models for predictive control:
  - piecewise affine – PWA
  - linear complementarity – LC
  - extended linear complementarity – ELC
  - maxmin-plus-scaling – MMPS

$$x(k+1) = \mathcal{M}_x(x(k), u(k), d(k))$$

$$y(k) = \mathcal{M}_y(x(k), u(k), d(k))$$

$$\mathcal{M}_c(x(k), u(k), d(k)) \leq c$$

$$f := x_i | \alpha | max(f_k, f_l) | min(f_k, f_l) | f_k + f_l | \beta f_k$$

$$\text{kjer je } i \in \{1, 2, \ldots, n\}$$

$$\text{in } \alpha, \beta \in \mathbb{R}$$

PWA or equivalent models

- Models for predictive control:
  - piecewise affine – PWA
  - linear complementarity – LC
  - extended linear complementarity – ELC
  - maxmin-plus-scaling – MMPS
  - mixed logical dynamical – MLD

$$x(k+1) = Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k)$$

$$y(k) = Cx(k) + D_1 u(k) + D_2 \delta(k) + D_3 z(k)$$

$$E_1 x(k) + E_2 u(k) + E_3 \delta(k) + E_4 z(k) \leq g_5$$

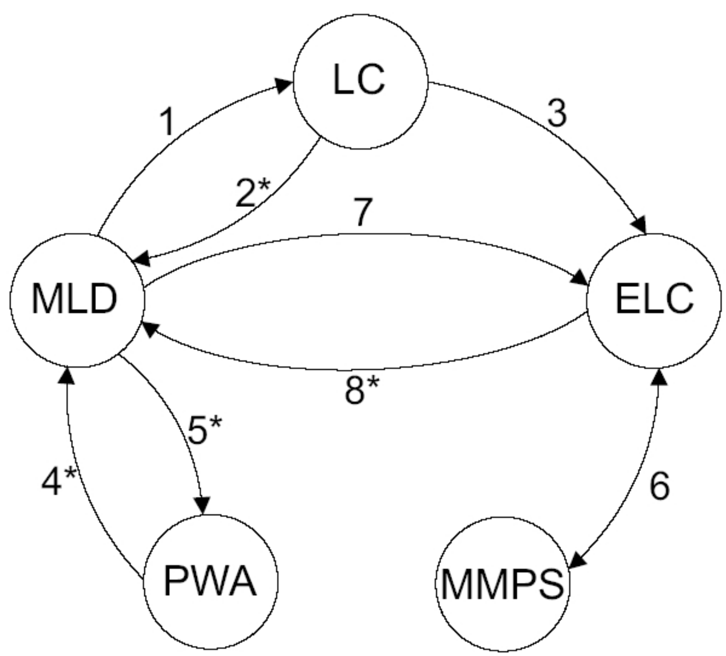
$$z(k) \in \mathbb{R}^{r_z}$$

$$\delta(k) \in \{0,1\}^{r_b}$$

PWA or equivalent models

- Models for predictive control:
  - piecewise affine – PWA
  - linear complementarity – LC
  - extended linear complementarity – ELC
  - maxmin-plus-scaling – MMPS
  - mixed logical dynamical – MLD

- Conversion possible:



PWA or equivalent models

- Predictive control:
  - Mostly based on PWA or MLD models
  - Basis for mixed-integer optimization problems formulation

- Drawbacks –> nonlinearities !

  - Theoreticaly allow arbitrary precise approximation;

  - Problems of fine state-space segmentation:
    - Modelling issues
    - Additional auxiuliary discrete variables for optimization => increase of computational complexity

Hybrid Fuzzy Model

- Hierarchy: discrete part is atop the continuous part.

$$\mathbf{x}(k+1) = \mathbf{f}_q(\mathbf{x}(k), \mathbf{u}(k)) \qquad \mathbf{x} \in R^n$$
$$q(k+1) = g(\mathbf{x}(k), q(k), \mathbf{u}(k)) \qquad \mathbf{u} \in R^m$$
$$q \in Q \qquad Q = \{1, \ldots, s\}$$

- Generalization of the strict Witsenhausen hybrid system formulation.

Takagi-Sugeno Fuzzy Formulation

- Approximates the nonlinear system.
- The rule base of the fuzzy system (*K* rules)

$$\mathbf{R}^{jd}:$$

$$\textbf{if } q(k) \; is \; Q_d \; and \; y(k) \; is \; A_1^j \; and \; \ldots \; and \; y(k-n+1) \; is \; A_n^j$$

$$\textbf{then } \hat{y}_p(k+1) = f_{jd}(y(k),\ldots,y(k-n+1),u(k),\ldots,u(k-m+1))$$

$$for \; j = 1,\ldots,K \; and \; d = 1,\ldots,s$$

- Consequences – affine functions

$$\hat{y}_p^{jd}(k+1) =$$

$$= f_{jd}(y(k),\ldots,y(k-n+1),u(k),\ldots,u(k-m+1)) =$$

$$= a_{1jd} \; y(k) + \ldots + a_{njd} \; y(k-n+1) +$$

$$+ b_{1jd} \; u(k) + \ldots + b_{mjd} \; u(k-m+1) + r_{jd}$$

Hybrid Fuzzy Model Output

- Output *y -> compact form*

$$\hat{y}_p(k+1) = \beta(k)\,\Theta^T(k)\,\psi(k)$$

Hybrid Fuzzy Model Output

- Output *y -> compact form*

$$\hat{y}_p(k+1) = \beta(k)\,\Theta^T(k)\,\psi(k)$$

- Normalized degree of fulfillment

$$\beta(k) = [\beta_1(k)\ \beta_2(k)\ \ldots\ \beta_K(k)]$$

$$\beta_j(k) = \frac{\mu_{A_1^j}(y(k)) \cdot \ldots \cdot \mu_{A_n^j}(y(k-n+1))}{\sum_{i=1}^{K} \mu_{A_1^i}(y(k)) \cdot \ldots \cdot \mu_{A_n^i}(y(k-n+1))}$$

Hybrid Fuzzy Model Output

- Output *y -> compact form*

$$\hat{y}_p(k+1) = \beta(k)\,\Theta^T(k)\,\psi(k)$$

- Matrix with the consequent fuzzyfied parameters (*n+m+1* rows and *K* columns)

$$\Theta(k) = \Theta(q(k)) = \begin{Bmatrix} \Theta_1 & \text{if} & q(k)=1 \\ \vdots & & \vdots \\ \Theta_s & \text{if} & q(k)=s \end{Bmatrix}$$

$$\Theta_d^T = \begin{bmatrix} a_{11d} & \cdots & a_{n1d} & b_{11d} & \cdots & b_{m1d} & r_{1d} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ a_{1Kd} & \cdots & a_{nKd} & b_{1Kd} & \cdots & b_{mKd} & r_{Kd} \end{bmatrix}$$

Hybrid Fuzzy Model Output

- Output *y -> compact form*

$$\hat{y}_p(k+1) = \beta(k)\,\Theta^T(k)\,\psi(k)$$

- Regressor

$$\psi^T(k) = \begin{bmatrix} y(k) & \cdots & y(k-n+1) & u(k) & \cdots & u(k-m+1) & 1 \end{bmatrix}$$

Identification of the Hybrid Fuzzy Model

- Dynamics of the system often not known well enough to determine suitable fuzzy sets (premise) -> undefined membership functions.
- Fuzzy c-means clustering algorithm.

Fuzzy clustering

- Carried out over the data in $\mathcal{D}_{IO}$
- Separates the data into fuzzy clusters
- Every piece of identification data is a member of a particular fuzzy cluster:

  membership degree <-> distance from center

Fuzzy c-means clustering

- Minimization of the criterion

$$J_{MR} = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^{m} \|x_i - c_j\|^2$$

Predefined real number 1≤m<∞

i-th data point

Euclidean norm

$$\|x\| = \sqrt{x^T \cdot x}$$

Value of the membership (i-th data point, j-th cluster)

Number of data points

Number of clusters

Identification of clusters

Centers

$$\mu_{ij} = \frac{}{\sum_{k=1}^{C}} \qquad c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} \mu_{ij}^{m}}$$

Fuzzy c-means clustering

Steps in the algorithm:

1. Set the number of clusters C and the parameter m and establish the initial membership matrix $\Upsilon(0) = [\mu_{ij}]$.
2. In kth iteration determine the centers of the clusters c_j for $j = 1, \ldots, C$ according to $\Upsilon(k)$.
3. Calculate the new membership matrix $\Upsilon(k+1)$.
4. If $\|\Upsilon(k+1) - \Upsilon(k)\| < \varepsilon$ stop the algorithm, otherwise continue from step 2.

Projections of the fuzzy clusters

- Problem: c_j and μ_{ij} are defined in $\mathcal{D}_{IO}$:
  - \+ μ_{ij} can directly be used for parameter estimation
  - \- not usable for predicting the behaviour of the system in MPC strategies – impossible to determine $\beta_j(k)$, because μ_{ij} depend on $\|x_i - c_j\|$, which are defined in $\mathcal{D}_{IO}$. <– Model for MPC: vectors in $\mathcal{D}_I$.
- Solution: mapping the information on membership functions from $\mathcal{D}_{IO}$ to $\mathcal{D}_I$.
- μ_{ij} are calcullated from $\|x_i - c_j\|$ => map the distances from $\mathcal{D}_{IO}$ to $\mathcal{D}_I$.

Projections of the fuzzy clusters

- The points that are equidistant from a chosen point in $\mathcal{D}_{IO}$: hypersphere in $\mathcal{D}_{IO}$:

$$[x - c_j]^T \cdot [x - c_j] = r_j^2 = \|x - c_j\|^2$$

Points on the hypersphere

Center of the hypersphere

Distance from the center of the hypersphere

Projections of the fuzzy clusters

- In the HFM, every rule represents a particular (affine) function in $\mathcal{D}_{IO}$: hyperplane in $\mathcal{D}_{IO}$:

$$[x - s_j]^T \cdot n_j = 0$$

Points on the hyperplane

Arbitrary point on the hyperplane

Normal vector

Projections of the fuzzy clusters

- Split the vector x into 2 components $x = \begin{bmatrix} x_I \\ x_O \end{bmatrix}$
  - MPC deals with x_I
  - x_O can be regarded as a parameter:

$$[x - c_j]^T \cdot [x - c_j] = r_j^2 = \|x - c_j\|^2$$
$$[x - s_j]^T \cdot n_j = 0$$

- The parametrized equations define a contour in $\mathcal{D}_{IO}$: 3 alternatives:
  - If $0<r_j<r_{j,min}$ => contour does not exist.
  - If $r_j=r_{j,min}$ => contour degenerates into a point.
  - If $r_j>r_{j,min}$ => contour is a hypercircle.

Projections of the fuzzy clusters

- The hypercircle defined in $\mathcal{D}_{IO}$ can be projected into $\mathcal{D}_I$: hyperellipse in $\mathcal{D}_I$:

$$[x_I - s_{j,I}]^T \; A_{r_j} \; [x_I - s_{j,I}] = r_j^2$$

Points on the hyperellipse

Center of the hyperellipse

Square matrix (orientation and lengths of semiaxes)

Projections of the fuzzy clusters

- The hypercircle defined in $\mathcal{D}_{IO}$ can be projected into $\mathcal{D}_I$: hyperellipse in $\mathcal{D}_I$:

$$[x_I - s_{j,I}]^T \ A_{r_j} \ [x_I - s_{j,I}] = r_j^2$$

- Derive C distance functions

$$r_j : \mathcal{D}_I \to [r_{j,min}, \infty),$$
$$r_j : x_I \mapsto r_j(x_I),$$
$$\text{where } j \in \{1, 2, \dots, C\}$$

- Using these functions we can assign a membership value to every x_I in $\mathcal{D}_I$:

$$\mu_j = \frac{1}{\sum_{k=1}^{C} \left(\frac{r_j}{r_k}\right)^{\frac{2}{m-1}}}$$

Estimation of parameters

- The regression matrix Ψ_{jd} for the rule $\mathbf{R}^{jd}$

$$\mathbf{\Psi}_{jd} = \begin{bmatrix} \beta_j(k_1)\,\psi^T(k_1) \\ \vdots \\ \beta_j(k_{Pjd})\,\psi^T(k_{Pjd}) \end{bmatrix}$$

$$q(k) = d$$

$$\beta_j(k) \geq \delta$$

- The output data vector for the rule $\mathbf{R}^{jd}$

$$\mathbf{Y}_{jd} = \begin{bmatrix} \beta_j(k_1)\,y(k_1+1) \\ \vdots \\ \beta_j(k_1)\,y(k_{Pjd}+1) \end{bmatrix}$$

Estimation of parameters

- The output contribution for the rule $\mathbf{R}^{jd}$

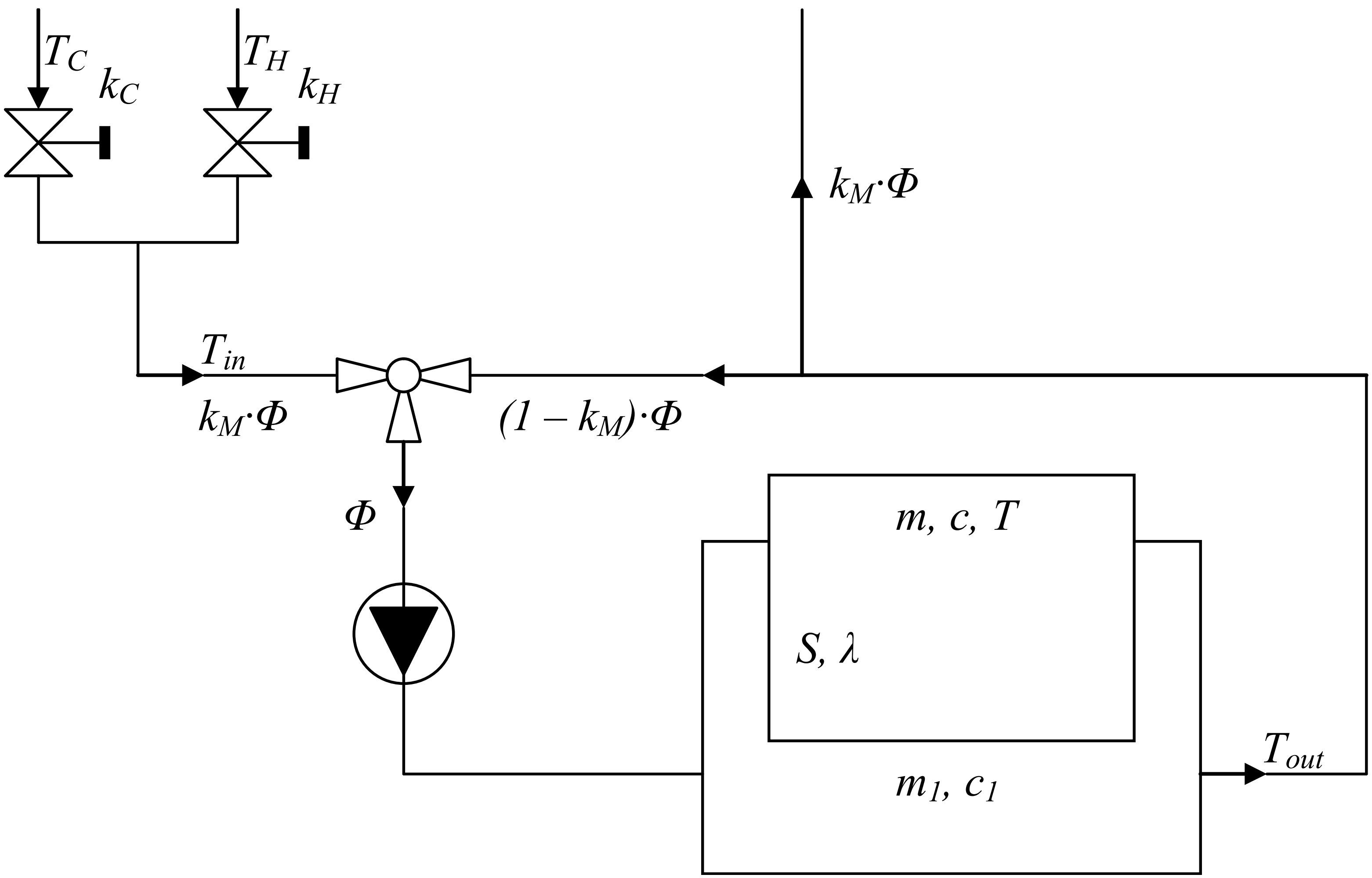
$$\beta^{j}(k_1)\,\hat{y}_p^{jd}(k+1) = \Theta_{jd}^{T}\,(\beta^{j}(k_1)\,\psi(k))$$

$$\Theta_{jd}^{T} = [a_{1jd}\ \dots\ a_{njd}\ \ b_{1jd}\ \dots\ b_{mjd}\ \ r_{jd}]$$

- Least-squares method -> estimation of parameters for each rule individually

$$\Theta_{jd} = (\Psi_{jd}^{T}\Psi_{jd})^{-1}\Psi_{jd}^{T}Y_{jd}$$

The Batch Reactor



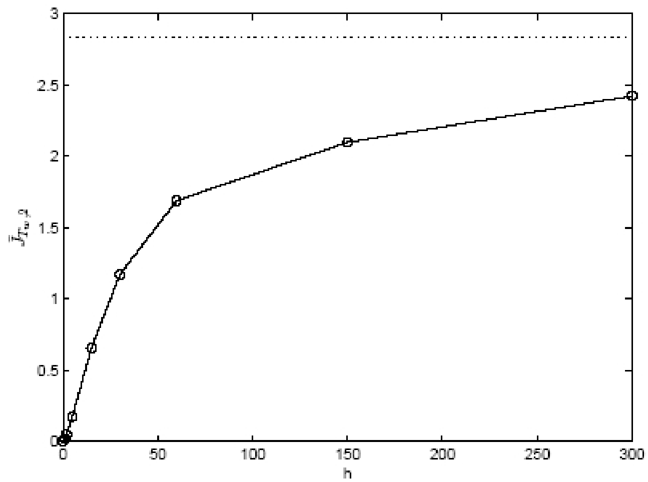
Modelling: PWA, HFM

- PWA model – 2 approaches:
  1. Heuristic partitioning of the input space
  2. Fuzzy clustering

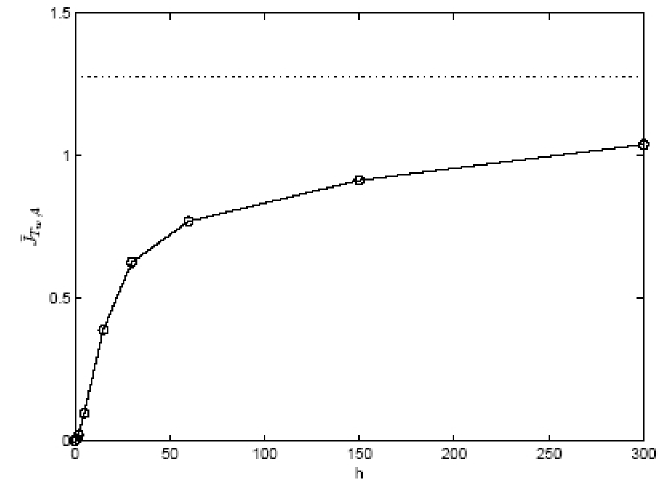
- HFM – 2 approaches :
  1. Heuristic partitioning of the input space
  2. Fuzzy clustering

- In every approach 10 partitions / membership functions have been established in the input space

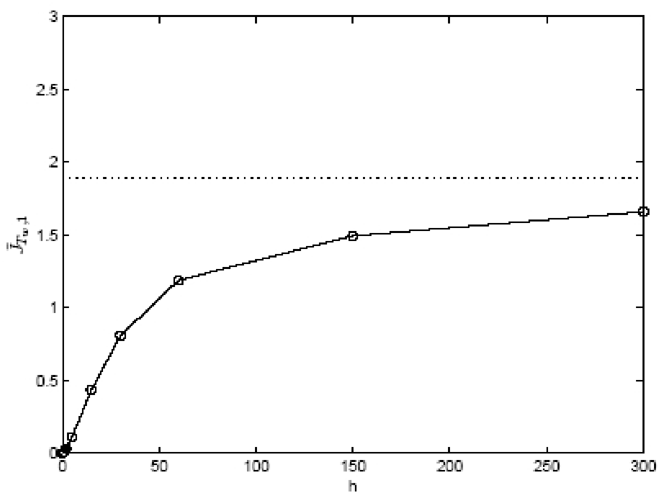
Validation: average square error



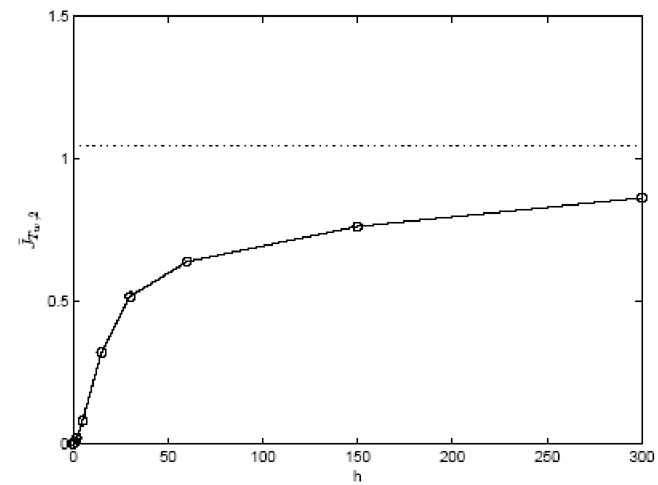
PWA – Heuristic partitioning



PWA – Fuzzy clustering



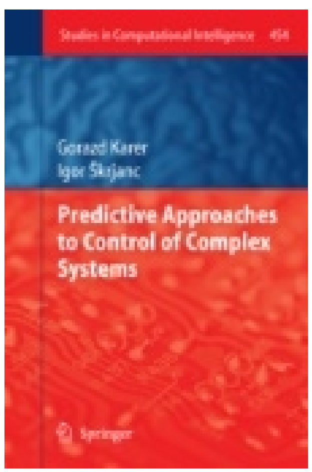
HFM – Heuristic partitioning

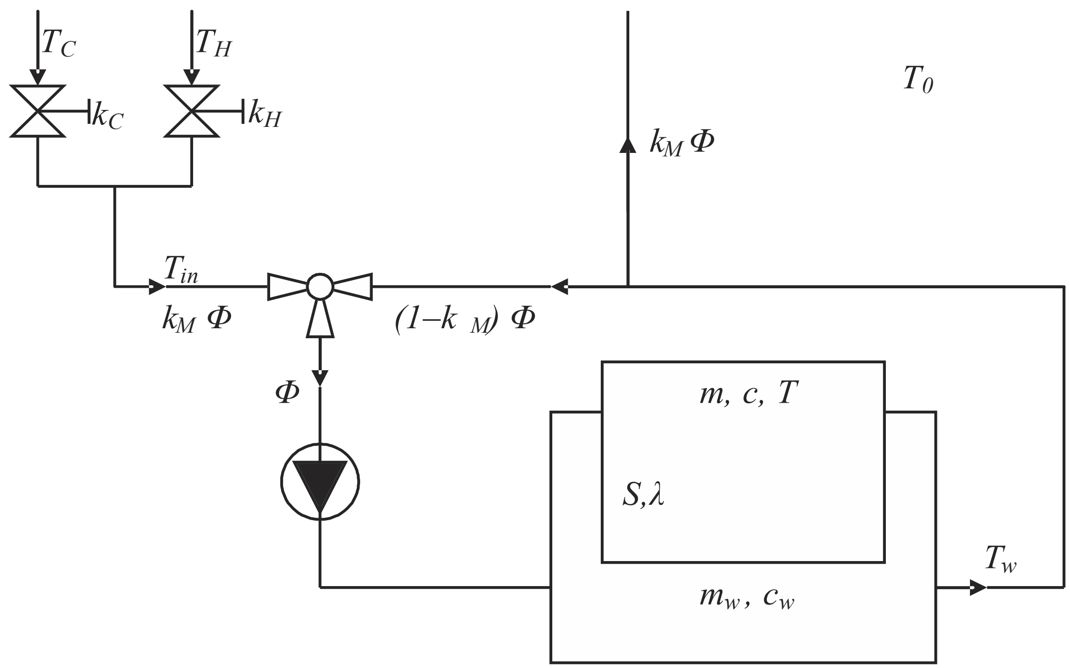


HFM – Fuzzy clustering

Predictive approaches to control of complex systems

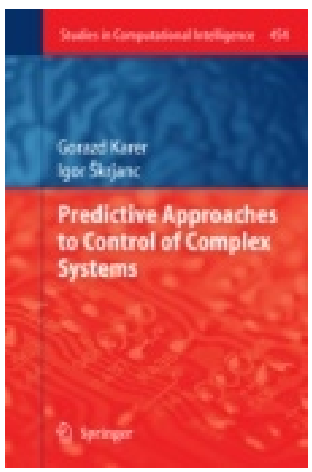
- Modeling of complex systems for predictive control
  - Complex dynamics: properties and formulations
  - Hybrid fuzzy model
  - Unsupervised learning methods for identification of complex systems
  - Modeling and identification of a batch reactor – case study

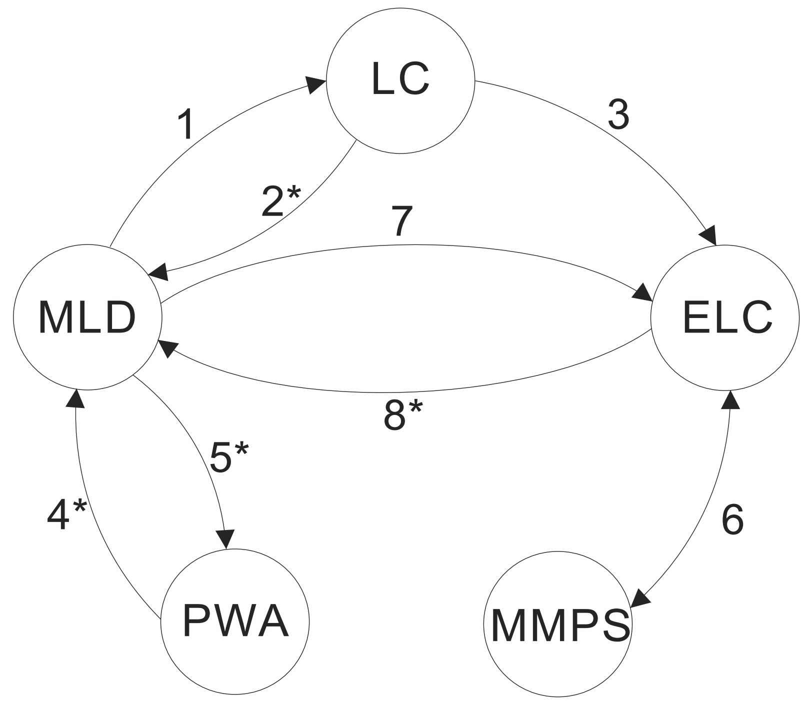




Predictive approaches to control of complex systems

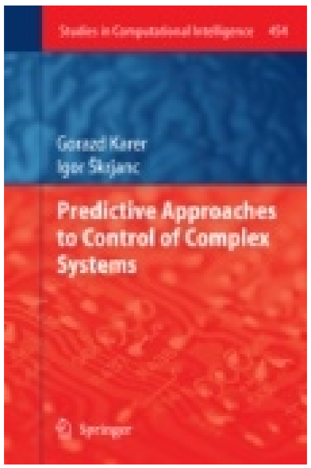
- Predictive control of complex systems
  - Solving mixed-integer optimization problems

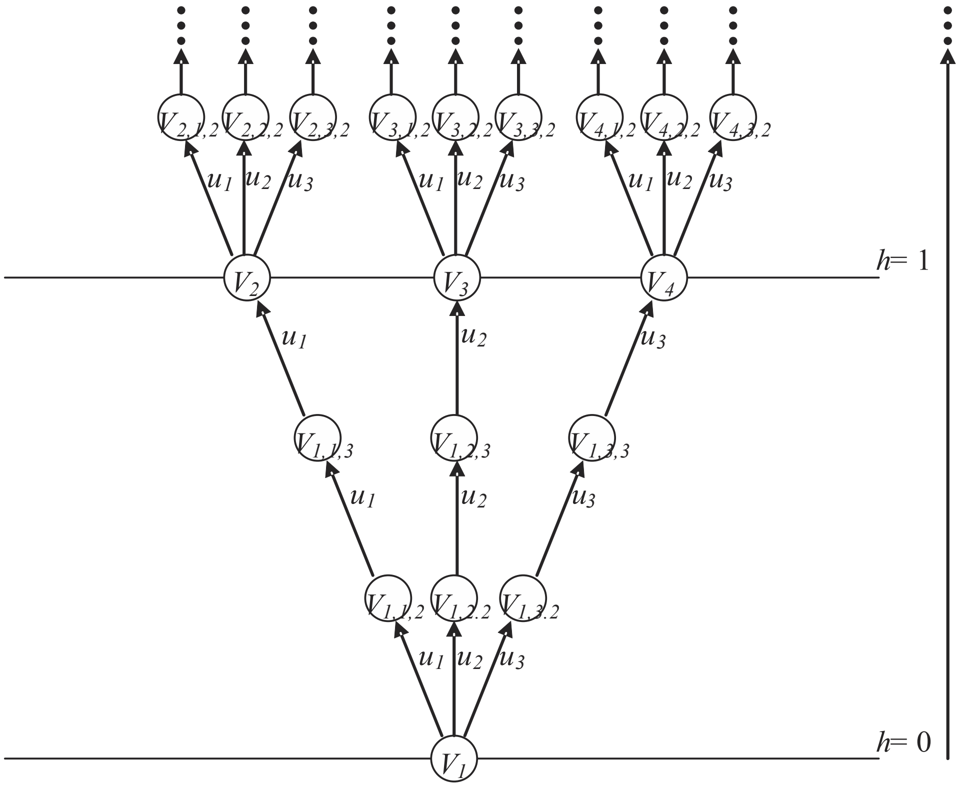




Predictive approaches to control of complex systems

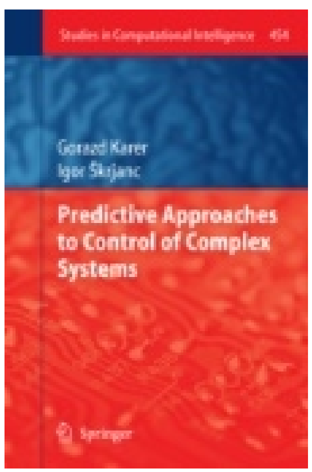
- Predictive control of complex systems
  - Solving mixed-integer optimization problems
  - Predictive control based on reachability analysis

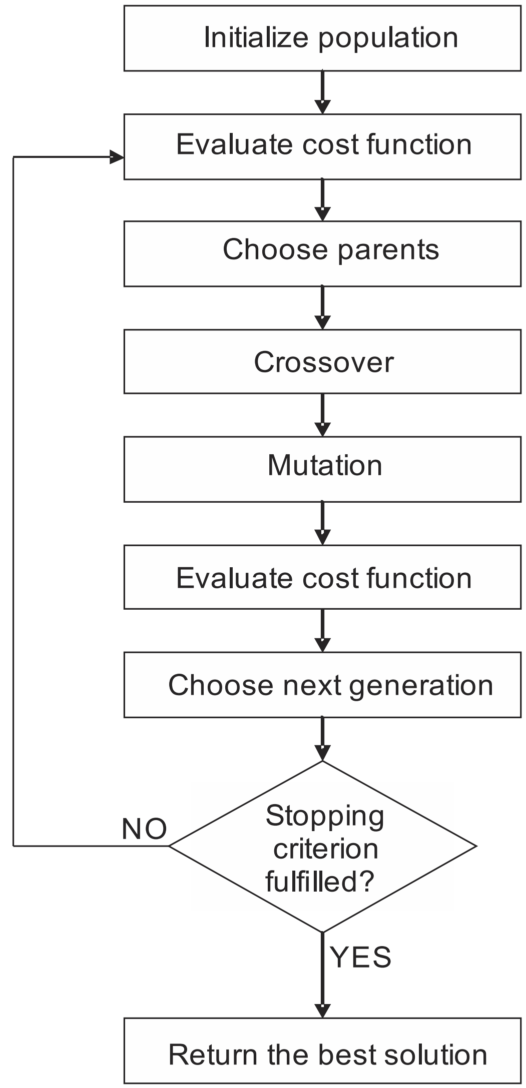




Predictive approaches to control of complex systems

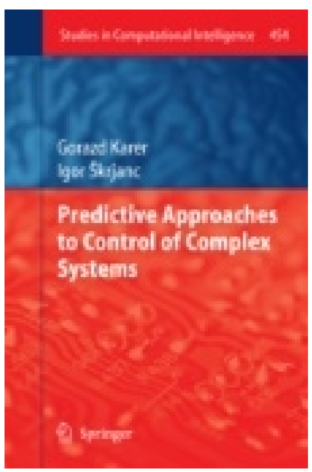
- Predictive control of complex systems
  - Solving mixed-integer optimization problems
  - Predictive control based on reachability analysis
  - Predictive control based on a genetic algorithm

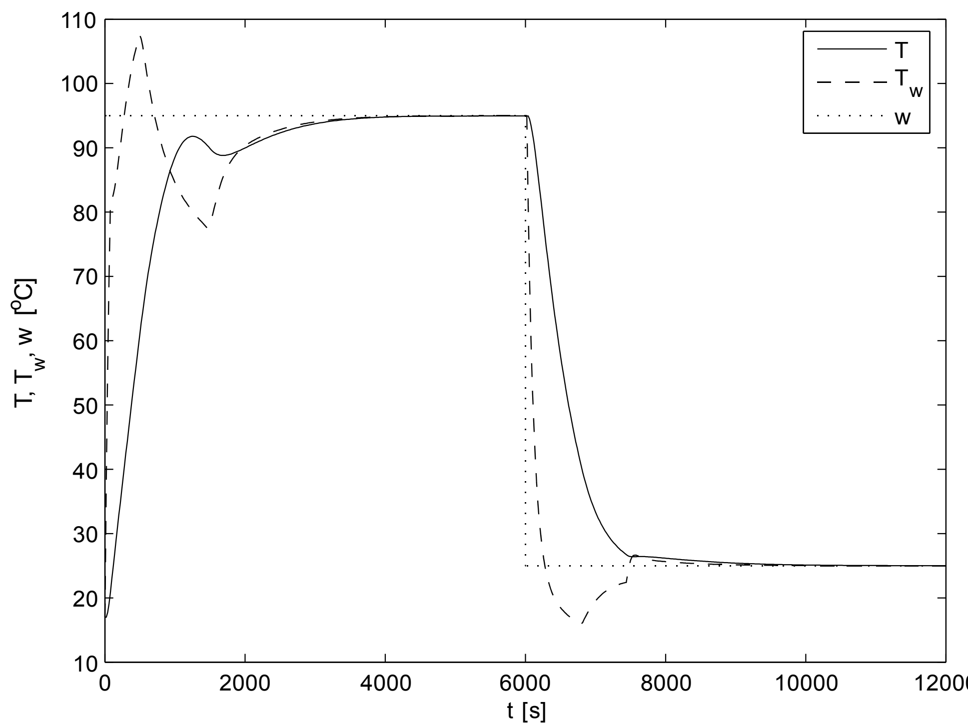




Predictive approaches to control of complex systems

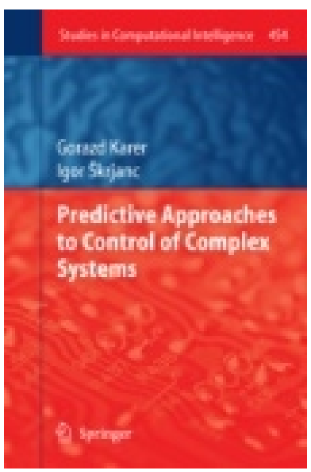
- Predictive control of complex systems
  - Solving mixed-integer optimization problems
  - Predictive control based on reachability analysis
  - Predictive control based on a genetic algorithm
  - Self-adaptive predictive control with an online local-linear-model identification

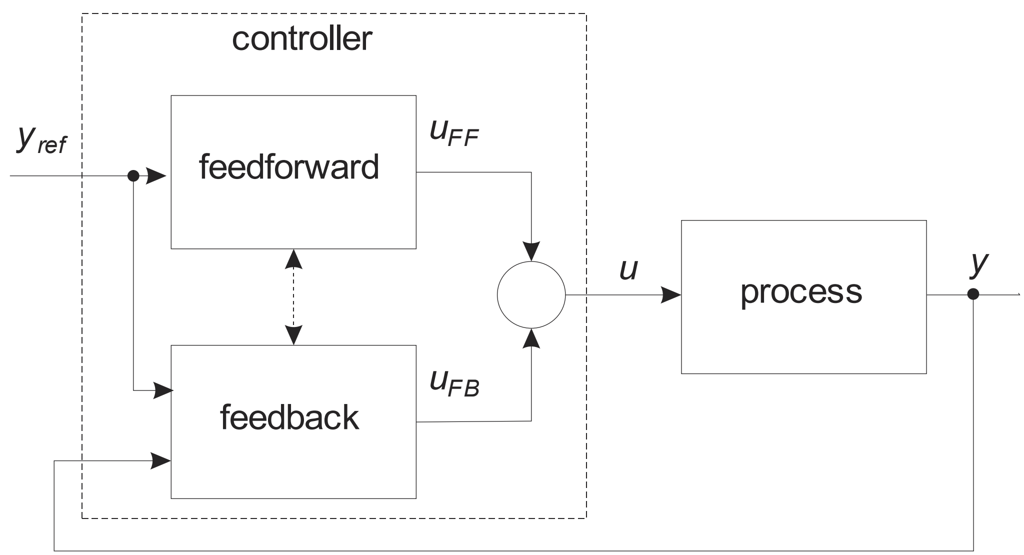




Predictive approaches to control of complex systems

- Predictive control of complex systems
  - Solving mixed-integer optimization problems
  - Predictive control based on reachability analysis
  - Predictive control based on a genetic algorithm
  - Self-adaptive predictive control with an online local-linear-model identification
  - Control using an inverse hybrid fuzzy model





Conclusion

- HFM: a convinient framework for modelling complex nonlinear hybrid systems for control purposes.

- Difficult identification of systems to be formulated as HFM.

- The presented identification method:
  - Fuzzy clustering algorithm.
  - Project the clusters from $\mathcal{D}_{IO}$ into $\mathcal{D}_I$ for MPC.

- Results have shown the usability of the algorithm – batch reactor efficiently identified and formulated as HFM.

- Further cooperation within CEEPUS: modelling, analysis, optimization, control design in technical and non-technical areas

University of Ljubljana

Faculty of Electrical Engineering

Model formulations of complex systems for predictive control design

Asst. Prof. Dr. Gorazd Karer

Sofia, October 2016