Adaptive control system of a robot manipulator based on a decentralized position-dependent PID controller

Jan Cvejn, Jiří Tvrdík

University of Pardubice, Faculty of Electrotechnics and Informatics, Studentská 95, 532 10, Pardubice, Czech Republic

jan.cvejn@upce.cz

Abstract: The paper describes an approach to adaptive feedback control of a robot manipulator, based on partitioning of the joint space into segments. Within each segment the robot is controlled as a decoupled linear system by means of conventional PID controllers. To achieve continuity of control variables the segments are represented as fuzzy sets. The controller settings are adapted by online identification from past measurements of position and control signals.

Keywords: Robot manipulator; Motion control; PID controller; Fuzzy modeling

1 Introduction

Control of robot manipulators is difficult in general, because robot dynamics is usually strongly non-linear. Although the influence of non-linear terms in the arm motion equations can be suppressed by using high-ratio gears in the actuators, is such cases the friction in the gears and bearings often degrades the actuator performance for high-velocity motions. Especially in the case of light-weight, high-velocity robot arms for pure manipulating purposes, the arm dynamics cannot be neglected to achieve optimal performance.

In this paper the problem of tracking a trajectory, provided by the motion planning layer of the robot control system, is discussed. It is assumed that the trajectory, defined in the robot operational space, is transformed into the robot joint space by the algorithm of inverse kinematics [1], before the motion task is performed. The motion control layer then works with the information on the robot joint positions, i.e. the relative positions of the robot links.

Multiple approaches to design of the feedback control system of robot manipulator are described in literature [1,2]. The simplest approach, suitable only for low-velocity motions, works with the actuators as with velocity generators and the effects of the robot dynamics are considered as unknown disturbances [2]. The feedback control can be then based on PI or PID controllers. To enhance the performance, cascade configuration with additional velocity or even acceleration feedback can be used. It is also possible to use a partial feed-forward compensation of non-linearities, if a partial knowledge of the robot mathematical model is available [1].

More advanced robot control architectures use actuators as torque generators [1,2]. This approach is utilized in centralized control systems, viewing the robot dynamics

in full complexity as a high-order, coupled and non-linear one. The centralized methods utilize some special features of the robot dynamics. In particular, dynamic inversion method transforms the controller design problem into a linear one by means of additional interior loop. However, applicability of this approach depends on precision of the mathematical model available, which often cannot be guaranteed due to unknown influences, such as backslashes and flexibilities in the gears or saturations of the action forces. Therefore, practical usability requires some extensions, guaranteeing at least closed-loop asymptotical stability [1]. Among alternative approaches e.g. the non-linear PID control based on Lyapunov stability theory or passive systems theory can be mentioned [1, 2].

An advantage of the decentralized control approach is that for the controller tuning only rough robot mathematical model is sufficient, describing approximate inertial effect on individual axes and damping effects. If we assume that terms in the robot model depend only on position, it is possible to improve the performance by dividing the joint space into segments with constant controller settings. The controller parameters can be changed during motion when the trajectory goes across the segment boundary. Within each segment the robot then can be controlled as a decoupled linear system by means of conventional PID controllers. It is possible to use initially the same settings in all the segments and to adapt the controller parameters in each segment automatically by processing past measurements of the kinematic and control variables. Such an algorithm adapts the controller parameters also when the robot manipulated load changes.

In this paper an extension of the decentralized robot control approach is proposed, based on the idea outlined above, including some additional enhancements. Practical implementation is indeed more complex than in the case of conventional decentralized control. Partitioning of the joint space into segments brings increased requirements on the control system hardware, as regards both performance and memory capacity. However, it reveals that these requirements can be fulfilled by using current 32-bit microcontroller-based platforms.

2 The robot mathematical model

The mathematical model of a robot arm consisting of n links in an open kinematic chain, moving freely in the operation space, can be considered in the form

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{f} \tag{1}$$

where $\mathbf{q}$ is the vector of joint positions and $\mathbf{f}$ the vector of total force effects of actuators. If K and P denote the total kinetic and potential energy, $\mathbf{B}(\mathbf{q}) = \partial^2 K / \partial \dot{\mathbf{q}}^2$ is a positive definite position-dependent inertia matrix,

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \frac{\partial^2 K}{\partial \mathbf{q} \partial \dot{\mathbf{q}}}\dot{\mathbf{q}} - \frac{\partial K}{\partial \mathbf{q}} \tag{2}$$

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

is a non-linear function corresponding to the effects of centrifugal and Coriolis forces and $\mathbf{g}(\mathbf{q}) = \partial P / \partial \mathbf{q}$ is the vector function corresponding to the gravity-force effects [1]. If we assume only electrical DC actuators, by neglecting the winding inductance and mechanical friction, we obtain

$$\mathbf{M} \approx \mathbf{K}_u \mathbf{u} - \mathbf{K}_v \boldsymbol{\omega} \tag{3}$$

where $\mathbf{M}$ is the vector of motor output torques, $\mathbf{u}$ the input voltages, $\boldsymbol{\omega}$ the vector of motor angular velocities and $\mathbf{K}_u > 0$, $\mathbf{K}_v > 0$ constant diagonal matrices. In principle, (3) allows using the motors as velocity generators, where the connected load is represented as a disturbance. Alternatively, the motor can play the role of a torque generator, where the term $\mathbf{K}_v \boldsymbol{\omega}$, corresponding to induced voltage in winding, is considered as electromagnetic friction. In this case, the motor is usually equipped with inner current feedback, which reduces the effect of $\mathbf{K}_v \boldsymbol{\omega}$ and protects from overload [1]. Although this paper is based on the idea of decentralized control, the motors are considered as torque generators, like at most centralized control methods. In this case

$$\mathbf{f} = \mathbf{K}_r \left(\mathbf{K}_u \mathbf{u} - \mathbf{K}_v \boldsymbol{\omega} \right) - \mathbf{F} \dot{\mathbf{q}} \tag{4}$$

where $\mathbf{K}_r > 0$ and $\mathbf{F} > 0$ are diagonal matrices. The term $\mathbf{F}\dot{\mathbf{q}}$ corresponds to viscous friction in bearings and gears and $\mathbf{K}_r$ is the mechanical gear ratio. Coulomb friction is not considered. Since $\dot{\mathbf{q}} = \mathbf{K}_r^{-1} \boldsymbol{\omega}$,

$$\mathbf{f} = \mathbf{K}_r \mathbf{K}_u \mathbf{u} - \mathbf{F}_r \dot{\mathbf{q}}, \quad \mathbf{F}_r = \mathbf{K}_r \mathbf{K}_v \mathbf{K}_r + \mathbf{F}. \tag{5}$$

The equation (1) then can be rewritten as

$$\mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \left(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}_r \right) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{K}\mathbf{u} \tag{6}$$

where $\mathbf{K} = \mathbf{K}_r \mathbf{K}_u$. Since the dependence of $\mathbf{F}_r$ on $\mathbf{K}_r$ is quadratic, the term $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ has low influence in the case of higher mechanical gear ratios and the model depends predominantly only on $\mathbf{q}$.

3 The decentralized control with partial knowledge of $\mathbf{B}(\mathbf{q})$

One possible version of the decentralized robot control algorithm uses partial knowledge of the inertia matrix $\mathbf{B}(\mathbf{q})$, which is decomposed as

$$\mathbf{B}(\mathbf{q}) = \overline{\mathbf{B}} + \Delta \mathbf{B}(\mathbf{q}) \tag{7}$$

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

where $\bar{\mathbf{B}}$ is a constant diagonal positive definite matrix, corresponding to approximate average inertial effects on individual axes [1]. The robot model for the controller design is in the form

$$\bar{\mathbf{B}}\ddot{\mathbf{q}} + \mathbf{F}_r\dot{\mathbf{q}} = \mathbf{K}\mathbf{u} - \mathbf{d} \tag{8}$$

where

$$\mathbf{d} = \Delta\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{9}$$

is disturbance. Since all matrices in (8) are diagonal, it is possible to write (8) as

$$b_i\ddot{q}_i + f_{ri}\dot{q} = k_i u_i - d_i, \quad i = 1,..,n \tag{10}$$

where b_i, f_{ri} and k_i denote the diagonal terms of $\bar{\mathbf{B}}$, $\mathbf{F}_r$ and $\mathbf{K}$, respectively. Eq. (10) can be rewritten as

$$T_i\ddot{q}_i + \dot{q} = K_i u_i - \delta_i \tag{11}$$

where $T_i = b_i / f_{ri}$, $K_i = k_i / f_{ri}$ and $\delta_i = d_i / f_{ri}$. The corresponding axis transfer function is in the form

$$F_i(s) = \frac{K_i}{s(T_i s + 1)}. \tag{12}$$

To compensate the effect of persistent input-type disturbance d_i of the system (10) and to achieve zero tracking error in the case of ramp reference trajectory, a controller with additional zero pole is needed. If we use the PID controller with the transfer function in Laplace transform

$$R_i(s) = r_i\left(1 + \frac{1}{T_{Ii}s} + T_{Di}s\right) = \frac{r_i}{T_{Ii}s}\left(T_{Ii}T_{Di}s^2 + T_{Ii}s + 1\right) \tag{13}$$

for generating the control signals $u_i(t)$, the characteristic polynomial of the *i*-th axis is in the form

$$Q_i(s) = \frac{T_{Ii}s^2(T_i s + 1)}{r_i K_i} + \left(T_{Ii}T_{Di}s^2 + T_{Ii}s + 1\right) = \frac{T_{Ii}T_i}{r_i K_i}s^3 + \frac{T_{Ii} + r_i K_i T_{Di}T_{Ii}}{r_i K_i}s^2 + T_{Ii}s + 1. \tag{14}$$

The PID controller parameters can be determined so that poles of $Q_i(s)$ are placed at desired locations [6]. If we assume $Q_i(s) = (T_{1i}s + 1)\left(T_{2i}^2 s^2 + 2\xi_i T_{2i}s + 1\right)$, where T_{1i}, T_{2i} are chosen closed-loop response time constants and ξ_i is the relative damping ratio, by comparison of the coefficients we obtain

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

$$r_i = \frac{(T_{1i} + 2\xi_i T_{2i})T_i}{T_{1i} T_{2i}^2 K_i},\ T_{Ii} = T_{1i} + 2\xi_i T_{2i},\ T_{Di} = \frac{2\xi_i T_{1i} T_{2i} + T_{2i}^2}{T_{Ii}} - \frac{1}{r_i K_i}. \quad (15)$$

If the disturbance d_i is partially known, its effect can be compensated in part by adding $K_i^{-1}\delta_i$ to the *i*-th axis controller output, where $\delta_i = d_i / f_{ri}$.

4 The extended decentralized robot control algorithm

Consider that the robot joint space S, i.e. the space of all possible joint positions $\mathbf{q} = [q_1, ..., q_n]^T$, is partitioned into *m* segments S_k, $k = 1, .., m$, such that

$$\bigcup_{k=1}^{m} S_k = S,\ \bigcap_{k=1}^{m} S_k = \varnothing. \quad (16)$$

Since the diagonal parts of $\mathbf{B}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q})$ are position-dependent, it is possible to approximate the robot dynamics in each segment by a different linear model. Then the tracking performance can be enhanced if to each segment there correspond different controller settings.

A basic approach is that the controller parameters are rewritten during motion when the trajectory goes across the segment boundary. Within each segment the robot is controlled as a decoupled linear system by means of conventional PID controllers. This extension is rather straightforward and can be efficiently implemented, although a sufficient amount of memory in the control system hardware is needed. If a space of each generalized coordinate is divided into d sub-intervals, the joint space will be divided into $m = d^n$ segments. To each segment S_k there corresponds a matrix of the parameters $\mathbf{P}_k = [\mathbf{K}_k, \mathbf{T}_k, \boldsymbol{\delta}_k]$ describing the plant dynamics as described in the previous section. The columns in $\mathbf{P}_k$ are vectors of *n* components, e.g. $\mathbf{K}_k = [K_{k1}, ..., K_{kn}]^T$. The corresponding PID controlled settings can be obtained directly by substitution into (15).

However, this concept has an important drawback, consisting in discontinuity of the action variables $u_i(t)$ caused by changes of the controller settings at the segments boundaries. Such discontinuities are undesirable, since they can lead to oscillations of the mechanical structure.

One possible solution is replacing the segments S_k by the fuzzy sets $\tilde{S}_k = \{R^n, \mu_k(\mathbf{q})\}$, where the membership functions $\mu_k(\mathbf{q})$ are chosen continuous and so that $\mu_k(\mathbf{q}) \in [0,1]$ and $\mu_k(\mathbf{c}_k) = 1$, where $\mathbf{c}_k$ denotes the centre of the segment S_k. The matrix of the plant parameters is then computed at each control step as

$$\mathbf{P} = \sum_{k=1}^{m} \mu_k(\mathbf{q}) \mathbf{P}_k \Big/ \sum_{k=1}^{m} \mu_k(\mathbf{q}). \quad (17)$$

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

The expression (17) is usually used as an inference rule in Takagi-Sugeno-type fuzzy modeling [4]. However, the computation of (17) at each control step can be time-consuming due to rather large number of segments *m*. It can be considered that the segments are for given parameter h defined by means of their centers $\mathbf{c}_k$ as

$$S_k = \left\{ \mathbf{q} \mid \left\| \mathbf{q} - \mathbf{c}_k \right\|_\infty \le h \right\}, \tag{18}$$

where $\left\| \mathbf{x} \right\|_\infty = \max \left| x_i \right|$ denotes the L_∞ - norm. Then it is possible to define

$$\mu_k \left(\mathbf{q} \right) = \mu \left(\mathbf{q} - \mathbf{c}_k \right) \tag{19}$$

where $\mu \left(\mathbf{q} \right) \in \left[0,1 \right]$ is a chosen continuous function, such that $\mu \left(\mathbf{0} \right) = 1$. Computation of (17) can be made much more efficient if $\mu \left(\mathbf{q} \right)$ is chosen as a function with compact support, see e.g. [5]. A simple possibility is to choose

$$\mu \left(\mathbf{q} \right) = \max \left\{ 1 - \left\| \mathbf{q} \right\|_\infty / \left(\left(1 + \lambda \right) h \right), 0 \right\} \tag{20}$$

where $\lambda > 0$, typically $\lambda \in \left[0.5, 2 \right]$. In this case the values of $\mu_k \left(\mathbf{q} \right)$ are zero, except for the k -the segment, where $\mu_k \left(\mathbf{q} \right)$ has the largest value, and several neighboring segments. This fact can be utilized for efficient implementation of the controller, although such a realization is more complex. A disadvantage of the choice (20) is that this function is not smooth, which will produce non-smooth histories of the control signals. Therefore, it might be preferable to construct $\mu \left(\mathbf{q} \right)$ as at least continuously differentiable. The compact-support choice of $\mu \left(\mathbf{q} \right)$ brings the risk of unbounded values in (17) for trajectories exceeding the boundaries of S, but this problem can be easily avoided, e.g. by increasing λ in the cases when $\sum_{k=1}^{m} \mu_k \left(\mathbf{q} \right) = 0$.

5 Adaptation of the controller settings

Since the number of segments can be rather large, it is necessary that the controller settings are computed automatically. Initially, the settings in all the segments are set to the same values corresponding to the decentralized PID controller design described in Section 3. During the robot operation the settings in the segments can be adapted by processing the measured values of $\mathbf{q}\left(t_k \right)$ and $\mathbf{u}\left(t_k \right)$, at the instants $t_k = k\Delta$, where Δ is the identification scan period. During motion in each segment it is needed to estimate the parameters K_i, T_i and δ_i. The index of segment is omitted below for simplicity, i.e. e.g. K_i should be written as K_{ki} in the *k*-th segment to be precise.

The continuous transfer function (12) has the corresponding transfer function in Z-transform

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

$$F_i(z) = \mathcal{Z}\left\{K_i T_i\left(-1 + t/T_i + e^{-t/T_i}\right);\, t = k\Delta\right\} =$$
$$= K_i T_i\left(-1 + \frac{\Delta}{T_i}\frac{1}{z-1} + \frac{z-1}{z-e^{-\Delta/T}}\right) = K_i\left(\frac{\Delta}{z-1} + T_i\frac{e^{-\Delta/T_i}-1}{z-e^{-\Delta/T_i}}\right). \tag{21}$$

If Δ/T_i is sufficiently low, $e^{-\Delta/T_i} \approx 1-\Delta/T_i$, so

$$F_i(z) \approx K_i\Delta\left(\frac{1}{z-1} - \frac{1}{z-e^{-\Delta/T_i}}\right) = K_i\Delta\frac{1-\alpha_i}{z^2-(1+\alpha_i)z+\alpha_i} \tag{22}$$

where $\alpha_i = e^{-\Delta/T_i}$. Eq. (22) corresponds to the data model

$$q_i^{[k+2]} - (1+\alpha_i)q_i^{[k+1]} + \alpha_i q_i^{[k]} = K_i\Delta(1-\alpha_i)u_i^{[k]} + \varepsilon^{[k]} \tag{23}$$

where $q_i^{[k]}$ denotes the value of q_i at k-th instant and $\varepsilon^{[k]}$ is the error process. After including the disturbance δ_i, defined by (11), (23) can be rewritten as

$$\Delta^{-1}\left(q_i^{[k]} - q_i^{[k+1]}\right)\alpha_i - u_i^{[k]}\tilde{K}_i + \tilde{\delta}_i = \Delta^{-1}\left(q_i^{[k+1]} - q_i^{[k+2]}\right) + \varepsilon^{[k]} \tag{24}$$

where $\tilde{K}_i = K_i(1-\alpha_i)$ and $\tilde{\delta}_i = \delta_i(1-\alpha_i)$. From (24) the value of $\boldsymbol{\theta}_i = \left[\alpha_i, \tilde{K}_i, \tilde{\delta}_i\right]^T$ can be estimated by means of the least-squares method. Then, α_i determines the value of T_i. The parameters K_i and δ_i can be computed directly from $\boldsymbol{\theta}_i$.

Note that to each segment there corresponds one data model (24) and a corresponding data structure have to exist in the control system for storing the measurements. It is advantageous to use the recursive version of the LS estimator [3], which need not store all the data, but only a 3x3 matrix and the vector $\boldsymbol{\theta}_i$ in each segment and for each axis. It is assumed that the settings are updated after a single task is performed, but the same approach can be used when the controller is updated during motion. Let

$$\mathbf{x}_i^{[k]} = w_j\left(\mathbf{q}^{[k]}\right)\left[\Delta^{-1}\left(q_i^{[k]} - q_i^{[k+1]}\right), -u_i^{[k]}, 1\right]^T,\quad y_i^{[k]} = w_j\left(\mathbf{q}^{[k]}\right)\Delta^{-1}\left(q_i^{[k+1]} - q_i^{[k+2]}\right) \tag{25}$$

where $w_j(\mathbf{q})$ is the weight of the measurement in the j-th segment, computed as

$$w_j(\mathbf{q}) = \mu_j(\mathbf{q}) / \sum_{l=1}^{m}\mu_l(\mathbf{q}). \tag{26}$$

The $(k+1)$-th estimate of $\boldsymbol{\theta}_i$ for the j-th segment is obtained as follows:

$$\boldsymbol{\theta}_i^{[k+1]} = \boldsymbol{\theta}_i^{[k]} + \frac{\mathbf{C}_i^{[k]}\mathbf{x}_i^{[k]}}{\gamma + \mathbf{x}_i^{[k]T}\mathbf{C}_i^{[k]}\mathbf{x}_i^{[k]}}\left(y_i^{[k]} - \mathbf{x}_i^{[k]T}\boldsymbol{\theta}_i^{[k]}\right) \tag{27}$$

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

$$\mathbf{C}_i^{[k+1]} = \frac{1}{\gamma}\left(\mathbf{I} - \frac{\mathbf{C}_i^{[k]}\mathbf{x}_i^{[k]}\mathbf{x}_i^{[k]T}}{\gamma + \mathbf{x}_i^{[k]T}\,\mathbf{C}_i^{[k]}\mathbf{x}_i^{[k]}} \right)\mathbf{C}_i^{[k]} \tag{28}$$

where γ is the forgetting coefficient, equal or very close to 1. In the considered case it seems to be necessary to require that $T_i > T_{i\min} > 0$ and $K_i > K_{i\min} > 0$, where the constants $T_{i\min}$, $K_{i\min}$ are suitably chosen. The recursive form of the estimator (27) enables to keep the values of the components of $\boldsymbol{\theta}_i$ in the corresponding range by updating only with feasible values. The matrix $\mathbf{C}_i^{[0]}$ is set as $\mathbf{C}_i^{[0]} = \mathrm{diag}\left(\sigma_1, \sigma_2, \sigma_3\right)$, where $\sigma_k > 0$ are chosen. Total memory requirements can be estimated as $12\,m\,n$ real numbers, which can occupy from tens to hundreds of kbytes of the control system memory.

6 Simulated results

Consider the 3-DOF anthropomorphic robot arm approximate model in Fig. 1, where $m_1 = m_2 = 1\ \mathrm{kg}$, $l_1 = l_2 = 0.5\,\mathrm{m}$ and $h = 1\mathrm{m}$. The terms of the diagonal matrices $\mathbf{K}_u, \mathbf{K}_v, \mathbf{K}_r$ and $\mathbf{F}$ were chosen as $k_{ui} = 1$, $k_{vi} = 0$, $k_{ri} = 10$ and $f_i = 3$, $i = 1,..,3$. The mathematical model for simulation, which is strongly non-linear, was obtained by expressing the terms $\mathbf{B}(\mathbf{q})$, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ and $\mathbf{g}(\mathbf{q})$ in (1), where $\mathbf{q} = \left[\varphi, \psi, \vartheta\right]^T$, from the expressions for kinetic and potential energy. Since the mathematical model is rather complex, the details had to be omitted due to paper length limitations.

First, the fixed axes PID controllers (13) were used. The diagonal matrix $\overline{\mathbf{B}}$ for setting-up the PID controllers, with the meaning of rough estimate of the inertia matrix, was chosen as

$$\overline{\mathbf{B}} = \mathrm{diag}\left(\frac{m_1 l_1^2 + m_2\left(l_1 + l_2\right)^2}{2},\ \left(m_1 + m_2\right) l_1^{\,2},\ m_2 l_2^{\,2} \right). \tag{29}$$

The reference trajectory was chosen as the step function, which can be viewed as the worst-case situation, since the robot will usually track a continuous trajectory. The controller parameters were computed so that $T_{1i} = T_{2i} = 0.075s$ and $\xi_i = 0.8$ in (15). The joint initial and target positions are considered as

$$\mathbf{q}_0 = \left[-1.5, 2, 3\right],\ \mathbf{q}_f = \left[1.5, -2, -1\right]. \tag{30}$$

Figure 2 shows the corresponding histories of the robot joint positions.

Further, the proposed adaptive controller has been used. The joint space has been divided into $m = 8^3$ segments, $\lambda = 1$ has been chosen in (20). The controller has been initialized to the same settings as in the previous case and executed 20 times the same trajectory with the scan period $\Delta = 0.002$ s to adapt. The desired closed-loop settings

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

$T_{1i} = T_{2i} = 0.075s$ and $\xi_i = 0.8$ were preserved. The matrices $\mathbf{C}_i$ have been initialized as $\mathbf{C}_i^{[0]} = 0.01 \times \mathrm{diag}(1, 0.3, 1)$ and $\gamma = 0.9^{\Delta}$, $T_{i\min} = 0.03$ and $K_{i\min} = 0.05$ has been chosen. Figure 3 shows the final simulated histories. It can be seen that significant enhancement has been achieved. Similar results were obtained also for lower values of m. In these cases the convergence was faster, but it seems that it is necessary to use higher values of $T_{i\min}$ and $K_{i\min}$ and the responses are a little slower.

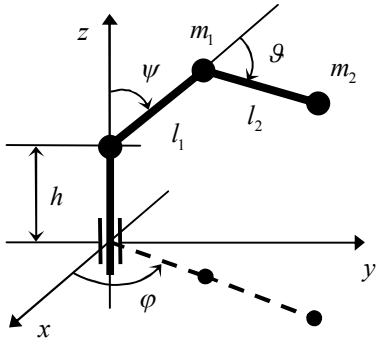

Fig. 1. The robot arm approximate model

7 Conclusions

The proposed adaptive control system is based on the principles of the PID controller-based decentralized control, where the joint space is divided into segments with different controller settings. To ensure continuity of the control signal at the segment boundaries, the segments are represented as fuzzy sets with a special choice of the membership function. Simulated results show that the approach can be used in the cases when the conventional decentralized control fails to produce good responses, although such situations seem to occur mainly in the cases of long-range and high-velocity movements. The control system memory requirements are large in comparison with conventional control algorithms, but can be fulfilled by using current 32-bit microcontrollers. Thus the worst problem from practical point of view seems to be proper initial settings of $\mathbf{C}_i$, $T_{i\min}$, $K_{i\min}$ and other parameters that influence convergence of the sequence of estimates (27).

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.

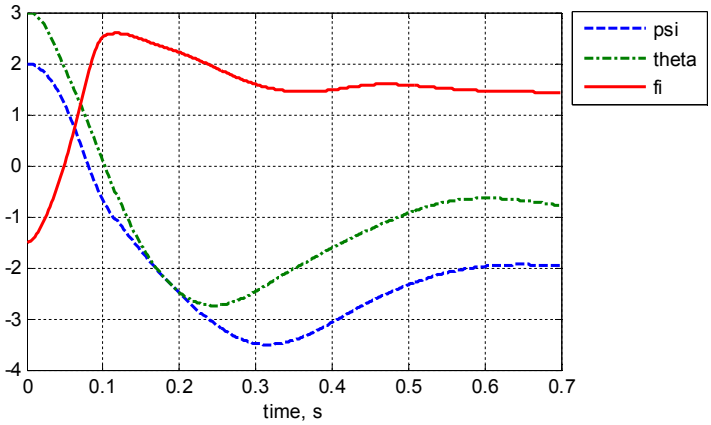

Fig. 2. The joint step responses - fixed axes PID controllers

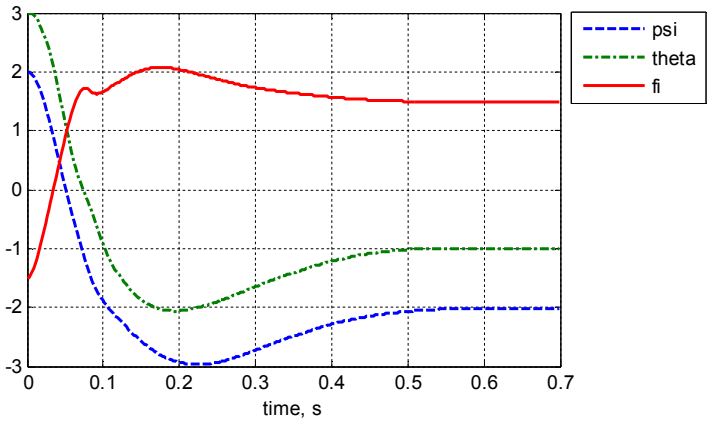

Fig. 3. The joint step responses - adaptive control system, $m = 8^3$

References

1. Sicilliano, B., Sciavicco, L., Oriollo, G. Robotics: Modelling, Planning and Control. Springer - Verlag (2009)
2. Siciliano, B., Khatib, O. (Eds.): Springer Handbook of Robotics. Springer-Verlag (2008)
3. Goodwin, G.C, Payne, R. L.: Dynamic System Identification. Experimental Design and Data Analysis. Academic Press, London (1977)
4. Sugeno, M., Industrial applications of fuzzy control, Elsevier Science Pub. Co. (1985)
5. Rektorys, K., Variational Methods in Mathematics, Science and Engineering. 2nd edition. Reidel, Dordrecht (1980)
6. Kiong, T. K., Quing-Guo, W., Chiech H. Ch. and Hägglund, T. J., Advances in PID Control. Springer-Verlag (1999)

Published in: R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2. Springer, 2017. DOI 10.1007/978-3-319-57264-2_10.