Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 57(3), 724–735(2016)

Mojtaba Hadi Barhaghtalab, Hassan Bayani, Armin Nabaei, Houman Zarrabi, Ali Amiri*

On The Design of The Robust Neuro-Adaptive Controller for Cable-driven Parallel Robots

DOI 10.7305/automatika.2017.02.1793
UDK [681.51.03.015.42-55:004.032.26]:621.865.8-831.2-025.41

Original scientific paper

In this study, a robust neuro-adaptive controller for cable-driven parallel robots is proposed. The robust neuro-adaptive control system is comprised of a computation controller and a robust controller. The computation controller containing a neural-network-estimator with radial basis function activator is the principal controller and the robust controller is designed to achieve tracking performance. An on-line tuning method is derived to tune the parameters of the neural network for estimating the controlled system dynamic function. To investigate the effectiveness of the robust adaptive control, the design methodology is applied to control a cable-driven parallel robot. Simulation results demonstrate that the proposed robust adaptive control system can achieve favorable tracking performances for the robot.

Key words: Robust control, Adaptive law, neural network, Cable-driven Parallel Robot

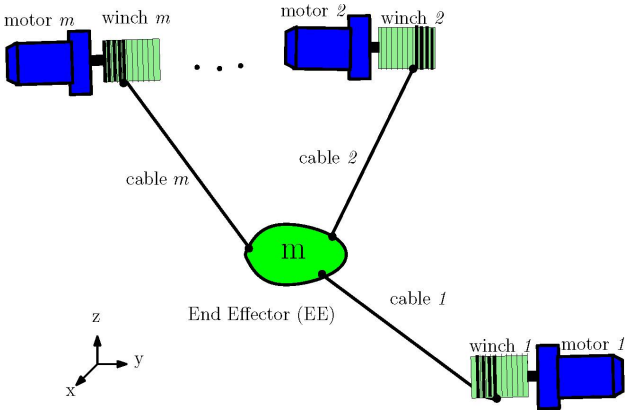
Dizajn robusnog neuro-adaptivnog regulatora za žično pogonjene paralelne robote. U ovome redu predstavljen je neuro-adaptivni regulator za žično pogonjene paralelne robote. Robusni neuro-adaptivni regulator sastoji se od regulatora zasnovanog na estimiranom modelu i robusnog regulatora. Prvi regulator sadrži estimator s neuronskom mrežom s radijalnom aktivacijskom funkcijom glavni je regulator u sustavu, a robusni je regulator dizajniran za slijeđenje. Izvedena je *on-line* metoda podešavanja parametara neuronske mreže za estimaciju dinamike sustava upravljanja. Efikasnost sustava adaptivnog, robusnog regulatora testirana je na na žično pogonjenom paralelnom robotu. Simulacijski rezultati pokazuju da se predloženim robusnim i adaptivnim regulatorom mogu dobiti zadovoljavajuće performanse prilikom slijeđenja.

Ključne riječi: robusno upravljanje, adaptivni upravljački zakon, neuronska mreža, žično pogonjeni paralelni robot

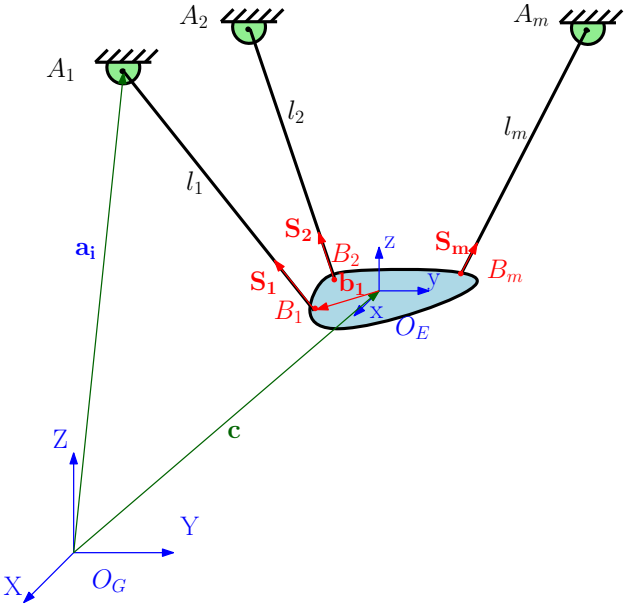
1 INTRODUCTION

In the recent decades, robots have been utilized in vast area of industries. However, many parts of manufacturing and engineering does not put upon robots, mainly due to the weakness of conventional robots [1] [2]. For instance, in many applications workspace requirements and load carrying capacity are so much higher than what the conventional robots can provide while cost of the robot should be considered [3] [4]. Toward resolving the latter issue, new class of parallel robots were introduced [5]. Cable-Driven Parallel Robots (CDPR) are structurally similar to parallel actuated robots but with the fundamental difference that cables can only pull the End-Effector (EE) but not push it. Figure 1 represents schematically a CDPR in a general arrangement. It consists of motor, winch system and the EE. From a scientific stand point, feedback control of CDPR is lot more challenging than their counterpart parallel-actuated robots due to the cables behavior. Several efforts had been exerted on modeling and control of CDPR for real-time purposes [6, 7]. For all kinetostatic model proposed for CDPR, there should be a static balancing between external forces and tension of the cables [8]. With assumption of no mass and no elongation for cables, most of the common control strategies used for conventional robots could be adapted for CDPRs. In case of classic controller, PD controller is applied in [9] and the results present acceptable performance over desired task. As modern control approaches, inverse control of cable-driven parallel mechanism using ANFIS is the main contribution in [10]. Nonlinear Sliding Mode Controller (SMC) and feasible workspace analysis for a cable suspended robot with input constraints is presented in [11].

All the above efforts for deriving kinematic and dynamic equations tend to present a simple model of CDPR that could works in an online manner with common control approaches while precision is not devastated [12, 13]. Many researches illustrates effectiveness of the precise modeling of the robot in the controlling procedure [14, 15], howe-



Slika 1. Schematic representation of a CDPR with m cables.



Slika 2. kinematic modeling of a CDPR with m cables.

ver controller capabilities may affect the tracking performance [16, 17]. The main contribution of this paper, compared to similar researches toward controlling of CDPR is taking benefits of Robust Neuro-Adaptive Controller (RNAC). The latter method allows the design of considerably more precise, energy-efficient and compliant controls for robots. It allows one to obtain a linear closed-loop equation in terms of the state variables [18, 19]. This fact has no precedent in the study of the controllers. On the other hand, RNAC is characterized for being a dynamic controller, that is, its complete control law includes additional state variables. Moreover, as another contribution radial basis function as an activator function is used in the structure of the neural network part of the adaptive law.

The organization of this paper is as follows. Section II describes the kinematic, static and dynamic equations of CDPR. The general forms of these equations are derived for any structure of the CDPR. Section III and IV outline a method for control based on RNAC which contains of obtaining the control rule. Finally, the paper concludes by presenting the simulation results and some hints for ongoing works.

2 CABLE-DRIVEN PARALLEL ROBOTS MODEL

This section outlines the kinematic, static and dynamic of the CDPR in general arrangement. The presented model is based on [20].

2.1 Kinematic Equations

Figure 2 illustrates schematically a 6-DOF spatial CDPR. In the latter figure, A_i denotes the attachment point of each cable to the base and $i = \{1, \cdots, m\}$ indicates the number of cable. The parameter B_i stands for the attachment point to the EE, eventually $\mathbf{a_i}$ and $\mathbf{b_i}$ are a constant vectors in O_G and O_E coordinates respectively. Note that O_G is the global coordinate and O_E is the attached coordinates on the EE. Referring to Fig. 2, the following equation is derived [20]:

$$\mathbf{c} = \mathbf{a_i} + \mathbf{l_i} - \mathbf{b_i} \tag{1}$$

Equation (1) could be written in the O_G coordinates as follows:

$$\mathbf{c} = \mathbf{a_i} + \mathbf{l_i} - \mathbf{R} \times \mathbf{b_i} \tag{2}$$

which could be rearranged in the following form:

$$\mathbf{l_i} = \mathbf{c} - \mathbf{a_i} + \mathbf{R} \times \mathbf{b_i} \tag{3}$$

where $\mathbf{R}$ is the transformation matrix from O_E to O_G which could be written as follows:

$$\mathbf{R} = \begin{bmatrix} c_\psi c_\phi & c_\psi s_1 s_\psi - c_\theta s_\psi & c_\theta c_\psi s_\phi + s_\theta s_\psi \\ c_\psi s_\psi & c_\theta c_\psi + s_\theta s_\phi s_\psi & -c_\psi s_\theta + c_\theta s_\phi s_\psi \\ -s_\phi & c_\phi s_\theta & c_\phi c_\theta \end{bmatrix} \tag{4}$$

In Eq. (4), c stands for $\cos(\cdot)$ and s indicates $\sin(\cdot)$. Also θ, ϕ and ψ are angle of rotation around x, y and z axis respectively. By taking the first derivative of Eq. (3) the following equation is obtained:

$$\mathbf{l_i}\dot{\mathbf{l}}_\mathbf{i} = [\mathbf{c} - \mathbf{a_i} + \mathbf{b_i}]^T [\dot{\mathbf{c}} - \dot{\mathbf{a}}_\mathbf{i} + \dot{\mathbf{b}}_\mathbf{i}] \tag{5}$$

Note that $\boldsymbol{\omega_i}$ is the rotational velocity of EE in the O_E coordinate. Obviously, by considering $\dot{\mathbf{a}}_\mathbf{i} = 0$ and $\dot{\mathbf{b}}_\mathbf{i} = \boldsymbol{\omega_i} \times \mathbf{b_i}$, Eq. (5) could be rewritten in the following form:

$$\mathbf{l_i}\dot{\mathbf{l}}_\mathbf{i} = [\mathbf{c} - \mathbf{a_i} + \mathbf{b_i}]^T [\dot{\mathbf{c}} - \boldsymbol{\omega_i}\dot{\mathbf{b}}_\mathbf{i}] \tag{6}$$

Rearranging Eq. (6), one could derive the following:

$$\dot{\mathbf{l}}_\mathbf{i} = \mathbf{S_i}^\mathbf{T}\dot{\mathbf{c}} + (\mathbf{b_i} \times \mathbf{S_i}^\mathbf{T})\boldsymbol{\omega_i} \tag{7}$$

where $\mathbf{S_i}$ is the unit vector along i^{th} cable and could be written in the following form:

$$\mathbf{S_i} = \frac{\mathbf{c} - \mathbf{a_i} + \mathbf{b_i}}{\sqrt{[\mathbf{c} - \mathbf{a_i} + \mathbf{b_i}]^\mathbf{T}[\mathbf{c} - \mathbf{a_i} + \mathbf{b_i}]}} \tag{8}$$

Therefor the kinematic equation for the robot has the following form:

$$\dot{\mathbf{L}} = \mathbf{Jt} \tag{9}$$

where $\mathbf{t} = [\dot{\mathbf{c}} \ \ \boldsymbol{\omega_i}]^T$ is the twist vector, $\dot{\mathbf{L}} = [\dot{\mathbf{l}}_\mathbf{1} \ \dot{\mathbf{l}}_\mathbf{2} \ \cdots \ \dot{\mathbf{l}}_\mathbf{n}]^T$ is the vector of cables displacement and Jacobian matrix, $\mathbf{J}$, is defined as follows:

$$\mathbf{J} = \begin{bmatrix} \mathbf{S_1} & \mathbf{Rb_1} \times \mathbf{S_1} \\ \mathbf{S_2} & \mathbf{Rb_2} \times \mathbf{S_2} \\ \cdot & \cdot \\ \mathbf{S_m} & \mathbf{Rb_m} \times \mathbf{S_m} \end{bmatrix} \tag{10}$$

It is worth noticing that $\mathbf{J}$ has dimension of $(m \times 6)$ in general form. With the above consideration in kinematic equation the next part investigates dynamic modeling of the CDPR.

2.2 Static Equations

As aforementioned, cables tension in CDPR should be always larger than a specific value. Moreover, it should not overtake the maximum permissible value due to the cable properties. The following inequality states the latter fact:

$$f_{\min} \leq f_i \leq f_{\max} \tag{11}$$

where f_i is the tension force in each cable. For the sake of static equilibrium, the sum of the tension forces exerted by the cables on the EE should be equal to the external force. It is assumed that there are no external forces which leads to the following equation for equilibrium of the forces and moments [21]:

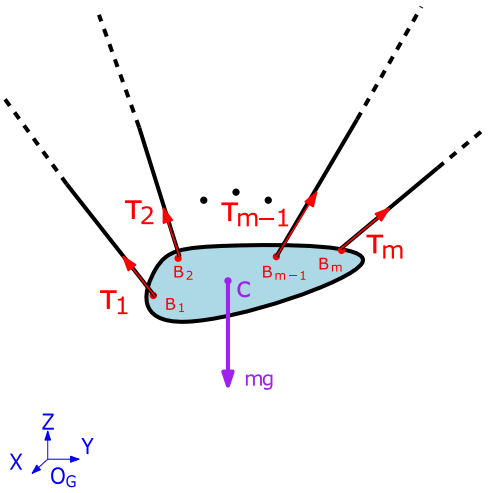
$$\mathbf{Kf} = \mathbf{0} \tag{12}$$

Based on Eq. (12) and Jacobian matrix derived in Eq. (10), one could derive a cable tension constraint for remaining larger than a specific value. There are fewer equations than which admits and there are infinite many solutions for its unknown parameters. Due to the latter fact, the following equation is derived [21]:

$$\mathbf{f} = (\mathbf{I}_{4\times4}\mathbf{K}^\dagger\mathbf{K})\mathbf{Z} \tag{13}$$

where $\mathbf{I_{4\times4}}$ indicates the identity matrix, $\mathbf{Z}$ denotes an arbitrary n-vector and $\mathbf{K}^\dagger$ is defined as follows:

$$\mathbf{K}^\dagger = \mathbf{K}^T(\mathbf{K}\mathbf{K}^T)^{-1} \tag{14}$$



Slika 3. Dynamic modeling of a CDPR with m cables.

Indeed, Eq. (14) defines a (4×3) under constrained Moore-Penrose pseudo inverse of $\mathbf{K}$. The latter is the homogeneous solution that maps $\mathbf{Z}$ to the null space of $\mathbf{K}$. Based on [21] and noticing that the spatial CDPR with eight cables has one degree of redundancy, Eq. (12) could be reformed as follows:

$$\mathbf{f} = a\mathbf{n} \tag{15}$$

where $\mathbf{n} = [n_1 \ n_2 \ n_3 \ n_4]^T$. In order to guarantee that there is a solution with positive cable tension, it is necessary and sufficient to show that all kernel vector components $(n_i, i = \{1,2,3,4\})$ have the same sign, i.e., for a given point to lie within the statics workspace, all arrays of $\mathbf{n}$ should be negative or positive simultaneously. If the latter condition is satisfied, withdrawing any particular solution, there is an scalar a in Eq. (15) which guarantees that all cables tension are positive by adding or subtracting enough homogeneous solution [20]. It could be concluded that the entire allowable kinematic workspace is also static workspace [20]. Thereupon, CDPRs have non-singular workspace, i.e., all proper workspace definitions of CDPR have one common condition which states that inside the boundaries of workspace there are no singular point [21] [22]. This property of CDPR is the result of the cable elastic behavior. Each cable link could be regarded as prismatic link with one major benefit. The values of tension in each cable could vary while EE does not have any displacement. However the same condition for prismatic (rigid) link will results in permanent deformation. In what follows the dynamic equations for CDPR are derived in order to be used in the proposed controller.

2.3 Dynamic Equations

In order to make sure that positive values of tension in each cable are derived, one should solve the inverse dynamic. Ignoring the mass of cables comparing to EE and motor winch masses while suggesting rigid string, based on Fig. 3, the dynamic equation of a CDPR could be regarded as follows [21] [22]:

$$\boldsymbol{\tau}_1\mathbf{S_1}+\boldsymbol{\tau}_2\mathbf{S_2}+\cdots+\boldsymbol{\tau}_n\mathbf{S_n}+m\begin{bmatrix}0\\0\\-g\end{bmatrix}=m\begin{bmatrix}\ddot{x}_c\\\ddot{y}_c\\\ddot{z}_c\end{bmatrix} \quad (16)$$

Moreover, the angular momentum of the EE around its mass center could be regarded as follows:

$$\mathbf{H}_c=\mathbf{I}_c\boldsymbol{\omega} \quad (17)$$

where $\mathbf{I}_c$ denotes the EE moment of inertia around its mass center. By taking the first derivative of Eq. (17) the following equation is derived:

$$\dot{\mathbf{H}}_c=\frac{\partial\mathbf{H}_c}{\partial t}+\dot{\boldsymbol{\omega}}\times\mathbf{H}_c=\dot{\mathbf{I}}_c\boldsymbol{\omega}+\mathbf{I}_c\dot{\boldsymbol{\omega}}+\boldsymbol{\omega}\mathbf{I}_c\boldsymbol{\omega}=\mathbf{I}_c\dot{\boldsymbol{\omega}}+\boldsymbol{\omega}\mathbf{I}_c\boldsymbol{\omega} \quad (18)$$

By using Eq. (17) and Euler equation, one could derive the following equation:

$$\mathbf{I}_c\dot{\boldsymbol{\omega}}+\boldsymbol{\omega}\mathbf{I}_c\boldsymbol{\omega}=\sum_{i=1}^{n}b_i\times\mathbf{S}_i\boldsymbol{\tau}_i \quad (19)$$

Note that $\boldsymbol{\omega}$ for the point c is defined as follows:

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\end{bmatrix}=\mathbf{P}\begin{bmatrix}\dot{\theta}\\\dot{\phi}\\\dot{\psi}\end{bmatrix} \quad (20)$$

where $\boldsymbol{\omega}_c=[\dot{\theta}\ \ \dot{\phi}\ \ \dot{\psi}]$ is the angular velocity of the mass center in EE and $\mathbf{P}$ is defined as follows:

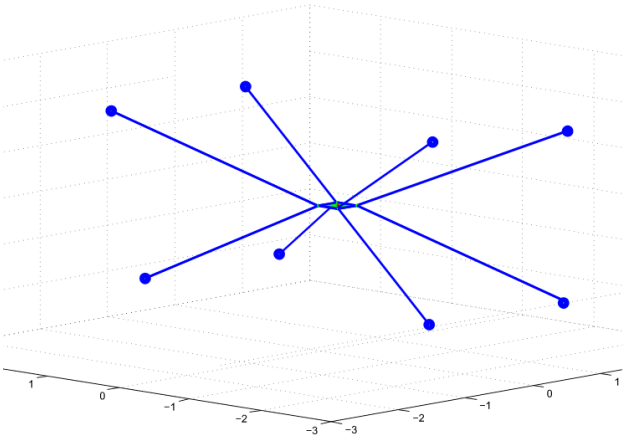
$$\mathbf{P}=\begin{bmatrix}c_\phi c_\psi & -s_\psi & 0\\ c_\phi s_\psi & c_\psi & 0\\ -s_\phi & 0 & 1\end{bmatrix} \quad (21)$$

Therefor the angular acceleration could be resolved from the following equation:

$$\begin{bmatrix}\alpha_1\\\alpha_2\\\alpha_3\end{bmatrix}=\dot{\mathbf{P}}\dot{\boldsymbol{\omega}}_c+\mathbf{P}\ddot{\boldsymbol{\omega}}_c \quad (22)$$

where $\alpha_i,\ i=\{1,2,3\}$ denotes the i^{th} components of angular acceleration. Using Eqs. (16) to (22), the following dynamic equation for the CDPR is derived:

$$-\mathbf{J}_{(\mathbf{q})}^T\boldsymbol{\tau}=\mathbf{M}_{(\mathbf{q})}\ddot{\mathbf{q}}+\mathbf{C}_{(\mathbf{q},\dot{\mathbf{q}})}\dot{\mathbf{q}}+\mathbf{M}_{(\mathbf{q})}\mathbf{G} \quad (23)$$



Slika 4. Parameters of the under study CDPR with eight cables.

where $G=[0\ 0\ 9.81\ 0\ 0\ 0]^T$ is the gravity vector, $\mathbf{q}$ is the vector of position, $\mathbf{M}_{(\mathbf{q})}$ denoted the mass matrix and is defined as follows [20] [22]:

$$\mathbf{M}_{(\mathbf{q})}=\begin{bmatrix}m\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{P}^T\mathbf{I}_\mathbf{P}\mathbf{P}\end{bmatrix} \quad (24)$$

where

$$\mathbf{I}_\mathbf{P}=\mathbf{R}\mathbf{I}\mathbf{R}^T \quad (25)$$

In addition, $\mathbf{C}_{(\mathbf{q},\dot{\mathbf{q}})}$ the so-called coriolis terms and centrifugal forces and $\mathbf{J}_{(\mathbf{q})}$ denoted the Jacobian matrix of the CDPR illustrated in Fig. 4, which compromise eight cables, are defined as follows:

$$\mathbf{C}_{(\mathbf{q},\dot{\mathbf{q}})}=\begin{bmatrix}0\\0\\0\\\mathbf{p}\mathbf{I}\dot{\mathbf{P}}\dot{\mathbf{q}}+\dot{\mathbf{P}}\dot{\mathbf{q}}\times\mathbf{I}\dot{\mathbf{P}}\dot{\mathbf{q}}\end{bmatrix} \quad (26)$$

$$\mathbf{J}_{(\mathbf{q})}=\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{P}\end{bmatrix} \quad (27)$$

The following equation represents the dynamic of the pulley [22]:

$$\mathbf{N}_\mathbf{z}\ddot{\boldsymbol{\theta}}+\mathbf{C}\dot{\boldsymbol{\theta}}=\boldsymbol{\tau} \quad (28)$$

where $\mathbf{N}_\mathbf{z}$ is the matrix of pulley inertia with respect to $z-$axis, θ is angel of each pulley and $\mathbf{C}$ is rotational viscous damping matrix which are defined due to the characteristics of the motor and system of winch and could be expressed as follows:

$$\mathbf{J}=\begin{bmatrix}J_1&0&0&0\\0&J_2&0&0\\0&0&J_3&0\\0&0&0&J_4\end{bmatrix},\quad \mathbf{C}=\begin{bmatrix}C_1&0&0&0\\0&C_2&0&0\\0&0&C_3&0\\0&0&0&C_4\end{bmatrix} \quad (29)$$

Based on the kinematic, static and dynamic equations derived here, the following part investigates the proposed controller for the eight cables CDPR. Next part provides the neural network estimator for the dynamic model with Taylor linearization method.

3 RADIAL BASIS FUNCTION NEURAL NETWORK

One of the promising estimators that could operates in real-time simulations are neural network with radial basis functions as the activation functions. They are highly accurate estimator and they could simulate any model including nonlinearities. A simple RBF neural network maps can be written according to the following equation [25]:

$$\mathbf{y} = \sum_{k=1}^{l} \mathbf{w}_k \boldsymbol{\Phi}_k(|x_i - m_{ik}|, \delta_{ik}) \tag{30}$$

where x_i, $i = 1,2,..,n$ and y contain the input variables and the output variable of the RBF neural network, respectively; w_k represents the connective weight between the hidden layer and the output layer; $\boldsymbol{\Phi}_k$ represents the firing weight of the k-th neuron in the hidden layer; and m_{ik} and δ_{ik} are the center and width of the activation function, respectively. The firing weight can be represented as:

$$\boldsymbol{\Phi}_k = e^{-net_k} \tag{31}$$

where

$$net_k = \sum_{i=1}^{n} \mathbf{s}_{ik}^2 [x_i - m_{ik}]^2 \tag{32}$$

in which $s_{ik} = 1 \,/\, \delta_{ik}$ is the inverse radius of the radial basis function. Toward simplification, the vectors m and s collecting all parameters of the hidden layer are defined as:

$$\mathbf{m} = [m_{11} \cdots m_{n1}\, m_{12} \cdots m_{n2} \cdots\cdots m_{1l} \cdots m_{nl}]^T \tag{33}$$

$$\mathbf{s} = [s_{11} \cdots s_{n1}\, s_{12} \cdots s_{n2} \cdots\cdots s_{1l} \cdots s_{nl}]^T. \tag{34}$$

Then, the output of the RBF neural network can be represented in a vector form as follows [26] and [27]:

$$\mathbf{y}(\mathbf{x}, \mathbf{m}, \mathbf{s}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\Phi}(\mathbf{x}, \mathbf{m}, \mathbf{s}) \tag{35}$$

where $\mathrm{x} = [x_1\, x_2\, ...\, x_n]^T$, $\mathbf{w} = [w_1\, w_2\, ...\, w_l]^T$ and $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1\, \boldsymbol{\Phi}_2\, ...\, \boldsymbol{\Phi}_l]^T$. It has been proven that there exists an RBF approximator of Eq.(35) such that it can uniformly approximate a nonlinear and even time-varying function $\boldsymbol{\Theta}$. Using the universal approximation theorem [28], there exists an optimal RBF approximator $\mathbf{y}^*$ such that [29]:

$$\begin{aligned} \boldsymbol{\Theta} &= \mathbf{y}^*(\mathbf{x}, \mathbf{m}^*, \mathbf{s}^*, \mathbf{w}^*) + \boldsymbol{\Delta} \\ &= \mathbf{w}^{*T} \boldsymbol{\Phi}^*(\mathbf{x}, \mathbf{m}^*, \mathbf{s}^*) + \boldsymbol{\Delta} \end{aligned} \tag{36}$$

where $\boldsymbol{\Delta}$ denotes an approximation error, $\mathbf{w}^*$ and $\boldsymbol{\Phi}^*$ are the optimal parameter vectors of $\mathbf{w}$ and $\boldsymbol{\Phi}$, respectively, and $\mathbf{m}^*$ and $\mathbf{s}^*$ are the optimal parameter vectors of $\mathbf{m}$ and $\mathbf{s}$, respectively. The optimal weighting vectors $\mathbf{w}^*$, $\mathbf{m}^*$ and $\mathbf{s}^*$ which are needed to best approximate a given nonlinear function $\boldsymbol{\Theta}$ are difficult to determine. An estimated RBF approximator is defined as the following form [29]:

$$\hat{\mathbf{y}} = \hat{\mathbf{w}}^T \hat{\boldsymbol{\Phi}}(\mathbf{x}, \hat{\mathbf{m}}, \hat{\mathbf{s}}) \tag{37}$$

where $\hat{\mathbf{w}}$ and $\hat{\boldsymbol{\Phi}}$ are the estimated vectors of $\mathbf{w}^*$ and $\boldsymbol{\Phi}^*$, respectively, and $\hat{\mathbf{m}}$ and $\hat{\mathbf{s}}$ are the estimated vectors of $\mathbf{m}^*$ and $\mathbf{s}^*$, respectively. The estimated error $\tilde{\mathbf{y}}$ is defined as:

$$\begin{aligned} \tilde{\mathbf{y}} &= \boldsymbol{\Theta} - \hat{\mathbf{y}} = \mathbf{y}^* - \hat{\mathbf{y}} + \boldsymbol{\Delta} \\ &= \tilde{\mathbf{w}}^T \hat{\boldsymbol{\Phi}} + \hat{\mathbf{w}}^T \tilde{\boldsymbol{\Phi}} + \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\Phi}} + \boldsymbol{\Delta} \end{aligned} \tag{38}$$

where $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ and $\tilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi}^* - \hat{\boldsymbol{\Phi}}$. In the following, some tuning laws will be developed to on-line tune the parameters of the RBF approximator to achieve favorable estimation. The Taylor expansion linearization technique is employed to transform the nonlinear function into a partially linear form [27]:

$$\tilde{\boldsymbol{\Phi}} = \begin{bmatrix} \tilde{\Phi}_1 \\ \tilde{\Phi}_2 \\ \vdots \\ \tilde{\Phi}_l \end{bmatrix} = \begin{bmatrix} \frac{\partial \Phi_1}{\partial m} \\ \frac{\partial \Phi_2}{\partial m} \\ \vdots \\ \frac{\partial \Phi_l}{\partial m} \end{bmatrix} |_{m=\hat{m}} \tilde{\mathbf{m}} + \begin{bmatrix} \frac{\partial \Phi_1}{\partial s} \\ \frac{\partial \Phi_2}{\partial s} \\ \vdots \\ \frac{\partial \Phi_l}{\partial s} \end{bmatrix} |_{s=\hat{s}} \tilde{\mathbf{s}} + \mathbf{H} \tag{39}$$

or

$$\tilde{\boldsymbol{\Phi}} = \mathbf{A}^T \tilde{\mathbf{m}} + \mathbf{B}^T \tilde{\mathbf{s}} + \mathbf{H} \tag{40}$$

where $\tilde{\mathbf{m}} = \mathbf{m}^* - \hat{\mathbf{m}}$, $\tilde{\mathbf{s}} = \mathbf{s}^* - \hat{\mathbf{s}}$, $\mathbf{H}$ is a vector of higher-order terms, $\mathbf{A} = \left[\frac{\partial \Phi_1}{\partial m} \frac{\partial \Phi_2}{\partial m} \cdots \frac{\partial \Phi_l}{\partial m}\right] |_{m=\hat{m}}$, $\mathbf{B} = \left[\frac{\partial \Phi_1}{\partial s} \frac{\partial \Phi_2}{\partial s} \cdots \frac{\partial \Phi_l}{\partial s}\right] |_{s=\hat{s}}$, $\frac{\partial \boldsymbol{\Phi}_k}{\partial \mathbf{m}}$ and $\frac{\partial \boldsymbol{\Phi}_k}{\partial \mathbf{s}}$ are defined as:

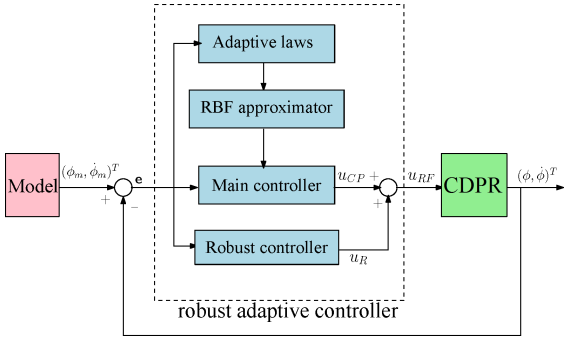
$$\left[\frac{\partial \boldsymbol{\Phi}_k}{\partial \mathbf{m}}\right]^T = \left[\underset{(k-1)\times l}{0 \cdots 0} \; \frac{\partial \Phi_k}{\partial m_{1k}} \cdots \frac{\partial \Phi_k}{\partial m_{nk}} \; \underset{(l-k)\times l}{0 \cdots 0}\right] \tag{41}$$

$$\left[\frac{\partial \boldsymbol{\Phi}_k}{\partial \mathbf{s}}\right]^T = \left[\underset{(k-1)\times l}{0 \cdots 0} \; \frac{\partial \Phi_k}{\partial s_{1k}} \cdots \frac{\partial \Phi_k}{\partial s_{nk}} \; \underset{(l-k)\times l}{0 \cdots 0}\right]. \tag{42}$$

Substituting Eq. (40) into Eq. (38), leads to:

$$\begin{aligned} \tilde{\mathbf{y}} &= \tilde{\mathbf{w}}^T \hat{\boldsymbol{\Phi}} + \hat{\mathbf{w}}^T (\mathbf{A}^T \tilde{\mathbf{m}} + \mathbf{B}^T \tilde{\mathbf{s}} + \mathbf{H}) + \tilde{\mathbf{w}}^T \tilde{\Phi} + \boldsymbol{\Delta} \\ &= \tilde{\mathbf{w}}^T \hat{\boldsymbol{\Phi}} + \tilde{\mathbf{m}}^T \mathbf{A} \hat{\mathbf{w}} + \tilde{\mathbf{s}}^T \mathbf{B} \hat{\mathbf{w}} + \varepsilon \end{aligned} \tag{43}$$

where $\hat{\mathbf{w}}^T \mathbf{A}^T \tilde{\mathbf{m}} = \tilde{\mathbf{m}}^T \mathbf{A} \hat{\mathbf{w}}$ and $\hat{\mathbf{w}}^T \mathbf{B}^T \tilde{\mathbf{s}} = \tilde{\mathbf{s}}^T \mathbf{B} \hat{\mathbf{w}}$, the sum of matching error $\varepsilon \equiv \hat{\mathbf{w}}^T \mathbf{H} + \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\Phi}} + \boldsymbol{\Delta}$. According to what was brought, in the early stages of the robot motion the obtained inaccurate model in Eq. (23) was used. The procedure of approximating the exact model by neural network



Slika 5. Block diagram for the proposed robust adaptive controller.

with RBF and its implementation is shown briefly in Algorithm 1, based on the input and output of ADAMAS MSC. The first line of the algorithm initials the formation of the neural network parameters according to Eqs. (30) to (33). Then, in the second line, the network weights in the range of one to minus one are defined. According to Taylor estimation method, the optimal values of the parameters of neural network based on the output of neural network training and model error is calculated. After that from the fifth to the last lines, the final neural network model is achieved based on the optimal error. Finally, the new model replaces with the model obtained in Eq. (23). It is worth noticing that this transmission should occur in an appropriate time with a change rate. This mainly depends on adaptive laws, rate of learning for the RBF and optimal coefficients of the designed neural network. The effectiveness of the adaptive control is to respond to changes in the system and modeling errors. The difference between adaptive control and robust control is the in the adaptive controller there is no need of knowing the range of the parameter or the errors. In other words, robust control leads to system stability in a certain range without having to change the rules of control, but with adaptive control method can be adapted to changing circumstances control rules so that the system remains stable. Adaptive controllers are divided into two categories, Direct and indirect methods. Nowadays, most of the articles like this article focuses on the direct adaptive control. Adaptive laws are experimentally determined according to the model of the robot. Usually, the exact model is obtained after five stages of identification based on inputs and outputs of the model. In the next section, robust adaptive control law is determined.

4 ROBUST ADAPTIVE CONTROL

Modern control theory is based on analysis of the differential equations in different systems. However, the stability of this type of system is sensitive to errors caused by

Algorithm 1 Procedure of obtaining exact model using RBF neural network.

Input: Φ_k, k, m_{ik} , δ_{ik} and ,*error*
Output: $\boldsymbol{Exact\ model}$

1. **Initialize** weight to small random values $[-1,1]$
2. **Find**$\hat{w}$ and $\hat{\Phi}$ using Taylor approximation.
3. **while** *error*> desired error
4. **For** each epoch (e)
5. **For** each input vector/pattern (p)
6. **Calculate** obtain (y) using Eq. (35)
7. **Calculate** weigths using Eq. (41)
8. **Calculate** optimal parameters of the network
9. **Update** network parameters using Eqs. (42) to (44).
10. **Update** model based on RBF m_{ik} and k
11. **End For**
12. **End For**
13. **End while**

the diffrences between the real system and its model. To solve the latter problem robust controller is used in most of the cases. In this type of controller, errors are considered from the beginning of the control procedure. Errors may occur and if they are in the expected range the system will remains in stable margins. Thus, the controller can handle the uncertainty in the model. This uncertainty can exist in both model and the measured values. This section outlines the robust adaptive controller law. The control problem can be equivalent to find a control law in such a way that the output ϕ can track the command trajectory, ϕ_m, where the tracking error vector is defined as [30]:

$$\mathbf{e} = [\ e(t) \quad \dot{e}\ (t)]^T = [\ \phi_m - \phi \quad \dot{\phi}_m - \dot{\phi}\ ]^T \qquad (44)$$

Toward representing some insights into the control procedure, the block diagram of the controller is illustrated in Fig. 5. Assume that the parameters of the under study CDPR in Eq. (44) are well defined, thus there exits an ideal controller as [31]:

$$u_{id} = -f(\phi, \dot{\phi}) + \ddot{\phi}_m + k_2\dot{e} + k_1 e \qquad (45)$$

Substituting Eq. (45) into Eq. (44), gives the following [31]:

$$\ddot{e} + k_2\dot{e} + k_1 e = 0 \qquad (46)$$

If k_1 and k_2 are chosen to correspond to the coefficients of a Hurwitz polynomial, it implies that $\lim_{t\to\infty} e(t) = 0$. However, the system dynamic function $f(\phi, \dot{\phi})$ in Eq. (45) is a nonlinear time-varying function and it cannot be exactly obtained, thus u_{id} cannot be implemented. An RBF approximator will be used to observe the unknown system dynamic function $f(\phi, \dot{\phi})$. By the universal approximation

theorem, there exists an optimal RBF neural network such that [28]:

$$f(\phi,\phi) = f^*(\ddot{}^*,t) + \varepsilon \tag{47}$$

where $\boldsymbol{\theta}^* = [\mathbf{w}^*\ \mathbf{m}^*\mathbf{s}^*]^T$ is the optimal weight vector of the RBF neural network and ε denotes the approximation error. However, the optimal weight vector is difficult to determine. Define the weight estimated error vector $\tilde{}$ as:

$$\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}} \tag{48}$$

where $\hat{\boldsymbol{\theta}} = [\hat{w}\,\hat{m}\,\hat{s}]^T$ is the estimated vector of the optimal vector $\boldsymbol{\theta}^*$. While ε appears, the following H^∞ tracking performance is requested [32]:

$$\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}dt \le \mathbf{e}^T(\mathbf{0})\mathbf{P}\mathbf{e}(\mathbf{0}) + \frac{1}{\kappa}\tilde{\boldsymbol{\theta}}^T(\mathbf{0})\tilde{\boldsymbol{\theta}}(\mathbf{0}) + \rho^2\int_0^T \varepsilon^2 dt \tag{49}$$

where $T \in [0,\infty]$ and $\varepsilon \in L_2[0,T]$. The $Q = Q^T$ and $P = P^T$ are given positive weighting matrices, κ is a design gain, and ρ is a prescribed attenuation level. If the system starts with initial conditions $e(0) = 0$ and $\tilde{\boldsymbol{\theta}}(0) = 0$, then the H^∞ performance in Eq. (49) can be rewritten as [33, 34]:

$$\sup_{\varepsilon\in L_2[0,T]} \frac{\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}\,dt}{\int_0^T \text{”}^2\,dt} \le \rho \tag{50}$$

where the L_2-gain from ε to the tracking error e should be equal or less than ρ. In order to achieving a favorable tracking performance and an arbitrarily small attenuation level simultaneously, a block diagram for robust adaptive controller is shown in Fig. 5.

$$u_{RF} = u_{CP} + u_R \tag{51}$$

in which the computation controller is chosen as [35]:

$$u_{CP} = -\hat{f}_{NN} + \ddot{\phi}_m + k_2\dot{\mathbf{e}} + k_1\mathbf{e} \tag{52}$$

with the dynamic approximator chosen as [36]:

$$\hat{f}_{NN}(\mathbf{e},\hat{\mathbf{m}},\hat{\mathbf{s}},\hat{\mathbf{w}}) = \hat{\mathbf{w}}^T\hat{\boldsymbol{\Phi}}(\mathbf{e},\hat{\mathbf{m}},\hat{\mathbf{s}}). \tag{53}$$

The estimated vectors $\hat{\mathbf{w}}$, $\hat{\mathbf{m}}$ and $\hat{\mathbf{s}}$ are the optimal vectors of $\mathbf{w}^*$, $\mathbf{m}^*$ and $\mathbf{s}^*$, respectively. By substituting Eq. (51) into Eq. (44) and using Eq. (45), the tracking error dynamic equation can be obtained as follows [33]:

$$\dot{\mathbf{e}} = \mathbf{A}_m\mathbf{e} - \mathbf{B}_m(f - \hat{f}_{NN} + u_R) \tag{54}$$

where $\mathbf{A}_m = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$, and $\mathbf{B}_m = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. Using Eq. (43), Eq. (54) can be rewritten as:

$$\dot{\mathbf{e}} = \mathbf{A}_m\mathbf{e} - \mathbf{B}_m(\tilde{\mathbf{w}}^T\hat{\boldsymbol{\Phi}} + \tilde{\mathbf{m}}^T\mathbf{A}\hat{\mathbf{w}} + \tilde{\mathbf{s}}^T\mathbf{B}\hat{\mathbf{w}} + \varepsilon + u_R) \tag{55}$$

Consider the proposed motion equations which are represented by Eq. (23) and (43). If the robust adaptive fuzzy control system is designed as Eq. (51), in which the adaptation laws of the system dynamic function approximator are designed as in Eq. (56) and Eq. (58), and the robust controller is designed as in Eq. (59). Henceforth, the stability of the system can be guaranteed [32]. The latter statement could be regarded as follows:

$$\hat{\mathbf{w}} = -\dot{\tilde{\mathbf{w}}} = -\eta_1\mathbf{e}^T P\mathbf{B}_m\hat{\boldsymbol{\Phi}} \tag{56}$$

$$\hat{\mathbf{m}} = -\dot{\tilde{\mathbf{m}}} = -\eta_2\mathbf{e}^T P\mathbf{B}_m\mathbf{A}\hat{\mathbf{w}} \tag{57}$$

$$\hat{\mathbf{s}} = -\dot{\tilde{\mathbf{s}}} = -\eta_3\mathbf{e}^T P\mathbf{B}_m\mathbf{B}\hat{\mathbf{w}} \tag{58}$$

$$u_R = \frac{1}{\kappa}\mathbf{B}_\mathrm{m}^T\mathbf{P}\mathbf{e} \tag{59}$$

where η_1, η_2 and η_3 are the learning rates with positive constants, κ is a positive weighting factor, and positive matrix $P = P^T$ is the solution of the following Riccati-like equation:

$$P\mathbf{A}_m + \mathbf{A}_m^T P + Q - \frac{2}{\kappa}P\mathbf{B}_m\mathbf{B}_m^T P + \frac{1}{\rho^2}P\mathbf{B}_m\mathbf{B}_m^T P = 0 \tag{60}$$

if and only if $\frac{2}{\kappa} - \frac{1}{\rho^2} \ge 0$ or $2\rho^2 \ge \kappa$ [35-37]. Therefore, for a prescribed ρ in H^∞ tracking control, in order to guarantee the solvability of H^∞ tracking performance, the weight κ on control law u_R of Eq. (59) should satisfy the above inequality. Then, the H^∞ tracking performance in Eq. (49) can be achieved for a prescribed attenuation level ρ. Aforementioned, the robust neuro-adaptive controller includes three different controlling efforts. These include the efforts to create stability by Lyapunov criterion, zero error and approximation of the system uncertainties. Robust controller approximate the uncertainties of the model while they are also decreased using RBF estimator. The robust adaptive controller design procedure is shown in algorithm 2 in general. In the first line, constants of neural network and robust controller are determined. Next, in the second line, the desired signal and its derivative are obtained. The third line consist of computing the new models based on the input and output of the model. Based on defined error control process begins. In lines 5 to 8, if five steps of identifying new model is past, the new model will be used to implement the controller; otherwise, the previous model is used. In the ninth line, the error vector of the identification is determined and based on this vector; future estimation for more accurate model is obtained. In the line ten, error of the controller signal is determined based Herwitz equation. The thirteenth line determines the controller output errors. The P matrix is determined for the next step. This procedure continues until achievement of the desired control. The parameters b_i for the dynamic coefficients in Eq. (44) are given in [32]. Solving the Riccati-like equation represented in Eq. (60) with $\rho^2 = \kappa$, one has:

$$\mathbf{Q} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \tag{61}$$

Algorithm 2 The designed robust adaptive controller

Input: ε $(desirederror)$, s, η_1, η_2, η_3 and ρ
Output: *Control signal for tracking desired path*

1. **Take** Input signal $(\phi_m, \dot{\phi}_m)$.
2. **Find** $\hat{f}_{NN}$ using RBF (New model) by η_1, η_2, η_3
3. **while** *error*$> \varepsilon$
4. **If** s < 5
5. **Use** the old model, $s = s+1$
6. **Else Use** the New model $\hat{f}_{NN}$
7. **End If**
8. **Define** θ error of the weight vector of RBF Eq. 41
9. **Define** model based on RBF m_{ik} and k
10. **Define** signal errors with Hurwitz Eqs. (45) and (51).
11. **Define** performance using ε and ρ with Eq. (52).
12. **Define** Obtain $u_R F$,$u_C P$ and with Eqs. (53) and (54).
13. **Compute** errors with Eq. (55) to (60)
14. **Calculate** P for the next input.
15. **End while**

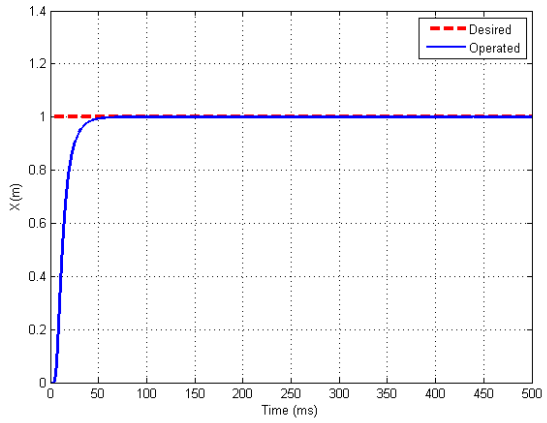
and

$$\mathbf{P} = \begin{bmatrix} 1.8 & 0.8 \\ 0.8 & 0.8 \end{bmatrix} \tag{62}$$

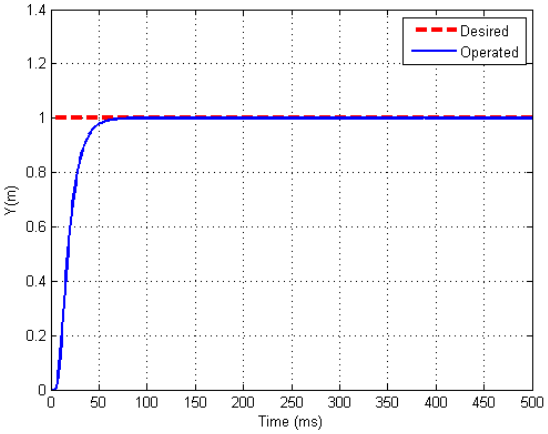
It should be emphasized that the derivation of controller equations does not need to use the dynamic parameters and the structure of the dynamic functions. The system parameters are used only for simulations. In order to investigate the effectiveness of the developed control system, initial conditions are simulated. The RBF neural network with seven neurons in hidden layer is utilized to approach the under study two-CDPR dynamics. The learning rates are selected as:

$$k_1 = 0.6, \;\; k_2 = 1.6, \;\; \eta_1 = \eta_2 = \eta_3 = 20 \tag{63}$$

In order to attenuate to a small level via H^∞ tracking design technique, the simulation results of the controller with $\kappa = 0.1$ are derived.



Slika 6. Desired and operated path for displacement in x-direction.



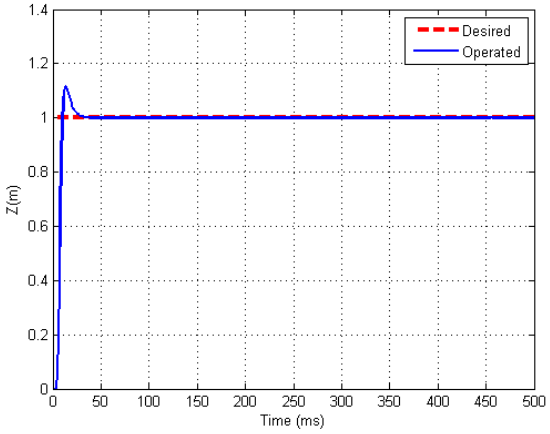
Slika 7. Desired and operated path for displacement in y-direction.

5 SIMULATION RESULTS

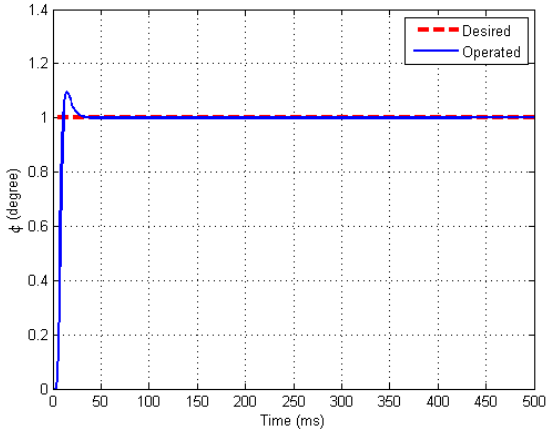
In this section, the proposed controller is implemented on the obtained plant. Figures 6 to 11 represents the operation of the robot in six degrees of freedom accordingly, which shows acceptable performance. The trajectory tracking for displacements are shown in Figs. 6, 7 and 8 in x, y and z directions accordingly. Moreover, Rotations around x, y and z axises are illustrated in Figs. 9, 10 and 11 respectively. The associated cables tensions are shown in Figs. 13. From these simulation results, it can be seen that robust tracking with high performance can be achieved without any knowledge of system dynamic functions. Moreover, obtained model verifies the trend of effort values obtained in Matlab and frequency response of the designed controller in constrained motion are represented in Fig. 12 which reveals acceptable range.

6 CONCLUSIONS

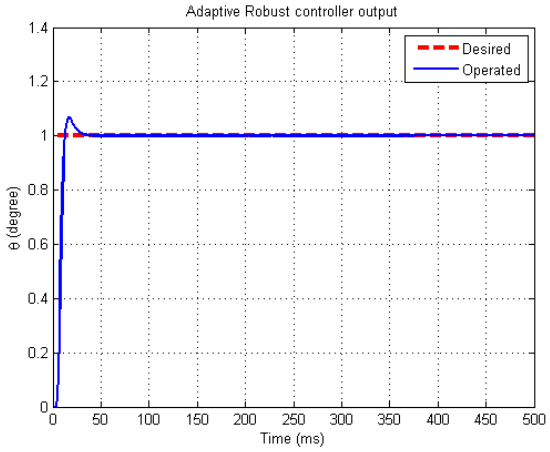
This paper developed a robust adaptive control system to attenuate the effects of the dynamic function approximation error on the tracking performance using H^∞ tracking technique. The robust adaptive control system is comprised of a computation controller and a robust controller. The computation controller including a neural-network-estimator is used to the system dynamic function and the robust controller is used to attenuate the effects of the approximation error. Furthermore, the proposed controller is entirely independent on the physical specifications of the



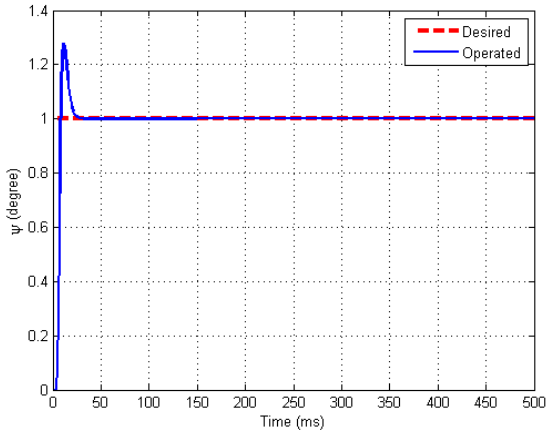
Slika 8. Step response for displacement in z-direction.



Slika 10. Step response for rotation around y-direction (ϕ).



Slika 9. Step response for rotation around x-direction (θ).



Slika 11. Step response for rotation around z-direction (ψ).

robot. In addition, the stability of the proposed controller was verified. Simulation results have shown effectiveness of this controller. Finally, the developed robust Neuro-adaptive control system was applied to control a CDPR, which demonstrates suitable performance. Ongoing works includes implementation of the proposed controller for real robot using vision procedure.

Literatura

[1] P. Dewdney, et, al. *"The Large Adaptive Reflector: A Giant Radio Telescope With An Aero Twist,", Canadian Aeronautics journal*,2002.

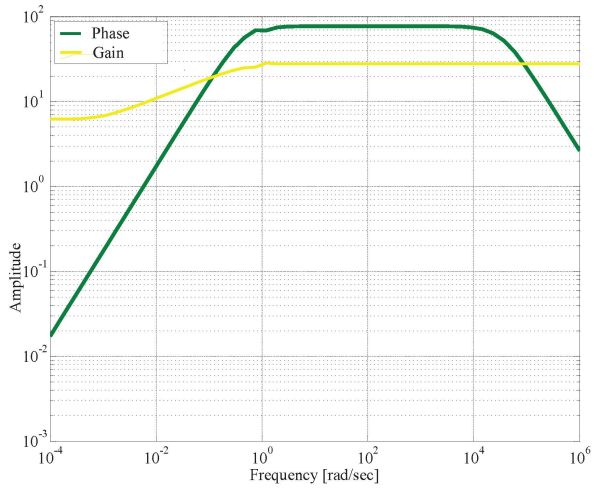
[2] Hassan Bayani, et, al. *"On the determination of the maximal inscribed ellipsoid in the Wrench-Feasible Workspace of the cable-driven parallel robots"*, Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), pp. 422-427, 2014.

[3] S. Kawamura, W. Choe, S. Tanaka, and S. R Pandian. *Development of an Ultrahigh Speed Robot Falcon Using Wire Drive System.*, International Conference on Robotics and Automation, Proceedings of IEEE,1995, volume 1, pp.215–220. IEEE, 1995.

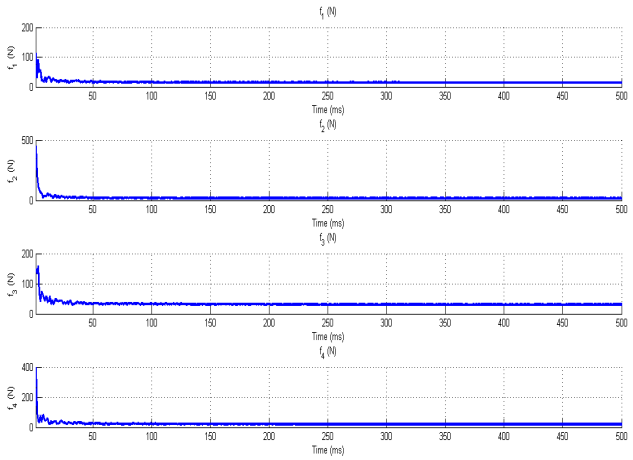
[4] Hassan Bayani, et, al. *"On the control of planar cable-driven parallel robot via classic controllers and tuning with intelligent algorithms "*, Third RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), pp. 623-628, 2015.

[5] Guo, Hongbo, et al. *"Cascade control of a hydraulically driven 6-DOF parallel robot manipulator based on a sliding mode," Control Engineering Practice 16.9 (2008): pp.1055-1068.*

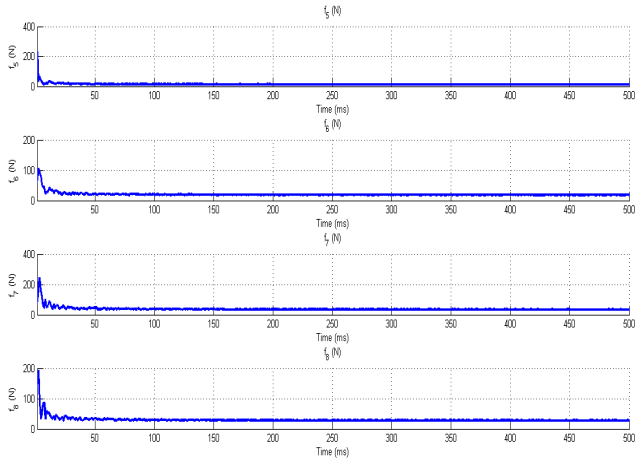
[6] J. Brest,et al. *Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems*, IEEE Transaction on computation, 2006.



Slika 12. The frequency response of the designed controller in constrained motion.



(a) Cables one to four tension.



(b) Cables five to eight tension.

Slika 13. Values of the cables tension.

[7] Maier, T., Woernle, C. *Dynamics and Control of a Cable Suspension Manipulator, The 9th German Japanese Seminar for Nonlinear Problems in Dynamical Systems , Germany* , 2000.

[8] Hiller, M., Fang, S., Mielczarek, S., Verhoeven, R., Franitza, D. *Design, Analysis and Realization of Tendon-based Parallel Manipulators*, Mechanism and Machine Theory, 40,425–445, 2005.

[9] Cheng-Dong, Li, Yi Jian-Qiang, Yu Yi, and Zhao Dong-Bin. *"Inverse Control of Cable-driven Parallel Mechanism Using Type-2 Fuzzy Neural Network." Acta Automatica Sinica 36, no. 3 (2010) pp.459-464.*

[10] R. L Williams and P.Gallina. Translational Planar Cable-direct-driven Robots. *Journal of Intelligent and Robotic systems*,2003.

[11] M. M. Eissa, G. S. Virk, A. M. AbdelGhany, E. S. Ghith, *Optimum Induction Motor Speed Control Technique Using Particle Swarm Optimization*, International Journal of Energy Engineering, 2013, pp.65–73.

[12] R. L Williams and P. Gallina. Planar Cable-direct-driven Robots, part i: Kinematic and Statics. ASME Conference, 2001.

[13] Ali Aflakiyan, et, al. *"Computed torque control of a cable suspended parallel robot"*, Third RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), pp. 749-754, 2015.

[14] Hassan Bayani, Mehdi Tale Masouleh, Ahmad Kalhor, *"Practical Performance Comparison of Pole placement and Sliding Mode Controller for Position Control of Cable-driven Parallel Robots Using Visual Servoing"*, Modares Mechanical Engineering, Vol.15, No.12, pp. 63-74, 2015.

[15] Hassan Bayani, Jalal Hajipour, *"On the design of an autopilot for roll motion in an airplane using LQR and flexible dynamic model"*, 3rd International Conference on Control, Instrumentation and Automation (ICCIA) , pp. 623-628, 2013.

[16] C., Harry and T, Dung , *"Web-based Interactive Analysis and Animation ofMechanisms," Journal of Computing and Information Science in Engineering, ASME*, vol. 6, no. 1, pp.84–90, 2006.

[17] Rasoul Sadeghian, Hassan Bayani, Mehdi Tale Masouleh, *"Design of an adaptive sliding mode controller for a novel spherical rolling robot"*, Third RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), pp. 529-534, 2015.

[18] Khosravi, Mohammad A., and Hamid D. Taghirad. *"Experimental Performance of Robust PID Controller on a Planar*

Cable Robot." In Cable-Driven Parallel Robots, pp. 337-352. Springer, 2013.

[19] Z. Jinhua , Z. Jian, D. Haifeng, W. Sunan, *Self-organizing genetic algorithm based tuning of PID controllers*, Journal of Information Sciences, 2009, VOL.179, NO.2, pp.1007–1018.

[20] Reza Babaghasabha, Mohammad A. Khosravi and Hamid D. Taghirad. *"Adaptive Control of KNTU Planar Cable-Driven Parallel Robot with Uncertainties in Dynamic and Kinematic Parameters"*,Mechanisms and Machine Science, 2011.

[21] Guo, Hongbo, et al. *"Cascade control of a hydraulically driven 6-DOF parallel robot manipulator based on a sliding mode," Control Engineering Practice 16.9 (2008): pp.1055-1068.*

[22] Korayem, Moharam Habibnejad and Tourajizadeh, H *"Maximum DLCC of Spatial Cable Robot For a Predefined Trajectory Within The Workspace Using Closed Loop Optimal Control Approach', Journal of Intelligent & Robotic Systems,springer*, vol. 63, no. 1, pp. 75–99, 2011.

[23] Middleton, RH and Goodwin, GC *"Adaptive computed torque control for rigid link manipulations"*, Elsevier,1988.

[24] Yang, Zhiyong, Jiang Wu, and Jiangping Mei. *"Motor-mechanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator."*Mechatronics 17.7 (2007).

[25] Reiner, Philip David. *"Algorithms for Optimal Construction and Training of Radial Basis Function Neural Networks"*, Diss. Auburn University, 2015.

[26] Raj, Devi R., I. Jacob Raglend, and M. Anand. *"Inverse kinematics solution of a five joint robot using feed forward and Radial Basis Function Neural Network"*, International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), 2015.

[27] Daachi, M.E., Madani, T., Daachi, B. and Djouani, K., *"A radial basis function neural network adaptive controller to drive a powered lower limb knee joint orthosis."*, Applied Soft Computing, pp.324-336. , 2015.

[28] Reiner, Philip David. *"Algorithms for Optimal Construction and Training of Radial Basis Function Neural Networks"*, Diss. Auburn University, 2015.

[29] Huang, Hsu-Chih, and Chih-Hao Chiang. *"An Evolutionary Radial Basis Function Neural Network with Robust Genetic-Based Immunecomputing for Online Tracking Control of Autonomous Robots."*, Neural Processing Letters, pp.1-17, 2015.

[30] Chen S, Yao B, Zhu X, Chen Z, Wang Q, Zhu S, Song Y. *"Adaptive Robust Backstepping Force Control of 1-DOF Joint Exoskeleton for Human Performance Augmentation."*, IFAC-PapersOnLine, No.480, Vol.19, 2016.

[31] Liang, Yang, Liu Zhi, and Zhang Yun. *"Adaptive fuzzy yaw moment control of humanoid robot based on ankle joint"*, 34th Chinese Control Conference (CCC), pp.5999-6004. IEEE, 2014.

[32] Fradkov, A.L., Miroshnik, I.V. and Nikiforov, V.O., *" Nonlinear and adaptive control of complex systems"*, Springer Science and Business Media, Vol. 491, 2013.

[33] Vukobratovic, Miomir, Dragan Stokic, and Nenad Kircanski. *"Non-adaptive and adaptive control of manipulation robots"*,Springer Science and Business Media, Vol. 5, 2013.

[34] Fateh, Mohammad Mehdi, and Saeed Khorashadizadeh. *"Robust control of electrically driven robots by adaptive fuzzy estimation of uncertainty"*, Nonlinear Dynamics, Vol.69, No.3, pp. 1465-1477, 2012.

[35] Schenk, Christian, Heinrich H. Bulthoff, and Carlo Masone. *"Robust adaptive sliding mode control of a redundant cable driven parallel robot"*, 19th International Conference on System Theory, Control and Computing (ICSTCC), 2015.

[36] Salas-Pena, Oscar, Herman Castaneda, and Jesus de Leon-Morales. *"Robust Adaptive Control for a DC Servomotor with wide Backlash Nonlinearity"*, Automatika: Journal for Control, Measurement, Electronics, Computing and Communications, Vol.56, No.4, 2015.

[37] Kuljaca, Ognjen, Krunoslav Horvat, and Bruno Borovic. *"Design of adaptive neural network controller for thermal power system frequency control"*, Automatika: Journal for Control, Measurement, Electronics, Computing and Communications, Vol.52, No.4, 2012.

M.H.Barhaghtalab Mojtaba Hadi Barhaghtalab received his B.Sc. degree in Electrical Engineering (Communication orientation) from Islamic Azad University, Science and Research Branch, Iran (2012) and M.Sc. degree in Electrical Engineering (Control orientation) from Persian Gulf University, Bushehr, Iran (2016). His main research interests include robotic, intelligent system, fuzzy control, neural network, adaptive control, nonlinear control, stability analysis of systems.

H.Bayani received the B.S. Eng. Degree in Mechanical Engineering from the Iran University of Science and Technology (IUST) Tehran, Iran, in 2011. He worked on designing & building of a model helicopter as Bachelor degree thesis design & build of an R/C helicopter with minimum plane vibration. Then he received M.Sc. of Mechatronics from Tehran University in 2015. His research interests include kinematics, dynamics and control of robotic mechanical systems with a particular emphasis on the Cable driven parallel mechanism. His works is based on applying new mathematical tools, such as convex optimization and heuristic algorithms to the end of providing real time algorithm for control and identification of this type of robot. Also parts of his works include image processing and machine vision.

Armin Nabaei Armin Nabaei is currently pursuing his Master of electrical engineering from Azad University in Tehran, Iran. His main research interest includes intelligent embedded systems and VLSI.

Houman Zarrabi received his doctoral of engineering from Concordia University in Montreal, Canada in 2011. Since then he has been involved in various industrial and research projects. His main expertise includes IoT, M2M, big data, embedded systems and VLSI. He is currently the national IoT program director and an assistant professor in ITRC.

Ali Amiri received the B.Sc. Degree in Electrical Engineering from the K. N. Toosi University of Technology Tehran, Iran, in 2011. Currently, he is pursuing his Ph.D. from Tehran University. His research interests include control of complex system. His work is mainly on using different control approaches like MPC in ROS.

AUTHORS' ADDRESSES

Mojtaba Hadi Barhaghtalab, M.Sc. Department of Electrical Engineering, School of Engineering, Persian Gulf University, Bushehr, Iran. mh.barhaghtalab@gmail.com
Hassan Bayani*, M.Sc. Mechatronic Engineering, FNST, University of Tehran, Tehran, Iran.mailto:hassan_bayani@ut.ac.ir
Armin Nabaei, Azad University, Tehran, Iran armin.nabaei@srbiau.ac.ir
Houman Zarrabi, ICT Research Institute, Tehran, Iran h.zarrabi@itrc.ac.ir
Ali Amiri, Department of Systems and Control, Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran. a.amiri@ut.ac.ir

Received: 2016-05-09
Accepted: 2017-01-10