PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 43, NO. 3, PP. 177–187 (1999)

GENETIC ALGORITHM-BASED OPTIMISATION OF FUZZY LOGIC SYSTEMS FOR DYNAMIC MODELLING OF ROBOTS

Attila NEMES* and Béla LANTOS**

Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
H–1117 Budapest, Pázmány Péter sétány 1/D, Hungary
*nemesa@seeger.iit.bme.hu
**lantos@iit.bme.hu

Received: November 20, 1999

Abstract

This paper reports a novel method for the choice and reduction of the training data set for dynamic modelling of robotic manipulators (RMs) by fuzzy logic systems (FLSs) that are evolved by a genetic algorithm (GA). A multi-population, multi-objective GA is used for structure evolution and optimisation of the FLSs and constants for the precise approximation of the dynamic model (DM) and the simplicity of the FLSs and the complete DM. The initial large set of training data is considerably reduced, while not loosing any of its representative quality.

Keywords: genetic algorithms, fuzzy logic systems, robot dynamic model

1. Introduction

The dynamic modelling of the robot manipulator (RM), mapping the position, velocity and acceleration of the joints into the driving torques exerted to the structure is based on the Lagrange formulation, which ensures the appropriate structure of the dynamic model that is commonly used in control algorithms. RMs are known to be highly non-linear multi-input multi-output systems. To preserve the known structure of the Lagrange formulation Grey-Box modelling is used [3]. It means that the forces exerted to RM joints are the sum of four components modelling consequently the torque resulting from the inertia (H), the Coriolis effects and centrifugal forces (C), the gravity forces (g) and the viscous friction (f). The separate knowledge of these components is crucial for precise, model based robot control algorithms. This was the main reason to address the problem as identification (ID) of the DM as a set of constants and fuzzy logic systems (FLSs) interconnected to result in the described physical structure having unknown building blocks (a grey-box), rather than one huge FLS common to black-box modelling. Advantage is taken of other commonly known facts of robotics like H and g are non-linear functions of joint positions only. The driving torque is linear in the joint accelerations. The centrifugal and Coriolis effects are quadratic in the joint velocities and non-linear in the joint positions and f is linear in joint velocity [1]. The DM identification method uses the measured resultant torque and joint variables along suitably chosen paths for every joint.

Every single building block of the first, preliminary part of the identification was approximated with a constant by the singular value decomposition (SVD) based linear least squares (LSQ) method. At the second step representative portions of the training data are extracted, on bases of input space coverage in the joint position space. The number of samples is reduced to a minimal value that is still representative which is monitored through the singular values of the sample matrix transformed for the SVD based linear LSQ method of parameter identification. These quantitatively reduced, but qualitatively representative portions are concatenated to give the new training data set [5]. At the third step the possibility of improving the model is investigated in form of FLS building blocks for the non-linear functions of the joint positions. Multi-input single-output complete first order Takagi–Sugeno–Kong (TSK) type FLSs are considered having Gaussian membership functions (MFs), since they are known to be universal function approximators [2]. In order to avoid the exponential explosion of the rules defining the FLSs (Rules) the introduction of extra parameters is used for every single MF and Rule. The proposal is to handle extra parameters as a switch on or a switch off command for MFs and as a weighting coefficient for the Rules. Using this method all the important input and output space domains can be covered, while having the minimal number of active MFs and Rules assuring the required precision [3].

The usual formulation of linguistic rules or the use of neuro-fuzzy like FLS parameter estimations are cut off by the fact that they cannot manage the physical structure consisting of H, C, g and f since the separate outputs of these functions are unknown. Only the resulting torque is available for measurement. A suitable search technique to find a possible solution for this problem is GA. Minimisation of the cost function consisting of the combination of multiple criteria is performed. The objective function is formulated to minimise the identification error on all joints as well as the underlying FLS structures – the number of inputs, membership functions and rules. The cost function ensures that if the identification error is high its minimisation has the priority over structure minimisation [3].

2. Dynamic Model of a Robot Manipulator

The application of Lagrange dynamic equations for a RM in the joint space formulates the resultant torque τ_i acting on the i^{th} joint from all the p joints of the RM as a function of joint positions, velocities and accelerations:

$$\sum_j (D_{ij}(\overline{q}) \cdot \ddot{q}_j) + \sum_j \sum_k (\dot{q}_j \cdot D_{ijk}(\overline{q}) \cdot \dot{q}_k) + D_i(\overline{q}) + f_i \cdot \dot{q}_i = \tau_i, \quad i = 1, 2, \ldots, p, \tag{1}$$

where $q_i, \dot{q}_i, \ddot{q}_i$ stand for the joint variables and their derivatives. Let:

$$H_{ik} = D_{ik}(\overline{q}), \quad C_{ik} = \sum_{j=1}^{p} \dot{q}_j \cdot D_{ijk}(\overline{q}), \quad g_i = D_i(\overline{q}), \quad f_i = \text{const}, \tag{2}$$

where D_{ik}, D_{ijk}, D_i are in general highly non-linear scalar functions of the joint positions. They may contain the $\sin(\cdot)$ and $\cos(\cdot)$ functions of the joint positions and/or of their products and sums. Thus there are $(p \times p) + (p \times p \times p) + p$ possible, highly non-linear functions, and p constants when modelling a RM of p joints. Considerable reductions in the number of unknown functions are possible

$$D_{ik} = D_{ki}, \quad D_{ijk} = -D_{kji}, \quad \forall i,\ k \geq j \quad \text{and} \quad D_{iji} = 0, \quad \forall i \geq j, \tag{3}$$

but still, the remaining number is significant for a general case of four or more joints [1], [8].

It should be noted that the measurement of a single component from (2) is not possible, the only information of the output of the system is the resultant torque $\underline{\tau}$ in (1). The identification of non-linear functions under these terms is a considerable problem. Fuzzy logic systems can be used for this purpose.

3. Fuzzy Logic Systems and Genetic Algorithms

TSK type FLSs, having n inputs and 1 output can be described as: $f(\overline{x}) = \sum_{l=1}^{M} \omega_l(\overline{x}) \cdot y_l(\overline{x}) \big/ \sum_{l=1}^{M} \omega_l(\overline{x})$, where M is the number of Rules, $\overline{x}$ is the vector of n input variables, y_l is a scalar function of n input variables, y is most commonly a linear function of inputs, thus having $(n+1)$ parameters. The premise part of the Rule is: $\omega_l(\overline{x}) = \prod_{i=1}^{n} \mu_{F_{l(i)}}(x_i)$, where $\mu_{F_{l(i)}}(x_i)$ is the membership function (MF) of the i^{th} input variable in the l^{th} Rule. Commonly used MF type is: $\mu_{F_{l(i)}}(x_i) = \exp(-(x_i - \tilde{x}_{l(i)})^2 \big/ \sigma_{l(i)}^2)$, which is the Gaussian one that has two parameters: $\tilde{x}_{l(i)}$ is the centre and $\sigma_{l(i)}$ is the spread (width) of the MF. The linguistic form of the l^{th} Rule in a first order TSK FLS is: IF (x_1 is $F_{l(1)}$) AND (x_2 is $F_{l(2)}$) AND (x_n is $F_{l(n)}$) THEN $y_l = \sum_{j=1}^{n} c_{l(j)} \cdot x_j + c_{l(0)}$. For each Rule a weight w_l can be assigned in the interval [0, 1], in order to represent its importance relative to other rules. The input spaces may be considered separately and covered by a certain number, let's say k_l of MFs, where $i = 1, 2, \ldots, n$. Thus there can be exactly $M = \prod_{j=1}^{n} k_j$ Rules for covering the complete output space. More complex systems with more than a few inputs ($n > 2$), and/or with non-trivial input-output relations are difficult to manage. The tuning of the MF $(\tilde{x}_i, \sigma_i)$ and Rule (c_j) parameters is in both cases very difficult, since there are many of them and some are in non-linear context. A significant difficulty arises when the precise weights of every Rule are to be set.

Evolution-based algorithms have proved their usefulness on many multimodal optimisation problems and these partially non-linear parameter estimations can be managed by GAs [3]. A GA is constructed on bases of imitating natural biological processes and natural Darwinian evolution [6]. Two parameters code one of every k_j MF on each of n inputs. Every single Rule of the possible M needs $(n+1)$ parameters. There is an extra parameter for each Rule, which is the weight. Since most of the Rules should have the maximal weight of 1 or should be turned off (by the weight 0), weights should be coded in wider interval than $[0, 1]$. The proposed interval is $[-1, 2]$. All weights from range $[-1, 0]$ should be transformed to 0, and those from $[1, 2]$ should be set to 1. This technique will ensure that the probability of having weights 0, 1 and intermediate values is 1/3. Another extra parameter is needed to implement the turning on and off for every single MF. The interval of $[-1, 1]$ will be used interpreting positive values as switch on. These intervals have been used as proposed in [3] since this choice proved to be very successful for a significant reduction of the complexity of the required FLSs without the loss of their required quality.

Using such extra parameters, as described above, the reduction of the number of MFs and Rules can be achieved as well. Rules referring to non-existent MFs are discarded, along with Rules weighted with 0. If all MFs on one input channel are turned off it means that this particular input is of no relevance to the function being approximated. These considerations result in $N = \sum_{j=1}^{n}(k_j \cdot 3) + \left(\prod_{j=1}^{n} k_j\right) \cdot (n+2)$ parameters for every single FLS.

Let us remind that we counted $O(p^3 + p^2)$ non-linear functions of p inputs, where p is the number of joints of the robot manipulator (usually $p \geq 4$). This means that $n = p$ and the total number of parameters to code every function in (2), using all the available simplification relations, is comparable to $N(p^3 + p^2)$, plus the p parameters for coding f_i. Fundamental schemata theory suggests that small alphabets are good, because they maximise the number of schema available for genetic processing, so binary coding is implemented [4]. In order to avoid Hamming cliffs Gray code is used. For minimising the disruptive effect of mutation an adequately not too low probability has been chosen since the upkeeping of diversity in later generations is required. As chromosomes are simply the concatenated bit strings of all the parameters with fixed position for every gene, high probability (0.8) simple two points crossover will ensure low disruptiveness and high rate of inheritance during the reproductive phase. Stochastic universal sampling having minimal spread and zero bias is used for selection with a rather low selection pressure. Continuous exploration of the search space is achieved along with consistent convergence by the combination of genetic operators in this manner.

The objective of the optimisation is the precision of the dynamic model. Another aspect is the size of the involved FLSs. The aim is to use the smallest number of inputs to the individual FLSs and to have a really small Rule base, since their complete (only theoretically parallel) evaluation is to be frequent. The fact of dealing with significant difficulties should be pointed out. Grey-box modelling was introduced [3] to take into consideration all the extra information about the structure of the system to be identified, unfortunately grey-box modelling does not simplify the identification in our case. The FLSs evolved by this method are approximating the functions in (2), while we have information only of the resultant torque τ_i in (1), that is the superposition of a large number of unknown functions, which can not be measured separately. This obviously complicates the training of the FLSs to a high degree.

4. Training Data and Preliminary Identification

The RM should be excited along an appropriately chosen path, with the help of a well-tuned simple PID controller. During a certain period of work under PID control the information of the excerted torque on every joint is available as the output of the PID controller. The input of the controller is the position and speed of a joint, more precisely the errors to the desired values. The torque on every joint, the position and velocity of the joints can thus easily be measured. The acceleration of the joint can be computed by fitting a polynomial to the suitably windowed position-velocity pairs of data. The path should be persistently exciting and cover the complete typical workspace [5]. Using sampling times of the millisecond order often results in couple of ten thousands samples.

The first part of the off-line identification of the dynamic model is the approximation of all the non-linear functions and parameters described in (2) as simple constants. One suitable hard-computing technique for the linear parameter estimation problem is the linear variant of the Least Squares method (LS) [7]. The *Eq.* (2) is linear in the functions and the frictions that we are looking for. Thus the simple linear LS equation can be used as:

$$\overline{\overline{A}} \cdot \overline{x} = \overline{b}, \tag{4}$$

where in our case $\overline{x}$ is the vector of all the required functions that are considered to be constants in this first approximation and the friction parameters:

$$\overline{x} = [D_{11}, \dots D_{pp}, D_{111}, \dots D_{ppp}, D_1, \dots D_p,\ f_1, \dots\ f_p]^T. \tag{5}$$

The vector $\overline{b} = [\overline{b}_1^T \mid \overline{b}_2^T \mid \dots \mid \overline{b}_K^T]^T$ is a hyper vector of vectors $\overline{b}_l$ that contain the torque for every joint for the l^{th} time sample, so the hyper vector $\overline{b}$ contains all the torque for all joints and all K time samples. Matrix $\overline{\overline{A}}$ is a hyper matrix consisting of the blocks $\overline{\overline{A}} = [A^l_{ij}]$, where $\overline{\overline{A}}_l$ belongs to the data in the time instant t_l satisfying:

$$\sum_j A^l_{ij} \cdot x_j = b_{l,i} = \tau_i(t_l), \quad l = 1, 2, \dots, K. \tag{6}$$

The solution of the linear parameter estimation is now:

$$\overline{x} = \overline{\overline{A}}^{+} \cdot \overline{b}, \tag{7}$$

where the matrix $\overline{\overline{A}}^{+}$ is a pseudo inverse of matrix $\overline{\overline{A}}$. One should always validate the results of the standard linear LS ID (7) by evaluating the condition number associated with the problem:

$$\text{cond}\ (\overline{A}) = \|\overline{\overline{A}}^{+}\| \cdot \|\overline{\overline{A}}\| = \sigma_1/\sigma_n, \tag{8}$$

where σ_1 is the maximal singular value of matrix $\overline{\overline{A}}$, σ_n is the minimal one and $\|\overline{\overline{A}}\|$ is a matrix norm of $\overline{\overline{A}}$. A very high condition number of matrix $\overline{\overline{A}}$ from (4) clearly identifies a badly conditioned linear equation, not worth solving like (7) since it would result in a distorted solution. The Singular Value Decomposition (SVD) based variant of the LS method can be applied for mathematically more robust parameter estimation:

$$\overline{x} = \overline{\overline{V}} \cdot \overline{\overline{S}}^{+} \cdot \overline{\overline{U}}^{T} \cdot \overline{b}, \tag{9}$$

where matrices $\overline{\overline{V}}$, $\overline{\overline{S}}$, $\overline{\overline{U}}$ are the result of the singular value decomposition of matrix $\overline{\overline{A}} = \overline{\overline{U}} \cdot \overline{\overline{S}} \cdot \overline{\overline{V}}^{T}$. The appearing zero or close to zero singular values point out the samples used for the ID do not provide enough information for the precise evaluation of the required parameters [7].

The conclusion is clear: one should pump up the $\overline{\overline{A}}$ matrix with more and more $\overline{\overline{A}}_l$ samples until all the (close to) zero singular values disappear, which is equivalent to having a small condition number on matrix $\overline{\overline{A}}$. In real applications the highest singular value is limited, since it is in correlation with physical parameters of a real system. On the other hand while having a small condition number one could freely disregard all those $\overline{\overline{A}}_l$ samples whose absence does not raise the problems condition number significantly. The minimal set of representative samples can be obtained by following the next simple algorithm:

Step1 Evaluate cond $(\overline{\overline{A}}_l)$ for every block of the $\overline{\overline{A}}$ matrix, see (6).

Step2 Sort the blocks by the condition numbers in a non-descending order. This step results in vector J of indices j_i, for which cond $(\overline{\overline{A}}_{j_i}) \leq$ cond $(\overline{\overline{A}}_{j_{i+1}})$ for all $i = 1, \ldots, K-1$, where K is the number of samples.

Step3 Initiate $M = \{j_1\}$ and ${}_{\min}\overline{\overline{A}} = [\overline{\overline{A}}_{j_1}]$.

Step4 While σ_n (the minimal singular value of ${}_{\min}\overline{\overline{A}}$) is higher than S_n times σ_1 (the maximal singular value of ${}_{\min}\overline{\overline{A}}$), do add a new index j_i from $J\backslash M$ to the set of M, where $\overline{\overline{A}}_{j_i}$ minimises the condition number of hyper matrix $[{}_{\min}\overline{\overline{A}} \mid \overline{\overline{A}}_{j_i}]$.

After the iterative Step4 ${}_{\min}\overline{\overline{A}}$ is the minimal required set of samples for the quality solution of (7). The quality of the solution can be improved by decreasing the value of S_n [5]. Further improvements of the quality of the representative samples can be introduced if the stopping criterion of the iterative Step4 is replaced. The improved stopping criterion is the occurrence of no significant change in both the maximal and minimal singular value of ${}_{\min}\overline{\overline{A}}$ in the last iteration.

5. Identification Procedure by Genetic Algorithm

The second part of the off-line identification of the DM involves a GA described previously to evolve a set of FLSs that are capable of forming *Eq.* (1). The objective value of the GA consists of two parts as in (10). The first part of the objective value regarding the precision can be adopted in a quadratic form of the error of the identified and measured torque as in (11). The second part of the objective is the structural complexity of the FLSs. For the different parts certain priorities can be introduced. Obviously the error of the identification is of higher importance.

Simple PID controllers are capable of regulating the robotic manipulator from one point to another point in the joint space along a suitably defined path. For high precision control model-based controllers can be suggested [1]. One such algorithm is the so-called Inverse Dynamics Control or Computed Torque method, which requires the knowledge of all of the functions in (2). This method can be made robust and thus can cope with small model misalignments and disturbances. The relative error of the model of less than 1% is usually satisfactory for robust model-based control algorithms, under this error level the structural complexity should have the preference in the process of optimisation. It should be underlined that both error and structural minimisation do occur at every stage of the GA process. The objectives are first normalised, then scaled in such a manner that the significant part of the objective value shifts from the error part to the structural part, after certain precision is reached:

$$\begin{aligned} ObjVal_i = \sum_{j=1}^{p} Err_j + (&NofMFs_i / \max NofMFs_i \\ &+ NofRules_i / \max NofRules_i), \end{aligned} \tag{10}$$

where the index 'i' refers to the i^{th} individual of the population, Err_j is the ID error part as in (11), $\max NofMFs_i$ and $\max NofRules_i$ are the maximal possible number of MFs and Rules, and $NofMFs_i$ and $NofRules_i$ are the actual number of MFs and Rules forming the i^{th} individual. Measure of torque error on all joints is evaluated as

$$Err_j = 500 \cdot \left(K \cdot \max_k(\tau_j(k)) \right)^{-1} \cdot \|\overline{e}_j\|_2^2, \tag{11}$$

where j is the index of the joint, K is the number of the signal samples, $\overline{\tau}_j$ is the torque sequence to be identified on the j^{th} joint, and $\overline{e}_j$ is the torque identification error vector sequence on the j^{th} joint. The multiplication by 500 in (11) assures that the second part of the objective value (10) is suppressed until the first part drops under the desired magnitude.

It is possible for certain RMs that an appropriate choice of the Denavit–Hartenberg matrices leads to some constant D_{ijk}s. FLSs can easily approximate constant functions, but the number of required FLS parameters considered as structural complexity and the necessary computational effort will be much higher than that of a simple constant, whose complexity is minimal. The used method assures that all unnecessary FLSs are discarded and are replaced by the appropriate constant value [3].

6. Implementation

For the implementation tool the Matlab environment was used because of the simplicity of programming and the presence of its many useful toolboxes. The SCARA type RM was chosen to be the test platform of this new general method. The initial path was chosen to be as general and representative as possible. The sampling time is 0.005 [s] and along the initial path there are $K > 55000$ sampling points to collect the training data.

The value of $S_n = 4$ in the iterative Step4 of the training data reduction algorithm is chosen. The stopping criteria for the iterative Step4 was extended with checking the modification of the smallest and the largest singular value of$_{\min}\overline{\overline{A}}$, for every iteration. The iteration was stopped when the condition number (8) dropped below S_n and the singular values did not change for more than the value of 0.0005. The Moore–Penrose pseudo inverse was used in (7).

The SCARA RM configuration is fairly simple after all the possible reductions and simplifications described in (3). For the modelling of the DM there are 15 unknown components, and 4 of them are constants without any doubt. The parameters of the DM, shortly referred to as $\overline{x}$ in (5), are listed in the first column of *Table 1*. The characteristics of the used GA and the evolved FLSs are the same as in [3]. For testing purposes a complex random path was generated.

7. Results

The value of $S_n = 4$ and the required precision of 0.0005 for the singular values in the iterative Step4 of the training data reduction algorithm results in a quite good solution that requires only 54 sampling points. The reduction of more than 55000 samples to 54 is really significant.

The results of the preliminary identification of constant parameters are in the second column of *Table 1*.

Table 1.

	Const.
D_{11}	5.0241
D_{12}	0.84815
D_{14}	0.004001
D_{22}	1.1445
D_{24}	0.004039
D_{33}	130.2521
D_{44}	0.409
D_{112}	0.023372
D_{114}	−0.002136
D_{214}	−0.004013
D_{3}	67.1985
f_1	14.5031
f_2	13.7999
f_3	3948.9
f_4	13.4002

Table 2.

	D_{11}	D_{12}	D_{112}
# MFs for q_1	0	0	0
# MFs for q_2	2	2	1
# MFs for q_3	0	0	0
# MFs for q_4	1	1	1

Because of the relative simplicity of the SCARA type RMs only three D_{ijk}s were recognised as necessary to be FLSs, namely: D_{11}, D_{12} and D_{112}, while the others should remain constant. This matches exactly the results of classical, hard-computing robotics analyses [8].

The rough approximation of the DM was built by the constants from the second column of *Table 1*, which are the LS values of the minimal training data set. To improve the model FLSs are added to the constants resulting in the three final D_{ijk}s, which are described in *Table 2* (number of MFs for variables) and *Table 3* (coefficients of the right side of TSK rules). The error of the torque ID on both joints is such that the maximal error of the model is less than 0.5%.

Table 3.

	D_{11}	D_{12}	D_{112}
MF_1	0.72582	2.2645	0.0036922
	-0.108	-1.4503	1.3798
MF_2	1.5936	3.0183	
	1.5899	-0.3475	
MF_3	0.60622	0.79941	2.3046
	-0.23437	2.242	-1.5499
Rule_1	4.6717	1.9401	4.2849
	-3.7732	1.4625	-3.5749
	1.2926	-3.2653	-7.7998
	1.0	1.0	0.30392
Rule_2	-4.2234	-3.6023	
	-7.2138	-1.694	
	1.401	5.7801	
	1.0	0.61765	

8. Conclusions

This paper proposes a method for significant reduction of the samples for the dynamic modelling of robotic manipulators (RMs) without loosing on the quality of the representative training data.

The torque error of the DM built from constant parameters and very simple fuzzy logic systems that are evolved by a genetic algorithm using only 54 samples is less than half a percent ($< 0.5\%$). A multi-population, multi-objective GA is used for structure evolution and optimisation of the FLSs and constants, for model fitting and structural simplicity improving the results in [3] and [5]. The fact that the number of samples in the representative set of the training data is reduced from over 55000 to 54, while the quality of the identification by grey-box modelling remained superb, shows the validity of this method for reducing the set of training data.

Considering future research the necessity of the investigation of more effective FLS coding for GA is obvious, the use of computer linguistic is an interesting possibility. The application of more up to date multi-objective optimisation techniques (Pareto-dominance based ranking for example) should be incorporated as well.

Acknowledgement

Support for the research was provided by the Hungarian National Research Programs under grant No. OTKA T 029072 and FKFP 0417/1997.

References

[1] The ZODIAC, *Theory of Robot Control*, Springer, 1996, ISBN 3540760547.

[2] LI-XIN WANG: *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*, Prentice Hall, 1994, ISBN 0-13-099631-9.

[3] NEMES, A. – LANTOS, B.: Optimization of Fuzzy Logic Systems for Gray-Box Dynamic Modeling of Robot Manipulators by Genetic Algorithms, *Proceedings of IEEE INES'99*. Poprad, 1999.

[4] JANG, J.-S. R. – SUN, C.-T. – MIZUTANI, E.: *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997, ISBN 0-13-287467.

[5] NEMES, A.: Training Data Reduction for Optimisation of Fuzzy Logic Systems for Dynamic Modelling of Robot Manipulators by Genetic Algorithms. Submitted for *IEEE SYROCO'2000*, Vienna.

[6] GOLDBERG, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989, ISBN 0-201-15767-5.

[7] PRESS, W. H. et. al: *Numerical Recipes in C, The Art of Scientific Computing*, Cambridge University Press, 1990, ISBN 052135465X.

[8] LANTOS, B.: New Principles and Methods in the Control and Identification of Robots. D.Sc. Thesis, Hungarian Academy of Sciences, 1993 (in Hungarian).