

POSITION/FORCE CONTROL OF MANIPULATOR IN CONTACT WITH FLEXIBLE ENVIRONMENT

Piotr GIERLAK*

*Faculty of Mechanical Engineering and Aeronautics, Department of Applied Mechanics and Robotics
Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland

pgierlak@prz.edu.pl

received 26 June 2018, revised 3 March 2019, accepted 7 March 2019

Abstract: The paper presents the issue position/force control of a manipulator in contact with the flexible environment. It consists of the realisation of manipulator end-effector motion on the environment surface with the simultaneous appliance of desired pressure on the surface. The paper considers the case of a flexible environment when its deformation occurs under the pressure, which has a significant influence on the control purpose realisation. The article presents the model of the controlled system and the problem of tracking control with the use of neural networks. The control algorithm includes contact surface flexibility in order to improve control quality. The article presents the results of numerical simulations, which indicate the correctness of the applied control law.

Key words: Robotics, robot control, nonlinear control systems

1. INTRODUCTION

The number of industrial applications, in which robots come in contact with the environment and where it has a significant influence on the robotised processes realisation quality, is constantly increasing (Birglen and Schlicht, 2018; Denkena et al. 2017; Hashemiet et al., 2013; Iglesias et al., 2015; Lotz et al., 2014; Mendes and Neto, 2015). The aforementioned processes include, among others, robotised mechanical processing such as grinding (Zhu et al., 2015), polishing (Gracia et al., 2018; Tian et al., 2016), or edge deburring (Burghardt et al., 2017b). Therefore, modelling and control of robots in interaction with the environment becomes crucial, particularly if the actual features of the environment are taken into account, such as flexibility or damping. Manipulator-environment system modelling is a complex issue, and the weakest point of the model is partly connected with the environment and model of manipulator's contact with the surface. A significant difficulty in environment modelling is its changeability and the lack of certainty or knowledge of its parameters, resulting from the geometrical complexity of structures with which the manipulator interacts, related to changing stiffness, damping and mass balance in various parts of the structure. Moreover, environment surface shape may be not known in detail, which increases system description uncertainty (Burghardt et al., 2017a; Capisani and Ferrara, 2012; Duan et al., 2018; Jafari and Ryu, 2016; Pliego-Jiménez and Arteaga-Pérez, 2015; Ravandi et al., 2018).

The paper presents the issue of manipulator position/force control in contact with an environment, taking into consideration its crucial feature, which is flexibility. The issue of position/force control itself consists of the realisation of manipulator end-effector motion on the environment surface with the simultaneous appliance of desired pressure on the surface (Gierlak, 2014; Hendzel et al., 2014). End-effector motion speed and pressure force result from technological process parameters. In the case of a deforma-

ble environment, its deformation occurs under pressure, which has a significant influence on control purpose realisation. Thus, in order to improve control quality, control algorithm takes into consideration the contact surface flexibility. The paper used artificial neural networks (Gierlak, 2012; Żylski and Gierlak, 2010) to realise control compensating system non-linearity, thanks to which, it is not necessary to know the structure and the parameters of the manipulator and environment models.

2. SYSTEM DYNAMICS

Manipulator dynamics expressed via configuration coordinates have the following formula:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \xi(t) = u + J(q)^T \lambda, \quad (1)$$

where: $q \in R^n$ – the vector of generalised coordinates, $M(q) \in R^{n \times n}$ – the inertia matrix, $C(q, \dot{q})\dot{q} \in R^{n \times n}$ – the vector of centrifugal and Coriolis forces, $F(\dot{q}) \in R^n$ – the friction vector, $G(q) \in R^n$ – the gravity vector, $\xi(t) \in R^n$ – the vector of disturbances bounded by $\|\xi\| \leq b$, $b > 0$, $u \in R^n$ – the control input vector, $J(q) \in R^{m \times n}$ – an analytical Jacobian matrix, $\lambda \in R^m$ – an interaction force vector expressed in the task space, n – the number of degrees of freedom of the manipulator, m – a workspace (task space) dimension.

Due to the control purpose, which is the manipulator end-effector motion on the environment surface with simultaneously influencing it with normal force, it is convenient to present manipulator dynamics in a task coordinate system related to the environmental surface. Cartesian coordinates c were selected as task coordinates, which are related to the configuration coordinates in the following way:

$$c = k(q) \in R^m, \quad (2)$$

where: $k(q)$ is the so-called kinematics function. Jacobian $J(q)$ describing the transformation of forces and speeds between the task space and the configuration spaces results from the formula:

$$J(q) = \frac{\partial c}{\partial q} = \frac{\partial k(q)}{\partial q}. \quad (3)$$

On the basis of equations (1) and (2), manipulator dynamics were described in task coordinates (Gierlak, 2018; Gierlak and Szuster, 2017)

$$\begin{aligned} A(q)\ddot{c} + H(q, \dot{q})\dot{c} + B(q, \dot{q}) + G(q, \dot{q}) + \Psi(q, t) \\ = U + \lambda. \end{aligned} \quad (4)$$

where specific vectors and matrices are as follows:

$$\left. \begin{aligned} J^{-T}M(q)J^{-1} &= A(q) \in R^{m \times m} \\ J^{-T}C(q, \dot{q})J^{-1} - J^{-T}M(q)J^{-1}J\dot{J}^{-1} &= H(q, \dot{q}) \in R^{m \times m} \\ J^{-T}(F(\dot{q}) + G(q)) &= B(q, \dot{q}) \in R^m \\ J^{-T}\xi(t) &= \Psi(q, t) \in R^m \\ J^{-T}u &= U \in R^m \end{aligned} \right\} \quad (5)$$

and the expression $J^{-T} = (J^{-1})^T = (J^T)^{-1}$. In the case of $m \neq n$, the Moore-Penrose pseudo-inverse J^+ instead of J^{-1} should be used (Barata and Hussein, 2012). Task space $\{C\}$ was divided into r -dimensional normal subspace $\{N\}$ and $(m-r)$ -dimensional tangent subspace $\{T\}$: $\{C\} = \{T\} \oplus \{N\}$ (Vukobratović et al., 2002). It leads to the decomposition of vector c into a part related to the tangent directions $c_\tau \in R^{m-r}$ and a part related to normal directions $c_n \in R^r$ so $c = [c_\tau^T \ c_n^T]^T$. Thus, it is now convenient to determine the environment features that are different on tangent and normal directions. For normal directions, the most important feature is flexibility, which is presented by the formula:

$$K_e c_n = F_{en}, \quad (6)$$

where: $F_{en} \in R^r$ – a normal force vector, $K_e \in R^{r \times r}$ – diagonal matrix of environment stiffness, so $K_e = K_e^T > 0$. On tangent directions, manipulator end-effector motion is hindered by resisting force $F_{e\tau}$, which may be modelled or compensated for on the basis of force detector measurements. Such an approach of interaction between the manipulator and the environment allows expressing interaction forces vector as $\lambda = [F_{e\tau}^T \ F_{en}^T]^T$.

Equation (6) was transformed into a form $c_n = P_e F_{en}$, in which $P_e = K_e^{-1} \in R^{r \times r}$ is flexibility matrix; so, $P_e = P_e^T > 0$. By adding decomposed vector c (taking into consideration $c_n = P_e F_{en}$) to equation (4), the dynamic equation of the system motion equation takes on the following form:

$$A(q)E\ddot{\theta} + H(q, \dot{q})E\dot{\theta} + B(q, \dot{q}) + \Psi(q, t) = U + \lambda, \quad (7)$$

where vector describing system state $\theta = [c_\tau^T \ F_{en}^T]^T \in R^m$ consists of position and force variables, while matrix E has the following form:

$$E = \begin{bmatrix} I_{(m-r) \times (m-r)} & 0 \\ 0 & P_e \end{bmatrix} \in R^{m \times m}, \quad (8)$$

and presents information on environment flexibility. Description (7) with vector θ is a more natural description of dynamics due to control purposes than equation (4), since the purpose of control of the presented system is to realise motion on tangent directions and to realise forces on normal directions.

3. TRACKING CONTROL

The purpose of control is to implement the desired trajectory $\theta_d(t) \in R^m$, $\dot{\theta}_d(t)$, $\ddot{\theta}_d(t)$, which consists of the trajectory of motion in the tangent plane $c_{\tau d}(t) \in R^{m-r}$, $\dot{c}_{\tau d}(t)$, $\ddot{c}_{\tau d}(t)$ and the trajectory of force on normal directions $F_{end}(t) \in R^r$, $\dot{F}_{end}(t)$, $\ddot{F}_{end}(t)$. So, it can be noted in the following way:

$$\theta_d = \begin{bmatrix} c_{\tau d} \\ F_{end} \end{bmatrix}, \dot{\theta}_d = \begin{bmatrix} \dot{c}_{\tau d} \\ \dot{F}_{end} \end{bmatrix}, \ddot{\theta}_d = \begin{bmatrix} \ddot{c}_{\tau d} \\ \ddot{F}_{end} \end{bmatrix}. \quad (9)$$

Tracking control was defined as an issue of stabilising tracking error, defined as:

$$\tilde{\theta} = \theta_d - \theta, \quad (10)$$

that is, regarding both motion error and force error stabilisation. In accordance with the classic theory of tracking control of nonlinear systems, a filtered tracking error was defined:

$$s = \dot{\tilde{\theta}} + \Lambda \tilde{\theta}, \quad (11)$$

which is a linear combination of tracking error and its derivative. In equation (11), project matrix $\Lambda \in R^m$ occurs, which fulfils the condition $\Lambda = \Lambda^T > 0$. By adding equation (11) to formula (7), the description of the dynamics in the filtered tracking error function:

$$A(q)E\dot{s} = -H(q, \dot{q})Es + A(q)E\dot{v} + H(q, \dot{q})Ev + B(q, \dot{q}) + \Psi(q, t) - U - \lambda, \quad (12)$$

where ancillary variable $v = \dot{\theta}_d + \Lambda \tilde{\theta}$ occurs. Nonlinear part of equation (12) was marked as:

$$A(q)E\dot{v} + H(q, \dot{q})Ev + B(q, \dot{q}) = f, \quad (13)$$

where $f \in R^m$ is a function that depends both on the manipulator model and the environment. It particularly depends on the unknown manipulator parameters and unknown environment stiffness. Thus, the final form of the dynamic system motion description is:

$$A(q)E\dot{s} = -H(q, \dot{q})Es + f + \Psi(q, t) - U - \lambda. \quad (14)$$

For this system, the control consists of conventional PD regulator, control compensating for system nonlinearity $\hat{f} \in R^m$, and control compensating for the interaction forces influence λ :

$$U = K_D s + \hat{f} - \lambda - r. \quad (15)$$

In the established control, function \hat{f} approximates f , and expression $K_D s$ is a form of PD control where $K_D \in R^{m \times m}$ is an amplification matrix that $K_D = K_D^T > 0$, while $r \in R^m$ is a robust control. A part of equation (15) expresses forces of interaction between the manipulator and the environment in all directions. It is assumed that its value is assessable via measurement and can be put in the control signal. Assessable via measurement are also vectors of angular displacements q and angular speeds \dot{q} , which, knowing kinematic equations (2), allow calculating motion parameters in task coordinates c . It is necessary to establish filtered tracking error s and ancillary signal v occurring in function \hat{f} . In order to implement compensation of nonlinear function f , it was decomposed into two parts: one $f_\tau \in R^{(m-r)}$ for tangent directions, and the other $f_n \in R^r$ for normal directions. Neural network linear in regards to RVFL weights were used to approximate the member functions. This type of neural network is a universal approximator (Galushkin, 2007; Pao et al., 1994), which, due to relatively simple structure, is frequently used in robot control systems for nonlinearity compensation (Kumar et al., 2011). If the selected network is a network with a set layer of input weights,

one hidden layer, and input weights layer undergoing adaptation, then nonlinear member functions may be noted as outputs from the ideal neural networks with limited approximation errors:

$$f = \begin{bmatrix} f_\tau \\ f_n \end{bmatrix} = \begin{bmatrix} W_\tau^T \Phi_\tau(x_\tau) + \varepsilon_\tau(x_\tau) \\ W_{n1}^T \Phi_{n1}(x_{n1}) + \varepsilon_{n1}(x_{n1}) \\ \vdots \\ W_{ni}^T \Phi_{ni}(x_{ni}) + \varepsilon_{ni}(x_{ni}) \\ \vdots \\ W_{nr}^T \Phi_{nr}(x_{nr}) + \varepsilon_{nr}(x_{nr}) \end{bmatrix}, \quad (16)$$

where: x_τ, x_{ni} – network input signals vectors, W_τ, W_{ni} – ideal output weights matrices, $\Phi_\tau(\cdot), \Phi_{ni}(\cdot)$ are neurons activation functions vectors, and $\varepsilon_\tau, \varepsilon_{ni}$ are vectors of errors of function mapping by networks that $\|\varepsilon_\tau\| \leq \varepsilon_{b\tau}, \|\varepsilon_{ni}\| \leq \varepsilon_{bni}$, where $\varepsilon_{b\tau} > 0, \varepsilon_{bni} > 0$. If neurons activation functions are selected in the form of a basic functions' group, then the network with ideal limited weights has the feature of approximation of any function defined on a compact set with a finite number of discontinuity points (Hertz et al., 1991). Since network ideal weights are unknown, function estimate (16) should be used, in form of:

$$\hat{f} = \begin{bmatrix} \hat{f}_\tau \\ \hat{f}_n \end{bmatrix} = \begin{bmatrix} \hat{W}_\tau^T \Phi_\tau(x_\tau) \\ \hat{W}_{n1}^T \Phi_{n1}(x_{n1}) \\ \vdots \\ \hat{W}_{ni}^T \Phi_{ni}(x_{ni}) \\ \vdots \\ \hat{W}_{nr}^T \Phi_{nr}(x_{nr}) \end{bmatrix}, \quad (17)$$

where $\hat{W}_\tau, \hat{W}_{ni}$ are network ideal weights estimates. As equation (16) suggests, each nonlinear function f_n was decomposed into functions f_{ni} , each of which corresponds to the normal i -th direction and is approximated by a separate neural network. This action was performed due to the possibility of proving closed system stability for network weights' estimation rule not requiring knowledge of environment elasticity.

Taking into account control rule (15), as well as equations (16) and (17) in the dynamics equation (14), the following closed system description was received:

$$A(q)E\dot{s} = -H(q, \dot{q})Es + \Psi(q, t) - K_D s + r + \begin{bmatrix} \hat{W}_\tau^T \Phi_\tau(x_\tau) \\ \hat{W}_{n1}^T \Phi_{n1}(x_{n1}) \\ \vdots \\ \hat{W}_{ni}^T \Phi_{ni}(x_{ni}) \\ \vdots \\ \hat{W}_{nr}^T \Phi_{nr}(x_{nr}) \end{bmatrix} + \begin{bmatrix} \varepsilon_\tau(x_\tau) \\ \varepsilon_{n1}(x_{n1}) \\ \vdots \\ \varepsilon_{ni}(x_{ni}) \\ \vdots \\ \varepsilon_{nr}(x_{nr}) \end{bmatrix}, \quad (18)$$

in which the interaction forces' influence was compensated for by the control, so that vector λ no longer occurs in equation (18). Closed-loop system dynamics is activated by the assessment errors of network and weights, which are as follows:

$$\tilde{W}_\tau = W_\tau - \hat{W}_\tau, \tilde{W}_{ni} = W_{ni} - \hat{W}_{ni}. \quad (19)$$

The errors are decreased as a result of the weights estimates adaptation, which occurs continuously during system operation. Weights adaptation rules were accepted in the following form (Lewis et al., 1995; Narendra and Annaswamy, 1987; Polycarpou and Ioannu, 1991):

$$\begin{cases} \dot{\hat{W}}_\tau = \Gamma_\tau \Phi_\tau(x_\tau) s_\tau^T - k_\tau \|s_\tau\| \Gamma_\tau \hat{W}_\tau \\ \dot{\hat{W}}_{ni} = \Gamma_{ni} \Phi_{ni}(x_{ni}) s_{ni} - k_{ni} |s_{ni}| \Gamma_{ni} \hat{W}_{ni} \end{cases}, \quad (20)$$

where: $\Gamma_\tau = \Gamma_\tau^T > 0, \Gamma_{ni} = \Gamma_{ni}^T > 0$ – adaptation amplification

matrices, $k_\tau > 0, k_{ni} > 0$ – constant project parameters, s_τ, s_{ni} – results from the decomposition of filtered tracking error $s = [s_\tau^T \ s_n^T]^T$, where $s_n = [s_{n1} \ \dots \ s_{ni} \ \dots \ s_{nr}]^T$. The second element on the right of formula (20) is responsible for 'attenuation' of weights estimates adaptation, while the 'attenuation' size is decided by the coefficients k_τ and k_{ni} . Assuming such rules of weights estimates actualisation guarantees their limitedness and protects the system from estimates 'drifting' and 'exploding', even without uniform system activation. When decomposing interruptions vector and robust control in the following way:

$\Psi(q, t) = [\Psi_\tau^T(q, t), \Psi_{n1}(q, t), \dots, \Psi_{ni}(q, t), \dots, \Psi_{nr}(q, t)]^T$ and $r = [r_\tau^T, r_{n1}, \dots, r_{ni}, \dots, r_{nr}]^T$, specific expressions of robust control were assumed as:

$$r_\tau = -\frac{K_\tau}{\|s_\tau\|} s_\tau, r_{ni} = -\frac{K_{ni}}{|s_{ni}|} s_{ni}, \quad (21)$$

where: $K_\tau > b_\tau \geq \|\Psi_\tau(q, t)\|, K_{ni} > b_{ni} \geq |\Psi_{ni}(q, t)|$.

Assumed neural networks control and weights adaptation rules guarantee the practical stability of the control system (Canudas et al., 1996). The proof of the stability is complex and was presented for a similar case by Gierlak and Szuster (2017) with regard to conditions given by Lewis et al. (1995), Narendra and Annaswamy (1987) and Polycarpou and Ioannu (1991).

4. NUMERICAL EXAMPLE

A two-link planar manipulator in contact with a flat surface was selected as an example presenting the work of control system of the manipulator in contact with a flexible environment (Fig. 1).

Two-link manipulator dynamics in articulated coordinates is described with an equation in the following form (1) of the following vectors and matrices (Gierlak and Szuster, 2017):

$$\left. \begin{aligned} M(q) &= \begin{bmatrix} a_1 & a_2 \cos(q_2 - q_1) \\ a_2 \cos(q_2 - q_1) & a_3 \end{bmatrix} \\ C(q, \dot{q}) &= \begin{bmatrix} 0 & -a_2 \sin(q_2 - q_1) \dot{q}_2 \\ a_2 \sin(q_2 - q_1) \dot{q}_1 & 0 \end{bmatrix} \\ F(\dot{q}) &= \begin{bmatrix} a_4 \dot{q}_1 + a_6 \operatorname{sgn}(\dot{q}_1) \\ a_5 \dot{q}_2 + a_7 \operatorname{sgn}(\dot{q}_2) \end{bmatrix} \\ G(q) &= \begin{bmatrix} a_8 \cos(q_1) \\ a_9 \cos(q_2) \end{bmatrix} \\ \xi(t) &= \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} \\ u &= \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ q &= \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \end{aligned} \right\} \quad (22)$$

Parameters characterising manipulator dynamics have the following significance:

$$\left. \begin{aligned} a_1 &= l_{c1}^2 m_1 + l_1^2 m_2 + I_1, \\ a_2 &= l_1 l_{c2} m_2, a_3 = l_{c2}^2 m_2 + I_2, \\ a_4 &= c_{v1}, a_5 = c_{v2}, a_6 = \kappa_1, a_7 = \kappa_2, \\ a_8 &= (l_{c1} m_1 + l_1 m_2) g, a_9 = l_{c2} m_2 g \end{aligned} \right\}. \quad (23)$$

where: m_i – a mass of i -th link, l_i – a length of i -th link, l_{ci} – the distance between centre of mass of i -th link and end of $i - 1$ link, I_i – a mass moment of inertia of i -th link relative to its centre of mass, c_{vi} – coefficient of viscous friction in i -th kinematic pair, κ_i – moment of force of dry friction in i -th kinematic pair.

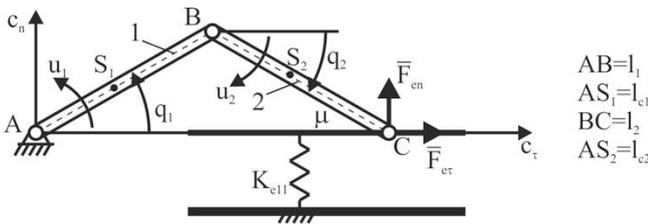


Fig. 1. The scheme of the 2-degrees-of-freedom manipulator with environment

Kinematic equation (2) describing manipulator end point position in the analysed case assume the following form:

$$c = \begin{bmatrix} c_\tau \\ c_n \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_2) \\ l_1 \sin(q_1) + l_2 \sin(q_2) \end{bmatrix} \quad (24)$$

Knowledge of kinematics equations allows determining Jacobian $J(q)$ and recording system dynamics in the task coordinates. On the basis of the definition (3), Jacobian $J(q)$ was determined as:

$$J(q) = \begin{bmatrix} -l_1 \sin(q_1) & -l_2 \sin(q_2) \\ l_1 \cos(q_1) & l_2 \cos(q_2) \end{bmatrix}. \quad (25)$$

This enabled the presentation of robot dynamics in the form of equation (4), in which matrices and vectors have the following form:

$$A(q) = \frac{1}{\sin^2(q_2 - q_1)} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad (26)$$

$$\left. \begin{aligned} A_{11} &= \frac{\cos^2(q_2)}{l_1^2} a_1 - \frac{2\cos(q_1)\cos(q_2)\cos(q_2 - q_1)}{l_1 l_2} a_2 + \frac{\cos^2(q_1)}{l_2^2} a_3 \\ A_{12} &= A_{21} = \frac{\sin(q_2)\cos(q_2)}{l_1^2} a_1 - \frac{2\sin(q_1 + q_2)\cos(q_2 - q_1)}{l_1 l_2} a_2 + \frac{\sin(q_1)\cos(q_1)}{l_2^2} a_3 \\ A_{22} &= \frac{\sin^2(q_2)}{l_1^2} a_1 - \frac{2\sin(q_1)\sin(q_2)\cos(q_2 - q_1)}{l_1 l_2} a_2 + \frac{\sin^2(q_1)}{l_2^2} a_3 \end{aligned} \right\} \quad (27)$$

$$H(q, \dot{q}) = \frac{1}{\sin^3(q_2 - q_1)} \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}, \quad (28)$$

$$\left. \begin{aligned} H_{11} &= \frac{g_1 \cos(q_2)}{l_1^2} a_1 + \frac{g_5 \cos(q_1) - g_6 \cos(q_2)}{l_1 l_2} a_2 + \frac{g_2 \cos(q_1)}{l_2^2} a_3 \\ H_{12} &= \frac{g_3 \cos(q_2)}{l_1^2} a_1 + \frac{g_5 \sin(q_1) - g_6 \sin(q_2)}{l_1 l_2} a_2 + \frac{g_4 \cos(q_1)}{l_2^2} a_3 \\ H_{21} &= \frac{g_1 \sin(q_2)}{l_1^2} a_1 + \frac{g_7 \cos(q_1) - g_8 \cos(q_2)}{l_1 l_2} a_2 + \frac{g_2 \sin(q_1)}{l_2^2} a_3 \\ H_{22} &= \frac{g_3 \sin(q_2)}{l_1^2} a_1 + \frac{g_7 \sin(q_1) - g_8 \sin(q_2)}{l_1 l_2} a_2 + \frac{g_4 \sin(q_1)}{l_2^2} a_3 \end{aligned} \right\} \quad (29)$$

$$\left. \begin{aligned} g_1 &= \cos(q_2)\cos(q_2 - q_1)\dot{q}_1 - \cos(q_1)\dot{q}_2 \\ g_2 &= \cos(q_2)\dot{q}_1 - \cos(q_1)\cos(q_2 - q_1)\dot{q}_2 \\ g_3 &= \sin(q_2)\cos(q_2 - q_1)\dot{q}_1 - \sin(q_1)\dot{q}_2 \\ g_4 &= \sin(q_2)\dot{q}_1 - \sin(q_1)\cos(q_2 - q_1)\dot{q}_2 \\ g_5 &= [\cos(q_2) + \cos(q_1)\cos(q_2 - q_1)]\dot{q}_2 \\ g_6 &= [\cos(q_1) + \cos(q_2)\cos(q_2 - q_1)]\dot{q}_1 \\ g_7 &= [\sin(q_2) + \sin(q_1)\cos(q_2 - q_1)]\dot{q}_2 \\ g_8 &= [\sin(q_1) + \sin(q_2)\cos(q_2 - q_1)]\dot{q}_1 \end{aligned} \right\}, \quad (30)$$

$$B(q, \dot{q}) = \begin{bmatrix} \frac{l_2 \cos(q_2)[a_4 \dot{q}_1 + a_6 \cos(q_1)] - l_1 \cos(q_1)[a_5 \dot{q}_2 + a_7 \cos(q_2)]}{l_1 l_2 \sin(q_2 - q_1)} \\ \frac{l_2 \sin(q_2)[a_4 \dot{q}_1 + a_6 \cos(q_1)] - l_1 \sin(q_1)[a_5 \dot{q}_2 + a_7 \cos(q_2)]}{l_1 l_2 \sin(q_2 - q_1)} \end{bmatrix} \quad (31)$$

$$U = \begin{bmatrix} \frac{l_2 \cos(q_2)u_1 - l_1 \cos(q_1)u_2(t)}{l_1 l_2 \sin(q_2 - q_1)} \\ \frac{l_2 \sin(q_2)u_1(t) - l_1 \sin(q_1)u_2(t)}{l_1 l_2 \sin(q_2 - q_1)} \end{bmatrix}, \quad (32)$$

$$\Psi(q, t) = \begin{bmatrix} \frac{l_2 \cos(q_2)\xi_1(t) - l_1 \cos(q_1)\xi_2(t)}{l_1 l_2 \sin(q_2 - q_1)} \\ \frac{l_2 \sin(q_2)\xi_1(t) - l_1 \sin(q_1)\xi_2(t)}{l_1 l_2 \sin(q_2 - q_1)} \end{bmatrix}. \quad (33)$$

In the analysed case, constant disturbance values were assumed as $\xi_1(t) = \xi_2(t) = \xi = const$. Forces of interaction between the manipulator and the environment are assumed as:

$$\lambda = \begin{bmatrix} F_{e\tau} \\ F_{en} \end{bmatrix} = \begin{bmatrix} \mu F_{en} \text{sgn}(\dot{c}_\tau) \\ K_e c_n \end{bmatrix}. \quad (34)$$

In the analysed example, the contact surface is one-dimensional, thus, there is only one tangent direction, along which the end-effector motion can occur. The flexibility of the surface also occurs only in one normal direction. Therefore, the matrix E has a form:

$$E = \begin{bmatrix} 1 & 0 \\ 0 & P_e \end{bmatrix}, \quad (35)$$

where $P_e = K_e^{-1}$.

The control law equation (15) contains the interaction forces vector λ defined by the formula (34) and the gain matrix in the form:

$$K_D = \begin{bmatrix} K_{D\tau} & 0 \\ 0 & K_{Dn} \end{bmatrix}. \quad (36)$$

The filtered tracking error s taking into account the diagonal matrix $\Lambda = \text{diag}\{\Lambda_\tau, \Lambda_n\}$, has the form:

$$s = \begin{bmatrix} s_\tau \\ s_n \end{bmatrix} = \begin{bmatrix} \dot{\tilde{c}}_\tau + \Lambda_\tau \tilde{c}_\tau \\ \dot{\tilde{F}}_{en} + \Lambda_n \tilde{F}_{en} \end{bmatrix}, \quad (37)$$

where \tilde{c}_τ – the position error in the tangent direction, \tilde{F}_{en} – the force error in the normal direction. The components of the robust control vector determined by equation (21) have the form:

$$r_\tau = -\frac{K_\tau}{|s_\tau|} s_\tau, r_n = -\frac{K_n}{|s_n|} s_n. \quad (38)$$

The most complex element of the control law (15) is the compensative control \hat{f} , the structure for which results from the applied neural network. Separate neural networks were used for the compensative control in the tangent and normal directions. The structure of the neural network that generates the output \hat{f}_τ is shown in Fig. 2.

The input-output dependency of the presented neural network has the form:

$$\hat{f}_\tau = \hat{W}_\tau^T \Phi_\tau(x_\tau) = \hat{W}_\tau^T S_\tau (V_\tau^T x_\tau) \quad (39)$$

where: $\Phi_\tau(x_\tau) = S_\tau (V_\tau^T x_\tau)$ – the output vector from the hidden layer, $S_\tau = [S_{\tau 1}, S_{\tau 2}, \dots, S_{\tau N}]^T$ – the vector of neuron activation function, $x_\tau = [x_{\tau 1}, x_{\tau 2}, \dots, x_{\tau M}]^T$ – the input vector, V_τ^T – the input weights matrix, in which the values of weights of connection of inputs with the hidden layer were grouped, \hat{W}_τ^T – estimation of the output weight matrix, which contains the values of the weights of connections of the hidden layer with the network output \hat{f}_τ . Elements of this matrix are adapted during the system operation. An analogous structure of the neural network was used to generate the compensative control \hat{f}_n – in the description of this net-

work, the denotation n should be used instead of τ . The input vectors for both neural networks have the same form, because the nonlinearities in the tangent and normal directions depend on the same signals, thus: $x_\tau = x_n = [q_1, q_2, \dot{q}_1, \dot{q}_2, v_\tau, v_n, \dot{v}_\tau, \dot{v}_n]^T$, where v_τ and v_n are elements of the auxiliary signal $v = [v_\tau, v_n]^T$, which was introduced in equations (12) and (13). The neuron activation functions were selected as sigmoidal bipolar functions described by the equation:

$$S_{\tau i} = \frac{2}{1 + \exp(-\beta z)} - 1, \quad (40)$$

where: $\beta > 0$ – the steepness factor of the function, z – the input value to the neuron.

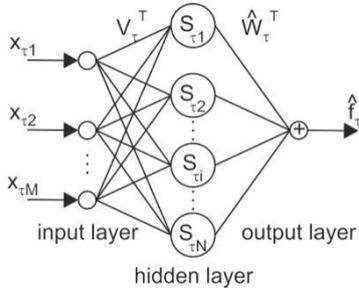


Fig. 2. Structure of neural network

The applied control law (15) leads to the description of a closed system (with feedback) in form (18), which, in the analysed example, takes the form:

$$A(q)E\dot{s} = -H(q, \dot{q})Es + \Psi(q, t) - K_D s + r + \begin{bmatrix} \hat{W}_\tau^T \Phi_\tau(x_\tau) \\ \hat{W}_n^T \Phi_n(x_n) \end{bmatrix} + \begin{bmatrix} \varepsilon_\tau(x_\tau) \\ \varepsilon_n(x_n) \end{bmatrix}. \quad (41)$$

The last two terms on the right side of the equation (41) require explanation. The terms $\hat{W}_\tau^T \Phi_\tau(x_\tau)$ and $\hat{W}_n^T \Phi_n(x_n)$ result from the inaccurate approximation of system nonlinearity by neural networks, which results from errors in the adaptation of the network weights \hat{W}_τ and \hat{W}_n . These adaptation errors are not known explicitly, because the ideal weights of neural networks are unknown. The last part results from the inaccuracy of approximation of nonlinear functions even by ideal neural networks that have finite mapping accuracy. This term is also not known in the open form. The last two expressions together with $\Psi(q, t)$ are understood as disturbances of the closed system and they constitute an excitation of error s .

The desired trajectory of motion was adopted in accordance with the following equation:

$$c_{\tau d} = c_{\tau d}(0) + \frac{v_{max}}{w_\tau} \sum_{i=1}^3 \ln \left(\frac{\left(\frac{1}{e^{-w_\tau(t-t_{c1i})} + 1} \right) \left(\frac{1}{e^{-w_\tau(t-t_{c4i})} + 1} \right)}{\left(\frac{1}{e^{-w_\tau(t-t_{c2i})} + 1} \right) \left(\frac{1}{e^{-w_\tau(t-t_{c3i})} + 1} \right)} \right) \quad (42)$$

where: $c_{\tau d}(0)$ – the initial position of the end-effector, v_{max} – the maximum velocity, w_τ – the coefficient corresponding to the velocity increase and decrease rate, $t \in (0,50)$ s. Function (42) and its first and second derivative in relation to time are limited.

The trajectory of system in the normal direction is the pressure force on the surface of contact, which should be continuous, limited and not negative, and should continuously have the first and second derivative in relation to time. Such conditions are met by the function:

$$F_{end} = F_{enmax} \sum_{i=1}^3 \left[\frac{1}{1 + e^{-w_n(t-t_{F1i})}} - \frac{1}{1 + e^{-w_n(t-t_{F2i})}} \right] \quad (43)$$

where: F_{enmax} – the maximum pressure force, w_n – the coefficient corresponding to the force increase and decrease rate, $t \in (0,50)$ s.

Parameters of the robot-environment dynamic system, parameters of the control system and the desired trajectory are summarized in Tables 1–3.

In the analysed example, the contact surface is one-dimensional; thus, there is only one tangent direction, along which the end-effector motion can occur. Figure 3 presents the desired motion path, displacement on the tangent direction, and pressure force in the normal direction.

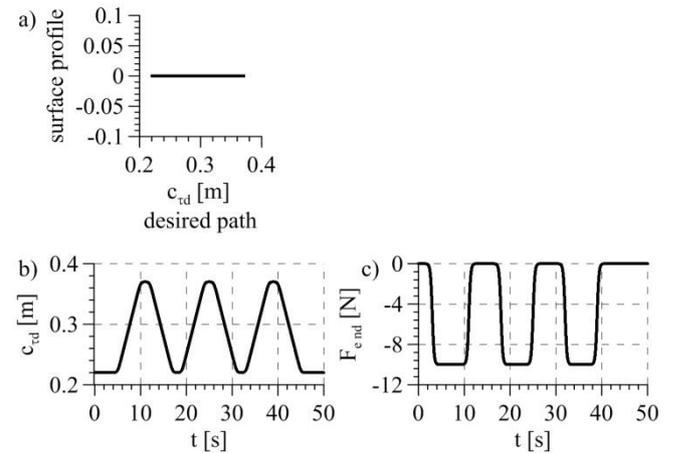


Fig. 3. Manipulator control purposes: a) motion path, b) desired displacement on the tangent direction, c) desired pressure force

Tab. 1. Parameters of robot-environment system used in numerical tests

Parameter	Unit	Value
a_1	kgm ²	0.036
a_2	kgm ²	6·10 ⁻⁵
a_3	kgm ²	0.031
a_4	Nms	0.54
a_5	Nms	0.51
a_6	Nm	0.02
a_7	Nm	0.02
a_8	Nm	0.05
a_9	Nm	0.025
l_1	m	0.22
l_2	m	0.22
K_e	N/m	1333
μ	-	0.04
ξ	Nm	0.01

The desired trajectory is planned so that during a move to the right, the manipulator end-effector exalts correct pressure, while during an end-effector return, there should be no pressure. Thus, there might be motion stages discerned: with pressure and without pressure. Fig. 4 presents the control signals generated by the specific control subsystems. The total control signal in the task space presents Fig. 4a. In accordance to equation (15), it consists of PD control (Fig. 4b), compensative control (Fig. 4c), robust control (Fig. 4d), as well as control of normal force λ_1 and control compensating for friction influence λ_2 (Fig. 4e).

Tab. 2. Parameters of control system used in numerical tests

Parameter	Unit	Value
Λ_τ	s^{-1}	3
Λ_n	s^{-1}	34.6
$K_{D\tau}$	kg/s	8
K_{Dn}	s	0.00173
k_τ	-	0.2
k_n	-	0.2
K_τ	N	0.001
K_n	N	0.01
number of neurons	-	15
Γ_τ	-	$4 \cdot I_{15 \times 15}$
Γ_n	-	$0.0008 \cdot I_{15 \times 15}$
$\hat{W}_\tau(0)$	-	0
$\hat{W}_n(0)$	-	0
$\hat{V}_\tau(0)$	-	rand $\{\pm 0.5\}$
$\hat{V}_n(0)$	-	rand $\{\pm 0.5\}$
β	-	2
v_{max}	m/s	0.03
w_τ	s^{-1}	5
F_{enmax}	N	-10
w_n	s^{-1}	5
$c_{\tau d}(0)$	m	0.22

Tab. 3. Parameters of desired trajectory used in numerical tests

Parameter	Unit	Value
t_{F11}	s	3
t_{F21}	s	16
t_{F12}	s	3
t_{F22}	s	28
t_{F13}	s	3
t_{F23}	s	3
t_{c11}	s	5
t_{c21}	s	10
t_{c31}	s	12
t_{c41}	s	17
t_{c12}	s	19
t_{c22}	s	24
t_{c32}	s	26
t_{c42}	s	31
t_{c13}	s	33
t_{c23}	s	38
t_{c33}	s	40
t_{c43}	s	45

PD control and compensative control pattern is typical for adapting systems, that is, PD control plays the largest role during the initial motion stage, when neural network weights estimates are unavailable and the compensative control is incorrect. Along with neural network weights estimates adaptation, the role of compensative control increases, while the role of PD control decreases. Force control is realised on the basis of the feedback loop and does not undergo adaptation. Fig. 4f presents the total control signal in the configuration space, which is a nonlinear mapping of the control signal in the task space. Due to practical

reasons, knowledge of it is necessary to control the drives of the controlled object.

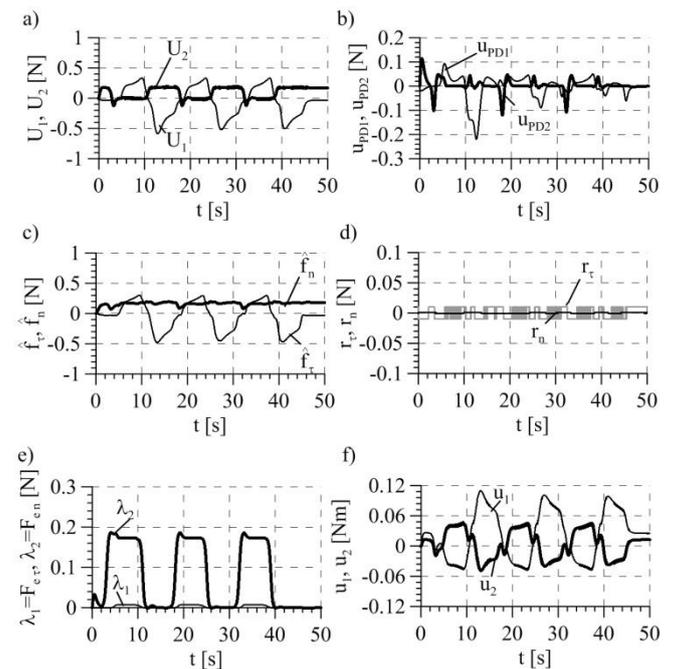


Fig. 4. Control signals: a) total control, b) PD control, c) compensative control, d) robust control, e) control of normal force λ_2 and control compensating for friction influence λ_1 , f) control signal in the configuration space

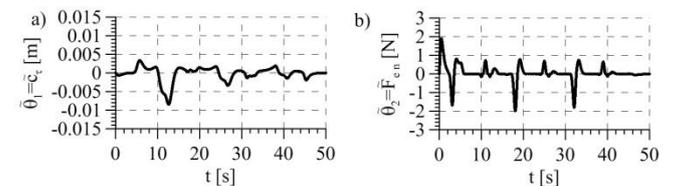


Fig. 5. Tracking errors: a) displacement on the tangent direction error, b) pressure force error

Tracking errors (Fig. 5) are typical for adapting systems, that is, the errors are the biggest during the initial control stage and then consequently they decrease, which is related to the neural network weights adaptation and increasingly better adjustment of the compensative control.

5. CONCLUSIONS

In the paper, the manipulator control algorithm is presented, which takes into account that the contact with flexible environment presented is unsusceptible to modelling imprecisions, both in terms of parameters and structure. It is possible due to the approximative features of artificial neural networks used for the compensation of system nonlinearity. Utilised system dynamics description and its control in a task space related to the environment space is very beneficial in practical applications, for which the main purpose is the realisation of the task of manipulator motion with a simultaneous pressure, utilised in process realisation.

REFERENCES

1. Barata J.C.A., Hussein M.S. (2012), The Moore–Penrose pseudoinverse: A tutorial review of the theory, *Brazilian Journal of Physics*, 42(1-2), 146–165.
2. Birglen L., Schlicht T. (2018), A statistical review of industrial robotic grippers, *Robotics and Computer-Integrated Manufacturing*, 49, 88–97.
3. Burghardt A., Kurc K., Szybicki D., Muszyńska M., Nawrocki J. (2017a), Software for the robot-operated inspection station for engine guide vanes taking into consideration the geometric variability of parts, *Tehnicki Vjesnik-Technical Gazette*, 24(2), 349–353.
4. Burghardt A., Szybicki D., Kurc K., Muszyńska M., Mucha J. (2017b), Experimental Study of Inconel 718 Surface Treatment by Edge Robotic Deburring with Force Control, *Strength Mater*, 49(4), 594–604.
5. Canudas de Wit C.A., Siciliano B., Bastin G. (Eds.) (1996), *Theory of robot control*, New York, Springer.
6. Capisani L. M., Ferrara A. (2012), Trajectory planning and second-order sliding mode motion/interaction control for robot manipulators in unknown environments, *IEEE Transactions on Industrial Electronics*, 59(8), 3189–3198.
7. Denkena B., Bergmann B., Lepper T. (2017), Design and optimization of a machining robot, *Procedia Manufacturing*, 14, 89–96.
8. Duan J., Gan Y., Chen M., Dai X. (2018), Adaptive variable impedance control for dynamic contact force tracking in uncertain environment, *Robotics and Autonomous Systems*, 102, 54–65.
9. Galushkin A. I. (2007). *Neural networks theory*, Springer Science & Business Media.
10. Gierlak P. (2012), Hybrid Position/Force Control of the SCORBOT-ER 4pc Manipulator with Neural Compensation of Nonlinearities, in: Rutkowski L., Korytkowski M., Scherer R., Tadeusiewicz R., Zadeh L.A., Zurada J.M. (eds) *Artificial Intelligence and Soft Computing. ICAISC 2012. Lecture Notes in Computer Science*, 7268, 433–441, Springer, Berlin, Heidelberg.
11. Gierlak P. (2014), Hybrid position/force control in robotised machining, *Solid State Phenomena*, 210, 192–199.
12. Gierlak P. (2018), Combined strategy for control of interaction force between manipulator and flexible environment, *Journal of Control Engineering and Applied Informatics*, 20(2), 64–75.
13. Gierlak P., Szuster M. (2017), Adaptive position/force control for robot manipulator in contact with a flexible environment, *Robotics and Autonomous Systems*, 95, 80–101.
14. Gracia L., Solanes J.E., Muñoz-Benavent P., Miro J.V., Perez-Vidal C., Tornero J. (2018), Adaptive Sliding Mode Control for Robotic Surface Treatment Using Force Feedback, *Mechatronics*, 52, 102–118.
15. Hashemi S.M., Gürcüoğlu U., Werner H. (2013), Interaction control of an industrial manipulator using LPV techniques, *Mechatronics*, 23(6), 689–699.
16. Hendzel Z., Burghardt A., Gierlak P., Szuster M. (2014), Conventional and fuzzy force control in robotised machining, *Solid State Phenomena*, 210, 178–185.
17. Hertz J., Krogh A., Palmer R.G. (1991), *Introduction to the theory of neural computation*, Boston, Addison-Wesley Longman Publishing Co.
18. Iglesias I., Sebastián M.A., Are, J.E. (2015), Overview of the state of robotic machining: Current situation and future potential, *Procedia engineering*, 132, 911–917.
19. Jafari A., Ryu J.H. (2016), Independent force and position control for cooperating manipulators handling an unknown object and interacting with an unknown environment, *Journal of the Franklin Institute*, 353(4), 857–875.
20. Kumar N., Panwar V., Sukavanam N., Sharma S.P., Borm J.-H. (2011), Neural network based hybrid force/position control for robot manipulators, *International Journal of Precision Engineering and Manufacturing*, 12(3), 419–426.
21. Lewis F.L., Liu K., Yesildirek A. (1995), Neural Net Robot Controller with Guaranteed Tracking Performance, *IEEE Transactions on Neural Networks*, 6(3), 701–715.
22. Lotz M., Bruhm H., Czinki A. (2014), An new force control strategy improving the force control capabilities of standard industrial robots, *Journal of Mechanics Engineering and Automation*, Vol. 4, 276–283.
23. Mendes N., Neto P. (2015), Indirect adaptive fuzzy control for industrial robots: a solution for contact applications, *Expert Systems with Applications*, 4 (22), 8929–8935.
24. Narendra K., Annaswamy A.M. (1987), A new adaptive law for robust adaptation without persistent excitation, *IEEE Transactions on Automatic Control*, 32(2), 134–145.
25. Pao Y.-H., Park G.-H., Sobajic D.J. (1994), Learning and generalization characteristics of the random vector functional-link net, *Neurocomputing*, 6(2), 163–180.
26. Pliego-Jiménez J., Arteaga-Pérez M.A. (2015), Adaptive position/force control for robot manipulators in contact with a rigid surface with uncertain parameters, *European Journal of Control*, 22, 1–12.
27. Polycarpou M.M., Ioannu P.A. (1991), *Identification and control using neural network models: design and stability analysis*, California, University of Southern California.
28. Ravandi A. K., Khanmirza E., Daneshjou K. (2018), Hybrid force/position control of robotic arms manipulating in uncertain environments based on adaptive fuzzy sliding mode control. *Applied Soft Computing*, 70, 864–874.
29. Tian F., Lv C., Li Z., Liu G. (2016), Modeling and control of robotic automatic polishing for curved surfaces, *CIRP Journal of Manufacturing Science and Technology*, 14, 55–64.
30. Vukobratović M., Ekalo Y., Rodič A. (2002), How to Apply Hybrid Position/Force Control to Robots Interacting with Dynamic Environment, In: Bianchi G., Guinot J.-C., Rzymkowski C. (Eds.) *Romansy*, 14, 249–258, Vienna.
31. Zhu D., Luo S., Yang L., Chen W., Yan S., Ding H. (2015), On energetic assessment of cutting mechanisms in robot-assisted belt grinding of titanium alloys, *Tribology International*, 90, 55–59.
32. Żylski W., Gierlak P. (2010), Verification of Multilayer Neural-Net Controller in Manipulator Tracking Control, *Solid State Phenomena*, 164, 99–104.