TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 6, December 2020, pp. 3191~3201
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i6.14913

Radial basis function neural network control for parallel spatial robot

Nguyen Hong Quang[1], Nguyen Van Quyen[2], Nguyen Nhu Hien[3]
[1,3]Thai Nguyen University of Technology, Vietnam
[2]Hanoi University of Science and Technology, Vietnam

Article Info

Article history:

Received Dec 9, 2019
Revised May 30, 2020
Accepted Jun 25, 2020

Keywords:

Inverse dynamics controller
Kronecker product
Numerical simulation
Parallel robot manipulator
RBF neural network control

ABSTRACT

The derivation of motion equations of constrained spatial multibody system is an important problem of dynamics and control of parallel robots. The paper firstly presents an overview of the calculating the torque of the driving stages of the parallel robots using Kronecker product. The main content of this paper is to derive the inverse dynamics controllers based on the radial basis function (RBF) neural network control law for parallel robot manipulators. Finally, numerical simulation of the inverse dynamics controller for a 3-RRR delta robot manipulator is presented as an illustrative example.

Corresponding Author:

Nguyen Hong Quang,
Thai Nguyen University of Technology,
No. 666, 3/2 Street, Tich Luong Ward, Thai Nguyen City, Vietnam.
Email: quang.nguyenhong@tnut.edu.vn

1. INTRODUCTION

In the past three decades, the theory on dynamics of constrained multibody systems has been developed to a high degree of maturity [1-4]. The parallel robot manipulators are constrained multibody structures [5-7]. The equations of motion for a multibody system are obtained as the end result of a sequence of mathematical operators. In general, the known methods to derive the equations of motion of multibody systems are Lagrange's equations, Newton–Euler equations, Kane's equations. Among these methods, the approach using Lagrange's equations with multipliers has become an attractive method to derive the equations of motion of constrained multibody systems. This approach provides a well analytical and orderly structure that is very useful for control purposes.

The control of treelike multibody systems is of interest to a number of research communities in a very of applications areas. Many advanced methods for control of robot manipulators based on the Lagrange's equations have been developed [8-19]. The application of modern control methods such as sliding mode control method, the neural network control method for controller design of the treelike robot manipulators is presented in the works [20-30]. In contrast to the rapid progress in control theory of treelike robot manipulators, the development of the control theory for parallel robots is still limited. Modern control methods have also been used in the control problem of plane parallel manipulators [31-34]. One has used the control methods such as the proportional derivative (PD) control and proportional integral derivative (PID) control for designing some controllers of spatial parallel robot manipulators [35-38]. However, the application of modern control methods such as sliding mode control method, the radial basis function (RBF) neural network control method for controller design of the spatial parallel robot manipulators is a new problem that has not been investigated.

Recently, N. H. Quang [39-41] proposed a control method using model predictive approach. AL-Azzawi [42] address the control problem for a class of nonlinear dynamical systems based on linear feedback control strategies. S. Riache [43] proposed adaptive nonsingular terminal super-twisting controller consists of the hybridization of a nonsingular terminal sliding mode control and an adaptive super twisting. Simulations with nonsingular terminal super-twisting control to prove the superiority and the effectiveness of the proposed approach. In [44], a new compound hierarchical sliding mode control and fuzzy logic control scheme has been proposed for a class of underactuated systems with mismatched uncertainties.

In the present study, we present a control method using neural network for controller design of spatial parallel robot manipulators. In the section 2, the application of the new matrix form of Lagrangian equations with multipliers for constrained multibody systems to establish a new expression for calculation of the driving torques of parallel robots will be discussed. The inverse dynamics controller for the parallel robot manipulator is considered in the section 3. In the section 4, numerical simulation of the inverse dynamics controller for a 3-RRR delta parallel spatial robot manipulator is presented as an illustrative example.

2. INVERSE DYNAMICS OF CONSTRAINED MULTIBODY SYSTEMS

Let us consider a scleronomic multibody system of $f = n_a$ degree of freedom containing p rigid-bodies with r holonomic constraints. Let $s = [s_1, s_2, \dots, s_n]^T$ be the vector of generalized coordinates, the motion equations of constrained holonomic multibody systems can be written as:

$$M(s)\ddot{s} + C(s,\dot{s})\dot{s} + g(s) + \Phi_s^T(s)\lambda + d(s,\dot{s}) = \tau \quad (1)$$

$$f(s) = 0 \quad (2)$$

where $M(s)$ is the $n \times n$ mass matrix, $C(s,\dot{s})$ is the $n \times n$ coriolis/centripetal matrix, f is $r \times 1$ vector of constraint equations, $\Phi_s(s)$ is the $r \times n$ Jacobian matrix of the vector f , $\boldsymbol{d}$ is the $n \times 1$ vector of friction force and disturbance, $\boldsymbol{\tau}$ is the $n \times 1$ vector of driving forces/torques, $\boldsymbol{\lambda}$ is the $r \times 1$ vector of Lagrangian multipliers. The Coriolis/Centripetal matrix $C(s,\dot{s})$ is determined from the mass matrix according the following formula [45, 46].

$$C(s,\dot{s}) = \frac{\partial M(s)}{\partial s}(E_n \otimes \dot{s}) - \frac{1}{2}\left[\frac{\partial M(s)}{\partial s}(\dot{s} \otimes E_n)\right]^T \quad (3)$$

The Jacobian matrix $\Phi_s(s)$ of the constrained equations is determined by the following formula;

$$\Phi_s = \frac{\partial f}{\partial s} = \begin{bmatrix} \frac{\partial f_1}{\partial s_1} & \cdots & \frac{\partial f_1}{\partial s_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_r}{\partial s_1} & \cdots & \frac{\partial f_r}{\partial s_n} \end{bmatrix} \quad (4)$$

Firstly, the generalized coordinates in vector $\boldsymbol{s}$ are divided into two subgroups: independent coordinates q_a, and redundant coordinates z. Then we have;

$$s = [q_a^T \quad z^T]^T, \boldsymbol{q}_a = \begin{bmatrix} q_1 & \dots & q_f \end{bmatrix}^T, \boldsymbol{z} = \begin{bmatrix} z_1 & \dots & z_r \end{bmatrix}^T, n = f + r \quad (5)$$

By differentiating in (2) with respect to vectors s, q_a, z, respectively, we obtain the following Jacobian matrices,

$$\boldsymbol{\Phi}_s = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{s}} \in \mathbb{R}^{r \times n}, \boldsymbol{\Phi}_z = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} \in \mathbb{R}^{r \times r}, \boldsymbol{\Phi}_a = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}_a} \in \mathbb{R}^{r \times f}, \Phi_s = [\Phi_a \quad \Phi_z] \quad (6)$$

By introducing the projection matrix [47]:

$$R(s) = \begin{bmatrix} E \\ -\Phi_z^{-1}\Phi_a \end{bmatrix} \in R^{n \times f} \quad (7)$$

one has:

$$R^T(s)\Phi_s^T(s) = 0, \quad (8)$$

where $\boldsymbol{E}$ is the $f \times f$ identity matrix.

Left multiplication of the motion in (1) with the matrix $\boldsymbol{R}^T\ \boldsymbol{s}$ yields,

$$\begin{aligned} R^T[M(s)\ddot{s} + C(s,\dot{s})\dot{s} + g(s) + \Phi_s^T(s)\lambda + d(s,\dot{s})] &= R^T\begin{bmatrix}\tau_a\\ \tau_z\end{bmatrix} \\ &= [E \quad [\Phi_z^{-1}\Phi_a(s)]^T]\begin{bmatrix}\tau_a\\ \tau_z\end{bmatrix} \\ &= \tau_a - [\Phi_z^{-1}\Phi_a(s)]^T\tau_z \end{aligned} \tag{9}$$

where $\boldsymbol{\tau}_a$ is the vector of the driving forces/torques in active joints and τ_z is the vector of the forces/torques in passive joints. Making use of in (8) and assuming that $\tau_z = 0$, the driving torques can be deduced from (9) as,

$$\tau_a = R^T[M(s)\ddot{s} + C(s,\dot{s})\dot{s} + g(s) + d(s,\dot{s})] \tag{10}$$

3. ADAPTIVE RBF NEURAL NETWORK CONTROL BASED ON INVERSE DYNAMICS FOR PARALLEL ROBOTS

3.1. Transformation of motion equations

To study the stability of the control algorithms, the motion equations of parallel robots are transformed into a suitable form. Let us consider a scleronomic constrained multibody system. From the constrained in (2) we get;

$$\boldsymbol{f}(s) = \boldsymbol{f}(\boldsymbol{q}_a, \boldsymbol{z}) = \boldsymbol{0}, \quad \dot{\boldsymbol{f}} = \boldsymbol{\Phi}_a\dot{\boldsymbol{q}}_a + \boldsymbol{\Phi}_z\dot{\boldsymbol{z}} = \boldsymbol{0} \tag{11}$$

Assuming that the Jacobian matrix Φ_z is nonsingular, det $(\Phi_z) \neq 0$. From (11) one may obtain,

$$\dot{z} = -\Phi_z^{-1}\Phi_a\dot{q}_a \tag{12}$$

It is noted that,

$$q_a = Eq_a \tag{13}$$

Combining (12) with (13) yields the following differential equation:

$$\dot{s} = R(s)\dot{q}_a \tag{14}$$

Differentiating in (14) with respect to time gives the acceleration relation as;

$$\ddot{s} = R(s)\ddot{q}_a + \dot{R}(s,\dot{s})\dot{q}_a = R(s)\ddot{q}_a + \frac{\partial R(s)}{\partial s}(E_p \otimes \dot{s})\dot{q}_a \tag{15}$$

Substituting in (14) and (15) into to (9) yields;

$$\boldsymbol{R}^T(s)\left[\boldsymbol{M}\ \ s\ (\boldsymbol{R}(s)\ddot{\boldsymbol{q}}_a + \frac{\partial \boldsymbol{R}(s)}{\partial \boldsymbol{s}}(\boldsymbol{E}_p \otimes \dot{\boldsymbol{s}})\dot{\boldsymbol{q}}_a) + \boldsymbol{C}\ \ s,\dot{s}\ \ \boldsymbol{R}(s)\dot{\boldsymbol{q}}_a + \boldsymbol{g}\ \ s\ \ + \boldsymbol{d}(s,\dot{s})\right] = \boldsymbol{\tau}_a \tag{16}$$

To simplify the description, we define;

$$\begin{aligned} \bar{M}(s) &:= R^T(s)M(s)R(s)\bar{C}(s,\dot{s}) \\ &:= R^T(s)\left[M(s)\frac{\partial R(s)}{\partial s}\left(E_p \otimes \dot{s}\right) + C(s,\dot{s})R(s)\right]\bar{g}(s): \\ &= R^T(s)g(s)\ \bar{d}(s,\dot{s}) := R^T(s)d(s,\dot{s}) \end{aligned} \tag{17}$$

In (9) and (11) now can be rewritten as follows;

$$\bar{M}(s)\ddot{q}_a + \bar{C}(s,\dot{s})\dot{q}_a + \bar{g}(s) + \bar{d}(s,\dot{s}) = \tau_a \tag{18}$$

$$f(s) = 0 \tag{19}$$

The motion equations of parallel robots (18) and (19) are called the motion equations in mixture form. Where s is the vector of redundant generalized coordinates and q_a is the vector of independent coordinates.

We will use this equation as the basis for designing the controller for parallel robots. For this purpose, we prove the following properties [33]:
– $\bar{M}$ is a symmetric positive definite matrix: $\bar{M}^T = \bar{M}$,
– $\dot{\bar{M}} - 2\bar{C}$ is a skew-symmetric matrix: $(\dot{\bar{M}} - 2\bar{C})^T = -(\dot{\bar{M}} - 2\bar{C})$.
Due to the symmetry of the matrix $\boldsymbol{M}$ is symmetric, one has;

$$\bar{M}^T(s) = [R^T(s)M(s)R(s)]^T = R^T(s)M(s)R(s) = \bar{M}(s)$$

Since $M(s)$ is positive definite, $\bar{M}(s)$ is also a positive definite matrix. Using the (17), one obtains;

$$\begin{aligned}\dot{\bar{M}}(s) - 2\bar{C}(s,\dot{s}) &= \dot{R}^T MR + R^T \dot{M}R + R^T M\dot{R} - 2R^T\left(M\dot{R} + CR\right)\\ &= \dot{R}^T MR + R^T \dot{M}R - R^T M\dot{R} - 2R^T CR\\ &= R^T(\dot{M} - 2C)R - R^T M\dot{R} + \dot{R}^T MR\end{aligned} \tag{20}$$

Since $\dot{M} - 2C$ is skew-symmetric [8], from in (20) one has;

$$\begin{aligned}\left[\dot{\bar{M}}(s) - 2\bar{C}(s,\dot{s})\right]^T &= \left[R^T(\dot{M} - 2C)R\right]^T - (R^T M\dot{R})^T + (\dot{R}^T MR)^T\\ &= -R^T(\dot{M} - 2C)R - \dot{R}^T MR + R^T M\dot{R} = -[\dot{\bar{M}}(s) - 2\bar{C}(s,\dot{s})]\end{aligned} \tag{21}$$

Thus, $\dot{\bar{M}}(s) - 2\bar{C}(s,\dot{s})$ is a skew symmetric matrix.

3.2. RBF neural network control law and stability analysis

In practice, the perfect robot model could be difficult to obtain, and external disturbances are always present in practice. The uncertain motion equations of parallel robots with $n_a = f$ active joints (18) can be described in the following form;

$$\hat{M}(s)\ddot{q}_a + \hat{C}(s,\dot{s})\dot{q}_a + \hat{g}(s) + \hat{d}(s,\dot{s}) = \tau_a \tag{22}$$

where $\hat{M}(s)$ is an $f \times f$ inertia matix, $\hat{C}(s,\dot{s})$ is an $f \times f$ matrix containing the centrifugal and Coriolis terms, $\hat{g}(\boldsymbol{s})$ is an $f \times 1$ vector containing gravitational forces and torques, $\boldsymbol{s}$ is the vector of generalized coordinates, $\boldsymbol{q}_a$ is active joint coordinates, and $\hat{\boldsymbol{d}}$ denotes disturbances. It is supposed that,

$$\begin{aligned}&\hat{M}(s) = \bar{M}(s) + \Delta\bar{M}(s)\\ &\hat{C}(s,\dot{s}) = \bar{C}(s,\dot{s}) + \Delta\bar{C}(s,\dot{s})\\ &\hat{g}(s) = \bar{g}(s) + \Delta\bar{g}(s)\\ &\hat{d}(s,\dot{s}) = \bar{d}(s,\dot{s}) + \Delta\bar{d}(s,\dot{s})\end{aligned} \tag{23}$$

where $\bar{\boldsymbol{M}}$, $\bar{\boldsymbol{C}}$, $\bar{\boldsymbol{g}}, \bar{\boldsymbol{d}}$ are the prior-known components and $\Delta\bar{\boldsymbol{M}}$, $\Delta\bar{\boldsymbol{C}}$, $\Delta\bar{\boldsymbol{g}}, \Delta\bar{\boldsymbol{d}}$ are modeling errors of $\hat{\boldsymbol{M}}$, $\hat{\boldsymbol{C}}$, $\hat{\boldsymbol{g}}$ and $\hat{\boldsymbol{d}}$ respectively. Assume that the modeling errors are bounded by some finite constants as;

$$\left\|\Delta\bar{\boldsymbol{M}}\right\| \le m_0,\ \left\|\Delta\bar{\boldsymbol{C}}\right\| \le c_0,\ \left\|\Delta\bar{\boldsymbol{g}}\right\| \le g_0,\ \left\|\Delta\bar{\boldsymbol{d}}\right\| \le \mathrm{d}_0 \tag{24}$$

where m_0, c_0, g_0, d_0 are known constants. Substituting in (23) into to (22) yields.

$$(\bar{M} + \Delta\bar{M})\ddot{q}_a + (\bar{C} + \Delta\bar{C})\dot{q}_a + \bar{g} + \Delta\bar{g} + \bar{d} + \Delta\bar{d} = \tau_a \tag{25}$$

From (25) one has;

$$\bar{M}(s)\ddot{q}_a + \bar{C}(s,\dot{s})\dot{q}_a + \bar{g}(s) + \bar{d}(s,\dot{s}) + \bar{h}(s,\dot{s}) = \tau_a \tag{26}$$

where $\bar{h}(s,\dot{s})$ is the sum of unknown terms of the dynamic system.

$$\bar{h}(s,\dot{s}) = \Delta\bar{M}\ddot{q}_a + \Delta\bar{C}\dot{q}_a + \Delta\bar{g} + \Delta\bar{d} \quad (27)$$

Assume that $\|\bar{h}(s,\dot{s})\| \leq h_0$. The sliding mode function is selected as;

$$n(t) = \dot{e}_a(t) + \Lambda e_a(t) \quad (28)$$

where $\boldsymbol{\Lambda}$ is the positive diagonal matrix.

$$\boldsymbol{\Lambda} = \text{diag}\ \lambda_1, \lambda_2, \ldots, \lambda_{na} \quad , \quad \lambda_i > 0\ ; \quad i = 1,2,\ldots,n_a \quad (29)$$

In (28) we define,

$$\boldsymbol{e}_a\ t\ = \boldsymbol{q}_a\ t\ - \boldsymbol{q}_a^d\ t \quad (30)$$

where $q_a^d(t)$ is the vector of desired trajectory and $q_a(t)$ is the vector of real trajectory. The function $\bar{h}(s,\dot{s})$ can be rewritten as:

$$\bar{h}(n) := \bar{h}(s,\dot{s}) \quad (31)$$

The function $\bar{h}(n)$ is the main reason for the degradation of the control quality. If this effect is compensated, the control accuracy can then be improved. According to Stone-Weierstrass theorem [23-24] one can choose an appropriate artifical neural network (ANN) with a limited number of neurals that can approximate an unknown nonlinear function with a given accuracy. For approximating function $\bar{h}(n)$we choose the following simple structure ANN:

$$\bar{h}(n) = Ws + e = \hat{h}(n) + e \quad (32)$$

where W is the $n_a \times n_a$ matrix, $\hat{h}(n) = [\hat{h}_1, \hat{h}_2, .., \hat{h}_{na}]^T = Ws$ is the approximation of $\bar{h}(n)$, e is the approximation error. If $\|\bar{h}(n)\| \leq h_0$, we have $\|e\| \leq \varepsilon_0$. Assuming that the matrix W has n_a column vectors w_i, we have;

$$\hat{h} = [\hat{h}_1, \hat{h}_2, .., \hat{h}_{na}]^T = W\sigma = \sum_{i=1}^{n_a} \sigma_i w_i \quad (33)$$

In this paper, the radial basis function (RBF) neural network was used as shown in Figure 1. This structure has been proved to satisfy the Stone-Weierstrass theorem [23]. If we choose the Gaussian activation function σ_i according to the formula,

$$\sigma_i = exp\left[-\frac{\|\boldsymbol{v}-\boldsymbol{c}_i\|^2}{\chi_i^2}\right] \quad (34)$$

where the vector c_i represents the coordinate value of the center point of the Gaussian function of neural net i, and χ_i is derivation parameter which is freely choosen, the function approximation $\hat{\boldsymbol{h}}$ has the following form;

$$\hat{h}_i = \sum_{j=1}^{n_a} \sigma_j w_{ji}\ ,\ i = 1..,n_a \quad (35)$$

where w_{ji} are the weights to be updated of the approximating neural network.

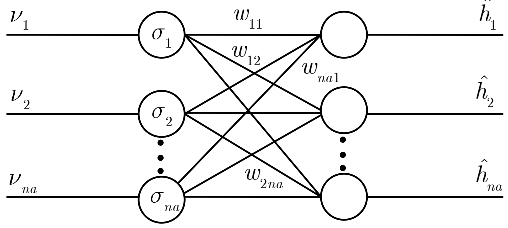

Figure 1. RBF neural network structure

The control problem is now to find the control torque $\boldsymbol{u}$ and learning algorithm of w_{ji} of the neural network so that $n \to 0$ and position errors $e \to 0$, granting $q_a(t) \to q_a^d(t)$.

Theorem: The trajectory $q_a(t)$ of dynamic system defined by (3.34) with RFB neural network according to (3.60), (3.62), and the sliding surface (3.40) will track the desired trajectory $q_a^d(t)$ with error $e(t) = q_a(t) - q_a^d(t) \to 0$ if the control law $\boldsymbol{u}$ the learning algorithm $\boldsymbol{w}_i$ are chosen as follows;

$$u = \bar{M}(s)\ddot{q}_a^d + \bar{C}(s,\dot{s})\dot{q}_a^d + \bar{g} + \bar{d} - \bar{M}(s)\Lambda\dot{e}_a - \bar{C}(s,\dot{s})\Lambda e_a - Kn - \gamma\frac{n}{\|n\|} + (1+\eta)Ws \tag{36}$$

$$\dot{w}_i = -\eta\sigma_i n \tag{37}$$

where K *is a* $n_a \times n_a$ symmetric positive matrix, and $\eta > 0, \gamma > 0$. Noting that the states $q_a, \dot{q}_a$ in the control law (36) are measured.

Proof: This theorem can be proved using the Lyapunov direct method. We choose the Lyapunov function as;

$$V(t) = \frac{1}{2}\left[n^T\bar{M}n + \sum_{i=1}^{n_a} w_i^T w_i\right] \tag{38}$$

Since $\bar{M}(s)$ is symmetric and positive definite $V(t) > 0$ for $n \neq 0, w_i \neq 0$ and $V(t) = 0$ if and only if $n = 0, w_i = 0$.

The derivative of the function $V(t)$ is,

$$\dot{V}(t) = n^T\bar{M}\dot{n} + \frac{1}{2}n^T\dot{\bar{M}}n + \sum_{i=1}^{n_a} w_i^T \dot{w}_i \tag{39}$$

Using the skew-symmetry of the matrix $\bar{M}(s) - 2\bar{C}(s,\dot{s})$, we have

$$n^T\left(\dot{\bar{M}} - 2\bar{C}\right)n = 0 \to n^T\dot{\bar{M}}n = 2n^T\bar{C}n \tag{40}$$

Substituting (40) into (39) gives:

$$\dot{V}(t) = n^T(\bar{M}\dot{n} + \bar{C}n) + \sum_{i=1}^{n_a} w_i^T \dot{w}_i \tag{41}$$

If we choose $\boldsymbol{u} = \boldsymbol{\tau}_a$, from (38) and (26) one has,

$$\bar{M}\dot{n} + \bar{C}n = -\left[Kn + \gamma\frac{n}{\|n\|} - (1+\eta)Ws + \bar{h}(n)\right] \tag{42}$$

Substituting in (42) into to (41) yields,

$$\dot{V}(t) = n^T\left[-Kn - \gamma\frac{n}{\|n\|} + \eta W\sigma - e\right] + \sum_{i=1}^{n_a} w_i^T \dot{w}_i \tag{43}$$

Using the learning algorithm (37), the last term in (43) has the following form,

$$\sum_{i=1}^{n_a} w_i^T \dot{w}_i = -\eta\sum_{i=1}^{n_a} w_i^T n\sigma_i = -\eta n^T ws \tag{44}$$

Substituting (44) into to (43) yields,

$$\dot{V}(t) = -n^T Kn - \gamma\frac{n^T n}{\|n\|} - n^T e \tag{45}$$

If we select $\gamma = \delta + \varepsilon_0$, and $\delta > 0$, one obtains: $\dot{V}(t) = -n^T Kn - \delta\|n\| - (\varepsilon_0\|n\| + n^T e)$ (46)

Since $\|e\| < \varepsilon_0, \dot{V}(t) < 0$ for all $n \neq 0$, and $\dot{V}(t) = 0$ if and only if $n = 0$. It follows from Lyapunov's theory that the system is asymptotically stable, or $n \to 0$ as $t \to \infty$, therefore,

$$\boldsymbol{e}_a\ t = \boldsymbol{q}_a\ t - \boldsymbol{q}_a^d\ t \to \boldsymbol{0} \tag{47}$$

4. SIMULATION EXAMPLE

From the RBF neural network control law presented in the above section, the resulting block scheme is illustrated in the Figure 2. For the use of the control law $\boldsymbol{u}$ based on (36) and (37), actual signals $q_a, \dot{q}_a, \ddot{q}_a$ are assumed to be known. Therefore, one can obtain actual values of generalized coordinates, velocities and accelerations $s, \dot{s}, \ddot{s}$. The numerical simulation may be a possible way to suggest an alternative choice for actual values of generalized coordinates, velocities and accelerations. Considering the dynamic equations of parallel robot manipulator as;

$$M(s)\ddot{s} + C(s,\dot{s})\dot{s} + g(s) + \Phi_s^T(s)\lambda + d(s,\dot{s}) = \tau \tag{48}$$

$$f(s) = 0 \tag{49}$$

where,

$$\tau = \tau(t, s, \dot{s}, s^d, \dot{s}^d) = [u^T, 0^T]^T \tag{50}$$

Differentiating in (49) with respect to time gives,

$$\dot{f}(s) = \frac{\partial f}{\partial s}\dot{s} = \Phi_s\dot{s} = 0 \tag{51}$$

$$\ddot{f}(s,\dot{s}) = \Phi_s\ddot{s} + \dot{\Phi}_s\dot{s} = 0 \tag{52}$$

where [45, 46]:

$$\dot{\Phi}_s(s) = \frac{\partial \Phi}{\partial s}(E_n \otimes \dot{s}) \tag{53}$$

Define,

$$p_1(s, \dot{s}, s^d, \dot{s}^d, t) = \tau - C(s,\dot{s})\dot{s} - g(s) - d(s,\dot{s}) \tag{54}$$

$$p_2(s, \dot{s}, t) = -\dot{\Phi}_s(s)\dot{s} = -\left[\frac{\partial \Phi_s}{\partial s}(E_n \otimes \dot{s})\right]\dot{s} \tag{55}$$

In (48) and (52) now can be written in the following form,

$$M(s)\ddot{s} + \Phi_s^T(s)\lambda = p_1(s, \dot{s}, s^d, \dot{s}^d, t) \tag{56}$$

$$\Phi_s(s)\ddot{s} = p_2(s, \dot{s}, t) \tag{57}$$

Left multiplication of (56) with the matrix R^T yields,

$$R^T(s)M(s)\ddot{s} + R^T(s)\Phi_s^T(s)\lambda = R^T(s)p_1(s, \dot{s}, t) \tag{58}$$

According to (8), (58) becomes:

$$R^T(s)M(s)\ddot{s} = R^T(s)p_1(s, \dot{s}, s^d, \dot{s}^d, t) \tag{59}$$

In (59) is a system of f second-order differential equations. Combining in (59) with (57) yields,

$$\begin{bmatrix} R^T(s)M(s) \\ \Phi_s(s) \end{bmatrix}\ddot{s} = \begin{bmatrix} R^T(s)p_1 \\ p_2 \end{bmatrix} \tag{60}$$

If the matrix,

$$A(s) = \begin{bmatrix} R^T(s)M(s) \\ \Phi_s(s) \end{bmatrix} \tag{61}$$

is nonsigular, from (60) one obtains the following diferential equation system,

$$\ddot{s} = \ddot{s}(s, \dot{s}, s^d, \dot{s}^d, t) \tag{62}$$

Then, solving the (62) we find $s, \dot{s}$ [48]. Therefore we can calculate control law according to (36).

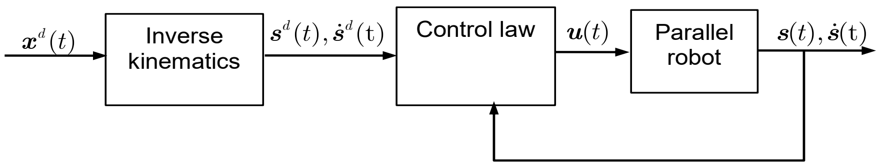

Figure 2. Block scheme of joint space control

A 3-RRR spatial parallel robot manipulator shown in Figure 3 is utilized in this study to verify the effectiveness of the proposed control scheme. The mechanical model for the 3-RRR delta robot manipulator is a system of rigid bodies connected by joints as Figure 4. The parallelogram mechanisms that connect the driving links to the mobile platform are modeled as homogeneous rods with universal and spherical joints at two ends. From Figures 4 and 5 it is followed that the configuration of the 3-RRR delta spatial parallel robot manipulator is represented by a vector of generalized coordinates as:

$$\boldsymbol{s} = \left[\psi_1 \;\; \psi_2 \;\; \psi_3 \;\; \gamma_1 \;\; \gamma_2 \;\; \gamma_3, x_P \;\; y_P \;\; z_P\right]^T$$

The differential-algebraic equations of the system are given in the Appendix. The kinematic and dynamic parameters of the robot manipulator are given in the Table 1. In the simulation, the center of the moving platform will be controlled to track the given trajectory defined by,

$$x_P = 0.3\cos \; 2\pi t \; ; \; y_P = 0.3\sin \; 2\pi t \; ; \; z_P = -0.7 \; (m)$$

The parameters of the neural network control law are chosen as follows,

$$\boldsymbol{K} = diag \;\; 80, 80, 80 \;\; ; \; \boldsymbol{L} = diag(80, 80, 80); \; \eta = 1.1; \; \gamma = 200;$$
$$\chi_1 = 1; \chi_2 = 2; \chi_3 = 3; \; c_1 = 0.01; c_2 = 0.02; c_3 = 0.03;$$

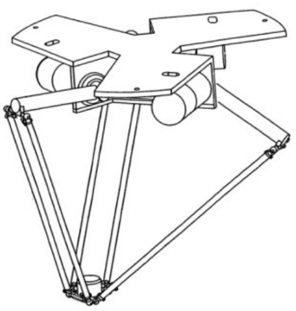

Figure 3. Delta robot with three parallelogram mechanisms

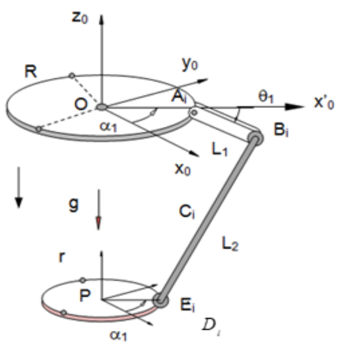

Figure 4. Model of 3-RRR Delta robot

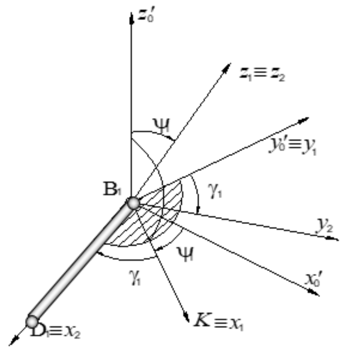

Figure 5. Position of the B_iD_i rod in the space

Table 1. The parameters of the Delta robot in Figure 3

L_1	L_2	R	r r	α_1 α_1	α_2	α_3	m_1 m_1	m_2 m_2	m_P
$0.3\ (m)$	$0.8\ (m)$	$0.266\ (m)$	$0.04\ (m)$	$0\ (rad)$	$\frac{2\pi}{3}(rad)$	$\frac{4\pi}{3}(rad)$	$0.42\ (kg)$	$2 \cdot 0.2\ (kg)$	$0.75\ (kg)$

In this paper the modeling errors are chosen to be 20% of the prior-known values of the nominal model as,

$$\Delta M(s) = 20\% M(s); \Delta C(s,\dot{s}) = 20\% C(s,\dot{s}); \Delta g(s) = 20\% g(s)$$

The disturbance vector is chosen as $\boldsymbol{d} = \left[\kappa_1 \sin 20t \quad \kappa_1 \cos 20t \; \ldots \; \kappa_6 \sin 20t \quad \kappa_6 \cos 20t\right]^T$. Some simulation results are given in the Figures from 6 to 9. The position errors of the moving platform are shown in Figures 6 and 7. The control torques are shown in Figures 8 and 9. The stationary errors in position of the platform are kept about 10^{-4} mm.

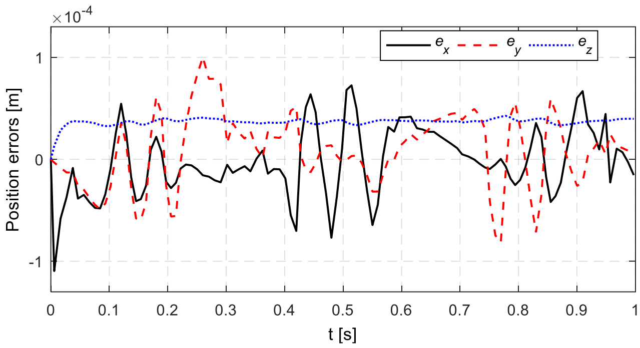

Figure 6. Position errors of the moving platform without the modeling errors and disturbance

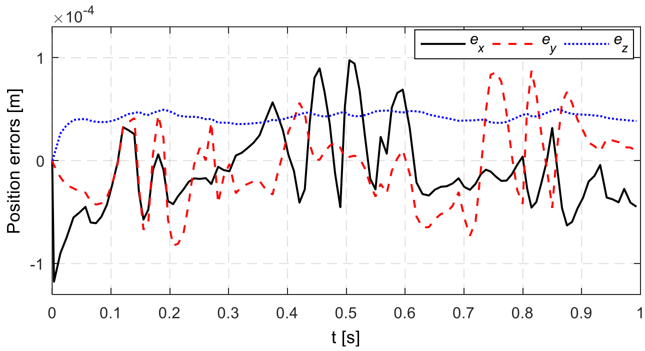

Figure 7. Position errors of the moving platform with the modeling errors and disturbance

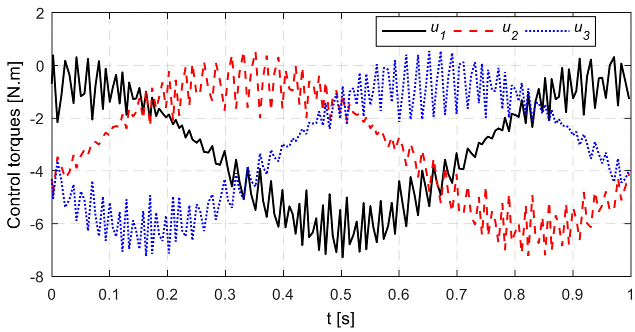

Figure 8. Control torques without the modeling errors and disturbance

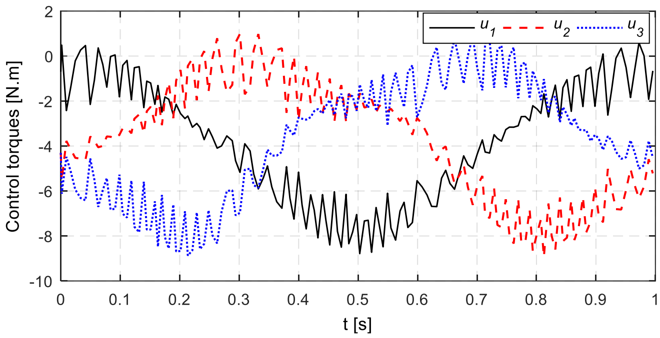

Figure 9. Control torques with the modeling errors and disturbance

5. CONCLUSIONS

Many modern methods for control of robot manipulators based on the Lagrangian multipliers have been developed. In contrast to the rapid progress in control theory of treelike robot manipulators, the development of the modern control theory for parallel robot manipulators is still limited. This paper presented the application of the RBF neural network control law to compensate uncertainties in the parallel robot manipulators. The new matrix form of Lagrangian equations with multipliers for constrained multibody systems was used to derive dynamic equations of spatial parallel robot manipulators based on computer software packages. Using adaptive RBF neural network control method, the controller for spatial robot manipulators based on inverse dynamics was developed. The stability of the control law using adaptive RBF neural network method for the control problem of spatial robot manipulators based on inverse dynamics was proven. Using Simulink program, the numerical simulation of the adaptive RBF neural network controller for a 3-PRR spatial parallel robot manipulator is studied. The application of modern methods for motion control of the constrained spatial multibody systems and spatial parallel robot manipulators will be presented in other works.

ACKNOWLEDGEMENTS

This research was supported by Research Foundation funded by Thai Nguyen University of Technology.

REFERENCES

[1] J. G. Jalon, E. Bayo, "Kinematic and Dynamic Simulation of Multibody Systems – The Real-Time Challenge," New York : *Springer-Verlag*, 1994.
[2] W. Schiehlen (Editor), "Multibody Systems Handbook," Berlin : *Springer*, 1990.
[3] A. A. Shabana, "Dynamics of Multibody Systems," 3rd ed. New York : *Cambridge University Press*, 2005.
[4] W. Schiehlen, P. Eberhard, "Applied Dynamics," Switzerland : *Springer International Publishing*, 2014.
[5] L.-W. Tsai, "Robot Analysis/The Mechanics of Serial and Parallel Manipulators," New York : *John Wiley & Sons*, 1999.
[6] J.-P. Merlet, "Parallel Robots," 2nd ed. Berlin : *Springer*, 2006.
[7] M. Ceccarelli, "Fundaments of Mechanics of Robotic Manipulation," Dordrecht : *Springer*, 2004.
[8] R. M. Muray, Z. Li, S. S. Sastry, "A Mathematical Introduction to Robotic Manipulation," Boca Raton : *CRS Press*, 1994.
[9] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, "Robotics/ Modelling, Planning and Control," London : *Springer-Verlag*, 2009.
[10] F. L. Lewis, D. M. Dawson, Ch. T. Abdallah, "Robot Manipulator Control/ Theory and Practice," 2nd ed. New York : *Marcel Dekker*, 2004.
[11] R. Kelly, V. Santibanez, A. Loria, "Control of Robot Manipulators in Joint Space," London : *Springer-Verlag*, 2005.
[12] H. Asada, J. E. Stotine, "Robot analysis and control," New York : *John Wiley and Sons*, 1985.
[13] J. E. Slotine, W. Li, "Applied Nonlinear Control," New York : *Prentice Hall*, 1991.
[14] V. I. Utkin, "Sliding Modes in Control and Optimization," Berlin : *Springer-Verlag*, 1992.
[15] A. Astolfi, D. Karagiannis, R. Ortega, "Nonlinear and Adaptive Control with Applications," London : *Springer-Verlag*, 2008.
[16] S. S. Ge, C. C. Hang, T. H. Lee, "Stable Adaptive Neural Network Control, " New York : *Springer*, 2002.
[17] S. S. Ge, T. H. Lee, C. J. Harris, "Adaptive Neural Network Control of Robotic Manipulators," Singapore : *World Scientific Publishing*, 1998.
[18] F. L. Lewis. S. Jagannathan, A. Yesildirek, "Neural Network Control of Robotic Manipulators and Nonlinear Systems," London : *Taylor & Francis*, 1999.
[19] An-Ch. Huang, M.-Ch. Chien, "Adaptive Control of Robot Manipulators," Singapore : *World Scientific*, 2010.
[20] J. Liu, X. Wang, "Advanced Sliding Mode Control for Mechanical System," *Tsinghua University Press*, 2012.
[21] J. Liu, "Radial Baisis Function (RBF) Neural Network Control for Mechanical Systems," *Tsinghua University Press*, 2013.
[22] H. A. Talebi, R. V. Patel, K. Khorasani, "Contol of Flexible-link Manipulators Using Neural Networks," London : *Springer-Verlag*, 2001.

[23] N. E. Cotter, "The Stone-Weierstrass theory and its applications to neural networks," IEEE *Transactions on Neural Network*, vol. 1, no. 4, pp. 290-295, 1990.
[24] T. Ozaki, S. Suzuki, T. Furuhashi, S. Okuma, Y. Uchikawa, "Trajectory control of robotic manipulators using neural networks," IEEE *Transactions on Industrial Electronics*, vol. 38, no. 3, pp. 195-202, 1991.
[25] O. Barambones, V. Etxebarria, "Robust neural control for robotic manipulators," *Automatica*, vol. 38, no.2, pp. 235-242, 2002.
[26] R. J. Wai, "Tracking control based on neural network strategy for robot manipulator," *Neurocomputing,* vol. 51, pp. 425-445, 2003.
[27] F. Sun, Z. Sun, P. Y. Woo, "Neural network-based adaptive controller design of robotic manipulators with an observer," *IEEE Transactions on Neural Networks*, vol. 12, no. 1, pp. 54-67, 2001.
[28] P. T. Cat, N. T. Hiep, "Robust PID sliding mode control of robot manipulators with online learning neural networks," *European Control Conference,* pp. 2187-2192, 2009.
[29] Z. H. Jiang, "A neutral network controller for trajectory control of industrial robot manipulators," *Journal of Computers*, vol. 3, no. 8, pp. 1-8, 2008.
[30] Z. H. Jiang, "Trajectory control of robot manipulators using a neural network controller," *Robot Manipulators Trends and Development*, pp. 361-376, 2010.
[31] G. Liu, Z. Li, "A unified geometric approach to modeling and control of constrained mechanical systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 574-587, 2002.
[32] N. V. Khang, L. A. Tuan, "On the sliding mode control of redundant parallel robots using neural networks," *3rd IFToMM International Symposium on Robotics and Mechatronics*, pp.168-177, 2013.
[33] A. Müller, T. Hufnagel, "Model-based control of redundantly actuated parallel manipulators in redundant coordinates," *Robotics and Autonomous Systems,* vol. 60, no. 4, pp. 563-571, 2013.
[34] A. Zubizarreta, *et al*., "A redundant dynamic model of parallel robots for model-based control," *Robotica*, vol. 31, no. 2, pp. 203-216, 2013.
[35] G. Lebret, K. Liu, F. L. Lewis, "Dynamic analysis and control of a Stewart platform manipulator," *Journal of Robotic Systems*, vol. 10, no. 5, pp. 629-655, 1993.
[36] S.-H. Lee, *et al*., "Position control of a Stewart platform using inverse dynamics control with approximate dynamics," *Mechatronics*, vol. 13, no. 6, pp. 605-619, 2003.
[37] A. Ghobakhloo, M. Eghtesad, M. Azadi, "Position control of a Stewart-Gough platform using inverse dynamics method with full dynamics," *IEEE Conference 9th Int. Workshop on Advanced Motion Control (AMC'06)*, pp. 50-55, 2006.
[38] A. Mohsen, M. A. Ardestani, A. Mersad, "Dynamics and control of a novel 3-DOF spatial parallel robot," *2013 RSI/ISM Int. Conference on Robotics and Mechatronics*, pp. 183-188, 2013.
[39] N. H. Quang, *et al.,* "Multi parametric model predictive control based on laguerre model for permanent magnet linear synchronous motors," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 2, pp. 1067-1077, 2018.
[40] N. H. Quang, *et al*., "On Tracking Control Problem for Polysolenoid Motor Model Predictive Approach," International *Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 849-855, 2020.
[41] N. H. Quang, *et al*., "Min Max Model Predictive Control for Polysolenoid Linear Motor," *International Journal of Power Electronics and Drive Systems,* vol. 9, no. 4, pp. 1666-1675, 2018.
[42] AL-Azzawi, *et al*., "Strategies of linear feedback control and its classification," *TELKOMNIKA Telecommunication Computing Electronics and Control,* vol. 17, no. 4, pp.1931-1940, 2019
[43] S. Riache, *et al*., "Adaptive robust nonsingular terminal sliding mode design controller for quadrotor aerial manipulator," *TELKOMNIKA Telecommunication, Computing, Electronics and Control,* vol. 17, no. 3, pp. 1501-1512, 2019.
[44] D. H. Vu, *et al*., "Hierarchical robust fuzzy sliding mode control for a class of simo under-actuated systems with mismatched uncertainties," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 6, pp. 3027-3043, 2019.
[45] N. V. Khang, "Consistent definition of partial derivatives of matrix functions in dynamics of mechanical systems," *Mechanism and Machine Theory*, vol. 45, no. 7, pp. 981-988, 2010.
[46] N. V. Khang, "Kronecker product and a new matrix form of Lagrangian equations with multipliers for constrained multibody systems," *Mechanics Research Communications*, vol. 38, no. 4, pp. 294-299, 2011.
[47] W. Blajer, *et al*., "A projective criterion to the coordinate partitioning method for multibody dynamics," *Archive of Applied Mechanics*, vol. 64, pp. 86-98, 1994.
[48] J. Baumgarte "Stabilization of constraints and integrals of motion in dynamioc systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 1, pp. 1-16, 1972.