KYBERNETIKA — VOLUME 28 (1992), NUMBER 5, PAGES 413 - 423

ALGORITHMS FOR BAYESIAN ESTIMATION OF SPLINE MODEL STRUCTURE

JAN SPOUSTA

A special case of model structure identification is studied. Convolution models with the kernel described by first order spline-functions are tested. Fast algorithm for finding the most probable structure of the model is described.

1. INTRODUCTION

There are two contradictory demands in practice of discrete adaptive control of continuous dynamic systems.

- For good knowledge of system behavior, we must choose a short sampling period.
- If an adaptive regression model based regulator with a given order is used, the numerical sensibility grows up with the sampling frequency. An increasing of the order which can improve the robustness is often not possible because of limited computing time, storage size etc.

One way to solve this antagonism is to use a continuous data filtration. In this paper we use the filtration based on the spline-function approximation of the convolution kernel in the convolution model of a linear dynamic system. The motivation is to obtain a flexible tool for modelling kernels, particularly those with limited supports. An approximating spline-function can be expressed as a linear combination of given base spline-functions. The problem is to find the set of base functions, their number and some other demands (order, defect) are given. The crucial demand is that they must give "good" approximation of the (slowly changing) kernel of the system for purposes of control.

We deal with the filtration derived from a spline-approximated convolution kernel in the convolution model of linear dynamic system. The kernel (denoted by $K(t)$) is parametrized through a fixed number of basic spline functions. This parametrization can be more flexible in comparison with usually used exponentials in the case of limited support of the kernel. The supports of the spline functions are namely limited, too.

If the basic spline functions are denoted by $f_{Ki}(t)$ and some real parameters θ_i for $i = 1, 2, \ldots, m$, we write $K(t) \approx \sum_{i=1}^{m} \theta_i f_{Ki}(t)$. The parameters θ_i are then estimated (and changed) on-line and through this estimation the adaptivity of the regulator is realized.

The problem solved in the paper is to define the functions $f_{Ki}(t)$ before the adaptive regulation starts. As a basis for this choice we have some knowledge of the system behavior, that is the data $d^{(N)}$ for some N.

Our solution is based on a Bayes decision algorithm, described in [2]. In our case, we must choose one hypothesis about the basis spline-functions from a set of all a priori defined hypotheses. In more detail we must:

- define the set of all hypothesis $\{\mathcal{H}^p\}_{p=1}^{M}$ about the bases. Any hypothesis $\mathcal{H}^p$ corresponds to some basis $\mathcal{B}^p$ for each p. From the data (or from a sufficient statistic V) and from the corresponding bases (or from the filter matrices S_p defined by the bases $\mathcal{B}^p$) we shall then need to compute the probabilities of all so given hypotheses in the Bayes manner. Therefore we must
- adapt the algorithm for computing probabilities of the above hypotheses i.e. probabilities of the hypotheses about filter matrices on given data (see subsection 2.5.) and
- find the optimal sequence of the hypotheses for the computation so that the results from one step could be used in the next one (see subsection 3.1.) and find how to do it (see 3.2.).

2. PRELIMINARIES

2.1. The System Equation

In this paper, we deal with the one-dimensional linear autonomous dynamic system described by the equation

$$y(t) = \int_0^t K(\tau)\, y(t-\tau)\, \mathrm{d}\tau + \theta_0 + e(t), \tag{1}$$

where
- $y(t)$ — a signal value at time t
- $K(.)$ — a convolution kernel
- θ_0 — an absolute term
- $e(t)$ — a Gaussian, zero mean term standing for uncertainty of the system behavior.

We have measured the system output $y(t)$ in N discrete equidistant time instants. We introduce a data set $d^{(N)} = \{y(t_1), y(t_2), \ldots y(t_N)\}$ which is, we suppose, all our information about the system. Further we shall denote $y_i = y(t_i)$ for simplicity.

Our problem is to estimate the structure of the kernel K.

2.2. Spline-approximation

There are different ways for description of convolution kernels of linear dynamic systems. One of them is the description through spline functions, piecewise polynomial functions. We choose the splines with degree 1 and defect 1 as the most simple. These splines are broken lines in fact.

The points of breaking are called nodes of the spline-function and the set of all nodes $\Delta = \{v_0, v_1, \ldots, v_{m+1}\}$ for some m is called splitting of the definition interval of the spline-function. Spline-functions with the same splitting create a linear functional space. The set of m "hat" functions

$$f_{Ki}(t) = \begin{cases} (x - v_{m-i-1})/(v_{m-i} - v_{m-i-1}) & \text{for } x \in \langle v_{m-i-1},\, v_{m-i}) \\ (v_{m-i+1} - x)/(v_{m-i+1} - v_{m-i}) & \text{for } x \in (v_{m-i},\, v_{m-i+1}\rangle \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

form a functional basis in the subspace of all first-order spline-functions, which satisfy the condition to be zero in v_0 and v_{m+1}. The space (and so the basis) is then determined through the number and the positions of nodes. In this paper we require to have the number m fixed.

A kernel $K(t)$ can be approximated as a superposition of the basis functions

$$K(t) \approx \sum_{i=1}^{m} \theta_i f_{Ki}(t), \tag{3}$$

where the weights $\{\theta_i\}_{i=1}^{m}$ parametrize now the corresponding kernel.

The structure estimation means estimation of suitable nodes in Δ and thus estimation of the corresponding basis.

The task will be solved through the Bayesian algorithm, described in [4]. We must design a set of hypothesis about the structure of K, i.e., a set of functional bases composed of the above-mentioned "hat" functions. Now, our idea is to shape properly comparatively long kernel with a compratively small number of parameters and so to be able to consider larger period of data sampling.

For approximation of a signal we take the first-order-splines, too. If the sampling period is equal to one, we have also a basis for signal description:

$$f_{Yi}(t) = \begin{cases} x - n + 1 & \text{for } x \in \langle n-1,\, n) \\ n + 1 - x & \text{for } x \in (n,\, n+1\rangle \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The coefficients for the description are then simply the sampled values of the system output y_i:

$$y(t) \approx \sum_{i=1}^{N} y_i f_{Yi}(t). \tag{5}$$

2.3. Sufficient Statistics

For the computation of hypothesis probabilities we shall use the ideas proposed for a multivariate regression model in [2]. Here, a sufficient matrix statistic $V_{(t)} \in \mathsf{R}^{m \times m}$ is described which is evaluated by the regression of "shifted" data:

$$V_{(t)} = V_{(t-1)} + \bar{f}_{(t)} \bar{f}_{(t)}^{\mathrm{T}}, \quad V_{t_0-1} = \varepsilon I, \tag{6}$$

where $\varepsilon > 0$ is some small number, $t = t_0, t_0 + 1, \ldots, t_0 + k$ and the vector $\bar{f}_{(t)}$ has the structure

$$\bar{f}_{(t)} = (y_{(t)}, y_{(t-1)}, \ldots, y_{(t-l)}, 1)^{\mathrm{T}}$$

for $l = m - 2$ which corresponds to the lenght of the kernel.

The positive definite matrix $V_{(t)}$ (we shall write only V) can be decomposed into the form

$$V = L D L^{\mathrm{T}} \tag{7}$$

where L is a unique lower triangular matrix with units on the diagonal and D is a positive diagonal matrix. The computing of the probabilities (see [2]) is based on in the decomposition obtained values D_{ii}.

2.4. Definition of the Hypotheses

Let us have an equidistant splitting Δ^* of the interval $\langle 0, T\rangle$, where T means the maximum a priori known lenght of the kernel $K(.)$ — i.e. we suppose $\mathrm{supp} K \subset \langle 0, T\rangle$. The splitting Δ^* consists of n nodes:

$$\Delta^* = \{v_1^*, v_2^*, \ldots, v_n^*\},$$

where $0 < v_1^* < v_2^* < \ldots < v_n^* < T$ and moreover

$$v_1^* - 0 = v_2^* - v_1^* = \ldots = T - v_n^*$$

(equidistant splitting). Let $m < n$. Let us choose some subset of the splitting: $\delta^p \subset \Delta^*$, $\delta^p = \{v_{i_k}^*\}_{k=1}^m$. Then the set $\Delta^p = \delta^p \cup \{0, T\}$ defines a spline basis on interval $\langle 0, T\rangle$ according to (2), if we adjoin $v_0^p = 0, v_k^p = v_{i_k}^*$ for $k = 1$ to m, and $v_{m+1}^p = T$. For all possible choices of the subsets δ^p we have

$$M = \binom{n}{m}$$

bases $\mathcal{B}^p = \{f_i^p(t)\}_{i=1}^m$ for $p = 1$ to M.

A hypothesis $\mathcal{H}^p$ corresponds to the basis $\mathcal{B}^p$: the hypothesis insists that the basis $\mathcal{B}^p$ is the most probable from all the M bases, if we know the data d^N (and we have no other informations).

2.5. The Computation of Hypothesis Probability

A spline linear dynamic model can be converted to regression one by filtering the data. Let the filter matrix S is given by the convolution at time n (see [4])

$$S_{ij} = [f_{Ki} * f_{Yj}](n), \qquad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{8}$$

It follows from the substitution of both approximated the kernel (3) and the signal (5) into the system equation (1). The convolution is then reduced into a matrix multiplying, where the middle term is the matrix S.

The "spline" model keeps the properties of multivariate regression models for filtered data: $\bar{f}_{\text{spline}(t)} = S\bar{f}_{\text{regression}(t)}$. (The index "spline" means the spline model and "regression"means the original data.) It holds

$$V_{\text{spline}(t)} = S\, V_{\text{regression}(t)}\, S^{\mathrm{T}} \tag{9}$$

and with the filtered statistics, we can compute the probability in the way of the following decomposition algorithm.

The positive definite matrix $V_{\text{spline}(t)}$ (we shall write only V) can be decomposed in form

$$V = L\, D\, L^{\mathrm{T}} \tag{10}$$

where L is a unique lower triangular matrix with units on the diagonal and D is a positive diagonal matrix.

About the kernel, we have a set of hypothesis $\{\mathcal{H}^1, \mathcal{H}^2, \ldots, \mathcal{H}^l, \ldots, \mathcal{H}^M\}$. For all the hypothesis we can compute the statistics $\{V_1, V_2, \ldots, V_l, \ldots, V_M\}$ based on the observed data:

$$V_l = S_l V_{\text{regression}} S_l^{\mathrm{T}}, \tag{11}$$

where the matrix S_l is given by (8), and decomposed them:

$$V_l = L_l D_l L_l^{\mathrm{T}}, \quad D_l = \operatorname{diag}(d_1, d_2, \ldots, d_{n+2}). \tag{12}$$

Then according to [2], it can be written:

$$p(\mathcal{H}^l | k+1 \text{ measured data}) \propto \frac{1}{\sqrt{(d_{n+2})^k \prod_{i=1}^{n+1} d_i}}. \tag{13}$$

Computing all M probabilities is in real cases extremly demanding on the computer time. The following chapter says how to carry out this computations as efficient as possible.

3. MAIN RESULTS

3.1. Idea of an Algorithm for the Sequential Computation of all the Hypotheses Probabilities

Computing all the M probabilities in the way of (8), (11), (12), (13) step-by-step takes too much time. But, it is possible to choose a sequence of computed hypothesis so that a

large part of computations-results in the previous step executed probability-computation (matrix rows, columns etc.) is utilized for the next steps.

The idea is simple: we choose the sequence of the computed probabilities (and also the corresponding bases) so that the next basis differs from the previous one only in the position of a single node. This implies that in the next basis $\{f_K^{(p+1)}\}$ there are maximally 3 basis-functions $f_{K,i-1}^{(p+1)}$, $f_{K,i}^{(p+1)}$ and $f_{K,i+1}^{(p+1)}$ different from the previous base $\{f_K^{(p)}\}$. It implies that in the filter matrix S_{p+1} at maximum three rows differ from rows the matrix S_p. Moreover, the elements of the three rows can be recalculated even more efficiently.

We shall see that the re-computation of the LDL^T decomposition after the change of one node is more efficient if we change a node with small index.

Example. We show the work of the algorithm on a very simple example with $n = 6$ and $m = 3$.

	Position					
No.	1st	2nd	3rd	4th	5th	6th
1	.	.	.	x	x	x
2	.	.	x	.	x	x
3	.	x	.	.	x	x
4	x	.	.	.	x	x
5	x	.	.	x	.	x
6	.	.	x	x	.	x
7	.	x	.	x	.	x
8	.	x	x	.	.	x
9	x	.	x	.	.	x
10	x	x	.	.	.	x
11	x	x	.	.	x	.
12	x	.	.	x	x	.
13	.	.	x	x	x	.
14	.	x	.	x	x	.
15	.	x	x	.	x	.
16	x	.	x	.	x	.
17	x	x	.	.	x	.
18	x	x	.	x	.	.
19	x	.	x	x	.	.
20	.	x	x	x	.	.
21	x	.	x	x	.	.
22	x	x	.	x	.	.
23	x	x	x	.	.	.

("x" means "node", "." means "the position is free". The standard defined nodes in the 0th and 7th positions are not displayed.) In this example $M = 20$, but we need 23 steps. Bases No. 17, 21 and 22 had been already computed in previous steps. The algorithm does not compute the probability in this cases; it changes only the matrices in the storage. □

On the assumption that the first node and the last one are fixed on positions 0 and $n + 1$ respectively and that p is the vector of the nodes positions, the algorithm works in following manner:

- Put all nodes as right as possible.
- Change step-by-step the position of the second node to all its possible positions.
- **While** the nodes are not as left as possible do:
  - **If** $p(3) = 2$,
    then find the lowest left shiftable node, shift it once to left and shift all the left neighbours of this node step-by-step as right as possible;
    else shift the 3rd node once to the left, but before that "clear" the place for it, if it is not empty.
    end of if
  - Change step-by-step the position of the second node to all its posible positions.

  end of while

end of the algorithm

3.2. The Data Matrices and Their Re-calculation

There are the auxiliary algorithms described in this section.

3.2.1. Storing of the Filter Matrix

The way of writing the filter matrix into the storage is described here.

Detailed analysis shows that the filter matrix S has not more than $L = 2(m + n - 2)$ elements different from zero. (For the proof see [6].) Between two non-zero elements in every column (row) are only non-zero elements. So, the whole matrix can be stored in four data vectors:

- real-vector **values**(L) containing rowvise values of the non-zero elements of the matrix;
- integer-vector **first**(m) containing in its ith element the column index of the first non-zero element in the ith row of the matrix;
- integer-vector **last**(m) containing in its ith element the column index of the last non-zero element in the ith row of the matrix;
- integer-vector **index**(m) containing in its ith element the index of the first non-zero element in the ith matrix row in the vector **values**.

This method does the computation with the matrix more efficient and economizes the storage.

Example. Let us have a filter matrix 12 × 5.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{47} & a_{48} & a_{49} & a_{4,10} & a_{4,11} & a_{4,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{59} & a_{5,10} & a_{5,11} & a_{5,12} \end{pmatrix}$$

The data vectors are then:

values(30) = $(a_{11}, \ldots, a_{14}, a_{23}, \ldots, a_{27}, a_{35}, \ldots, a_{39}, a_{47}, \ldots, a_{4,12}, a_{59}, \ldots, a_{5,12}, 0, 0, 0, \ldots, 0)$;

first(5) = (1, 3, 5, 7, 9);

last(5) = (4, 7, 9, 12, 12);

index(5) = (1, 5, 10, 15, 21). □

3.2.2. Re-calculation of the Elements of Filter Matrix

If the position of one node is changed, maximally three rows of the filter matrix are changed: the node is an element of supports of three basic spline-functions, in maximum, and one basic function corresponds with one row. Moreover, according to the definition it is not necessary to compute all elements of the changed row. This subsection shows how to re-calculate most of elements without computing of convolutions.

An element of a filter matrix S is defined by the convolution $s = [f_K * f_Y](\tau)$. The function f_K is the 1st-order spline, linear everywhere except the nodes $n_{\text{left}} < n_{\text{center}} < n_{\text{right}}$, continuous everywhere and $f_K(n_{\text{left}}) = f_K(n_{\text{right}}) = 0$ and $f_K(n_{\text{center}}) = 1$.

- Suppose, that in the next step the node n_{center} is changed: $\bar{n}_{\text{center}} = n_{\text{center}} + \xi$, $\xi \in (n_{\text{left}} - n_{\text{center}}, n_{\text{right}} - n_{\text{center}})$. The nodes $n_{\text{left}}, \bar{n}_{\text{center}}, n_{\text{right}}$ define a new function $\bar{f}_K$ and $\bar{s} = [\bar{f}_K * f_Y](\tau)$. Then the following implications are valid:

$$\operatorname{supp} f_Y \subset (-\infty, \min\{n_{\text{center}}, \bar{n}_{\text{center}}\}) \Rightarrow \bar{s} = \frac{n_{\text{center}} - n_{\text{left}}}{\bar{n}_{\text{center}} - n_{\text{left}}} s, \tag{14}$$

$$\operatorname{supp} f_Y \subset (\max\{n_{\text{center}}, \bar{n}_{\text{center}}\} + \infty) \Rightarrow \bar{s} = \frac{n_{\text{right}} - n_{\text{center}}}{n_{\text{right}} - \bar{n}_{\text{center}}} s, \tag{15}$$

- Suppose that in the next step the node n_{right} is changed: $\bar{n}_{\text{right}} = n_{\text{right}} + \xi$, $\xi \in (n_{\text{center}} - n_{\text{right}}, +\infty)$. The nodes $n_{\text{left}}, n_{\text{center}}, \bar{n}_{\text{right}}$ define a new function $\bar{f}_K$ and $\bar{s} = [\bar{f}_K * f_Y](\tau)$. Then the following implications are valid:

$$\operatorname{supp} f_Y \subset (-\infty, n_{\text{center}}\rangle \cup \langle\max\{n_{\text{right}}, \bar{n}_{\text{right}}\}, +\infty) \Rightarrow \bar{s} = s, \tag{16}$$

$$\operatorname{supp} f_Y \subset \langle n_{\text{center}}, \min\{n_{\text{right}}, \bar{n}_{\text{right}}\}\rangle \Rightarrow \tag{17}$$

$$\Rightarrow \bar{s} = \frac{n_{\text{right}} - n_{\text{center}}}{\bar{n}_{\text{right}} - n_{\text{center}}} s + \frac{\xi}{\bar{n}_{\text{right}} - n_{\text{center}}} \int_{-\infty}^{+\infty} f_Y(t)\,\mathrm{d}t.$$

- Suppose, that in the next step the node n_{left} is changed: $\bar{n}_{\text{left}} = n_{\text{left}} + \xi$, $\xi \in (-\infty, n_{\text{center}} - n_{\text{left}})$. The nodes $\bar{n}_{\text{left}}, n_{\text{center}}, n_{\text{right}}$ define a new function $\bar{f}_K$ and $\bar{s} = [\bar{f}_K * f_Y](\tau)$. Then the following implications are valid:

$$\operatorname{supp} f_Y \subset (-\infty, \min\{n_{\text{left}}, \bar{n}_{\text{left}}\}) \cup \langle n_{\text{center}}, +\infty) \Rightarrow \bar{s} = s, \tag{18}$$

$$\operatorname{supp} f_Y \subset \langle \max\{n_{\text{left}}, \bar{n}_{\text{left}}\}, n_{\text{center}}\rangle \Rightarrow \tag{19}$$

$$\Rightarrow \bar{s} = \frac{n_{\text{center}} - n_{\text{left}}}{n_{\text{center}} - \bar{n}_{\text{left}}} s - \frac{\xi}{n_{\text{center}} - \bar{n}_{\text{left}}} \int_{-\infty}^{+\infty} f_Y(t)\,\mathrm{d}t.$$

Thus we may simplify the re-computation of the matrix.

Example. Three rows of a filter matrix are given. We change the common node of the three corresponding basic functions. Letters in the rows mean the implications: A means (14), B means (15), C means (16), D means (17), E means (18). K means "the convolution is computed per definition".

$$\begin{pmatrix} \ddots & \vdots & & & & & & & & \\ \dots & C & C & K & K & D & D & K & K & \\ & & & B & B & B & B & K & K & A \\ & & & & & & & K & K & K & E & \dots \\ & & & & & & & & & \vdots & \ddots \end{pmatrix}$$

The equation (19) is in this example not used. □

3.2.3. The Product $S\,V\,S^{\mathrm{T}}$ and Its $L\,D\,L^{\mathrm{T}}$-decomposition.

There are only three rows and three columns changed in the actualized $S\,V\,S^{\mathrm{T}}$ matrix product after one step. If the changed node is the $(m+1-i)$-th the changed rows/columns are the $i-1$, i, $i+1$.

Example. for $m = 12$, $i = 7$ ("x" means "modified", "." means "not modified", omitted means "zero".):

$$SVS^{\mathrm{T}} = \begin{pmatrix}
. & . & . & . & . & . & x & x & x & . & . & . & . \\
. & . & . & . & . & . & x & x & x & . & . & . & . \\
. & . & . & . & . & . & x & x & x & . & . & . & . \\
. & . & . & . & . & . & x & x & x & . & . & . & . \\
. & . & . & . & . & . & x & x & x & . & . & . & . \\
x & x & x & x & x & x & x & x & x & x & x & x & x \\
x & x & x & x & x & x & x & x & x & x & x & x & x \\
x & x & x & x & x & x & x & x & x & x & x & x & x \\
. & . & . & . & . & . & x & x & x & . & . & . & . \\
. & . & . & . & . & . & x & x & x & . & . & . & . \\
. & . & . & . & . & . & x & x & x & . & . & . & . \\
. & . & . & . & . & . & x & x & x & . & . & . & .
\end{pmatrix}$$

$$LD = \begin{pmatrix}
. & & & & & & & & & & & & \\
. & . & & & & & & & & & & & \\
. & . & . & & & & & & & & & & \\
. & . & . & . & & & & & & & & & \\
. & . & . & . & . & & & & & & & & \\
x & x & x & x & x & x & & & & & & & \\
x & x & x & x & x & x & x & & & & & & \\
x & x & x & x & x & x & x & x & & & & & \\
. & . & . & . & . & x & x & x & x & & & & \\
. & . & . & . & . & x & x & x & x & x & & & \\
. & . & . & . & . & x & x & x & x & x & x & & \\
. & . & . & . & . & x & x & x & x & x & x & x &
\end{pmatrix}$$

□

Even, it is not necessary to change all elements of matrices of the $L\,D\,L^{\mathrm{T}}$-decomposition are all changed. In the matter of fact, the non-changed elements are all elements in the first $i-2$ columns of the matrix L with exception of the rows $i-1$, i, $i+1$ and the $i-2$ "first" (in index) elements of the matrix D.

This implies that our algorithm for the next-step-search is correct: it changes mostly the low-index nodes.

4. CONCLUSIONS

This work links up the results from papers [2] (automatic estimation of the model order) and [4] (spline approximation in adaptive controllers). The solved problem is to estimate the most probable structure of spline description of the convolution kernel.

The above algorithm computes m-times faster compared with the computation "per definition" (m means the given number of base functions). The structure of the main algorithm is open to parallelisation and/or adaptable to consider an additional information (omitting of hypothesis known as non-probable etc.).

A remaining problems are to extend the results to a more general linear system and to restrict the big number of prior hypotheses by using some another additional prior knowledge.

(Received July 18, 1991.)

REFERENCES

[1] M. Kárný: Bayesian estimation of model order. Problems Control Inform. Theory *9* (1980), 1, 33–46.

[2] M. Kárný: Algorithms for determining the model structure of a controlled system. Kybernetika *19* (1983), 164–178.

[3] M. Kárný: Quantification of prior knowledge about global characteristic of linear normal model. Kybernetika *20* (1984), 164–178.

[4] M. Kárný, I. Nagy, J. Böhm and A. Halousková: Design of spline-based self-tuners. Kybernetika *26* (1990), 1, 17–30.

[5] V. Peterka: Bayesian approach to system identification. In: Trends and Progress in System Identification (P. Eykhoff, ed.), Pergamon Press, Oxford 1981, pp. 239–304.

[6] J. Spousta: Structure Estimation of Spline-Description of Dynamic Systems (in Czech). Ph.D. Dissertation, Institute of Information Theory and Automation, Czechoslovak Academy of Sciences, Prague 1990.

Ing. Jan Spousta, Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation – Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 182 08 Praha 8. Czechoslovakia.