SOFT COMPUTING MODELING AND REAL TIME MODEL BASED ADAPTIVE CONTROL

Tamás Péni **Gábor Rödönyi**

Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
H-1117 Budapest, Pázmány Péter sétány 1/D, Hungary
peni@ seeger.iit.bme.hu rodonyi@seeger.iit.bme.hu

Abstract: This paper deals with identification and real time model based adaptive control of a special class of nonlinear systems. The identification and controller design procedure is presented on the example of a two degree of freedom robot arm and consists of three steps. In the first step a not too precise controller is designed and the robot is derived along a special path covering the full workspace. In the next step the functions of the dynamic model are identified by using LS method and the samples collected in the previous step. In the third step the models are re-sampled and simpler and precise models are generated, which are used in adaptive control. *Copyright© 2000 IFAC*

Keywords: robot arms, fuzzy modelling, identification, adaptive control.

1. INTRODUCTION

The control of robot arms is often difficult to handle by only conventional mathematical and control theory and many times the combination of 'hard' mathematics and 'soft' human experiences seem more effective. The identification and controller design procedure introduced in this work tries to follow this approach. First (Section 2.) the fuzzy model of the robot is created by using samples which have been collected previously along a workspace-covering path. This model can be simplified (Section 3.) by applying Multivariate Adaptive Regression Splines (MARS) procedure. Section 4 shows how the obtained approximation can be used in adaptive control and Section 5 discusses the possibility of usage of this method in real-time environment.

2. DYNAMICS OF ROBOT ARM AND FORMULATION OF ITS IDENTIFICATION

The examined robot is a two degree-of-freedom arm, which has revolute joints and its segments move in a vertical plane. If the mass, length, position of center of gravity and inertia of the segments are indicated with m_1, m_2, l_1, l_2, l_{c1}, l_{c2}, I_1, I_2 then by introducing

$$D_{11}(q) = I_1 + m_1 l_{c1}^2 + I_2 + m_2 (l_1^2 + 2 l_1 l_{c2} c_2 + l_{c2}^2)$$

$$D_{12}(q) = I_2 + m_2 \left(l_{c2}^2 + l_1 l_{c2} c_2\right)$$

$$D_{22}(q) = I_2 + m_2 l_{c2}^2$$

$$D_{112}(q) = -m_2 l_1 l_{c2} s_2$$

$$D_1(q) = m_1 g l_{c1} c_1 + m_2 g (l_1 c_1 + l_{c2} c_{12})$$

$$D_2(q) = m_2 g l_{c2} c_{12}$$

$$\alpha = 2 q_1' q_2' + q_2'^2 \qquad \beta = q_1'^2 \tag{1}$$

the dynamic equation of this system can be described as follows (Lantos 1997):

$$q'' = G(q)\tau + f(q, q') =$$

$$= \begin{bmatrix} C_{11}(q) & C_{12}(q) \\ C_{12}(q) & C_{22}(q) \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \tag{2}$$

$$\begin{bmatrix} C_{112}(q) & C_{111}(q) & C_1(q) \\ -C_{111}(q) & C_{211}(q) & C_2(q) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix}$$

where τ is the torque (input, command signals), q is the joint variable (output, controlled signal) and q', q'' are its derivatives.

$$C_{11}(q) = \frac{D_{22}(q)}{D_{11}(q) D_{22}(q) - D_{12}^2(q)}$$

$$C_{12}(q) = \frac{-D_{12}(q)}{D_{11}(q) D_{22}(q) - D_{12}^2(q)}$$

$$C_{22}(q)=\frac{D_{11}(q)}{D_{11}(q)D_{22}(q)-D_{12}^2(q)}$$
$$C_{112}(q)=-C_{11}(q)D_{112}(q)$$
$$C_{111}(q)=C_{12}(q)D_{112}(q)$$
$$C_1(q)=-C_{11}(q)D_1(q)-C_{11}(q)D_2(q)$$
$$C_{211}(q)=C_{22}(q)D_{112}(q)$$
$$C_2(q)=-C_{12}(q)D_1(q)-C_{22}(q)D_2(q) \qquad (3)$$

The parameters of robot used in numerical calculations are: $m_1=5, m_2=5, l_1=1, l=1$, $l_{c1}=0.5$, $l_{c2}=0.5$, $I_1=1$, $I_2=1$ in SI units.

During the identification the system is handled as a gray box, i.e. the structure of its dynamical description is assumed to be known, except the eight $C_k, k\in\{1,2,11,12,22,112,111,211\}$ functions of which determination is the task of the identification procedure. In order to solve this problem at first a suitable path has to be designed which covers the entire workspace of the arm, along which moving the robot the collected samples contain enough information for precise identification. This path can be designed heuristically. See Fig1.

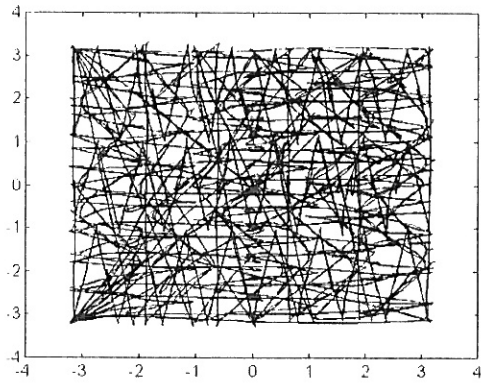


Fig. 1. Collected sample points in the **q** space

The next step is a selection of a simple control algorithm, which can move the robot along this path. This controlling may be poor, since it is used only for data collection. In this work a badly set computed torque controller is used for the operation of an inaccurate controller, i.e. the command signal was calculated as follows:

$$\tau=\hat{G}^{-1}\left(q_a''+K_P(q_a-q)+K_I\int(q_a-q)\mathrm{d}t+K_D(q_a'-q')\right)+$$
$$+\hat{G}^{-1}\hat{f} \qquad (4)$$

Along the path 42865 samples were collected. Every sample consists of four parts: torque (τ) and position (q), velocity (q') and acceleration (q'') of segments, (the last two components were calculated from q by numerical differentiation). Unfortunately these samples can not be used directly to identify C_k functions so a special model is needed, which can be tuned with the given samples and gives approximations for the individual functions (Lantos 1998a). In order to get an appropriate one at first (2) has to be rewritten in a following form:

$$q''=\begin{bmatrix} q_1'' & q_2'' & 0 & \beta & \alpha & 0 & l & 0 \\ 0 & q_1'' & q_2'' & -\alpha & 0 & \beta & 0 & l \end{bmatrix}\begin{bmatrix} C_{11}(q) \\ C_{12}(q) \\ C_{22}(q) \\ C_{111}(q) \\ C_{112}(q) \\ C_{211}(q) \\ C_1(q) \\ C_2(q) \end{bmatrix}=$$
$$=\Omega\Delta \qquad (5)$$

where Ω is known (can be calculated if q, q' and q'' are given) but Δ is unknown. Each C_k function in Δ is modeled by a Sugeno-fuzzy system. The output of the model can be described as $C_k(q)=\varphi_k^T(q)\vartheta_k$, where φ_k denotes the input dependent part of rules and ϑ indicates the parameters of linear consequences. Furthermore if the input dependent parts (φ_k) of each C_k functions are fixed and common (i.e. $\varphi_k=\varphi$, not adjustable) then the condition parts can be converted into Ω and the entire structure is rewritten as

$$q''=\Omega\varphi(q)\begin{bmatrix} \vartheta_{11}^{(1)} \\ \vdots \\ \vartheta_{11}^{(s)} \\ \vdots \\ \vartheta_{2}^{(1)} \\ \vdots \\ \vartheta_{2}^{(s)} \end{bmatrix}=\Omega\varphi\vartheta \qquad (6)$$

The inputs of this model are q, q', τ and output is q'' so the previously collected samples are suitable for its tuning. Furthermore the output depends on the adjustable parameters linearly so Least Squares (LS) method can set them. After the training, the parameters belonging to individual functions can be separated and models for C_k functions are obtained. In this work the domain of q ($[-\pi;\pi]\times[-\pi;\pi]$) was covered by a grid of size 5x5 so the condition parts (φ) consisted of 25 rules and the dimension of ϑ was 25*8*3=600. From the collected samples 5 reduced sample sets were selected, each set contained approximately 2000 samples. By using them 5 models were created of which average gave the final model. The original functions and the results of identification can be seen on Fig 2. (The order is C_{11}, C_{12}, C_{22}, C_{111} C_{112}, C_{211}, C_1, C_2). The only problem with these models is that they contain unnecessarily too many parameters. In order to

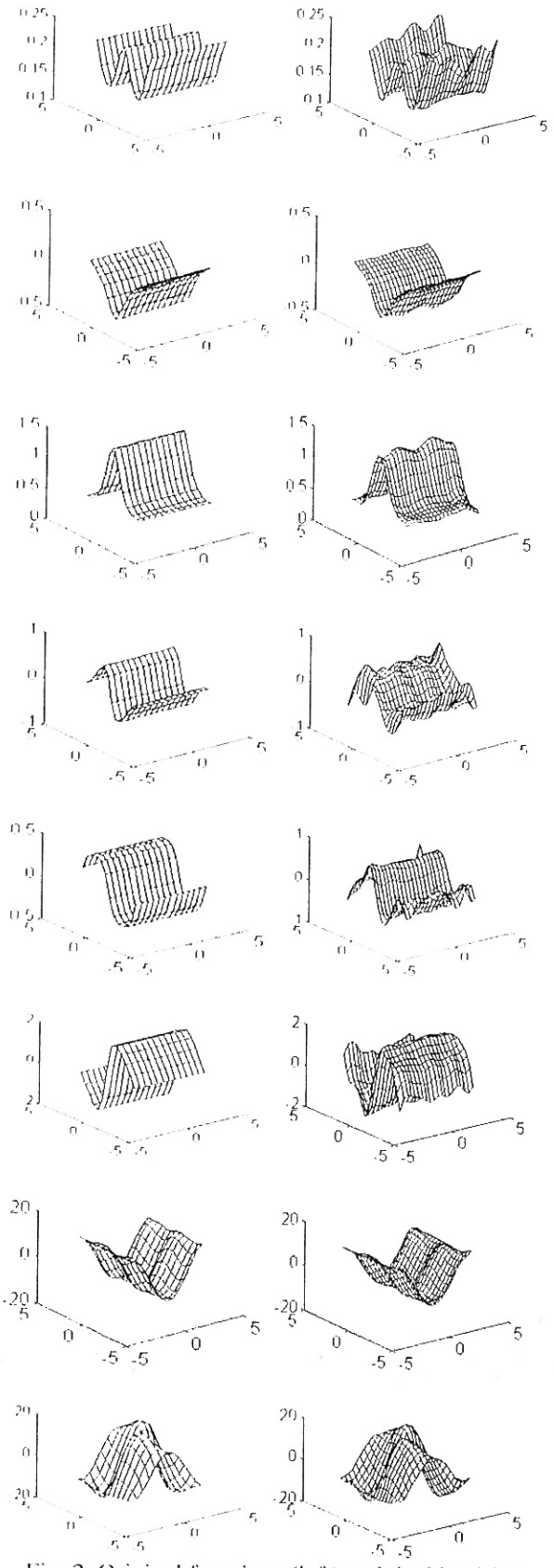


Fig. 2. Original functions (left) and the identified fuzzy models (right)

reduce the complexity the Multivariate Adaptive Regression Splines (MARS) method was used, since it is able to recognize the inputs on which the output really depends.

3. MULTIVARIATE ADAPTIVE REGRESSION SPLINES (MARS)

The MARS algorithm (Friedman 1991) is a reliable method for constructing models to approximate a given set of samples. Its main advantage is that it tries to find the ANOVA (ANalysis Of Variance) decomposition of the modeled function, i.e. the output of the created structure is a sum of local models, which do not necessarily depend on all inputs. Unfortunately the use of MARS is limited, since it applies exhaustive search so in the case of large data set it can be too slow. Formally the model generated by MARS can be written as follows:

$$\hat{f}(q) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{K_m} \left[s_k \left(x_{v(k,m)} - t_{km} \right) \right]_+ = \\ = a_0 + \sum_{m=1}^{M} a_m B_m(q) \qquad (7)$$

where

$B_m(q)$: local model

K_m : number of input variables in m-th model.

s_{km} : sign (+1 or −1)

$v(k,m)$= identifies the k-th input of model m among the system inputs.

t_{km}, a_m = parameters of the m-th model. t_{km} identifies the place of the local model (set by exhaustive search) and a_{km} is a linear, adjustable parameter, (set by LS method).

$$z_+ = \begin{cases} z & \text{if } 0 \le z \\ 0 & \text{otherwise} \end{cases}$$

If D and M_{max} indicate the training set and the limit of number of local models the MARS algorithm can be described as follows:

$B_0(q) := 1$, $M := 2$ (let the first model be constant.)

Loop until $M > M_{max}$: $lof^* := \infty$

for $m=1$ to $M-1$ do

for $v \notin \{v(k,m) \mid 1 \ldots K\}$

for $t \in \{q_{vj} \mid B_m(q_j) > 0, q \in D\}$

$$g := \sum_{i=1}^{M-1} a_i B_i(q) + a_M B_m(q)[+(q_v - t)]_+ + \\ a_{M+1} B_m(q)[-(q_v - t)]_+$$

(the place of the next model is assigned by one of the samples)

$$lof := min_{a_1 \ldots a_{M+1}} \left[\frac{1}{\#D} \sum_{i=1}^{\#D} \left| y_i - g(x_i) \right|^2 \right]$$

(the parameters of the new structure are set by minimizing a quadratic loss function)

if $lof < lof^*$ then $lof^* := lof$; $p^* := p$

$v^* := v$; $t^* := t$

end for

end for

end for

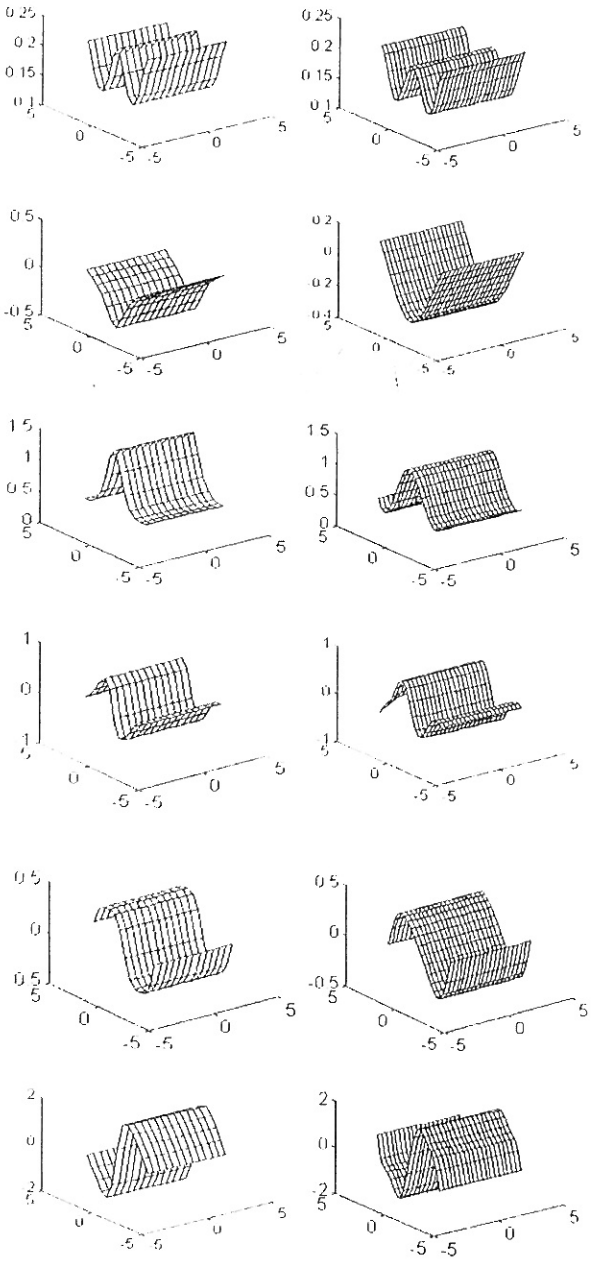
$B_M(x) := B_{m^*}(x)\left[+\left(x_{v^*} - t^*\right)\right]_+$

$B_{M+1}(x) := B_{m^*}(x)\left[-\left(x_{v^*} - t^*\right)\right]_+$

$M := M + 2$

end Loop

In our identification problem the previously generated C_k models were re-sampled by a grid of size 20x20 and these data sets were given to MARS. Fig. 3 shows the results are satisfactorily precise and the number of linear parameters (a_l) were reduced to 130.



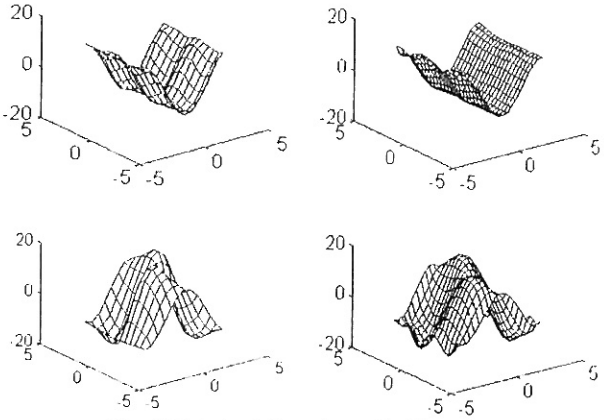


Fig.3 Original functions (left) and MARS models (right)

4. THE WANG-LANTOS SELFTUNING ADAPTIVE CONTROLLER

The following algorithm was developed by Wang (1994) for SISO systems and was generalized for the MIMO case by Lantos (1998b). Assume the system equations in the form

$$y^{(n)} = f(x) + G(x)u \tag{8}$$

where $u \in R^m$ is the control signal, $y^{(n)}$ is the n-th derivative of the output $y \in R^m$, $x = \left(y^T \quad \ldots \quad y^{(n-1)T}\right)^T \in R^{nm}$ is the state variable, $G=G^T>0$ is positive definite matrix. Note the equations of the robot arm can be expressed in this manner. If this system equation were known exactly then the control

$$u = G^{-1}\left(y_d^{(n)} - f + \tilde{K}\tilde{e}\right) \tag{9}$$

would decrease the error $\tilde{e} = \left(e^T \quad \ldots \quad e^{(n-1)T}\right)^T$ exponentially, where $e = y_d - y$, y_d is the desired output,

$$\tilde{K} = \left[K_n \quad K_{n-1} \quad \ldots \quad K_1\right] = \begin{bmatrix} k_{1n} & & \cdots & k_{11} & \\ & \ddots & & & \ddots \\ & & k_{mn} & \cdots & & k_{m1} \end{bmatrix} \tag{10}$$

and k_{ij} are to be choosen so that the roots of $s^n + k_{i1}s^{n-1} + \ldots + k_{in} = 0$ have stable and fast dynamics, since in the closed loop we get $e^{(n)} = -\tilde{K}\tilde{e}$. In fact only an approximate model of f and G is available: $\hat{f}(x)$ and $\hat{G}(x)$. Based on these models let the nominal control be

$$u_c = \hat{G}^{-1}\left(y_d^{(n)} - \hat{f} + \tilde{K}\tilde{e}\right) \tag{11}$$

which results in the closed loop error dynamics $e^{(n)} = -\tilde{K}\tilde{e} + \left(\hat{f} - f\right) + \left(\hat{G} - G\right)u_c$. The inaccuracy of $\hat{f}(x)$ and $\hat{G}(x)$ can lead to expansion of error e and unstability, therefor we need a supervisor control law that can be derived from Lypunov stability theory in the following way. Consider the dynamics of $\tilde{e}$

$\dot{\tilde{e}} = A_c \tilde{e} + B_c \left\{ \left(\hat{f} - f\right) + \left(\hat{G} - G\right) u_c \right\}$, where

$$A_c = \begin{bmatrix} 0 & I & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & I \\ -K_n & -K_{n-1} & \cdots & -K_1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \end{bmatrix}. \qquad (12)$$

Let $Q>0$ matrix and P be the solution of the equation $A_c^T P + P A_c = -Q$. The derivative of the Lyapunov function $V = \frac{1}{2} \tilde{e}^T P \tilde{e}$ is

$$\dot{V} = -\frac{1}{2} \langle Q\tilde{e}, \tilde{e} \rangle + \left\langle B_c^T P \tilde{e}, \left(\hat{f} - f\right) + \left(\hat{G} - G\right) u_c \right\rangle. \qquad (13)$$

The effect of the second term can be compensated by the supervisor control u_s, which is turned on as the Lypunov function V exceeds a limit V_e, otherwise $u_s=0$. The control signal is then $u=u_c+u_s$ and we get

$$\begin{aligned} e^{(n)} &= -\tilde{K}\tilde{e} + \left(\hat{f} - f\right) + \left(\hat{G} - G\right) u_c - G u_s \\ \dot{\tilde{e}} &= A_c \tilde{e} + B_c \left\{ \left(\hat{f} - f\right) + \left(\hat{G} - G\right) u_c - G u_s \right\} \\ \dot{V} &= -\frac{1}{2} \langle Q\tilde{e}, \tilde{e} \rangle + \left\langle B_c^T P \tilde{e}, \left(\hat{f} - f\right) + \left(\hat{G} - G\right) u_c \right\rangle - \\ &\quad - \left\langle B_c^T P \tilde{e}, G u_s \right\rangle \end{aligned} \qquad (14)$$

Since $G(x)$ is positive definite, for its λ_i eigenvalues we have $g_L \le \lambda_{min} \le \|G\| \le \lambda_{max} \le g_U$. If we know the limits g_L, g_U and $f_U \ge \left|\hat{f}(x)\right|$ then

$$u_s = \frac{g_L^{-1} B_c^T P \tilde{e} \left\{ \left|\hat{f}(x)\right| + f_U + \left|\hat{G}(x) u_c\right| + \left|g_U u_c\right| \right\}}{\left|B_c^T P \tilde{e}\right|} \qquad (15)$$

makes the derivative of Lyapunov function negative. Next we derive the parameter tuning rules. Suppose $\hat{f}(x) = \varphi^T(x)\vartheta_f$ and $\hat{G}(x)u = \Psi^T(x,u)\vartheta_G$. Note the fuzzy and also the MARS model of the robot arm can be written in this form. Let $\Gamma_1>0$ and $\Gamma_2>0$ positive definite matrices and denote ϑ_f^* and ϑ_G^* the theoretically reachable optimal model parameters. Introduce the notation

$$w = \hat{f}\left(x, \vartheta_f^*\right) - f(x) + \left[\hat{G}\left(x, \vartheta_G^*\right) - G(x)\right] u_c$$
$$\Delta\vartheta_f = \vartheta_f - \vartheta_f^*, \quad \Delta\vartheta_G = \vartheta_G - \vartheta_G^* \qquad (16)$$

and choose the Lyapunov function

$$V = \frac{1}{2} \tilde{e}^T P \tilde{e} + \frac{1}{2} \left\langle \Gamma_1^{-1} \Delta\vartheta_f, \Delta\vartheta_f \right\rangle + \frac{1}{2} \left\langle \Gamma_2^{-1} \Delta\vartheta_G, \Delta\vartheta_G \right\rangle. \qquad (17)$$

Since

$$\varphi^T(x)\Delta\vartheta_f = \hat{f}\left(x, \vartheta_f\right) - \hat{f}\left(x, \vartheta_f^*\right)$$

and

$$\Psi^T(x, u_c)\Delta\vartheta_G = \left[\hat{G}\left(x, \vartheta_G\right) - \hat{G}\left(x, \vartheta_G^*\right)\right] u_c,$$

$$\begin{aligned} \dot{V} &= -\frac{1}{2} \langle Q\tilde{e}, \tilde{e} \rangle - \left\langle B_c^T P \tilde{e}, G u_s \right\rangle + \left\langle B_c^T P \tilde{e}, w \right\rangle + \\ &\quad + \left\langle \varphi B_c^T P \tilde{e} + \Gamma_1^{-1} \Delta\dot{\vartheta}_f, \Delta\vartheta_f \right\rangle + \\ &\quad + \left\langle \Psi B_c^T P \tilde{e} + \Gamma_2^{-1} \Delta\dot{\vartheta}_G, \Delta\vartheta_G \right\rangle. \end{aligned} \qquad (18)$$

Making the last two term zero we obtain the adaptive parameter tuning rules

$$\Delta\dot{\vartheta}_f = -\Gamma_1 \varphi(x) B_c^T P \tilde{e},$$
$$\Delta\dot{\vartheta}_G = -\Gamma_2 \Psi(x, u_c) B_c^T P \tilde{e}. \qquad (19)$$

The third positive term in $\dot{V}$ represents a strict condition on the applicability of the adaptive tuning and depends on the choosen model structure.

Now we compared the fuzzy model based and MARS model based control.

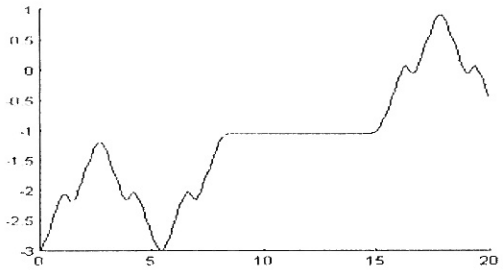


Fig 4. Reference signal for q_1 joint

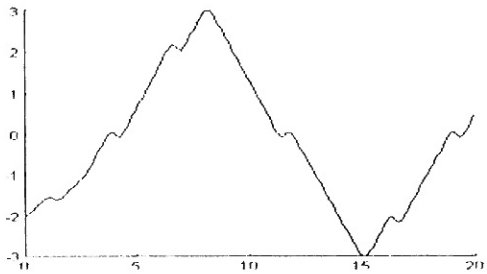


Fig 5. Reference signal for q_2 joint

The motion of robot arm is simulated with its true nonlinear model by 4-th order Runge-Kutta method on MATLAB. Fig. 4 and 5 shows the reference signals of the two joint variables q_1 and q_2 respectively. The robot arm has to follow these command signals with Ts=1ms sampling time.

The parameters of Wang-Lantos controller are $Q=125I$, $g_L=1.1$, $g_U=1.3$, $f_U=10$, $\Gamma_1=\Gamma_2=0.0007$, $V_e=1.2$, $k_{i,1} = 2/T$, $k_{i,2} = 1/T^2$, $T=0.0045$.

The simulation results with the fuzzy model are shown in Fig. 6 and 7, that shows the error of joint variables e_1 and e_2 respectively. The peak values of e_1 are of the order of 4e-5, and e_2 of 5e-5. The error signals of MARS model based control are drawn in Fig. 8 and 9. In this case the peak values of e_1 are of the order of 6e-5, and e_2 of 2e-4. The control in the fuzzy case resulted 1.5-2.5 times smaller peak values and much smaller variance, but has 600 linear parameters contrary to MARS model, wich has 130 linear parameters.

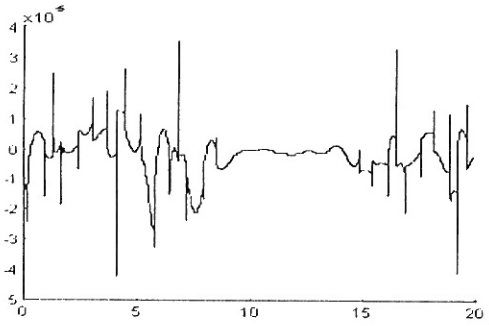


Fig 6. e_1 with fuzzy model

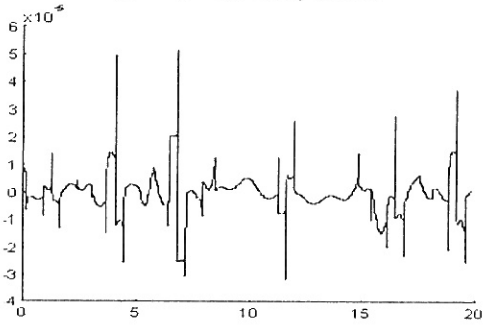


Fig 7. e_2 with fuzzy model

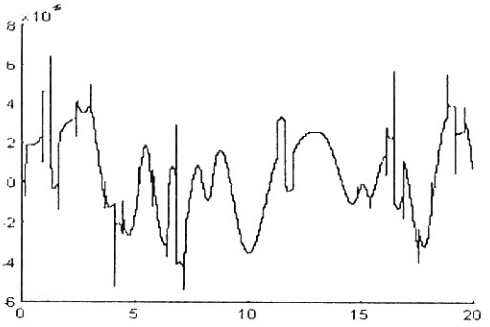


Fig 8. e_1 with MARS model

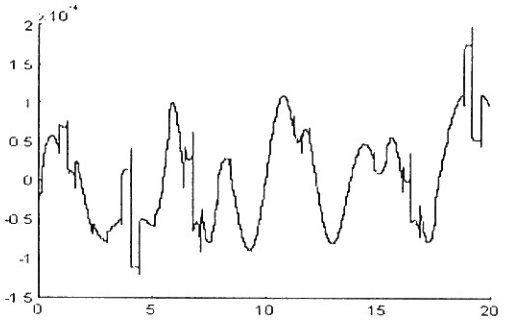


Fig 9. e_2 with MARS model

5. APPLICATION OF ADAPTIVE CONTROL IN REAL-TIME

The presented control method is applicable in real time environment if all nonlinear functions of the model are sampled over the region of interest, evaluated at the sampling points off-line and the results are put into a look-up table. Therefor the nonlinear (and time consuming) calculations are reduced to a reading from this table. A real-time environment was built under NI-LabView. This system consists of a PC compatible computer (host) and a NI-PXI7030 real-time card attached to its PCI bus. The card has 16 analogue inputs, 2 analogue outputs and contains a 486DX processor. Furthermore it guarantees the 1000 Hz sampling frequency. The control algorithm has to be written under LabView and it has to be downloaded onto the card, where it runs independently from the operating system of the host. During the control the host is used only for monitoring the working by collecting information on the control loop via dual port RAM.

6. CONCLUSIONS

It was developed and tested an identification and control strategy for a class of nonlinear systems having the form like (8) and (5). The measurable or computable output (in our robot example q^*) is a linear combination of known functions (left matrix in (5)'s right side) and unknown nonlinear functions, that have common set of arguments. First a Sugeno-fuzzy model with common rule-base was chosen for the nonlinear functions to separate them, then each of the fuzzy models were simplified by MARS method. The simplification proved to be meaningful under marginal rise of error in control. The success of this strategy depends on whether learning data contain enough information for separation of the functions in the nonlinear system model.

ACKNOWLEDGEMENT

Support for the research was provided by the Hungarian National Research Program under grant No. FKFP 0270/1999 and OTKA T 029072.

REFERENCES

Lantos B. (1997): *Robot Control*, Akadémiai Kiadó,

Lantos B. (1998a): Some Possibilities to increase the intelligence in robot control systems. Plenary paper. In: *Proceedings of INES'98, Vienna, pp.7-8.*

Lantos B. (1998b). Some Applications of Soft Computing Methods in System Modeling and Control, In: *Journal of Advanced Computational Intelligence, Vol.2 No.3, pp. 82-87.*

Friedman J. H. (1991). Multivariate Adaptive Regression Splines. In:*The Annals of Statistics 1991, Vol.19, No. 1, pp.1-141.*

Wang L. X. (1994): *Adaptive fuzzy systems and control*, Prentice Hall.