sciendo

DOI 10.2478/ama-2023-0050

acta mechanica et automatica, vol.17 no.3 (2023)

NEURAL CONTROL OF A ROBOTIC MANIPULATOR IN CONTACT WITH A FLEXIBLE AND UNCERTAIN 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 13 December 2022, revised 15 February 2023, accepted 27 March 2023

Abstract: This article presents the synthesis of a neural motion control system of a robot caused by disturbances of constraints limiting the movement, which are the result of flexibility and disturbances of the contact surface. A synthesis of the control law is presented, in which the knowledge of the robot's dynamics and the parameters of a susceptible environment is not required. Moreover, the stability of the system is guaranteed in the case of an inaccurately known surface of the environment. This was achieved by introducing an additional module to the control law in directions normal to the surface of the environment. This additional term can be interpreted as the virtual viscotic resistance and spring force acting on the robot. This approach ensured the self-regulation of the robot's interaction force with the compliant environment, limiting the impact of the geometrical inaccuracy of the environment.

Key words: robotics, robot control, nonlinear control systems

1. INTRODUCTION

The development of industrial robotics is related to robots performing more and more complex tasks that often require simultaneous execution of a desired trajectory and the interaction force of the end effector with the environment [1-4]. This refers to, among other, tasks regarding the robotisation of mechanical processing. The performance of the above-mentioned tasks is possible with simultaneous movement in one direction and exerting forces on directions perpendicular to them [5-7]. Currently, two basic control strategies are widespread in relation to robotisation of machining [8]. In the first strategy, the desired interaction force is maintained at a constant speed of movement, while the motion path of the robot end effector automatically adjusts to the shape of the contact surface. In the second control strategy [8], the robot end effector moves along the desired trajectory, regardless of the shape of the surface. A variable value is the speed, which is reduced with increasing resistance to motion resulting from the surface allowance. However, the force is not a controlled value.

The main disadvantage of the discussed solutions is that the control algorithm takes into account only one criterion, which is either minimising the error of the interaction force or minimising the deviation from the desired motion path. However, in some applications, it is necessary to consider both criteria simultaneously. This applies, for example, to the machining of castings, which have a large change in shape due to the phenomenon of foundry shrinkage [9]. Another example of difficulties is the processing of thin-walled elements, which show high compliance and deformability during machining [10]. Then, taking into account only the pressure force or only the desired path of movement of the robot end effector leads to large errors in the shape of the machined surface. An important and current issue is the development of a robot control strategy that will ensure the appropriate quality of machining, despite the presence of unmodelled effects, such as inaccurately known geometry of the workpieces and their flexibility [11,12].

This paper deals with the synthesis of the control system in cases of disturbances in the robot's contact with the environment, which result from the flexibility of the surface and the inaccuracy of its description. This article is an extension and development of the topics presented in [6,7,10]. In the paper [10], an adaptive control system was used, which was stable in the case of contact of the robot with a flexible surface, and the algorithm did not require the knowledge of the surface elasticity coefficient. The article [6] solves a similar problem with the use of neural networks, which was aimed at eliminating the need to know the dynamics model of the robot–environment system. In [7], in addition to the flexibility of the contact surface, the possibility of its inaccuracy was taken into account, and the control problem was solved with the use of an adaptive system. Finally, this article deals with control taking into account the flexibility and inaccuracy of the contact surface, and the control system uses an artificial neural network in order to eliminate the need to know the mathematical model of the robot–environment system.

In order to solve the problem, a cooperative control strategy was used, combining two elementary strategies. One of them aims to achieve the desired force and the other aims to implement the desired path of motion of the end effector. In the absence of surface inaccuracy, only the first strategy is active. Otherwise, the second strategy is activated with the first strategy running simultaneously. This leads to the "competition" of the two strategies, which allows the control algorithm to include two criteria, rather than simply switching between strategies.

2. LITERATURE REVIEW

The issue of position–force control of robots has its genesis in the practical applications of robots for the implementation of pro-

DOI 10.2478/ama-2023-0050

cesses in which the interaction between the robot and the environment is important. The most common applications where this interaction is crucial are robotic machining processes, such as milling [13,14], deburring [9,15], grinding [16,17] and polishing [2,18]. Implementation of the aforementioned processes requires the movement of the tool along a given path, which results from the shape of the machined surface, while exerting pressure on this surface, ensuring contact between the tool and the workpiece. Control algorithms that simultaneously perform position and force control belong to the hybrid group [19,20].

An important element of designing a robot control algorithm for such a task is to formulate an appropriate model of the robot–environment system. In general, various classes of models of both robots and the environment with which the robot interacts are known in the literature. Depending on the purpose of modelling and the significance of the phenomena, the following robot models are distinguished: robots as connections of rigid bodies without the flexibility of joints [21], robots as connections of rigid bodies with the flexibility of joints [22,23] and robots with the flexibility of both joints and links [24,25,26]. Similarly, there are many ways to model the environment with which the robot physically interacts. The simplest models describe the interaction surface as rigid and smooth [27]. In justified cases, flexibility and roughness of surface are taken into account [6,7]. In more complex analyses, surface models are used, described by differential equations that take into account the mass, damping and stiffness of the object with which the robot is in contact [28]. In the control synthesis issues, the models contain the most important features of robot–environment systems and, at the same time, are simple enough to be used to demonstrate the stability of a closed system [27].

The formulation of an uncomplicated model of the robot–environment system allows for the decomposition of the problem into a control task on directions tangential to the interaction surface and on normal directions. In tangential directions, the positional control is responsible for moving the robot tool along the path. In normal directions, the interaction force of the robot with the surface is controlled. In the position control part, linearisation is typically used in the inner loop, and the outer loop uses a PID (proportional–integral–derivative), PD (proportional–derivative) or PI (proportional–integral) controller. Feedback linearisation can be implemented, e.g., in the form of calculated torque [27], adaptive control [28], neural control [29], fuzzy control [30], neuro-fuzzy control [31], reinforcement learning [32] and many others. In the part of the system responsible for controlling the force, a simple proportional or proportional integral controller is usually used, sometimes in the form of a fuzzy controller.

There is a significant number of works devoted to the use of neural networks in the control of robots interacting with the environment. Neural networks are used, for example, to compensate for the non-linearity of the robot [33] or in the structure of the state observer [34]. Numerous works concern the use of fuzzy or neuro-fuzzy systems in the positional control of robots [31,35]. In order to increase the efficiency of controller tuning, advanced methods are used, using, for example, genetic algorithms [36] or particle swarm optimisation [37]. When it comes to hybrid position–force control of robots using neuro-fuzzy systems, the basics of the theory are included in the article [38], while more advanced issues taking into account the uncertainty of the environment or robot kinematics are described in papers [12,39,40]. Among the cited works, only [39] presents the results of the experimental verification of the solution. Usually, only theoretical and simulation results are described, especially for more advanced solutions.

3. APPROACH TO THE PROBLEM

The article proposes an approach to the problem of robot control in interaction with the environment, which consists in combining two elementary control strategies (Fig. 1).

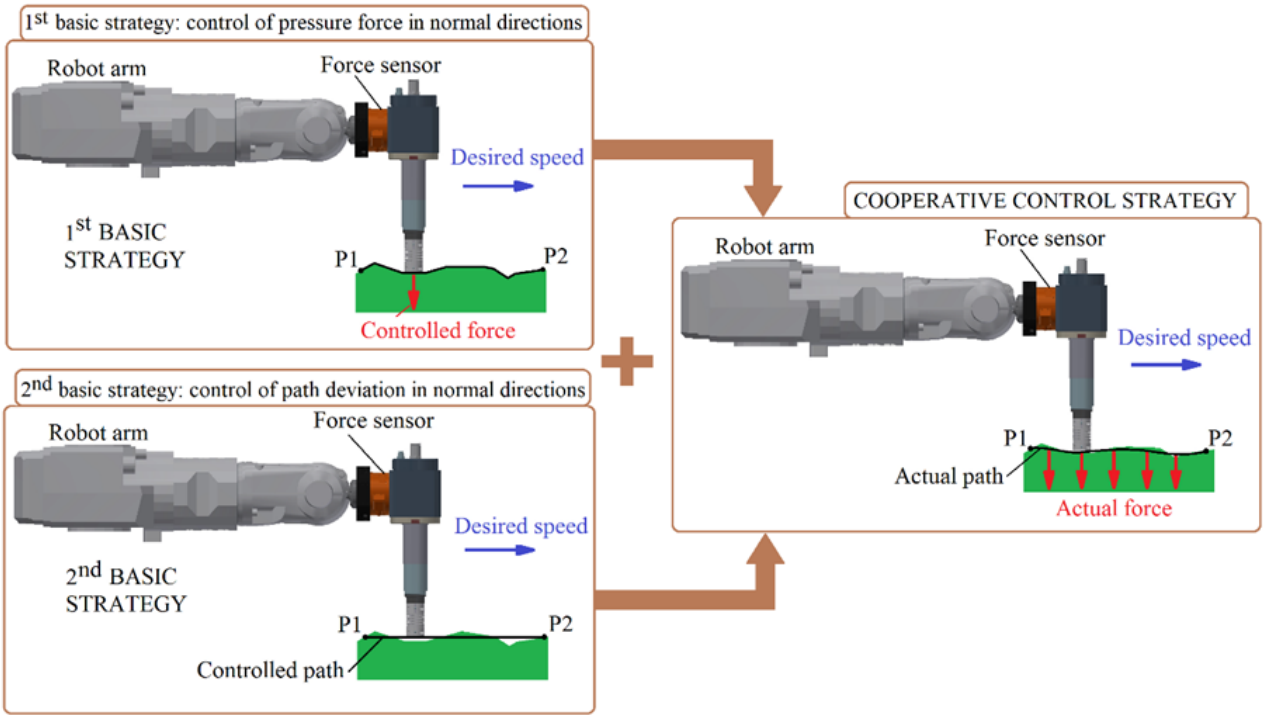


Fig. 1. Cooperative strategy for robot control

The task of the first one is to maintain a given interaction force at a given speed of movement. As a result, the path of the robot's end effector adjusts itself to the shape of the contact surface. The task of the second strategy is to execute the given motion path of the robot's end effector, regardless of the actual shape of the machined surface. The cooperative combination of these strategies leads to their interaction in such a way that the goals of each strategy are realised in a "soft" manner. The resulting operation of the control system may be closer to the first or second strategy depending on the adopted cooperation coefficient, which is responsible for adjusting the priority of force or position execution.

If the adopted surface model coincides with the actual shape of the machined surface, the first elementary strategy (Fig. 2a) is implemented, consisting in maintaining the given force in the direction normal to the surface. It is obvious that in the event of an inaccuracy of the machined surface in relation to the adopted surface model, and this is most often the case in fact, the operation of the elementary force control strategy will be weakened in favour of the strategy of minimising the deviation of the robot's end effector from the given path. Thus, neither strategy will be fully implemented. The activation of the second strategy causes the system to operate in such a way that in the place where the real surface was disturbed, virtual elastic and damping forces acted on the robot's end effector, replacing the interaction surface (Fig. 2b). Thanks to this, the end effector is attracted to the given path of movement.

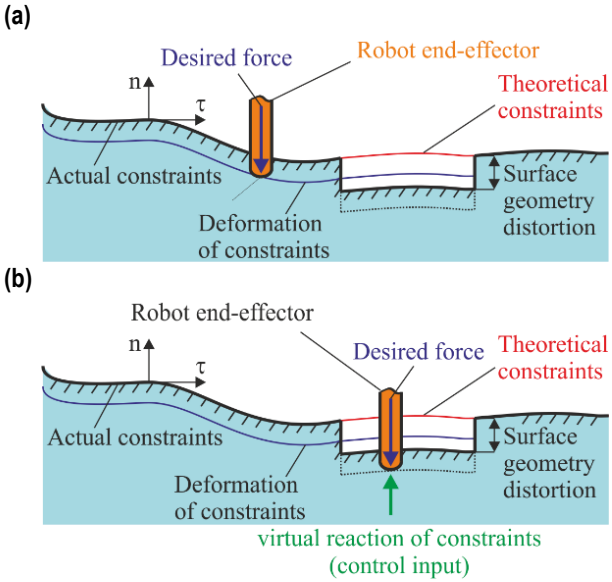


Fig. 2. The activity of the control strategy in the case of distortions in the surface: (a) movement on the surface without distortion and (b) movement on the distorted surface

The combination of strategies prevents extreme cases of system operation and at the same time ensures a smooth transition between strategies. It takes place automatically depending on the condition of the machined surface, which is an unquestionable advantage of the proposed method.

4. DYNAMICS OF THE ROBOT – FLEXIBLE ENVIRONMENT SYSTEM

In industrial practice, robotic tasks are usually not defined in the robot's configuration space but in a task space, which is usually Cartesian space. Therefore, in the further discussion, the description of the robot's dynamics in the task space was used, which has the following form [10]:

$$A(q)E\ddot{\theta} + H(q,\dot{q})E\dot{\theta} + B(q,\dot{q}) + \Psi(q,t) = U + \lambda \tag{1}$$

where $q \in R^n$ is the vector of generalised coordinates (joint coordinates), $\theta \in R^m$ is the vector of task variables, $A(q) \in R^{m\times m}$ is inertia matrix, $H(q,\dot{q}) \in R^{m\times m}$ is the matrix of coefficients of centrifugal and Coriolis forces, $B(q,\dot{q}) \in R^m$ is the vector of friction and gravitational forces, $\Psi(q,t) \in R^m$ is the vector of limited interference, $U \in R^m$ is the vector of control inputs, $\lambda \in R^m$ is the vector of interaction forces, $E \in R^{m\times m}$ is the matrix of system vulnerability and m is the dimension of the task space (it was assumed that $m = n$).

The vector of interaction forces can be represented by two components:

$$\lambda = \begin{bmatrix} F_{e\tau} \\ F_{en} \end{bmatrix} \tag{2}$$

where $F_{en} \in R^r$ is the vector of normal forces and $F_{e\tau} \in R^{m-r}$ is the vector of tangent forces. Similarly, the vector of task variables can be expressed with two components:

$$\theta = \begin{bmatrix} c_\tau \\ F_{en} \end{bmatrix} \in R^m \tag{3}$$

where $c_\tau \in R^{m-r}$ is the vector of tangential displacement. Decomposition in Eqs (2) and (3) results from the decomposition of the m-dimensional task space $\{C\}$ into r-dimensional normal subspace $\{N\}$ and $(m-r)$ – dimensional tangential subspace $\{T\}$. The vulnerability of the interaction surfaces, which limits the movement of the robot's end effector, is included in the dynamics description and the system vulnerability matrix has the following form:

$$E = \begin{bmatrix} I_{(m-r)\times(m-r)} & 0 \\ 0 & P_e \end{bmatrix} \in R^{m\times m} \tag{4}$$

where $P_e = R^{r\times r}$ is the environmental vulnerability matrix. Eq. (1) describes the dynamics of the system in task coordinates using kinematic motion parameters in tangent directions and forces in normal directions. Such a description facilitates the definition and implementation of the task, i.e., the execution of movement on the surface with the pressure force. Introducing the interaction forces into the vector of task variables enables effective control of these forces with the use of techniques known in the field of position control. The paper [10] provides information on the transformation of the dynamic description from the robot's configuration space to the task space.

5. NEURAL POSITION/FORCE TRACKING CONTROL

This section proposes a cooperative control strategy, which combines two different control strategies on the basis of cooperation (Fig. 3). The purpose of this approach is to supplement one strategy by the other in situations in which a given elementary strategy applied individually leads to unfavourable behaviour of the robot. The neural control algorithm presented in this section does not require the knowledge of the system dynamics model. Implementation of a cooperative control strategy requires the assumption of nominal contact surface geometry (theoretical constraints), desired trajectory of motion and force and knowledge of the current position of the robot's end effector.

Assumption 1: For dynamical system (1), the following trajectories are given:

DOI 10.2478/ama-2023-0050

- limited trajectory of motion of the robot's end effector in the tangential plane $c_{\tau d}(t) \in R^{m-r}$, $\dot{c}_{\tau d}(t)$, $\ddot{c}_{\tau d}(t)$,
- limited force trajectory in normal directions $F_{end}(t) \in R^{r}$, $\dot{F}_{end}(t)$, $\ddot{F}_{end}(t)$,
- limited nominal trajectory of motion of the robot's end effector in normal directions $c_{n\,nom}(t) \in R^{r}$, $\dot{c}_{n\,nom}(t)$, $\ddot{c}_{n\,nom}(t)$, which results from the assumed surface shape.

Assumption 1 concerning simultaneous knowledge of the description of the nominal motion path and the force trajectory in the same direction (normal) is a significant difference in comparison to the assumptions formulated in typical issues related to position/force control. It also allows the definition of a modified control objective, by an appropriate definition of the filtered tracking error, which in the case of taking into account the inaccuracy of constraints must be different than in the case of knowing the environment surface or omitting its inaccuracy.

To define the control objective, control errors were introduced, where

$$\tilde{c}_{\tau} = c_{\tau d} - c_{\tau} \tag{5}$$

is the error of motion in the tangential plane, and

$$\tilde{F}_{en} = F_{end} - F_{en} \tag{6}$$

is the error of force in the normal direction. An auxiliary variable $\delta \in R^{r}$ is defined such that

$$\delta = c_n - c_{n\,nom} - \delta_0 \tag{7}$$

which is related to the difference between the nominal position of the robot end effector $c_{n\,nom}$ resulting from the theoretically existing constraints and the real position c_n in the normal direction (Fig. 3).

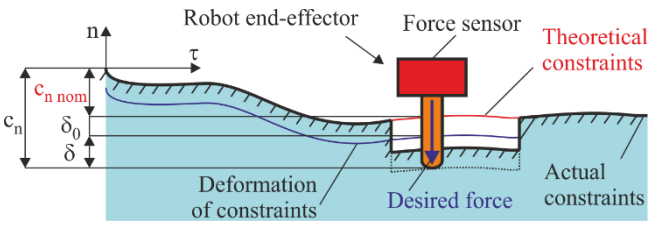


Fig. 3. Variables related to the position of the robot's end effector in the normal direction

That is, δ is the deviation of the end effector from the assumed constraints in the normal direction. In detail, it should be added that the expression $\delta_0 = K_e^{-1} F_{end}$ is the predicted surface deformation derived from the pressure force, F_{en}. With high surface rigidity, this deformation is negligible.

To achieve the control objective, a filtered tracking error was introduced:

$$s = \begin{bmatrix} s_{\tau} \\ s_n \end{bmatrix} \tag{8}$$

in which

$$s_{\tau} = \dot{\tilde{c}}_{\tau} + \Lambda_{\tau} \tilde{c}_{\tau} \tag{9}$$

$$s_n = \dot{\tilde{F}}_{en} - w_{\delta} \dot{\delta} + \Lambda_n \left(\tilde{F}_{en} - w_{\delta} \delta \right) \tag{10}$$

where $\Lambda_{\tau} \in R^{(m-r)\times(m-r)}$ and $\Lambda_n \in R^{r\times r}$ are diagonal gain matrices and $w_{\delta} \in R^{r\times r}$ is the cooperation gain matrix. Expression (10) introduces a new approach to the problem of control in the normal direction to the surface of constraints. Compared with the approach presented in papers [6,10], in which the generalised error s_n for normal directions depends on the force error and its derivative, here it also depends on deviation δ from the desired nominal motion path in the normal direction and from the derivative $\dot{\delta}$.

Eqs (8)–(10) make it possible to write the description of the dynamics as a function of the filtered tracking error:

$$A(q) E \dot{s} = -H(q, \dot{q}) E s + f + \Psi(q, t) - U - \lambda \tag{11}$$

with the non-linear part $f \in R^{m}$ dependent on both the robot mathematical model and the environment designated as follows:

$$f = A(q) E \dot{v} + H(q, \dot{q}) E v + B(q, \dot{q}) \tag{12}$$

where there is an auxiliary variable that has the following form:

$$v = \begin{bmatrix} \dot{c}_{\tau d} \\ \dot{F}_{end} - w_{\delta} \dot{\delta} \end{bmatrix} - \begin{bmatrix} \Lambda_{\tau} & 0 \\ 0 & \Lambda_n \end{bmatrix} \begin{bmatrix} \tilde{c}_{\tau} \\ \tilde{F}_{en} - w_{\delta} \delta \end{bmatrix} \tag{13}$$

Next, the control law including the PD controller (term $K_D s$), the non-linearity compensating control $\hat{f} \in R^{m}$, the term compensating the interaction force λ and the robust term $r \in R^{m}$ was assumed:

$$U = K_D s + \hat{f} - \lambda - r \tag{14}$$

where $K_D \in R^{m\times m}$ is a gain matrix such that $K_D = K_D^T > 0$, and function $\hat{f}$ approximates f. Regarding the first control part, it is possible to decompose the K_D matrix according to the equation:

$$K_D = \begin{bmatrix} K_{D\tau} & 0 \\ 0 & K_{Dn} \end{bmatrix} \tag{15}$$

where $K_{D\tau} \in R^{(m-r)\times(m-r)}$ and $K_{Dn} \in R^{r\times r}$ are diagonal gain matrices.

The part $\hat{f}$ was introduced into the control law of Eq. (14) to compensate for the non-linear function f, which depends on, inter alia, the inaccuracy of the surface δ, not just from the force error $\tilde{F}_{en}$. The function f can be decomposed into two parts, one of which corresponds to the tangential directions and the other the normal direction:

$$f = \begin{bmatrix} f_{\tau} \\ f_n \end{bmatrix} \tag{16}$$

The constituent functions f_{τ} and f_n described by Eq. (17) can be approximated with the help of various techniques. The nonlinear components f_{τ} and f_n can be written as outputs from ideal RVFL (Random Vector Functional Link) neural networks with limited approximation errors as follows [41,42]:

$$f = \begin{bmatrix} f_{\tau} \\ f_n \end{bmatrix} = \begin{bmatrix} W_{\tau}^T \Phi_{\tau}(x_{\tau}) + \varepsilon_{\tau}(x_{\tau}) \\ W_{n1}^T \Phi_{n1}(x_{n1}) + \varepsilon_{n1}(x_{n1}) \\ \vdots \\ W_{ni}^T \Phi_{ni}(x_{ni}) + \varepsilon_{ni}(x_{ni}) \\ \vdots \\ W_{nr}^T \Phi_{nr}(x_{nr}) + \varepsilon_{nr}(x_{nr}) \end{bmatrix} \tag{17}$$

where x_{τ} and x_{ni} are network input signals vectors, W_{τ} and W_{ni} are ideal output weights matrices, $\Phi_{\tau}(.) \text{and } \Phi_{ni}(.)$ are neurons activation functions vectors and ε_{τ} and ε_{ni} are vectors of errors of function mapping by networks that $\|\varepsilon_{\tau}\| \leq \varepsilon_{b\tau}$ and $|\varepsilon_{ni}| \leq \varepsilon_{bni}$ where $\varepsilon_{b\tau} > 0$, $\varepsilon_{bni} > 0$. If neuron 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. Since network ideal weights are unknown, function estimate Eq. (19) should be used, in the following form:

$$\hat{f} = \begin{bmatrix} \hat{f}_\tau \\ \hat{f}_n \end{bmatrix} = \begin{bmatrix} \widehat{W}_\tau^T \Phi_\tau(x_\tau) \\ \widehat{W}_{n1}^T \Phi_{n1}(x_{n1}) \\ \vdots \\ \widehat{W}_{ni}^T \Phi_{ni}(x_{ni}) \\ \vdots \\ \widehat{W}_{nr}^T \Phi_{nr}(x_{nr}) \end{bmatrix} \tag{18}$$

where $\widehat{W}_\tau$ and $\widehat{W}_{ni}$ are network ideal weights estimates. 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. The decomposition was performed in order to facilitate the proving of closed-loop system stability.

Assuming the control law of Eq. (14) and taking into account relationships of Eqs (16)–(18), a description of a closed system was obtained in the form:

$$AE\dot{s} = -HEs - K_D s + r + \begin{bmatrix} \widetilde{W}_\tau^T \Phi_\tau(x_\tau) \\ \widetilde{W}_{n1}^T \Phi_{n1}(x_{n1}) \\ \vdots \\ \widetilde{W}_{ni}^T \Phi_{ni}(x_{ni}) \\ \vdots \\ \widetilde{W}_{nr}^T \Phi_{nr}(x_{nr}) \end{bmatrix} + \begin{bmatrix} \varepsilon_\tau(x_\tau) \\ \varepsilon_{n1}(x_{n1}) \\ \vdots \\ \varepsilon_{ni}(x_{ni}) \\ \vdots \\ \varepsilon_{nr}(x_{nr}) \end{bmatrix} + \Psi(q,t) \tag{19}$$

where

$$\widetilde{W}_\tau = W_\tau - \widehat{W}_\tau \tag{20}$$

$$\widetilde{W}_{ni} = W_{ni} - \widehat{W}_{ni} \tag{21}$$

are errors of weights estimates.

Appropriate assumptions were made to prove the stability of the control system.

Assumption 2: There is limited interference on the dynamic system (1):

$$\Psi(q,t) = \begin{bmatrix} \Psi_\tau(q,t) \\ \Psi_n(q,t) \end{bmatrix} \tag{22}$$

where $\Psi_\tau(q,t) \in R^{m-r}$, $\Psi_n(q,t) \in R^r$ and b_τ, b_{ni} are known constants such that $\|\Psi_\tau(q,t)\| \le b_\tau$ and $\|\Psi_n(q,t)\| \le b_{ni}$.

Assumption 3: The vector of filtered tracking error in the form of Eq. (8) can be decomposed according to the following equation:

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_\tau \\ \boldsymbol{s}_n \end{bmatrix} \tag{23}$$

where $\boldsymbol{s}_\tau \in R^{m-r}$ and $\boldsymbol{s}_n = [s_{n1} \quad \dots \quad s_{ni} \quad \dots \quad s_{nr}]^T \in R^r$.

Assumption 4: A robust term can be decomposed in the following way:

$$r = \begin{bmatrix} r_\tau \\ r_n \end{bmatrix} \tag{24}$$

where

$$r_\tau = -\frac{K_\tau}{\|s_\tau\|} s_\tau \tag{25}$$

$$r_{ni} = -K_{ni} \frac{s_{ni}}{|s_{ni}|} \tag{26}$$

and $K_\tau > b_\tau \ge \|\Psi_\tau(q,t)\|$ and $K_{ni} > b_{ni} \ge |\Psi_{ni}(q,t)|$.

Assumption 5: The weights matrices are limited so that

$$\|\boldsymbol{W_\tau}\|_F \le \boldsymbol{W_{\tau\, max}} \tag{27}$$

$$\|\boldsymbol{W_{ni}}\|_F \le \boldsymbol{W_{ni\, max}} \tag{28}$$

Assumption 6: The weights adaptation law takes the form of the equations [43]:

$$\dot{\widehat{W}}_\tau = \Gamma_\tau \Phi_\tau(x_\tau) s_\tau^T - k_\tau \|s_\tau\| \Gamma_\tau \widehat{W}_\tau \tag{29}$$

$$\dot{\widehat{W}}_{ni} = \Gamma_{ni} \Phi_{ni}(x_{ni}) s_{ni} - k_{ni} |s_{ni}| \Gamma_{ni} \widehat{W}_{ni} \tag{30}$$

where $\Gamma_\tau = \Gamma_\tau^T > 0$, $\Gamma_{ni} = \Gamma_{ni}^T > 0$, $k_\tau > 0$ and $k_{ni} > 0$ are design parameters.

Theorem 1: If the system described by Eq. (1) is controlled by Eq. (14) and Assumptions 1–6 are fulfilled, the filtered tracking errors s_τ and s_{ni} and estimation errors $\widetilde{W}_\tau$ and $\widetilde{W}_{ni}$ are uniformly ultimately bounded with practical limits given by the right-hand side of the Eqs (41)–(44), respectively.

Proof of Theorem 1. The description of the system given by Eq. (1) was transformed into a description in terms of the filtered tracking error of Eq. (11), and after the introduction of the control law of Eq. (14), a closed-loop system description [Eq. (20)] was obtained, including disturbances and a robust term. To demonstrate the stability of the closed system, the Lyapunov stability theory was used. The following function was assumed:

$$V = \frac{1}{2} s^T E^T A(q) E s + \frac{1}{2} tr\left[\widetilde{W}_\tau^T \Gamma_\tau^{-1} \widetilde{W}_\tau\right] + \frac{1}{2} \sum_{i=1}^r P_{eii}\, tr\left[\widetilde{W}_{ni}^T \Gamma_{ni}^{-1} \widetilde{W}_{ni}\right] \tag{31}$$

where P_{eii} is an element of the matrix of environmental vulnerability. By calculating the derivative of function Eq. (31) with respect to time and taking into account Eq. (19), the following was obtained:

$$\dot{V} = s^T E^T \left(-K_D s + r + \Psi(q,t)\right) + s^T E^T \left(\begin{bmatrix} \widetilde{W}_\tau^T \Phi_\tau(x_\tau) \\ \widetilde{W}_{n1}^T \Phi_{n1}(x_{n1}) \\ \vdots \\ \widetilde{W}_{ni}^T \Phi_{ni}(x_{ni}) \\ \vdots \\ \widetilde{W}_{nr}^T \Phi_{nr}(x_{nr}) \end{bmatrix} + \begin{bmatrix} \varepsilon_\tau(x_\tau) \\ \varepsilon_{n1}(x_{n1}) \\ \vdots \\ \varepsilon_{ni}(x_{ni}) \\ \vdots \\ \varepsilon_{nr}(x_{nr}) \end{bmatrix} \right) + tr\left[\widetilde{W}_\tau^T \Gamma_\tau^{-1} \dot{\widehat{W}}_\tau\right] + \sum_{i=1}^r P_{eii}\, tr\left[\widetilde{W}_{ni}^T \Gamma_{ni}^{-1} \dot{\widehat{W}}_{ni}\right] \tag{32}$$

where the following property [10] was used:

$$s^T E^T \left[\dot{A}(q) - 2H(q,\dot{q})\right] E s = 0 \tag{33}$$

Given that $E^T = E$ and taking into account Eq. (8), the fourth element of Eq. (32) was transformed into the following form:

$$s_\tau^T \widetilde{W}_\tau^T \Phi_\tau(x_\tau) + \sum_{i=1}^r s_{ni} P_{eii} \widetilde{W}_{ni}^T \Phi_{ni}(x_{ni}) \tag{34}$$

The disturbance vector and robust control vector were also decomposed as follows:

$$\Psi(q,t) = [\Psi_\tau^T(q,t) \quad \Psi_{n1}(q,t) \quad \dots \quad \Psi_{ni}(q,t) \quad \dots \quad \Psi_{nr}(q,t)]^T \tag{35}$$

$$r = [r_\tau^T \quad r_{n1} \quad \dots \quad r_{ni} \quad \dots \quad r_{nr}]^T \tag{36}$$

Taking into account Eqs (4), (15), (23) and (34)–(36) and the weights adaptation laws of Eqs (29) and (30), the following equation was obtained:

$$\dot{V} = -s_\tau^T K_{D\tau} s_\tau - s_n^T P_e K_{Dn} s_n + k_\tau \|s_\tau\| tr\left[\widetilde{W}_\tau^T \widehat{W}_\tau\right] + \sum_{i=1}^r P_{eii} k_{ni} |s_{ni}| tr\left[\widetilde{W}_{ni}^T \widehat{W}_{ni}\right] + s_\tau^T [\varepsilon_\tau(x_\tau) + \Psi_\tau(q,t)] + s_\tau^T r_\tau + \sum_{i=1}^r s_{ni} P_{eii} \left(\varepsilon_{ni}(x_{ni}) + \Psi_{ni}(q,t)\right) + \sum_{i=1}^r s_{ni} P_{eii} r_{ni} \tag{37}$$

The ideal weights of neural networks are by definition limited,

DOI 10.2478/ama-2023-0050

which means that they meet the conditions (28) and (29), which were used for the transformations:

$$tr\left[\widetilde{W}_\tau^T\widehat{W}_\tau\right] = tr\left[\widetilde{W}_\tau^T\left(W_\tau - \widetilde{W}_\tau\right)\right] \le \left\|\widetilde{W}_\tau\right\|_F W_{\tau max} - \left\|\widetilde{W}_\tau\right\|_F^2 \quad (38)$$

$$tr\left[\widetilde{W}_{ni}^T\widehat{W}_{ni}\right] = tr\left[\widetilde{W}_{ni}^T\left(W_{ni} - \widetilde{W}_{ni}\right)\right] \le \left\|\widetilde{W}_{ni}\right\|_F W_{nimax} - \left\|\widetilde{W}_{ni}\right\|_F^2 \quad (39)$$

Taking into account network weight limitations, robust term elements of Eqs (25) and (26), Eq. (37) was transformed into the form:

$$\dot{V} \le -\|s_\tau\|\left[K_{D\tau min}\|s_\tau\| + k_\tau\left\|\widetilde{W}_\tau\right\|_F\left(\left\|\widetilde{W}_\tau\right\|_F - W_{\tau max}\right) - \varepsilon_{b\tau}\right] - \sum_{i=1}^{r} P_{eii}|s_{ni}|\left[K_{Dnii}|s_{ni}| + k_{ni}\left\|\widetilde{W}_{ni}\right\|_F\left(\left\|\widetilde{W}_{ni}\right\|_F - W_{nimax}\right) + \varepsilon_{bni}\right] \quad (40)$$

The function $\dot{V}$ is negative if the expressions in square brackets are positive. They are positive if the errors meet the following inequalities:

$$\|s_\tau\| > \frac{k_\tau W_{\tau max}^2 + 4\varepsilon_{b\tau}}{4K_{D\tau min}} \equiv b_{s\tau} \quad (41)$$

$$|s_{ni}| > \frac{k_{ni}W_{nimax}^2 + 4\varepsilon_{bni}}{4K_{Dnii}} \equiv b_{sni} \quad (42)$$

or

$$\left\|\widetilde{W}_\tau\right\|_F > \frac{W_{\tau max}}{2} + \sqrt{\frac{W_{\tau max}^2}{4} + \frac{\varepsilon_{b\tau}}{k_\tau}} \equiv b_{W\tau} \quad (43)$$

$$\left\|\widetilde{W}_{ni}\right\|_F > \frac{W_{nimax}}{2} + \sqrt{\frac{W_{nimax}^2}{4} + \frac{\varepsilon_{bni}}{k_{ni}}} \equiv b_{Wni} \quad (44)$$

It follows that $\dot{V}$ is negative outside the compact sets defined by Eqs (41)–(44). According to the extension of the standard Lyapunov theory, it can be concluded that $\|s_\tau\|$, $|s_{ni}|$, $\left\|\widetilde{W}_\tau\right\|_F$ and $\left\|\widetilde{W}_{ni}\right\|_F$ are uniformly ultimately bounded, and the control system is stable. Therefore, the filtered tracking error s and its derivative are limited, similar to the matrices of weight estimates $\widehat{W}_\tau$ and $\widehat{W}_{ni}$. □

6. SIMULATION RESULTS

The manipulator model was used in the simulation tests; the diagram of which is shown in Fig. 4. Its arm has three links, used to achieve the position, and the other links responsible for orientation are not used in these studies. Details on kinematics, dynamics, parameters and path planning used in the simulation are given in Appendix A in a paper [7].

End effector of the robot (point D) should move on a surface lying in the plane parallel to the plane x_0y_0 and simultaneously exert pressure force perpendicular to the surface. The Fig. 5 shows the desired motion path of point D (Figs. 5a,b) and the desired velocity (Fig. 5c). To test the properties of the control system, the simulation was carried out assuming an inaccuracy in the surface, consisting in a depression in the surface of 0.001 m, which is shown in Fig. 6a. Changing the surface can also be represented as a change of surface profile in terms of time during the motion of the end effector (Fig. 6b).

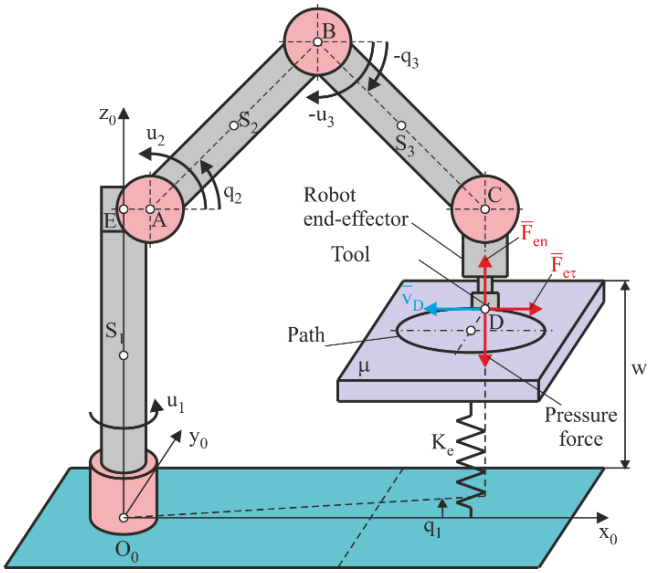
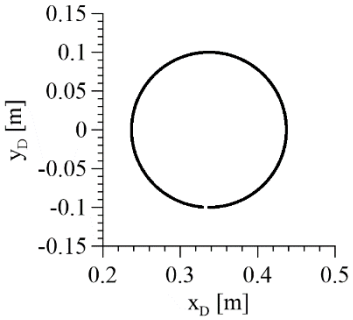
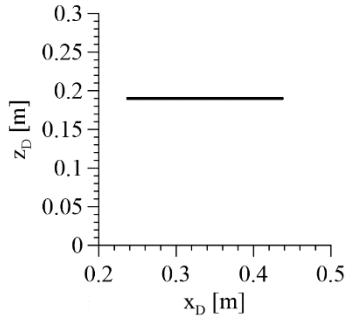


Fig. 4. Model of a robotic manipulator in contact with a flexible environment: $O_0E = d_1$, $EA = l_1$, $AB = l_2$, $BC = l_3$ and $CD = d_5$ are geometrical parameters characterising the robot arm; q_1, q_2 and q_3 are angles of link rotation assumed as generalised coordinates; u_1, u_2 and u_3 are input moments; K_e is stiffness coefficient and μ is coefficient of dry friction

a)



b)



c)

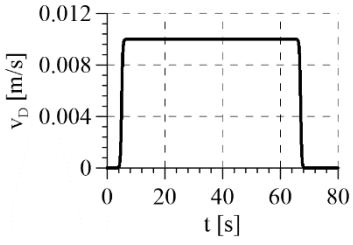


Fig. 5. The desired motion: (a) motion path of point D in the xy plane; (b) motion path of point D in the xz plane and (c) the desired velocity of motion of point D

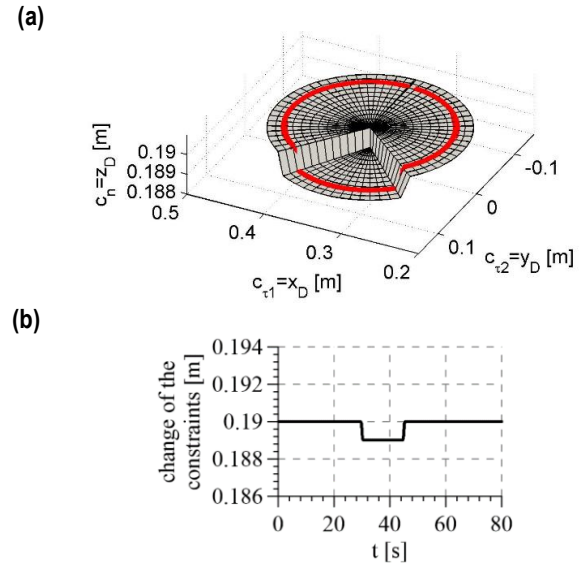


Fig. 6. Disruption of the surface of constraints: (a) defect in the surface and (b) change of the surface of constraints in time along the desired motion path

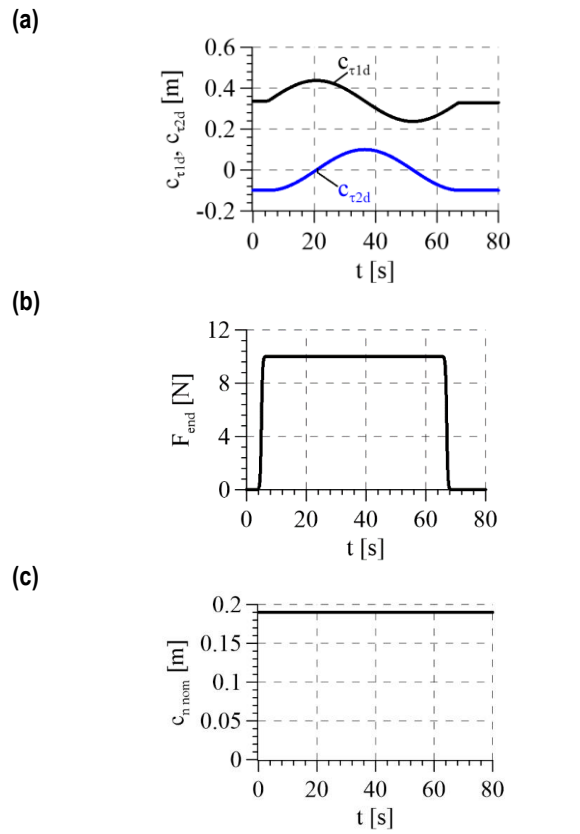


Fig. 7. The desired trajectory: (a) coordinates of point D; (b) pressure force and (c) nominal coordinate of point D in normal direction

The desired positional trajectory is shown in Fig. 7a, the desired force trajectory is shown in Fig. 7b and the nominal coordinate of point D in the normal direction is shown in Fig. 7c. The developed control algorithm requires the nominal trajectory of motion in the normal direction $c_{n\,nom}$ to be given. A trajectory deviation from nominal trajectory in the normal direction will activate the second control strategy. Moreover, due to the flexibility of contact surface, it will be deformed proportional to the pressure force.

To implement the robot's task, a control given by Eq. (14) was used in which the gain matrices have the form $K_D = diag\{K_{D\tau1}, K_{D\tau2}, K_{Dn}\}$ and $\Lambda = diag\{\Lambda_{\tau1}, \Lambda_{\tau2}, \Lambda_n\}$, and in the analysed case, w_δ is a one-dimensional coefficient, which determines the behaviour of the system in the presence of surface disturbances. The environmental vulnerability matrix is one dimensional; the vulnerability coefficient is $P_e = 0.0001$ m/N. In order to compensate for nonlinearities in each of the directions, neural networks with 15 bipolar sigmoid neurons were used. Table 1 shows the parameters of the control system.

Tab. 1. Parameters of control system used in numerical tests

Parameter	Unit	Value
$K_{D\tau1}$	kg/s	1
$K_{D\tau2}$	kg/s	1
K_{Dn}	s	0.002
$\Lambda_{\tau1}$	s^{-1}	3
$\Lambda_{\tau2}$	s^{-1}	3
Λ_n	s^{-1}	3.5
$k_{\tau1}$	-	0.1
$k_{\tau2}$	-	0.1
k_n	-	0.1
K_τ	N	0.0001
K_n	N	0.00001
$\boldsymbol{\Gamma}_{\tau1}$	-	$5 \cdot I_{15x15}$
$\boldsymbol{\Gamma}_{\tau2}$	-	$5 \cdot I_{15x15}$
$\boldsymbol{\Gamma}_n$	-	$0.003 \cdot I_{15x15}$
w_δ	N/m	10,000

Fig. 8a shows the realised trajectory in the tangent directions. The pressure force in the normal direction together with the desired force (indicated by the dashed line) is shown in Fig. 8c. In the area of surface disturbance, the actual pressure force was reduced, which fulfills the goal of the control strategy. The surface profile (dashed line) and surface deformation under pressure force are shown in Fig. 8b. Fig. 8d shows the deviation of the robot end effector from the desired path in the normal direction.

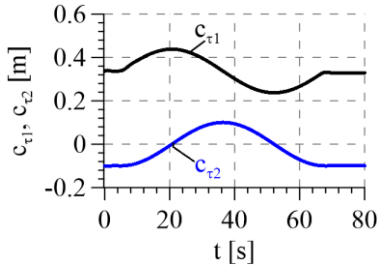
The overall control signals in the task space are shown in Fig. 9. It is clearly seen in Fig. 9b that, in the surface disturbance region, the control changes to reduce the end effector pressure on the surface (from 30 s to 45 s). The components of the control signals resulting from Eq. (14) are shown in Fig. 10: PD control (Fig. 10a,b), compensatory control (Fig. 10c,d), interaction force compensation (Fig. 10e,f) and robust control (Fig. 10g,h). The most sensitive to surface inaccuracies are the PD controls (Fig. 10b) and the control compensating for the effect of normal force (Fig. 10f).

Control errors in tangential directions are shown in Fig. 11a. These are typical waveforms showing oscillations resulting from the acceleration and deceleration phases in the initial and final phases of the movement. The pressure force error (Fig. 11b) also shows oscillations in these movement phases, but more importantly, it increases when the surface disturbance occurs. It is related to the deviation of the robot end effector from the desired motion path (Fig. 8d). The component goals of minimising the force error and minimising the robot's deviation from the path are not fully achieved because they are competitive and cannot be achieved simultaneously in a surface disturbance situation. However, this is consistent with the definition of the filtered tracking

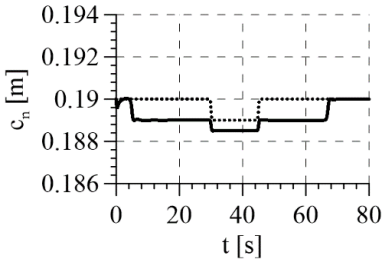
DOI 10.2478/ama-2023-0050

error given by Eq. (10), which indicates that the simultaneous occurrence of a force error and a position error in the normal direction does not contradict the possibility of minimising the filtered tracking error. Such formulation of the control objective is conducive to achieving a "balance" between minimising the force error and minimising the deviation from the nominal surface.

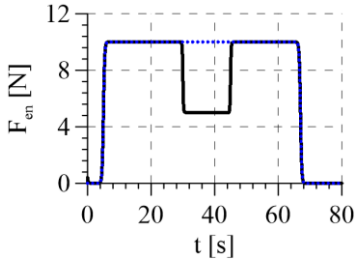
a)



b)



c)



d)

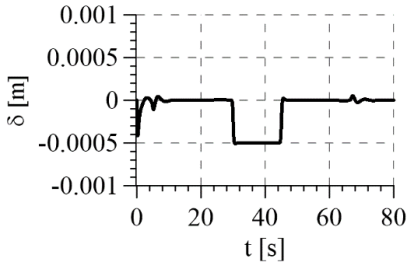
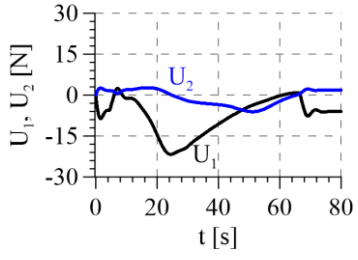


Fig. 8. Realised trajectory: (a) coordinates of point D in the tangential directions; (b) coordinates of point D in the normal direction related to surface deformation; (c) pressure force and (d) deviation of robot's end effector from assumed constraints in the normal direction

a)



b)

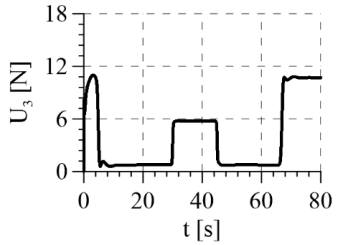
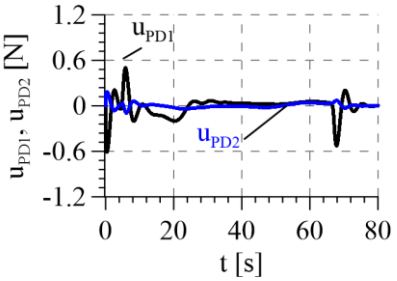
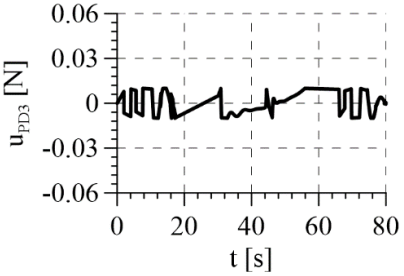


Fig. 9. The overall control signals: (a) in tangential directions and (b) in the normal direction

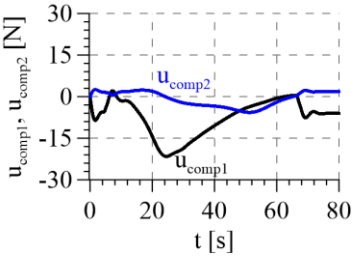
a)



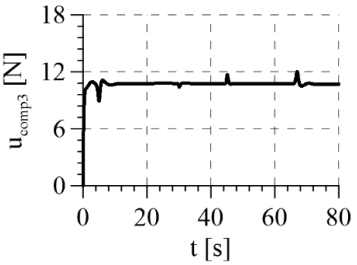
b)



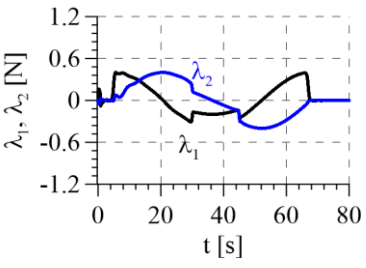
c)



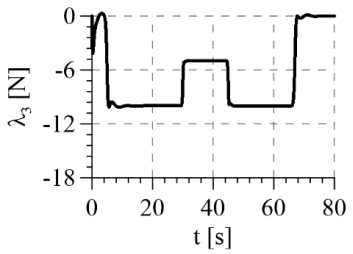
d)



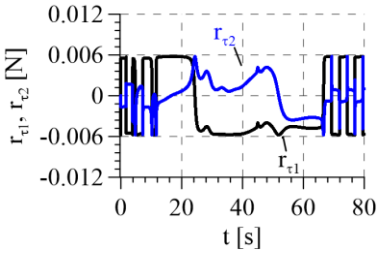
e)



f)



g)



h)

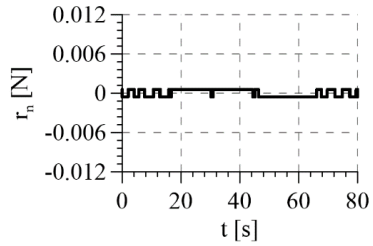
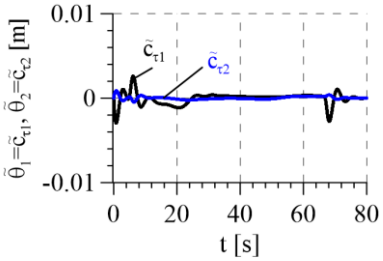


Fig. 10. Control signals: (a) PD control in tangential directions, where $u_{PD1} = K_{D\tau1}s_{\tau1}$ and $u_{PD2} = K_{D\tau2}s_{\tau2}$; (b) PD control in the normal direction, where $u_{PD3} = K_{Dn}s_n$; (c) compensatory control in tangential directions, where $u_{komp1} = \hat{f}_{\tau1}$ and $u_{komp2} = \hat{f}_{\tau2}$; (d) compensatory control in the normal direction, where $u_{komp3} = \hat{f}_n$; (e) control compensating for the influence of friction forces; (f) control compensating for normal force; (g) robust control in tangential directions and (h) robust control in the normal direction

(a)



(b)

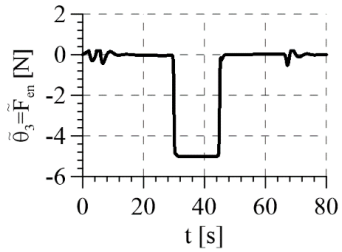


Fig. 11. Tracking errors: (a) motion errors in tangential directions and (b) normal force error

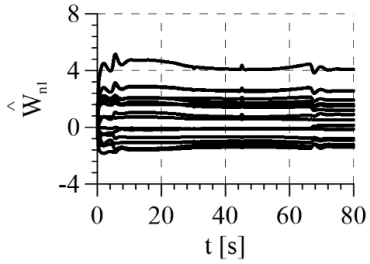


Fig. 12. Estimates of the weights of neural network compensating nonlinearities in the normal direction

The estimates of the weights of the neural network generating the compensatory control for the normal direction are shown in Fig. 12. They are limited, according to the stability proof, and their greatest variability occurs in the initial phase of motion, which is also the stage of the most intensive training of the neural network.

7. CONCLUSIONS

The article presents the synthesis of the control system for the robot-flexible environment system. The influence of the contact surface disturbance on the robot's behaviour was taken into account. The results of the simulation tests show that the requirements for the control system were met, i.e., firstly, the control system was stable, and secondly, the cooperative control strategy was correctly implemented. By appropriately combining the two elemental control strategies, a compromise is ensured between the execution of the desired pressure force and the maintenance of the desired movement path.

The main contribution of the article is as follows: the use of additional control components, which can be interpreted as reactions of virtual constraints, ensures a self-regulation of the robot's interaction force with a flexible environment, minimising the influence of the geometric inaccuracy of the environment. The presented method was developed for the use in robotic machining of elements with imprecise shape, e.g., thin-walled castings or parts made of plastics, which have low precision and are very flexible. The difficulty in its application in practice is the need to modify standard industrial robot controllers [44].

REFERENCES

1. Iglesias I, Sebastián MA, Ares JE. Overview of the State of Robotic Machining: Current Situation and Future Potential. Procedia Eng. 2015;132:911–7.
2. Tian F, Lv C, Li Z, Liu G. Modeling and control of robotic automatic polishing for curved surfaces. CIRP J Manuf Sci Technol. 2016;14:55–64.
3. Denkena B, Bergmann B, Lepper T. Design and optimization of a machining robot. Procedia Manuf. 2017;14:89–96.
4. Gracia L, Solanes JE, Muñoz-Benavent P, Valls Miro J, Perez-Vidal C, Tornero J. Adaptive Sliding Mode Control for Robotic Surface Treatment Using Force Feedback. Mechatronics. 2018;52:102–18.
5. Vukobratovič M, Ekalo Y, Rodič A. How to Apply Hybrid Position/Force Control to Robots Interacting with Dynamic Environment. W: Bianchi G, Guinot JC, Rzymkowski C, Eds. Romansy 14 [Internet]. Vienna: Springer Vienna; 2002 [cited 13 december 2022]. 249–58. Available from: http://link.springer.com/10.1007/978-3-7091-2552-6_27
6. Gierlak P. Position/Force Control of Manipulator in Contact with Flexible Environment. Acta Mech Autom. 2019;13(1):16–22.

DOI 10.2478/ama-2023-0050

7. Gierlak P. Adaptive Position/Force Control of a Robotic Manipulator in Contact with a Flexible and Uncertain Environment. Robotics. 12 2021;10(1):32.
8. Application Manual. Force Control for Machining. Zürich: ABB Robotics; 2011.
9. Burghardt A, Szybicki D, Kurc K, Muszyñska M, Mucha J. Experimental Study of Inconel 718 Surface Treatment by Edge Robotic Deburring with Force Control. Strength Mater. 2017;49(4):594–604.
10. Gierlak P, Szuster M. Adaptive position/force control for robot manipulator in contact with a flexible environment. Robot Auton Syst. 2017;95:80–101.
11. Duan J, Gan Y, Chen M, Dai X. Adaptive variable impedance control for dynamic contact force tracking in uncertain environment. Robot Auton Syst. 2018;102:54–65.
12. Ravandi KA, Khanmirza E, Daneshjou K. Hybrid force/position control of robotic arms manipulating in uncertain environments based on adaptive fuzzy sliding mode control. Appl Soft Comput. 2018;70:864–74.
13. Guo K, Zhang Y, Sun J. Towards stable milling: Principle and application of active contact robotic milling. Int J Mach Tools Manuf. 2022;182:103952.
14. Wang W, Guo Q, Yang Z, Jiang Y, Xu J. A state-of-the-art review on robotic milling of complex parts with high efficiency and precision. Robot Comput-Integr Manuf. 2023;79:102436.
15. Chen SC, Tung PC. Trajectory planning for automated robotic deburring on an unknown contour. Int J Mach Tools Manuf. 2000;40(7):957–78.
16. Robotic Grinding Process of Turboprop Engine Compressor Blades with Active Selection of Contact Force. Teh Vjesn - Tech Gaz [Internet]. 2022 Feb 15 [cited 2022 Dec 7];29(1). Available from: https://hrcak.srce.hr/269299.
17. Wang Z, Zou L, Luo G, Lv C, Huang Y. A novel selected force controlling method for improving robotic grinding accuracy of complex curved blade. ISA Trans. 2022;129:642–58.
18. Ke X, Yu Y, Li K, Wang T, Zhong B, Wang Z, et al. Review on robot-assisted polishing: Status and future trends. Robot Comput-Integr Manuf. 2023;80:102482.
19. Gierlak P. 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 LA, Zurada JM, editors. Artificial Intelligence and Soft Computing [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012 [cited 2020 Nov 21]. p. 433–41. (Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, et al., editors. Lecture Notes in Computer Science; vol. 7268). Available from: http://link.springer.com/10.1007/978-3-642-29350-4_52.
20. Gierlak P. Hybrid Position/Force Control in Robotised Machining. Solid State Phenom. 2013;210:192–9.
21. Dwivedy SK, Eberhard P. Dynamic analysis of flexible manipulators, a literature review. Mech Mach Theory. 2006;41(7):749–77.
22. Do TT, Vu VH, Liu Z. Linearization of dynamic equations for vibration and modal analysis of flexible joint manipulators. Mech Mach Theory. 2022;167:104516.
23. Cheng X, Zhang Y, Liu H, Wollherr D, Buss M. Adaptive neural backstepping control for flexible-joint robot manipulator with bounded torque inputs. Neurocomputing. 2021;458:70–86.
24. Endo T, Kawasaki H. Bending moment-based force control of flexible arm under gravity. Mech Mach Theory. 2014;79:217–29.
25. Thomsen DK, Søe-Knudsen R, Balling O, Zhang X. Vibration control of industrial robot arms by multi-mode time-varying input shaping. Mech Mach Theory. 2021;155:104072.
26. Cheong J, Youm Y. System mode approach for analysis of horizontal vibration of 3-D two-link flexible manipulators. J Sound Vib. 2003;268(1):49–70.
27. Lewis FL, Jagannathan S, Yeşildirek A. Neural network control of robot manipulators and nonlinear systems. London: Taylor & Francis; 1999. 442 p. (The Taylor & Francis systems and control book series).
28. Wei B. Adaptive Control Design and Stability Analysis of Robotic Manipulators. Actuators. 2018;7(4):89.
29. Gupta P, Sinha NK. Intelligent control of robotic manipulators: experimental study using neural networks. Mechatronics. 2000;10(1–2):289–305.
30. Yin X, Pan L, Cai S. Robust adaptive fuzzy sliding mode trajectory tracking control for serial robotic manipulators. Robot Comput-Integr Manuf. 2021;72:101884.
31. Pham DT, Fahmy AA. NEURO-FUZZY MODELLING AND CONTROL OF ROBOT MANIPULATORS FOR TRAJECTORY TRACKING. IFAC Proc Vol. 2005;38(1):170–5.
32. Szuster M, Gierlak P. Approximate Dynamic Programming in Tracking Control of a Robotic Manipulator. Int J Adv Robot Syst. 2016;13(1):16.
33. Kumar N, Rani M. Neural network-based hybrid force/position control of constrained reconfigurable manipulators. Neurocomputing. 2021;420:1–14.
34. Yang Z, Peng J, Liu Y. Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer. Neurocomputing. 2019;331:263–80.
35. de Campos Souza PV. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. Appl Soft Comput. 2020;92:106275.
36. Refoufi S, Benmahammed K. Control of a manipulator robot by neuro-fuzzy subsets form approach control optimized by the genetic algorithms. ISA Trans. 2018;77:133–45.
37. Vijay M, Jena D. PSO based neuro fuzzy sliding mode control for a robot manipulator. J Electr Syst Inf Technol. 2017;4(1):243–56.
38. Fanaei A, Farrokhi M. ADAPTIVE NEURO-FUZZY CONTROLLER FOR HYBRID POSITION/FORCE CONTROL OF ROBOTIC MANIPULATORS. IFAC Proc Vol. 2005;38(1):127–32.
39. Wang Z, Zou L, Su X, Luo G, Li R, Huang Y. Hybrid force/position control in workspace of robotic manipulator in uncertain environments based on adaptive fuzzy control. Robot Auton Syst. 2021 Nov;145:103870.
40. Garcia-Rodriguez R, Parra-Vega V. Normal and tangent force neuro-fuzzy control of a soft-tip robot with unknown kinematics. Eng Appl Artif Intell. 2017 Oct;65:43–50.
41. Pao YH, Park GH, Sobajic DJ. Learning and generalization characteristics of the random vector functional-link net. Neurocomputing. 1994;6(2):163–80.
42. Kumar N, Panwar V, Sukavanam N, Sharma SP, Borm JH. Neural network based hybrid force/position control for robot manipulators. Int J Precis Eng Manuf. 2011;12(3):419–26.
43. Lewis FL, Liu K, Yesildirek A. Neural net robot controller with guaranteed tracking performance. IEEE Trans Neural Netw. 1995;6(3):703–15.
44. Obal P, Gierlak P. EGM Toolbox—Interface for Controlling ABB Robots in Simulink. Sensors. 2021;21(22):7463.

Piotr Gierlak: https://orcid.org/0000-0003-4545-8253