Control
and Cybernetics

VOL. 15 (1986) No. 3-4

Decentralised adaptive controllers for robotic manipulators

by

KRZYSZTOF MALINOWSKI

Institute of Automatic Control
Technical University of Warsaw
Nowowiejska 15/19
00-665 Warszawa, Poland

The paper presents and discusses possible applications of adaptive controllers to continuous path control (CPC) of a robotic arm. The attention is focused on decentralised adaptive controllers since in practical applications the decentralised motion controllers are preferred to multi-input-multi-output controllers. The reason is fairly obvious: a decentralised control structure is much more simple and provides for easy real-time implementation of multi-microprocessor facilities. One possible solution of a CPC algorithm design which would preserve a simple decentralised form of the controller and yet could be capable of achieving a required performance of continuous path control is offered by the use of adaptive decentralised controllers with adjustable gains. In the paper several possible ways of introducing such controllers are presented when basing upon simple "conceptual" models of joint dynamics. The approach of Dubowski and Des Forges as well as some other approaches to design model reference decentralised adaptive controllers are discussed and relevant modifications to these control schemes are proposed. Finally simple sampled-data adaptive controllers for a hydraulic robot are presented together with discussion of simulation results regarding possible implementation to the industrial robot RIMP-1000.

1. Introduction

Numerous applications of robots require robotic arm to follow a desired continuous path in order to perform the prescribed tasks (e.g. painting, arc welding etc.). Design of continuous path controllers is made difficult by extremely complicated nonlinear dynamics of manipulators, varying weights of manipulated objects and by stringent requirements on the performance (precision of motion) of the manipulator systems.

Before a brief discussion of possible approaches to the CPC design let us consider the basic model of a robotic arm with n joints. Each joint is driven by an actuator (an electric d–c motor or a hydraulic actuator) via a transmission system. If we define a ψ-coordinate system, where ψ denotes e.g. a vector of the schaft angles of the d–c motors or a vector of the joint angles, then (transmission equations can be treated as holonomic constraints) the vector ψ can be used as a vector of generalized coordinates and one can obtain a mathematical model of a robotic arm in the Lagrange form:

$$\frac{d}{d_t}\left(\frac{\partial T}{\partial \dot{\psi}_i}\right)-\frac{\partial T}{\partial \psi_i}+\frac{\partial U}{\partial \psi_i}=Q_i, \quad i=1, ..., n, \tag{1}$$

where T, U denote the kinetic and the potential energies and Q_i, s are the generalized forces (including friction and the external forces e.g. those exercised by the actuators). Balestrino et. al. (1983) and Stokič, Vukobratovič (1984) show that eqn. (1) can be written in the following form:

$$\dot{x}=A(x)\cdot x+B(x)\cdot \boldsymbol{u}, \quad x=[\psi^T, \dot{\psi}^T]^T, \tag{2}$$

where $\boldsymbol{u}$ denotes the vector of control inputs.

Now, the task of a continuous path controller will consist of applying such control inputs so as keep an error $\boldsymbol{\epsilon}(t)=\boldsymbol{r}(t)-\psi(t)$ (where $\boldsymbol{r}(\cdot)$ is a desired trajectory of motion specified in ψ-coordinate system over, say, time interval $[t_0, t_f]$) within a prescribed region $E(t)$ (i.e. we want to have $\boldsymbol{\epsilon}(t)\in E(t)$). This time-dependent requirement represents an objective of the CPC design. As far as the desired trajectory $\boldsymbol{r}(\cdot)$ is concerned the following two cases are worth mentioning:

A. $\boldsymbol{r}(\cdot)$ is known in advance,

B. $\boldsymbol{r}(\cdot)$ is specified (by higher control layers) during the motion of a robotic arm (e.g. $\boldsymbol{r}(t)$ is specified at time $t-\Delta t$).

One possibility of designing a CPC system in case A is to use eqn. (1) or eqn. (2) and $\boldsymbol{r}(\cdot)$ to compute a nominal control $\boldsymbol{u}_0(t)$ for $t\in[t_0, t_f]$ (this is time consuming task and requires a very good model) and then to use a linearized (or otherwise simplified) manipulator model in order to design a controller using deviations of $\psi(t)$ from $\boldsymbol{r}(t)$ to provide for control input corrections $\delta\boldsymbol{u}(t)$ (then $\boldsymbol{u}(t)=\boldsymbol{u}_0(t)+$ $+\delta\boldsymbol{u}(t)$).

Another possibility is to design a sliding-mode control algorithm. In such case the reference trajectory has to be specified as the intersection of several sliding surfaces (see eq. Balestino et. al (1983) and (Slotine (1984)) and the controller itself will be nondiagonal (nondecentralized) and will have a complicated structure (difficult for real-time microprocessor-based implementation).

In case B the situation is even more difficult. It is also clear that any controller capable of doing its job in this case will be also applicable to case A. One, theoretically obvious, possibility to design a (CPC) controller is to perform this design in two stages, that is to build a complicated compensator in order to decouple nonlinear dynamics of eqn. (2) and then to design "decentralized" control algorithm for a decoupled system. This, however, requires a good knowledge of the robot dynamics, adjustment of compensators as the weight of manipulated object changes and leads to a very complicated nondiagonal controller. Implementation of such control algorithm requires then quite powerful computing facilities. On the other hand "classical" decentralized linear controllers (like P or PD controllers), which are simple and easy to implement and therefore are still commonly used in an industrial robot design (see Paul (1981)), are not capable of yielding a high performance of continuous path control over a wide range of manipulator tasks.

One possible solution which would preserve a simple decentralized form of the controller and yet could be capable of achieving a required performance of continuous path control is offered by the use of adaptive decentralized controllers with adjustable gains. In the next sections of this paper we will be concerned with several possible ways of introducing such controllers. In section 2 we consider simplified (conceptual) models of joint dynamics and an adaptive simple control law—thinking in terms of an analog (continuous) implementation of this law. In section 3 we present briefly the approach of Dubovski, Des Forges (1979) and the approach of Narendra et. al. (1978, 1980 a, b, c) to adjust the gains of the control laws. We also suggest some modifications to those schemes and we briefly address the stability issue. Finally, in section 4 simple sampled-data adaptive controllers for a hydraulic robot are presented together with the simulation results.

2. Conceptual models of joint dynamics and decentralized adaptive control laws

One way to approach the problem of designing a decentralized controller for a robotic arm is to assume that each joint together with an associated actuator can be described by the following "conceptual" model:

$$J_i(t)\,\ddot{\psi}_i + g_i(t)\,\dot{\psi}_i(t) = u_i(t), \tag{3}$$

where time-dependent functions $J_i(\cdot)$ and $g_i(\cdot)$ represent changing dynamical properties of the joint due e.g. to the motion of the other joints. $J_i(t)$ represents the inertial (dynamic) properties (e.g. an effective moment of inertia as seen at the rotor of the i-th d–c motor) and $g_i(t)$ represents e.g. viscous friction and the other effects. In the case when one type of the effects (inertial or velocity) dominate the other it is possible to consider further simplified models of the joint, namely

$$\text{a)}\quad J_i(t)\,\ddot{\psi}_i = u_i(t) \tag{4}$$

or

$$\text{b)}\quad g_i(t)\,\dot{\psi}_i = u_i(t) \tag{5}$$

In particular the "conceptual" model as given by eqn. (4) will be useful for joints driven by d–c motors (especially for large-size robots like IRb-60 and for fast motions) and the model given by eqn. (5) can be used for hydraulic robots driven by powerful hydraulic actuators (like RIMP-1000 robot).

As far as a decentralized adaptive control law is concerned one may propose the following simple form of such law:

$$u_i(t) = K_{pi}(t)\,[r_i(t) - \psi_i(t)] - K_{vi}(t)\cdot\dot{\psi}_i(t), \tag{6}$$

where $K_{pi}(t)$ and $K_{vi}(t)$ are adjustable position and velocity gains. $K_{vi}(t)$ can be equal to zero in the case when velocity measurements are not directly available (especially in the case when the model (5) is an adequate one). Implementation of the control law (6) requires specification of rules for adjustment of the gains $K_{pi}(t)$ and $K_{vi}(t)$. Such rules are described in the next sections. Since further on we are going to consider each joint separately then the subscript index i is dropped for convenience.

3. Model reference adaptive controllers

In order to develop the rules according to which K_p and K_v are to be adjusted one can use the reference model, say

$$a\ddot{y}(t)+b\dot{y}(t)+y(t)=r(t) \tag{7}$$

describing desired dynamics of a closed-loop system. Let us assume that the joint dynamics may be expressed by eqn. (4). Then, using control law as given in eqn. (6) the closed-loop system is described by the following equation:

$$\alpha(t)\cdot\ddot{\psi}(t)+\beta(t)\dot{\psi}(t)+\psi(t)=r(t), \tag{8}$$

where

$$\alpha(t)=J(t)/K_p(t), \quad \beta(t)=K_v(t)/K_p(t).$$

The error $e(t)=y(t)-\psi(t)$ between the output of the reference model and the joint coordinate $\psi(t)$ is then given by an equation:

$$\dot{\xi}(t)=A_M\xi(t)+d\cdot\omega^T(t)\cdot[J^{-1}(t)K(t)-\varphi_0], \tag{9}$$

where

$$\xi(t)=[e(t),\dot{e}(t)]^T, \quad K(t)=[K_p(t),K_v(t)]^T$$

and

$$\omega(t)=\begin{bmatrix}\psi(t)-r(t)\\ \dot{\psi}(t)\end{bmatrix}, \varphi_0=\begin{bmatrix}a^{-1}\\ ba^{-1}\end{bmatrix}, d=\begin{bmatrix}0\\ 1\end{bmatrix}, A_M=\begin{bmatrix}0 & 1\\ -a^{-1} & -ba^{-1}\end{bmatrix}$$

Dubovski and Des Forges (1979) introduce the error function to be minimized as:

$$f(e,\dot{e})=\frac{1}{2}(q_0e+q_1\dot{e})^2, \quad q_0,q_1>0 \tag{10}$$

and propose the following relations to be achieved:

$$\dot{\alpha}=-\frac{\partial f}{\partial\alpha}\simeq\frac{\partial f}{\partial a}, \dot{\beta}=-\frac{\partial f}{\partial\beta}\simeq\frac{\partial f}{\partial b} \tag{11}$$

— assuming that $\alpha\simeq a$ and $\beta\simeq b$. Then it is also assumed that $J(t)$ changes "slowly" in comparison to $\alpha(t)$ and $K(t)$. This leads to the following rules for the adjustment of K_p and K_v

$$\begin{aligned}\dot{K}_p&=-\frac{K_p}{a}(q_0e+q_1\dot{e})(q_0u+q_1\dot{u})\\ \dot{K}_v&=[K_p(q_0w+q_1\dot{w})-\frac{K_v}{a}(q_0u+q_1\dot{u})](q_0e+q_1\dot{e})\end{aligned} \tag{12}$$

where

$$u=\partial e/\partial a \quad \text{and} \quad a\ddot{u}+b\dot{u}+u=-\ddot{y}$$
$$w=\partial e/\partial b \quad \text{and} \quad a\ddot{w}+b\dot{w}+w=-\dot{y} \tag{13}$$

In the above algorithm the sensitivity functions u, w do not depend on the actual behaviour of $\psi(t)$. Therefore if the manipulator dynamics change suddenly (e.q. when the manipulator picks up some heavy object) then the functions u and w will still remain the same. The alternative approach to introduce the gain adjustment rule could be as follows:

Let us write the error eqn. (9) in the form

$$\dot{\xi}(t)=A_M\cdot\xi(t)+d\cdot\omega^T(t)\cdot z(t), \quad \text{where} \quad z(t)=J^{-1}(t)\cdot K(t)-\varphi_0 \tag{14}$$

If we define a sensitivity matrix Λ as:

$$\Lambda=\frac{\partial\xi}{\partial z}, \quad \Lambda\in R^{2\times 2} \tag{15}$$

then the following equation can be obtained

$$\dot{\Lambda}(t)=A_M\Lambda(t)+d\cdot\omega^T(t). \tag{16}$$

Thus $\Lambda(t)$ can be computed using either analog or digital facilities. Now we can consider a function V of the error ξ in the form $V=(1/2)\xi^T P\xi$, where $P=P^T>0$, and to propose the fulfilment of the following relationship:

$$\dot{\mathbf{z}}(t)=-c\Lambda^T\cdot\frac{\partial V}{\partial\xi}, \quad \text{where} \quad c>0$$

i.e.

$$\dot{\mathbf{z}}(t)=-c\Lambda^T\mathbf{P}\,\xi \tag{17}$$

From eqn. (14) it results that

$$\dot{\mathbf{z}}(t)=J^{-1}(t)\cdot\dot{\mathbf{K}}(t)+J^{-2}(t)\,[-\dot{J}(t)]\,K(t) \tag{18}$$

If we assume that $\dot{J}(t)$ is very small, i.e., that the second term in eqn. (18) is much smaller than the first one and we also assume that $z(t)=J^{-1}(t)\cdot K(t)-\varphi_0\simeq 0$ (i.e. the controlled system tracks the model closely), then we may obtain from (14), (17) and (18) the following adjustment rule:

$$\dot{K}=-c\,\|\varphi_0\|^{-2}\,(K^T\cdot\varphi_0)\cdot\Lambda^T\cdot P\cdot\xi \tag{19}$$

The properties of the above adaptive scheme are being investigated at present. An alternative way to specify the adaptive rule could be to consider the error function $V=(1/2)\,(\xi^T P\xi+z^T z)$. and to try to achieve the condition $\dot{\mathbf{V}}(t)<0$. Using similar assumption as above we can obtain the following rule:

$$\dot{\mathbf{K}}=-c\,\|\varphi_0\|^{-2}\,(K^T\varphi_0)\cdot\omega\cdot d^T\cdot P\cdot\xi, \quad c>1 \tag{20}$$

In fact the above rule is somewhat similar to the adjustment rule as proposed by Narendra et. al. (1978, 1980 a, b, c). If we consider the case when $J(t)=J_0=\text{const}>0$, then one can use the following gain adjustment rule

$$\dot{K}=-\Gamma\cdot\omega\cdot h^T\xi \tag{21}$$

where $\Gamma=\Gamma^T>0$ and h is a constant vector. Assuming that the transfer function

$$G(s)=h^T[sI-A_M]^{-1}d$$

is strictly positive real one can prove that $\xi \underset{t\to\infty}{\to} 0$. In order to do this it is sufficient to consider the Lyapunov function $V(\xi, z)=(1/2)(\xi^T P\xi+z^T(J_0\Gamma^{-1})z)$ and to use the Kalman-Yakubovich lemma.

It should be noted that in case of a real importance, that is when $J(t)\neq\text{const}$, the stability issue is much more complicated. Some attempts to investigate the stability of adaptive schemes with the gain adjustment rules like (20), (21) (in the case when $J(t)\neq\text{const}$) have been made in [Malinowski and Masłowski 1987]. Let us mention finally that it is possible to consider an adaptive decentralized control law (eqn. (6)) augmented with an integral feedback. In such case the gain adjustment rules have to be accordingly changed.

4. Simple adaptive sampled-data controller for a hydraulic robot

The above model reference adaptive controllers were developed essentially under the assumption of an analog or a very fast sampled-data implementation. Also the simple model in the form as given by eqn. (4) has been used. Now we consider the case when the term $g(t)\cdot\dot{\psi}(t)$ is dominating in the left hand side of eqn. (3), i.e. when eqn. (5) can be used. Such representation of the joint dynamics was found adequate for the hydraulic robot RIMP-1000 [Kuzan and Pilat 1984]. For this robot a sampled-data controller has to be used — with direct measurements of joint positions being provided from n-coders at every sampling time. The microprocessor based controller hardware make it possible to introduce a sampling interval Δt of several miliseconds. In the particular adaptive controller design the interval $\Delta t=50$ ms was found to be satisfactory. At the same time the evaluation of the parameters of each joint model of the form (3) has shown that in every possible situation the dynamics were very fast ("time constants" were at most 1–2 ms) and that the term $g(t)\,\dot{\psi}(t)$ was dominating in eqn. (3) with $g_i(t)$ changing significantly along typical trajectories of motion.

The proposed adaptive control laws take into account the sampled data ($\psi(t)$) in the following ways:
case (i)

$$u(t+\tau)=K_p(t)\cdot[V_1+2(V_2-V_1)\cdot\tau/\Delta t] \quad \text{for} \quad 0<\tau\leqslant\frac{\Delta t}{2} \tag{22a}$$

and

$$u(t+\tau)=K_p(t)\cdot\left[V_2+2(V_3-V_2)\left(\frac{\tau}{\Delta t}-\frac{\Delta t}{2}\right)\right] \quad \text{for} \quad \frac{\Delta t}{2}<\tau\leqslant\Delta t \tag{22b}$$

where

$$V_1=\dot{\psi}(t)$$

$$V_3=\frac{r(t+2\Delta t)-\psi(t)}{2\Delta t}$$

$$V_2=\frac{2[r(t+\Delta t)-\psi(t)]}{\Delta t}-\frac{V_1+V_3}{2}$$

The basic idea of the algorithm (22a, b) is depicted in Fig. 1. The objective of the control law is to provide for smooth velocity changes while meeting the basic

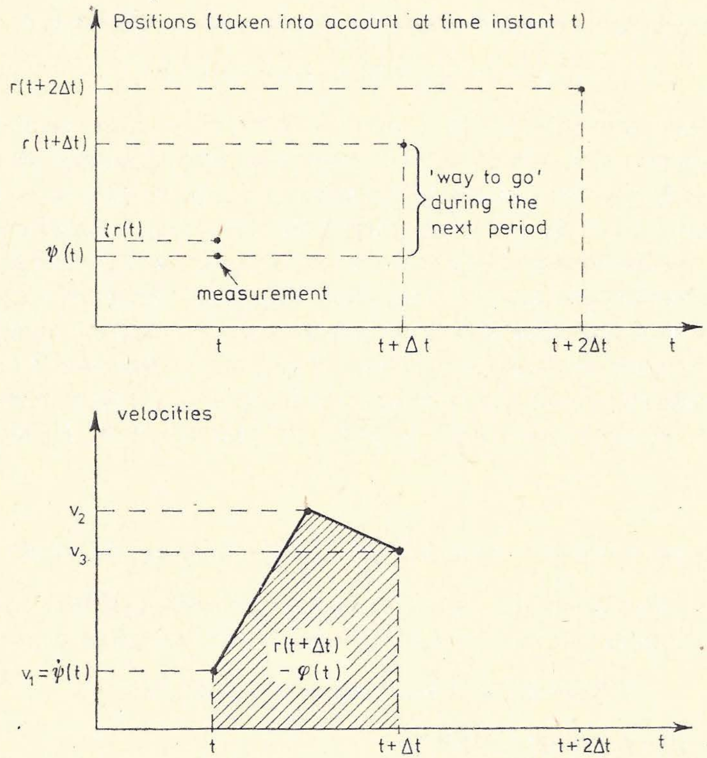

Fig. 1

target of reaching the position $r(t+\Delta t)$ at time $t+\Delta t$. The velocity V_3 to be reached at $t+\Delta t$ is an average velocity required over next two sampling intervals.

The adaptive gain $K_{pi}(t)$ is adjusted according to the following simple law:

$$K_p(t+\Delta t)=K_p(t)+\gamma K_p(t)\cdot\operatorname{sgn}\varepsilon(t), \tag{23}$$

where

$$\varepsilon(t)=r(t)-\psi(t), \qquad \gamma>0.$$

Application of the above algorithm (with suitably chosen γ) was found to be very satisfactory during several simulations of movement of a robotic arm — using very accurate dynamic model of the robot for simulation purposes. The parameters of the model were significantly changed during simulations (e.g. oil pressure was suddenly decreased by 20%). Yet, it has appeared that the errors $\varepsilon(t)$ at sampling times very rarely exceeded one bit error (at the lowest n-coder position).

It should be noted, however, that the control law (22a, b) is quite complicated for real time microprocessor-based implementation.

case (ii)

$$u(t+\tau)=K_p(t)\cdot\frac{r(t+\Delta t)-\psi(t)}{t}, \; 0<\tau\leqslant\Delta t \tag{24}$$

The adaptive gain $K_p(t)$ is adjusted according to the same simple law as in case (i), i.e

$$K_p(t+\Delta t)=K_p(t)+\gamma\cdot K_p(t)\cdot \operatorname{sgn}\varepsilon(t)$$

The application of this algorithm was also found to be quite successful during several simulations of movement of a robotic arm. The parameters of the model were changed during the simulations in the same way as in the case (i) above and it has appeared that errors $\varepsilon(t)$ at sampling times very rarely exceeded one bit error at the lowest n-coder position. In fact the results obtained were as good as in case (i) and the practical implementation of algorithm (24) should be much easier than the implementation of algorithm (22). It should be observed, that the implementation of the adaptive controller (24, 23) requires still very fast ("instantenous" computing of $u(t+\tau)$ (input current to a servo-valve of each hydraulic actuator) from eqn. (24) and between sampling times there is plenty of time to adjust K_p according to eqn. (23).

case (iii)

$$u(t+\tau)=u^{(1)}(t) \text{ if } l\cdot\Delta\psi\leqslant r(t+\Delta t)-\psi(t)<(l+1)\,\Delta\psi, \; 0<\tau\leqslant\Delta t \tag{25}$$

In the control law above $\Delta\psi$ denotes the sensor resolution, $\left(u^{(1)}(t), \ldots, u^{(l)}(t), \ldots, u^{(l_{\max})}(t)\right)$ is the table of control values adjusted at each sampling time as follows:

$$u^{(l)}(t+\Delta t)=u^{(l)}+\alpha u^{(l)}\operatorname{sgn}\left(r(t)-\psi(t)\right), \; \alpha>0 \tag{26}$$

if $\quad l\cdot\Delta\psi\leqslant r(t)-\psi(t-\Delta t)<(1+l)\,\Delta\psi$

and $\quad u^{(l)}(t+\Delta t)=u^{(l)}(t) \quad$ otherwise.

The above algorithm (25) is very convenient for a practical implementation. It also allows for nonlinear dependence of $u(t+\tau)$ on $r(t+\Delta t)-\psi(t)$. On the other hand the adjustment rule (eq. (26)) is rather "slow" (updating only one value $u^{(l)}$ at at time). Yet, the algorithm was also found to be very satisfactory and has been used for a practical implementation. The simulation results are given in [Kuzan and Pilat 1984].

References

[1] Balestrino A., De Maria G., Sciavicco L. An adaptive model following control for robotic manipulators, *ASME Trans. Journ. of Dynamic Systems, Measurement and Control*, September 1983.

[2] Dubovski S., Des Forges D. T. The application of model referenced adaptive control to robotic manipulators. *ASME Trans. Journ. of Dynamic Systems, Measurement and Control*, September 1979.

[3] Kuzan P., Pilat Z. Dynamical properties of robot RIMP-1000 and design of continuous path controllers (in Polish) MSc thesis, Inst. of Automatic Control, Techn. Univ. of Warsaw, 1984.

[4] Malinowski K., Masłowski P. On model reference adaptive controllers for a robotic arm. 1987 (In preparation).

[5] Narendra K. S., Valavani L. S. Stable adaptive controller design-direct control. *IEEE Trans. on Aut, Control.*, August 1978.

[6] Narendra K. S., Lin Y-H., Valavani L. S. Stable adaptive controller design — proof of stability. *IEEE Trans on Aut. Contr.*, June 1980.

[7] Narendra K. S., Lin Y-H. A new error model for adaptive systems. *IEEE Trans. on Aut. Contr.*, June 1980.

[8] Narendra K. S., Valavani L. S. A comparison of Lyapunov and hyperstability approaches to adaptive control of continuous systems. *IEEE Trans. on Aut. Contr.*, April 1980.

[9] Paul R. P. Robot manipulators. MIT Press, 1981.

[10] Slotine J-J. E. Sliding controller design for non-linear systems. *Int. Journ. on Contr.*, **40** (1984) 2.

[11] Stokič D., Vukobratovič M. Practical stabilization of robotic systems by decentralized control. *Automatica* (GB), **20** (1984) 3.

Zdecentralizowane adaptacyjne sterowanie manipulatorów ramienia robota

Praca prezentuje i omawia zastosowanie sterownika adaptacyjnego do sterowania ciągłego CPC ramieniem robota. Skoncentrowano się na sterowaniu zdecentralizowanym ze względu na jego znaczenie praktyczne. Wynika ono z prostoty i łączącej się z tym możliwości sterowania w czasie rzeczywistym. W pracy zaprezentowano kilka możliwych sposobów sterowania bazujących na prostych modelach dynamiki. Omówiono uzyskane wyniki symulacyjne oraz możliwości zastosowania do sterowania robota przemysłowego RIMP-1000.

Децентрализированное адаптивное управление манипуляторов рычага робота

В работе рассматривается использование адаптивного командо-контроллера для непрерывного управления CPC рычагом робота. Основное внимание обращено на децентрализированное управление ввиду его практического значения. Это вытекает из простоты и связанной с этим возможности управления в реальном масштабе времени. В работе представлено несколько возможных способов управления, базирующих на простых моделях динамики. Рассмотрены полученные результаты моделирования, а также возможности использования управления промышленным роботом RIMP-1000.