Fuzzy Model-based Robot Motion Control, Conventional and Decoupled Realization

Andreja Rojko, Karel Jezernik

Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
E-mail: andreja.rojko@uni-mb.si, karel.jezernik@uni-mb.si

Abstract: *The paper considers the problem of the robot motion control, where fuzzy logic dynamic model is used in the controller scheme. Two realizations are investigated. In first we implement decoupled control, that is one independent controller for the each robot joint. Here coupling effects are treated as the disturbances. In second realization there is also one fuzzy logic systems for each robot joint, but with input information from other joints so that the effects of coupling are also estimated. Simulation results are shown for both cases. Experimental results are shown for the decoupled approach. Control object is three degree of freedom robot without gears.*

Keywords: *robot, motion control, fuzzy logic, decoupled control, adaptive control*

I INTRODUCTION

There is a lot of effective motion control algorithms for the nonlinear mechanisms such as direct drive robots. Many of those algorithms require realistic dynamic model which is often the problem. Lately, instead of mathematical dynamic model, neural networks or adaptive fuzzy logic systems are used a lot in the different control schemes to estimate unknown nonlinear dynamic [1], [4], [6], [7]. Because the adaptive parameters are used, there is no problem with the unknown, poorly known or variable parameters.

In this paper we study the merging of the decentralized realization of conventional computed torque control and the adaptive fuzzy logic (neuro-fuzzy system), where fuzzy logic system - FLS is used instead of the dynamic model needed for realization of computed torque control.

The paper is organized as follows. Section II defines the problem of the robot motion control and describes the control design. In section III the general form of rules in rule base is described, together with the structure of FLS. In section IV simulation and experimental results are shown. Conclusions are drawn in section V.

II CONTROL DESIGN

Dynamics of m-degree of freedom stiff direct drive robot is described with:

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

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

Let us define the position and velocity error vectors as

$$\mathbf{e}_q(t) = \left[q_1^d(t) - q_1(t),..,q_m^d(t) - q_m(t)\right]^T$$

$$\dot{\mathbf{e}}_q(t) = \left[\dot{q}_1^d(t) - \dot{q}_1(t),..,\dot{q}_m^d(t) - \dot{q}_m(t)\right]^T \cdot$$

The reference trajectory is defined with the position $\mathbf{q}^{\mathbf{d}}(t) \in \Re^m$, velocity $\dot{\mathbf{q}}^{\mathbf{d}}(t) \in \Re^m$ and acceleration $\ddot{\mathbf{q}}^{\mathbf{d}}(t) \in \Re^m$ vectors.

Conventional computed torque control is one of often used motion controllers, that are model based [5]. The control torques in this scheme are calculated:

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}}^c + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_t(\boldsymbol{q},\dot{\boldsymbol{q}}) \quad (2)$$

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

$$\ddot{\boldsymbol{q}}^c = K_p\boldsymbol{e}_q + K_v\dot{\boldsymbol{e}}_q + \ddot{\boldsymbol{q}}^{\boldsymbol{d}} . \quad (3)$$

Here K_p and K_v are *m*x*m* diagonal matrixes of the velocity and position gains. For this control the realistic dynamic model of the robot is necessary. In that case the control decouples and linearizes the system (1). In continuation we use this control scheme (2), (3) to derive decentralized control.

a) Decentralized Control

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

$$\tau_k = \bar{J}_{kk}\ddot{q}_k + \Delta J_{kk}(\boldsymbol{q})\ddot{q}_k + \sum_{j=1, j\neq k}^{m} J_{kj}(\boldsymbol{q})\ddot{q}_j + \\ + \sum_{j=1}^{m}\sum_{l=1}^{m} C_{jl,k}(\boldsymbol{q})\dot{q}_j\dot{q}_l + G_k(\boldsymbol{q}) + \tau_{t,k}(\boldsymbol{q},\dot{\boldsymbol{q}}) \quad (4)$$

where $\bar{J}_{kk}$ is the constant inertia part and $\Delta J_{kk}(\mathbf{q})$ is the variable part of the joint inertia, and $J_{kj}(\mathbf{q})$ stands for the coupling inertias. Let us denote the whole dynamics of the robot joints (4), with exception of the constant part of the inertias as w_k:

$$w_k = \Delta J_{kk}(\boldsymbol{q})\ddot{q}_k + \sum_{j=1, j\neq k}^{m} J_{kj}(\boldsymbol{q})\ddot{q}_j + \\ + \sum_{j=1}^{m}\sum_{l=1}^{m} C_{jl,k}(\boldsymbol{q})\dot{q}_j\dot{q}_l + G_k(\boldsymbol{q}) + \tau_{t,k}(\boldsymbol{q},\dot{\boldsymbol{q}}) . \quad (5)$$

Considering (4) in (5), the model of the *k*-th robot's joint with the joint drive torque as an inputs is following:

$$\tau_k = \bar{J}_{kk}\ddot{q}_k + w_k . \quad (6)$$

Next we use FLS, one for each robot joint. Its output is estimated torque $\hat{w}_k$, which is calculated only on the basis of the signals for that joint. Finally the control law is:

$$\tau_k = \bar{J}_{kk}\ddot{q}_k^c + \hat{w}_k\left(q_k, \dot{q}_k, \ddot{q}_k\right) . \quad (7)$$

b) Control with Coupling Estimation

Again we have the same structure of control low as in case a). But now we want FLS to estimate the coupling effects too; coupling is neglected in the scheme a). Accordingly the signals from other joints are necessary as inputs in FLS. However we still have one controller for each robot joint. The control (7) rewritten for this case is:

$$\tau_k = \bar{J}_{kk}\ddot{q}_k^c + \hat{w}_k\left(q_1, \dot{q}_1, \ddot{q}_1, ..q_m, \dot{q}_m, \ddot{q}_m\right) \quad (8)$$

III DESIGN OF FUZZY LOGIC SYSTEM

a) Decentralized Control

As we have completely decentralized control (7), the inputs in FLS can be only information regarding that joint status. We have available actual position, actual velocity and desired trajectory information. Rules R_k^l can have the following 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 $\hat{w}_k = \bar{y}_k^l$

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 used for the input membership functions (MF). The output of the resulting FLS can be calculated as, [6]:

$$\hat{w}_k = \frac{\sum_{l=1}^{M} \bar{y}_k^l \prod_{i=1}^{n} \mu_{R_k^{i,l}}\left(x_k^i\right)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{R_k^{i,l}}\left(x_k^i\right)}, \tag{9}$$

$$\mu_{R_k^{i,l}}\left(x_k^i\right) = \left(1 \middle/ \left(1 + \left| \frac{x_k^i - \bar{x}_k^{i,l}}{\sigma_k^{i,l}} \right|^{2 \cdot b_k^{i,l}} \right) \right). \tag{10}$$

If we collect the positions of the output MF in the vector $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} = \left[\bar{y}_k^1, \dots, \bar{y}_k^M\right]^T$ and the rest of expression (9) in another vector $\xi_k(\boldsymbol{x}_{\boldsymbol{k}}) = \left[\xi_k^1(\boldsymbol{x}_{\boldsymbol{k}}), \dots, \xi_k^M(\boldsymbol{x}_{\boldsymbol{k}})\right]^T$, then (9) can be rewritten as

$$\hat{w}_k = \sum_{l=1}^{M} \bar{y}_k^l \cdot \varphi_{\boldsymbol{k}}^{\boldsymbol{l}}(\boldsymbol{x}_{\boldsymbol{k}}) = \hat{\boldsymbol{\theta}}_{\boldsymbol{k}}^{\boldsymbol{T}} \cdot \xi_{\boldsymbol{k}}(\boldsymbol{x}_{\boldsymbol{k}}). \tag{11}$$

In the theory of adaptive systems and according to [6], the adequate adaptation algorithm for positions of the output MF's for this class of systems is:

$$\dot{\hat{\boldsymbol{\theta}}} = -\alpha f_e \boldsymbol{\xi}(\boldsymbol{x}). \tag{12}$$

Here we use $fe_k = a_{1,k} \cdot e_k + a_{2,k} \cdot \dot{e}_k$.
a_1 and a_2 are positive constants, and α as an adaptation rate.

Positions and widths of the membership functions were chosen, so that they cover the whole physically possible values (position, velocity, acceleration) of the mechanism on which the control was implemented. Three membership functions for each of three FLS input were used. Their shape and distribution is depictured on Figure 1. The implemented rule base consists of 15 rules for each joint and is shown in Table 1.

b) Control with Coupling Estimation

Here the same structure of FLS as in case a) is implemented, only that inputs are also signals (positions, velocities, accelerations) from other joints so that the coupling effects can also be estimated. Membership functions are the same as in case a), shown on Figure 1. The rule base for each robot joint is written in Tables 2, 3 and 4. The inputs into each rule were chosen so that each rule is meant to estimate one specific part of robot dynamics. For example; the rules where the input is acceleration from other joints are meant to estimate the inertia coupling. Next the rules where two different velocities are used as input are working toward estimation of the Coriollis force effect. Adaptation algorithm is the same as in case a).

IV SIMULATION AND EXPERIMENTAL RESULTS

For the test object three degree of freedom Puma configuration robot without the gears was used, shown on Figure 2. The implemented simulation parameters are α=0.2; $Kp_{,1,2,3}$=[1000, 2400, 1200], velocity gains $Kv_{1,2,3}$=[64, 98, 70] and parameters of the average inertia matrix J=diag([3.5, 2.5, 0.13]) kgm^2. The sampling time was 2ms. All the parameters were same for both realizations and for simulation and experiment. The reference movement was also the same for all cases; it was point to point movement. End position of movement was 1.5 rad, velocity 0.5 rad/s and acceleration was 0.1 rad/s^2.

The simulation results are depicted on the Figure 3. The robot tip position error is little lower when FLS with estimation of the coupling is used as in case when decoupled control is used. This is expected result, as in this case higher number of rules is used and the rules estimate also coupling effects. Stability was achieved in the both cases.

For the experiment decoupled approach was used, as the controller algorithm has lower computational complexity. The result is shown on Figure 3. Maximum robot tip position error is 2.6 mm and there is no steady state error. The control was stable for all possible accelerations and velocities.

Conclusion

The presented motion controller presents the case of merging of the conventional control techniques and new soft computing approaches. It is also shown, that the simple FLS with effective adaptation algorithm can be successful in replacing dynamic model of the direct drive robot. Simulations and the experiments show good performance of the proposed approaches.

	RULE	POSITION k-th joint	VELO- CITY k-th joint	ACCELE- RATION k-th joint
	R^1	neg.	-	neg.
	R^2	zero	-	zero
	R^3	pos.	-	pos.
	R^4	neg.	neg.	-
	R^5	zero	zero	-
	R^6	pos.	pos.	-
	R^7	pos.	zero	zero
	R^8	neg.	zero	zero
	R^9	zero	neg.	pos.
	R^{10}	zero	zero	zero
	R^{11}	zero	zero	pos.
	R^{12}	zero	neg.	neg.
Not	R^{13}	pos.	zero	pos.
used	R^{14}	neg.	neg.	neg.
on 1. axis	R^{15}	pos.	pos.	pos.

Table 1
Rule base for decoupled FLS, for k-th robot joint

1. JOINT		1. INPUT	2. INPUT	3. INPUT
Rules for coup-lnig trough inertia	R^{16}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_2 = zero$
	R^{17}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_2 = neg.$
	R^{18}	$q_2 = pos.$	$q_3 = pos.$	$\ddot{q}_2 = pos.$
	R^{19}	$q_2 = neg.$	$q_3 = neg.$	$\ddot{q}_2 = zero$
	R^{20}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_3 = pos.$
Rules for coupling trough Coriollis and centre-fugal forces	R^{21}	$q_2 = zero$	$\dot{q}_1 = zero$	$\dot{q}_2 = zero$
	R^{22}	$q_2 = zero$	$\dot{q}_1 = pos.$	$\dot{q}_2 = pos.$
	R^{23}	$q_3 = neg.$	$\dot{q}_1 = neg.$	$\dot{q}_2 = neg.$
	R^{24}	$q_3 = pos.$	$\dot{q}_1 = pos.$	$\dot{q}_2 = pos.$
	R^{25}	$q_2 = zero$	$\dot{q}_2 = zero$	$\dot{q}_3 = zero$
	R^{26}	$q_3 = zero$	$\dot{q}_2 = zero$	$\dot{q}_3 = zero$

Table 2
Additional rules for FLS with coupling estimation, 1. robot joint

2. JOINT		1. INPUT	2.INPUT	3.INPUT
Rules for coupling trough inertia	R^{16}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_1 = zero$
	R^{17}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_1 = neg.$
	R^{18}	$q_2 = pos.$	$q_3 = pos.$	$\ddot{q}_1 = pos.$
	R^{19}	$q_2 = neg.$	$q_3 = neg.$	$\ddot{q}_1 = zero$
	R^{20}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_3 = pos.$
Rules for coupling trough Coriollis and centre-fugal forces	R^{21}	$q_1 = zero$	$\dot{q}_1 = zero$	$\dot{q}_2 = zero$
	R^{22}	$q_3 = pos.$	$\dot{q}_2 = pos.$	$\dot{q}_3 = pos.$
	R^{23}	$q_3 = neg.$	$\dot{q}_2 = neg.$	$\dot{q}_3 = neg.$
	R^{24}	$q_3 = pos.$	$\dot{q}_1 = pos.$	$\dot{q}_2 = pos.$
	R^{25}	$q_1 = zero$	$\dot{q}_1 = zero$	$\dot{q}_1 = zero$
	R^{26}	$q_3 = zero$	$\dot{q}_3 = zero$	$\dot{q}_3 = zero$

Table 3
Additional rules for FLS with coupling estimation, 2. robot joint

3. JOINT		1. INPUT	2. INPUT	3. INPUT
Rules for coupling trough inertia	R^{16}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_1 = zero$
	R^{17}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_1 = neg.$
	R^{18}	$q_2 = pos.$	$q_3 = pos.$	$\ddot{q}_1 = pos.$
	R^{19}	$q_2 = neg.$	$q_3 = neg.$	$\ddot{q}_1 = zero$
	R^{20}	$q_2 = zero$	$q_3 = zero$	$\ddot{q}_2 = pos.$
Rules for coupling trough Coriollis and centre-fugal forces	R^{21}	$q_3 = zero$	$\dot{q}_1 = zero$	$\dot{q}_1 = zero$
	R^{22}	$q_2 = zero$	$\dot{q}_1 = zero$	$\dot{q}_1 = zero$
	R^{23}	$q_3 = zero.$	$\dot{q}_2 = zero$	$\dot{q}_2 = zero$
	R^{24}	$q_3 = pos.$	$\dot{q}_2 = pos.$	$\dot{q}_2 = pos.$
	R^{25}	$q_2 = zero$	$\dot{q}_2 = zero$	$\dot{q}_2 = zero$
	R^{26}	$q_2 = neg.$	$\dot{q}_2 = neg.$	$\dot{q}_2 = pos$

Table 4
Additional rules for FLS with coupling estimation, 3. robot joint

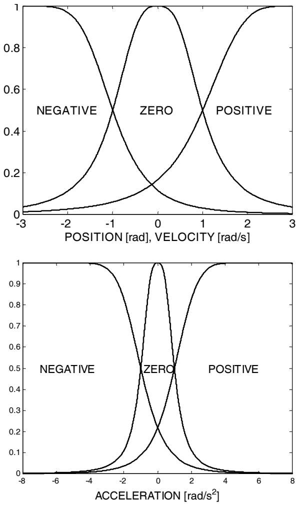

Figure 1
Membership functions of input variables position, velocity and acceleration of FLS

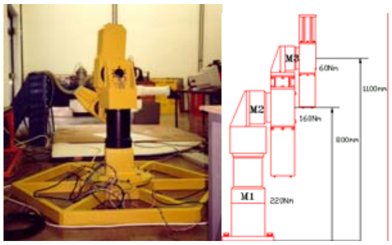

Figure 2
Direct drive robot

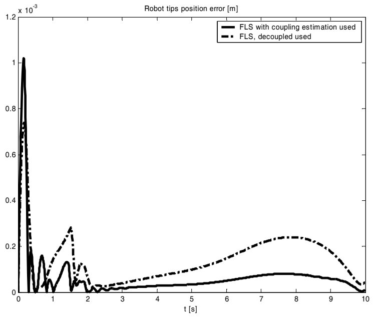

Figure 3
Simulation results for conventional and decoupled approach

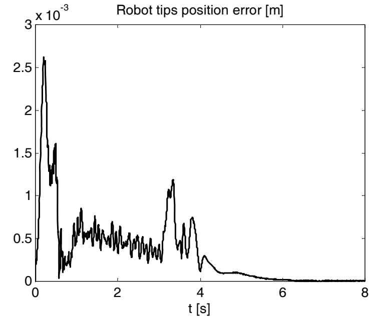

Figure 4
Experimental results for decoupled approach

References

[1] O. Kaynak, K. Erbatur, M. Ertugrul: The Fusion of Computationally Intelligent Methodologies and Sliding-Mode Control - A Survey, IEEE Trans. Industrial Electronics, Vol. 48, No. 1, pp. 4-17, 2001

[2] Ljung L.: System Identification: Theory for the Users, Prentice Hall, New Jersey, 1987

[3] Narendra K. S., Annaswamy A. M.: Stable Adaptive Systems. Englewood Cliffs, Prentice-Hall International, 1988

[4] Rojko, A., Jezernik, K.: Sliding-mode Motion Controller with Adaptive Fuzzy Disturbance Estimation. IEEE Trans. Ind. Electron.), 2004, Vol. 51, No. 5, pp. 963-971

[5] R. J. Schilling: Fundamentals of Robotics, Prentice Hall International, 1900

[6] Wang L. X.: Adaptive Fuzzy Systems and Control, Englewood Cliffs, NJ: Prentice Hall, New Jersey, 1994

[7] B. K. Yoo, W. C. Ham: Adaptive Control of Robot Manipulator Using Fuzzy Compensator, IEEE Trans. Fuzzy Syst., 2000, Vol. 8, pp. 186-199