FUZZY ESTIMATION OF THE ROBOT LOAD

Andreja Rojko, Karel Jezernik

University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia

Abstract: This paper considers the problem of the robot motion control in a presence of the major uncertainties such as the varying load. The proposed adaptive fuzzy system (FLS) is employed to estimate dynamic influence of the varying load and also the major part of the mechanism dynamics. Carefully designed membership functions and rules preserve the interpretability of FLS as each rule works toward estimation of the specific part of dynamics. The fuzzy estimator is implemented in the control scheme similar to the computed torque control. The effectiveness of the approach is demonstrated through the application on the three degree of freedom direct drive robot.

Keywords: motion control, adaptive control, robot dynamics, fuzzy system.

1. INTRODUCTION

The unknown varying load is one of the major dynamic uncertainties in the robotic systems. Its influence on the accuracy of the robot motion control is much greater than the influence of the other parameter uncertainties and unmodelled dynamics and therefore presents a significant problem in the design of the robot motion control. Additionally all other dynamic influences with exception of load changes can be quite successfully identified in advance or online by using a number of conventional identification methods (Ljung, 1987, Armstrong 1988), improved conventional identification methods (Grotjahn, et al., 2001) or a dynamic model described by Euler-Lagrange equations. Some recently proposed methods are also successful in identification of the friction with Stribeck effect (Kim, et al., 2004).

A problem of varying payload can be solved by using a robust or an adaptive control (Narendra and Annaswamy, 1989) or also by using the disturbance estimators, as for example the first order PI estimator (Jezernik, et al., 1994) or the second order PI estimator (Curk and Jezernik, 2001). However those estimators give only the information about the torque needed to compensate changes, but give no information about the specific dynamic parameters as the system identification does.

Recently a soft computing based systems are used a lot in the motion control approaches. Most of those systems include some kind of the parameter adaptation, therefore they can also solve the problem of load variation. Those methods are fuzzy logic systems, artificial neural networks (NN) or combinations of different soft computing approaches (Wong, et al,. 1999), (Ha et al., 2001). An adaptive FLS is often a preferable choice because it can be constructed by using both linguistic and numerical information. It is also proven, that FLS are universal aproximators, so they are capable to approximate any real, continuous function on a compact set (Wang L. X., 1994). Accordingly carefully designed adaptive FLS can be implemented as a 'white box' nonlinear online identification method for the robot dynamics. However using the adaptive FLS for solving this problem can be inappropriate in some cases because the FLS algorithm is often very computational demanding and consequently unsuitable for the real time applications. Therefore in the literature only a few real time application of adaptive fuzzy logic systems in the robot motion control can be found (Wu and Liu, 1996), (Chen and Chen, 1998), (Ha et al., 2001), (Boukezzoula et al., 2004).

In this paper we propose an adaptive FLS for compensation of the robot dynamics in the decentralized control scheme with a structure similar to the conventional computed torque motion control scheme. Rules that are implemented in FLS have different number of the inputs and therefore each group of the rules estimates one part of the robot dynamics. For example some rules estimate the inertia effects, other rules the gravitation. The control algorithm does not present a high computational burden. Its effectiveness is demonstrated by the application on the three degree of freedom direct drive robot. Emphasis is on the cases of motion control where the mechanism load is changing.

This paper is organized as follows. Section II first defines the problem of the robot motion control. Then the proper control law is derived. Design of the FLS and the adaptation law is described in the Section III. An optimization of FLS (transparency, computational requirements) is achieved trough the introduction of the subsystems. In Section IV the application results are shown. Conclusions and the future work intentions are drawn in the Section V.

2. CONTROL PLANT AND MOTION CONTROL SCHEME

The dynamic model of a rigid direct drive robot mechanism, with m degrees of freedom can be described as

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_t(\boldsymbol{q},\dot{\boldsymbol{q}}), \qquad (1)$$

where $\boldsymbol{q} = [q_1, q_2, .., q_m]^T \in \Re^m$ is the vector of the positions of the robot joints, $\dot{\boldsymbol{q}} \in \Re^m$ is the velocity vector and $\ddot{\boldsymbol{q}} \in \Re^m$ is the acceleration vector. $\boldsymbol{J}(\boldsymbol{q}) \in \Re^{mxm}$ is the symmetric, positive definite matrix of the inertias of the robot mechanism and actuator's rotors, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \Re^m$ is the vector of the Coriollis and centrifugal torques, $\boldsymbol{G}(\boldsymbol{q}) \in \Re^m$ is the vector of the gravitation torques, $\boldsymbol{\tau}_t(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the vector of the friction torques. $\boldsymbol{\tau} \in \Re^m$ is the vector of the joint drive torques.

The problem of the motion control for a robot is equal to finding the control torques $\boldsymbol{\tau}$ in (1) so that the equilibrium point $\boldsymbol{e} = 0$ defined as $\boldsymbol{e} = \left[\boldsymbol{e}_q^T(t), \dot{\boldsymbol{e}}_q^T(t)\right]^T \in \Re^{2m}$ is globally asymptotically stable. The error vector components are defined with

$$\begin{aligned} \boldsymbol{e}_q(t) &= \left[q_1^d(t) - q_1(t), .., q_m^d(t) - q_m(t)\right]^T \\ \dot{\boldsymbol{e}}_q(t) &= \left[\dot{q}_1^d(t) - \dot{q}_1(t), .., \dot{q}_m^d(t) - \dot{q}_m(t)\right]^T . \end{aligned} \qquad (2)$$

The reference trajectory is smooth function prescribed with the position $\boldsymbol{q}^d(t) \in \Re^m$, velocity $\dot{\boldsymbol{q}}^d(t) \in \Re^m$ and acceleration $\ddot{\boldsymbol{q}}^d(t) \in \Re^m$.

Many nonlinear control algorithms were developed to solve this problem, from a sliding mode control, an adaptive control to the soft computing techniques and others. One possibility is also a conventional computer torque control (Schilling, 1990). The control torques in this scheme are calculated as

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}}^c + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_t(\boldsymbol{q},\dot{\boldsymbol{q}}) \qquad (3)$$

where $\ddot{\boldsymbol{q}}^c$ is a calculated acceleration

$$\ddot{\boldsymbol{q}}^c = K_p \boldsymbol{e}_q + K_v \dot{\boldsymbol{e}}_q + \ddot{\boldsymbol{q}}^d . \qquad (4)$$

Here K_p and K_v are mxm diagonal matrixes of the velocity and position gains. For control (3) the complete dynamic model of the robot is necessary. Any discrepancy between the real and implemented parameters and the structure of dynamic model causes the position error, while using a very good estimation decouples and linearizes the system (1). This control method in practice often gives a poor result. As the goal of this work is to develop a FLS capable of estimating the mechanism dynamics which would be also applicable to other control schemes, we chose this simple control approach as the starting point for deriving our control algorithm. Next we carry out some necessary changes.

To facilitate the derivation of the decentralized control scheme, we rewrite (1) for the k-th robot joint, $k=1..m$, as

$$\begin{aligned} \tau_k = \bar{J}_{kk}\ddot{q}_k + \Delta J_{kk}(\boldsymbol{q})\ddot{q}_k + \sum_{j=1, j\neq k}^{m} J_{kj}(\boldsymbol{q})\ddot{q}_j + \\ + \sum_{j=1}^{m}\sum_{l=1}^{m} C_{jl,k}(\boldsymbol{q})\dot{q}_j\dot{q}_l + G_k(\boldsymbol{q}) + \tau_{t,k}(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{aligned} \qquad (5)$$

where $\bar{J}_{kk}$ is the constant inertia part and $\Delta J_{kk}(\boldsymbol{q})$ is the variable part of the joint inertia, and $J_{kj}(\boldsymbol{q})$ are the coupling inertias. Let us denote the whole dynamics of the robot joints (5), with exception of the constant part of the inertias as w_k

$$\begin{aligned} w_k = \Delta J_{kk}(\boldsymbol{q})\ddot{q}_k + \sum_{j=1, j\neq k}^{m} J_{kj}(\boldsymbol{q})\ddot{q}_j + \sum_{j=1}^{m}\sum_{l=1}^{m} C_{jl,k}(\boldsymbol{q})\dot{q}_j\dot{q}_l \\ + G_k(\boldsymbol{q}) + \tau_{t,k}(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{aligned} \qquad (6)$$

Considering (6) in (5) the model of the k-th robot's joint with the joint drive torque as an inputs is

$$\tau_k = \bar{J}_{kk}\ddot{q}_k + w_k . \qquad (7)$$

Applying the computed torque control approach (3), (4) for the system (7) gives following control law

$$\tau_k = \bar{J}_{kk}\ddot{q}_k^c + \hat{w}_k , \qquad (8)$$

where $\hat{w}_k$ is by the FLS estimated part of the mechanism dynamics w_k (6). The error dynamics is then equal to

$$\ddot{e}_{q,k} + K_{v,k}\dot{e}_{q,k} + K_{p,k}e_{q,k} = \bar{J}_{kk}^{-1}(\hat{w}_k - w_k). \qquad (9)$$

From (9) it can be seen that for the system linearization and therefore a good tracing accuracy, it is necessary that following is fullfield

$$\lim_t (\hat{w}_k - w_k) = 0 \cdot \qquad (10)$$

In this case the system dynamics depends only from the chosen position and velocity gains. The development of the FLS estimator to fulfill (10) is a subject of the next section.

3. DESIGN OF FUZZY ESTIMATOR

First let us consider the limitations that we have in the design of FLS. Our approach is decentralized control, so one FLS needs to be designed for the each robot joint. The inputs in each FLS can be only information regarding that joint status. Available information are measured position and velocity and desired trajectory.

As the task of FLS is to estimate the joint dynamics (6) a suitable vector of the inputs of FLS seems to be $\boldsymbol{x_k} = [q_k, \dot{q}_k, \ddot{q}_k^d]$, where $\ddot{q}_k^d$ is the desired acceleration used instead of the unknown actual acceleration. Fuzzy rule base on the k-th robot joint then consist of IF-THEN rules R_k^l with the following general form:

IF $q_k = X_k^{q,l}$ AND $\dot{q}_k = X_k^{\dot{q},l}$ AND $\ddot{q}_k^d = X_k^{\ddot{q}^d,l}$ THEN $\hat{w}_k = \bar{y}_k^l$ (11)

Superscript l refers to the l-th rule $l=1..M$. $X_k^{q,l}$, $X_k^{\dot{q},l}$, $X_k^{\ddot{q}^d,l}$ are input fuzzy sets, $\hat{w}_k$ are output linguistic variables and $\bar{y}_k^l$ are the positions of output singleton fuzzy sets.

We applied the following structure of the FLS: singleton output membership functions, singleton fuzzifier, product-operation rule of fuzzy implication and center of average deffuzifier. Bell shaped function form was chosen for input membership functions (MF). The output of the resulting FLS can be calculated as (Wang, 1994):

$$\hat{w}_k = \frac{\sum_{l=1}^{M} \bar{y}_k^l \prod_{i=1}^{n} \mu_{R_k^{i,l}}\left(x_k^i\right)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{R_k^{i,l}}\left(x_k^i\right)} \tag{12}$$

$$\mu_{R_k^{i,l}}\left(x_k^i\right) = \left(1 \Big/ \left(1 + \left| \frac{x_k^i - \bar{x}_k^{i,l}}{\sigma_k^{i,l}} \right|^{2 \cdot b_k^{i,l}} \right) \right), \tag{13}$$

where $\bar{x}_k^{i,l}$ are the centers of the input MFs, $\bar{y}_k^l$ are positions of the output MFs, $\sigma_k^{i,l}$ determine the width of the bell function and $b_k^{i,l}$ its slope. x_k^i refers to the i-th element of the vector of input variables.

After choosing the structure of the FLS also the parameters have to be determined. Parameters of the input MFs were designed to cover all possible values of the inputs. For example the MFs for position are distributed over the area that is limited by the joint end switches. MFs for velocity and acceleration are concentrated over the interval of working values but also defined outside this interval. The problem appears when we try to set the positions of the singleton output MFs $\bar{y}_k^l$ s. In the case when only one rule is fulfilled, they present the reference for the motor torque. As the complete dynamic model is not known and the additional disturbance of changing load can appear, it is obvious that these parameters must be adaptive. In order to derive adaptation law, we introduce a parameter vector $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} = \left[\bar{y}_k^1, .., \bar{y}_k^M\right]^T$, which includes all adaptive parameters and a vector of the known nonlinear functions $\xi_k(\boldsymbol{x_k}) = \left[\xi_k^1(\boldsymbol{x_k}), .., \xi_k^l(\boldsymbol{x_k}), .., \xi_k^M(\boldsymbol{x_k})\right]^T$, which are defined

$$\xi_k^l = \left(\prod_{i=1}^{n} \mu_{R_k^{i,l}}\left(x_k^i\right) \right) \Big/ \left(\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{R_k^{i,l}}\left(x_k^i\right) \right). \tag{14}$$

By using (14), the FLS (12) can be written in the parameter vector – regressor form (15), known from the classical theory of the system identification (Ljung, 1987).

$$\hat{w}_k = \sum_{l=1}^{M} \bar{y}_k^l \cdot \varphi_{\boldsymbol{k}}^l(\boldsymbol{x_k}) = \hat{\boldsymbol{\theta}}_{\boldsymbol{k}}^{\boldsymbol{T}} \cdot \xi_k(\boldsymbol{x_k}) \tag{15}$$

We also rewrite the joint error dynamics (9) as:

$$\ddot{e}_{q,k} + K_{v,k}\dot{e}_{q,k} + K_{p,k}e_{q,k} = \bar{J}_{kk}^{-1}\left(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}^{\boldsymbol{T}} - \boldsymbol{\theta}_{\boldsymbol{k}}^{\boldsymbol{T}}\right)\xi_{\boldsymbol{k}}(\boldsymbol{x_k}) = \bar{J}_{kk}^{-1}\tilde{\boldsymbol{\theta}}^{\boldsymbol{T}}\xi_k(\boldsymbol{x_k}), \tag{16}$$

where $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ is the parameter vector error. By introducing a new vector $\boldsymbol{v}=[0,1]^T$ and matrix $\boldsymbol{\Lambda}=\begin{bmatrix} 0 & 1 \\ -K_{pk} & -K_{v,k} \end{bmatrix}$ in (16) we obtain

$$\dot{\boldsymbol{e}} = \boldsymbol{\Lambda} \cdot \boldsymbol{e} + \bar{J}_{kk}^{-1} \cdot \boldsymbol{v} \cdot \tilde{\boldsymbol{\theta}}^{\boldsymbol{T}} \cdot \boldsymbol{\xi}_k(\boldsymbol{x}_k). \tag{17}$$

This error equation falls into the class of '*Error model 2'*, as classified by (Narendra and Annaswamy, 1989). Suggested adaptive law that guarantees the global asymptotic stability is (18)

$$\dot{\hat{\boldsymbol{\theta}}} = -\alpha f_e \boldsymbol{\xi}(\boldsymbol{x}) \tag{18}$$

under the condition

$$a_1 \cdot K_{p,k} > 0, \; a_2 > a_1 / K_{v,k}, \tag{19}$$

where a new term for linear combination of position and velocity error is introduced

$$fe_k = a_{1,k} \cdot e_k + a_{2,k} \cdot \dot{e}_k. \tag{20}$$

$a_{1,k}$ and $a_{2,k}$ are positive parameters and α is a learning rate. The exact stability proof of the adaptation is given by (Rojko and Jezernik, 2004).

FLS structure optimization: Because the developed algorithm must be calculated on-line, the computational complexity must be considered next.

The size of the fuzzy model is an exponential function of number of its inputs. This problem is in literature often referred to as the curse of dimensionality. Beside high computational requirements this also causes the loss of the transparency, that is interpretability of the knowledge comprehended in the FLS. For example, if we chose for all rules form (11) then we can not say which dynamic effect is compensated by each rule and the rule base is not interpretable. But if we use for example in one set of rules for inputs only acceleration and position, then we know that this rules will compensate inertia torque, as this is only dynamic effect that depends only of those two signals (1). So our approach for reducing the complexity and improving interpretability was dividing FLS into the three fuzzy logic subsystems (FLSB), each for the approximation of one part of the actual joint dynamics. All three FLSBs have the structure that we already described and the same learning algorithm, but different number of inputs.

First FLSB inputs are position and desired acceleration, $\boldsymbol{x}_{k,1.FLSB} = \left[q_k, \ddot{q}_k^d\right]$. Therefore its task is estimation of the varying part of the torque caused by inertia, which is a product of the position dependent inertia and acceleration.

$$\hat{w}_{1.FLSB,k} = \left(J_{kk}\left(q_k\right) - \bar{J}_{kk}\right) \cdot \ddot{q}_k^d \tag{21}$$

Second FLSB inputs are the position and velocity, $\boldsymbol{x}_{k,2.FLSB} = \left[q_k, \dot{q}_k\right]$ and it is designed for the estimation of the Coriollis, centrifugal forces and velocity dependent friction:

$$\hat{w}_{2.FLSB,k} = C_k\left(q_k, \dot{q}_k,\right) + \tau_{t,k}\left(q, \dot{q}_k\right). \tag{22}$$

The rules from the first two subsystems, where for the second input MF zero is considered, are working toward compensation of the gravitation effects.

Third FLSB inputs are the actual position, velocity and desired acceleration, $\boldsymbol{x}_{k,3.FLSB} = \left[q_k, \dot{q}_k, \ddot{q}_k^d\right]$ and it is used to compensate the rest of the dynamics.

Control scheme with main control algorithm (8), fuzzy dynamic estimation (12), (13) with presented subsystems (FLSB) and the implemented adaptation law (18) is shown in the Fig.2.

4. APPLICATION

The algorithm was implemented on a three-degree of freedom Puma like configuration direct drive robot, Fig.1. (Jezernik, *et. al*. 1997). Robot is equipped with AC motors with resolvers and has a transputer based multiprocesor controller. The sampling time of the controller is 2 ms. Joint velocities are not measured, but calculated by simple differentiation from the measured position. The controller parameters are mostly chosen by trial and error method and satisfy the stability condition (19). Position gains were set to $Kp_{,1,2,3}$=[1000, 2400, 1200], velocity gains to $Kv_{1,2,3}$=[64, 98, 70] and parameters of average inertia matrix to J=diag([3.5, 2.5, 0.13]) kgm^2.

FLSs implemented in the decentralized control of each robot joint have the same form and use the rules from the Table 1. Membership functions of the input variables are shown on Fig.3. Adaptation parameters were set to $a_{k=1}$=[44,0.7] for the first robot joint, to $a_{k=2}$=[50,1] for the second and to $a_{k=3}$=[7,0.2] for the third robot joint and for all joints $\alpha_{k=1,2,3}$=1. Note that small values of the learning factors $a_{k=1,2}$, $a_{k=2,\ 2}$ and $a_{k=3,2}$ were applied. Those parameters are multiplied by the velocity errors and then used in the adaptation (18), (20). This can be problem as the information about the joint velocities and therefore also velocities errors are quite noisy and can destabilize the adaptation. By using small values of the parameters we minimize this possibility. Additionally we also stop adaptation in cases when the position error is very small as for example at idle. This approach is known as a dead zone technique (Narendra and Annaswamy,1989) and guarantees the stability of adaptation also when the system is not persistently excited. Our dead zone area is limited with the value of linear combination of the position and velocity error $\left|fe_k\right| < 10^{-5}$ that we defined in (20).

Experiment 1: With first experiment we test the control stability and tracking accuracy for the slower movement with the variable load. The reference trajectory was point-to-point movement with the end position 0.8 rad, maximal velocity 0.04 rad/s and maximal acceleration 0.5 rad/s^2, the same for all three robot joints, Fig. 4. The load of 5 kg has been attached and released three times. Fig. 5. shows the robot tips position error, with shaded areas for loaded robot and clear areas for unloaded. At the end of shaded area the load is released and again applied at the beginning of next shaded area. At those moments (the moments of load change) the position error has peaks, but it returns to the normal value after short transition time. Notice that the amplitude of the position error by the load change increases with time. This is because the robot arm is lifting, so the influence of the load gravitation is also increasing. Total applied controller torques for the second and the third robot joint are depicted in Fig. 6. High torque values for the second joint are because this joint is not balanced and is together with third joint and its motor quite heavy. From this test it can be concluded that a good tracking accuracy and stability of the proposed scheme is preserved also in the presence of the major uncertainties, in our case the variable load. For a comparison, when a non adaptive control method (as computed torque control) is applied for the same test, then each time the robot is loaded the position error increases and stays high until the load is released.

Experiment 2: In this experiment we study the variation of the adaptive parameters, to find out if each group of the rules works toward estimating one specific part of the dynamics, as it was planned in the design of the FLS. First we performed the fast movement. Then when the robot was idle at the joints end position of 0.8 rad, we included the variable load of 5kg. The only dynamic effect that must be compensated at idle is gravitation due to the joints, motors and the changing load weights.

Fig. 7. and Fig. 8. show the adaptation of the parameters of the FLS for the 2. robot. From those two pictures can be seen that the most variable parameters belong to the rules R^2 and R^5, which are both depicted with short line-dot type of line. The small variation is also noticeable for parameters of R^7 and R^{12}. Fulfillment for both R^2 and R^5 depends from the fulfillment of the MF for position zero, while the inputs for velocity and acceleration are either not used or the MF zero is considered. The fulfillment of the rule R^7 depends from the MF for position positive and the MFs for velocity and acceleration zero. The only dynamic effect that depends only from robot position is gravitation, so this three rules should clearly cover most of the load change effect. A minor changes also appear in the some other rule's parameters. The reason is that the MFs cover a wide interval of the input values.

Based on this experiment, it can be concluded, that the proposed FLS rule base enables distinction which dynamic effect is compensated by each group of the rules at least for gravitation. Therefore regardless of the complicated nonlinear problem, the proposed FLS still reflects the physical background of the problem and enables the inclusion of the linguistic knowledge for the initial parameter setting.

5. CONCLUSION

In this paper we proposed an adaptive fuzzy logic system for estimation of robot dynamics, including varying payload. Unlike other adaptive FLS for identification of robot dynamics the proposed FLS preserve the transparency and therefore enables the use of linguistic knowledge for initial setting of adaptive parameters. Implemented fuzzy subsystems estimate specific part of dynamics and also optimize the number of needed rules and its inputs. This makes the algorithm less computational demanding and suitable for real-time implementation. For the test we employ the FLS instead of Lagrange dynamic model in the decentralized motion control scheme similar to the computed torque approach.
The presented application results show the features of the proposed algorithm. The direct drive robot has been used as the test object, as it have no gears that can reduce the influence of the load changes and dynamics on the motors and therefore present an extra challenge for the motion control design.

	RULE	POSITION	VELOCITY	ACCELERATION
1.	R^1	negative	-	negative
subsystem	R^2	zero	-	zero
	R^3	positive	-	positive
2.	R^4	negative	negative	-
subsystem	R^5	zero	zero	-
	R^6	positive	positive	-
3.	R^7	positive	zero	zero
subsystem	R^8	negative	zero	zero
	R^9	zero	negative	positive
	R^{10}	zero	zero	zero
	R^{11}	zero	zero	positive
	R^{12}	zero	negative	negative
Not	R^{13}	positive	zero	positive
used	R^{14}	Negative	negative	negative
on 1. axis	R^{15}	positive	positive	positive

Table 1 Rule base

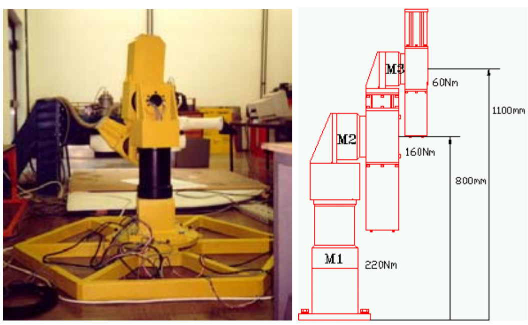

Fig. 1. Direct drive robot

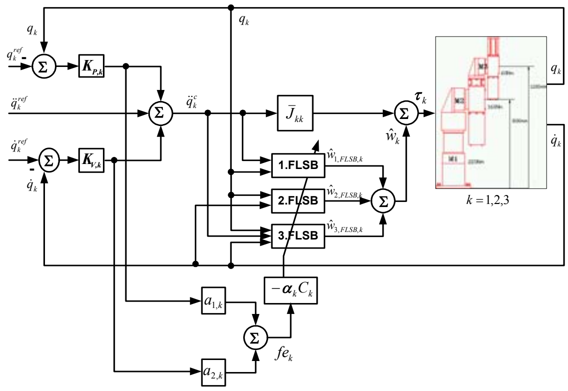

Fig. 2. Motion control scheme with FLS estimation of dynamics

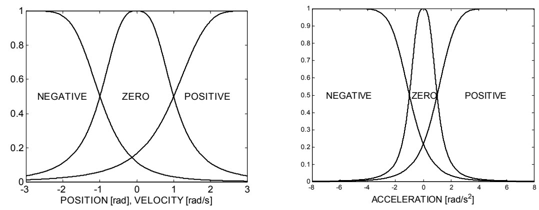

Fig. 3. Membership functions of input variables position, velocity and acceleration of FLS

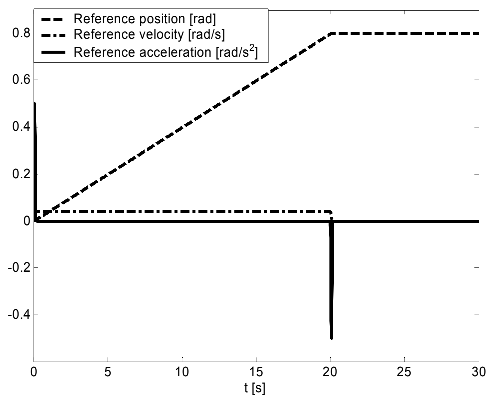

Fig. 4. Reference position, velocity and acceleration, point to point movement

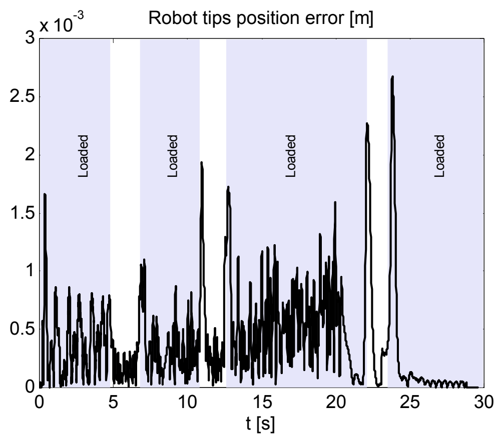

Fig. 5. *Experiment 1*; Robot tip's position error: test of varying payload between robot motion

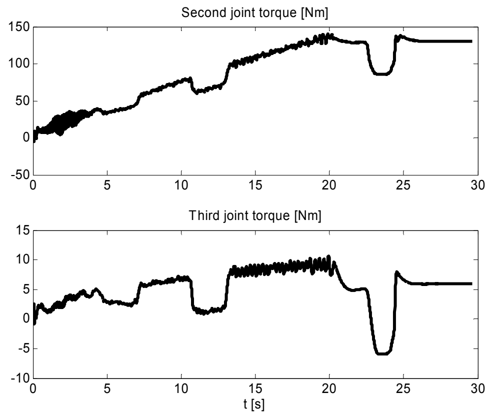

Fig. 6. *Experiment 1*; Applied robot joint torques for second and third robot joint

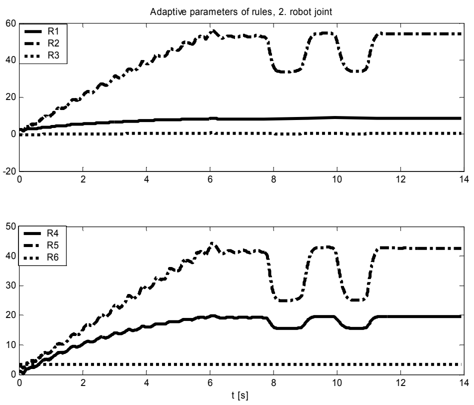

Fig. 7. *Experiment 2*; Some adaptive parameters for 2. robot joint, 2. subsystem
R^1, R^4 depicted with full line
R^2, R^5 depicted with short line-dot
R^3, R^6 depicted with dotted line

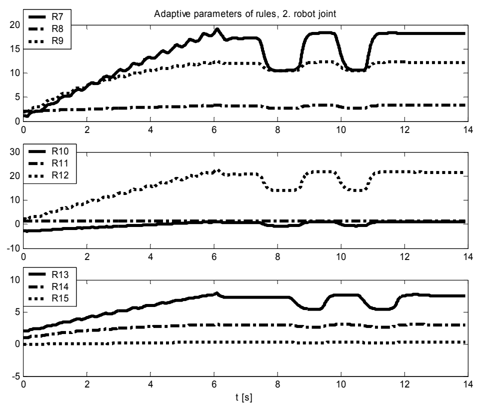

Fig. 8. *Experiment 2;* Some adaptive parameters for 2. robot joint, 3. subsystem
R^7, R^{10}, R^{13} depicted with full line
R^8, R^{11}, R^{14} depicted with short line-dot
R^9, R^{12}, R^{15} depicted with dotted line

REFERENCES

Armstrong B.(1988), Dynamics for robot control: friction modeling and ensuring excitation during parameter identification, Stanford University, Staford, CA.

Boukezzoula R., Galichet S., Foulloy L., (2004), Observer-Based Fuzzy Adaptive control for a Class of Nonlinear Systems: real Time Implementation for a Robot Wrist, *IEEE Trans. On Control Systems Technology*, **Vol 12**. No 3, pp. 340-351.

Chen C. S., Chen W. L., (1998), Robust Adaptive Sliding-Mode Control Using Fuzzy Modeling for an Inverted-Pendulum System, *IEEE Trans. On Industrial Electronics*, **Vol. 45**, No.2, pp. 297-306.

Curk B., Jezernik K., (2001), Sliding mode control with perturbation estimation: application on DD robot mechanism, *Robotica*, **Vol. 19**, pp. 641-648.

Grotjahn M., Daemi M., Heimann B. (2001), Friction and rigid body identification of robot dynamics, International Journal of Solids and Structures 38, pp. 1889-1902.

Ha Q. P., Nguyen Q. H., D.C. Rye, Durrant-White H. F., (2001), Fuzzy Sliding Mode Controller with Applications, *IEEE Trans. On Industrial Electronics*, **Vol. 48**, No. 1.

Jezernik K., Curk B., Harnik J. (1994), Observer Based Sliding Mode Control of Robotic Manipulator, *Robotica*, **Vol. 12**, pp.443-448.

Jezernik K., Terbuc M., Rojko A., Uran S. (1997), Mathematical model of Direct drive robot, http://www.ro.feri.uni-mb.si/projekti/4_dd_drive.html

Kim S.J., Kim S.Y., Ha I.J., (2004), An Efficient Identification Method for Friction in Single-DOF Motion Control System, *IEEE Trans. On Control Systems Technology*, **Vol 12**. No 4, pp. 555-563.

Ljung L. (1987), System identification: theory for the Users, prentice Hall, New Jersey.

Narendra K.S., Annaswamy A.M.,(1998) *Stable Adaptive Systems*. Englewood Cliffs, Prentice-Hall International.

Rojko, A., Jezernik, K., (2004) Sliding-mode motion controller with adaptive fuzzy disturbance estimation. *IEEE trans. ind. electron.)*, **Vol. 51**, No. 5, pp. 963-971

R. J. Schilling (1990), Fundamentals of Robotics, Englewood Cliffs, NJ: Prentice Hall, New Jersey.

Q. P. Ha, Q. H. Nguyen, D.C. Rye, H. F. Durrant-White, (2001) Fuzzy Sliding Mode Controller with Applications, *IEEE Trans. On Industrial Electronics*, **Vol. 48**, No.1.

Wong L. K, Leung ., F. H. F., Tam P. K.S., (2001), A Fuzzy Sliding Controller for Nonlinear Systems, IEEE Trans. On Industrial Electronics, **Vol. 48**, No.1, pp. 32-37.

Wang L. X., (1994) *Adaptive Fuzzy Systems and Control*, Englewood Cliffs, NJ: Prentice Hall, New Jersey.