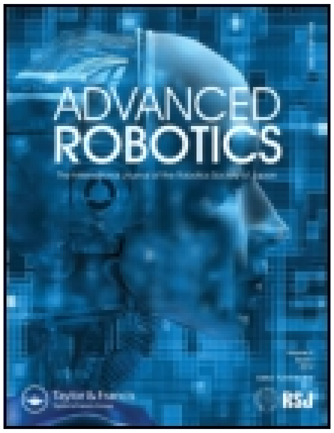


Advanced Robotics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tadr20

An improved hybrid method for forward kinematics analysis of parallel robots

Iman Kardan[a] & Alireza Akbarzadeh[a]

[a] Center of Excellence on Soft Computing and Intelligent Information Processing, Mechanical Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran

Published online: 23 Mar 2015.

To cite this article: Iman Kardan & Alireza Akbarzadeh (2015) An improved hybrid method for forward kinematics analysis of parallel robots, Advanced Robotics, 29:6, 401-411, DOI: 10.1080/01691864.2014.994034

To link to this article: http://dx.doi.org/10.1080/01691864.2014.994034

Advanced Robotics, 2015
Vol. 29, No. 6, 401–411, http://dx.doi.org/10.1080/01691864.2014.994034

FULL PAPER

An improved hybrid method for forward kinematics analysis of parallel robots

Iman Kardan* and Alireza Akbarzadeh

Center of Excellence on Soft Computing and Intelligent Information Processing, Mechanical Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran

(*Received 9 January 2014; revised 10 May 2014, 11 September 2014 and 13 November 2014; accepted 26 November 2014*)

This paper combines a new structure of artificial neural networks (ANNs) with a 3rd-order numerical algorithm and proposes an improved hybrid method for solving forward kinematics problem (FKP) of parallel manipulators. In this method, an approximate solution of the FKP is first generated by the neural network. This solution is next considered as an initial guess for the 3rd-order numerical technique which solves the nonlinear forward kinematics equations and obtains the answer with a desired level of accuracy. To speed up the method, a new structure is proposed for designing the ANN which is called Same Class One Network. In this structure, the outputs of the ANN are classified into classes of similar variables with an individual network designed for each class. The proposed method is then applied to a planar 3-RPR parallel manipulator and a spatial 3-PSP parallel robot. The results show that using this method will lead to a 55% reduction in required iterations and a 20% reduction in the FKP analysis time, while maintaining a high level of solution accuracy.

Keywords: forward kinematics of parallel robots; improved hybrid method; artificial neural networks; SCON structure; 3rd-order numerical algorithm

1. Introduction

Because of their unique properties like high precision and high load to weight ratio, parallel robots have found wide applications in different industries such as simulators,[1] machine tools,[2] and CNCs.[3] Obtaining a precise and fast forward kinematics model is an essential step in modeling and control of the parallel robots, especially for real-time applications. By decreasing the required time for calculation of kinematic parameters, more time can be devoted to calculations of the control algorithm. Therefore, complex algorithms with more computation loads and better performances can be employed to control the parallel robots. Thus, finding a way to reduce the computation time of the forward kinematics analysis of parallel robots has always been in the focus of researchers.

The forward kinematics problem (FKP) of parallel robots usually includes highly nonlinear equations which cannot be analytically solved. Therefore, in most cases numerical techniques are used to solve this problem.[4–10] Among different numerical algorithms, the Newton–Raphson method is widely used in the forward kinematics analysis of parallel robots.[5–10] The main disadvantage of this method is its high sensitivity to initial guesses. In fact, for some initial guesses the algorithm may diverge, resulting in a decrease in performance. Various methods like the robust Newton–Raphson,[11] the global Newton–Raphson,[12] and the modified global Newton–Raphson [13] are proposed to overcome this limitation. Although the divergence problem is somehow resolved in these methods, the algorithms will take a long time to converge if the initial guess falls away from exact solution.

An effective way for increasing the convergence speed of numerical algorithms is to provide them with more accurate initial guesses. Guez and Ahmad [14] have proposed a hybrid strategy which uses artificial neural networks (ANNs) to prepare suitable initial guesses for inverse kinematics analysis of serial robots. By combining ANNs and the Newton–Raphson algorithm, this method largely resolves the problems with the accuracy of ANNs and the convergence speed of numerical algorithms.

Parikh and Lam [15] extended the hybrid method to the forward kinematics analysis of parallel robots and applied it to a Gough–Stewart parallel platform. They showed that using the hybrid strategy reduces the duration of the FKP analysis to two-thirds of the time required for solving the same problem by the Newton–Raphson technique.

It is well known that the Newton–Raphson method is a 2nd order numerical algorithm.[16] Thus, by using higher order methods it is possible to achieve the desired accuracy in less iteration and consequently increase the solution speed. In this paper by combining ANNs and a 3rd-order numerical algorithm,[17] an improved hybrid

*Corresponding author. Email: Iman.Kardan@stu.um.ac.ir

strategy is proposed for solving the FKP of parallel robots. In the proposed method, first the ANNs generate an approximate solution of the problem. Then, the 3rd-order numerical technique uses this approximate solution as an initial guess to find the exact answer with the desired level of accuracy.

ANNs are commonly designed based on two different strategies of ACON[1] and OCON[2]. Generally the OCON approach provides better accuracies but increases time and resource use.[15] In order to keep a balance between speed and accuracy, in this paper a new ANN structure (SCON[3]) is proposed in which outputs are categorized based on their nature and one individual network is designed for each output category.

The performance of the improved hybrid method is evaluated for a planar 3-RPR parallel manipulator and a spatial 3-PSP parallel robot. It is shown that compared with the hybrid strategy,[15] the proposed method achieves the desired accuracy in less iteration and shorter time. Therefore, the main contributions of this paper can be summarized as:

- The Newton–Raphson algorithm of hybrid strategy [15] is replaced with a 3rd-order numerical technique to speed up the method.
- The SCON structure for ANNs is proposed for the first time in this paper. This structure provides a balance between network's accuracy and its computation time.

This paper is organized as follows: Section 2 proposes the improved hybrid strategy. The 3rd-order numerical technique and the proposed SCON structure of ANNs are described in Subsections 2.1 and 2.2, respectively. Structure and FKP of the two sample robots, 3-RPR parallel manipulator, and 3-PSP robot is briefly introduced in Section 3. Section 4 evaluates the performance of the proposed method in comparison with the hybrid strategy [15] and the Newton–Raphson algorithm and demonstrates the advantages of the improved hybrid method. Finally, Section 5 concludes the paper.

2. Improved hybrid method for solving the FKP of parallel robots

It is well known that forward kinematics model of parallel robots is generally composed of some closed kinematic chains and can be summarized as:

$$\mathbf{F}(\mathbf{q}^{ac}, \mathbf{x}) = 0 \tag{1}$$

in which $\mathbf{F}^{n\times 1} : \Omega \subseteq \Re^n \rightarrow \Re^n$ is a nonlinear vector function and $\mathbf{x}^{n\times 1} = [x^1, x^2, \dots, x^n]^{\mathrm{T}}$ is the vector of unknown kinematic parameters of the robot, namely unactuated joints variables as well as end-effector position and orientation. $\mathbf{q}^{ac}$ is an m by 1 vector of actuated joints variables which is known as the input to the forward kinematics problem. Therefore, the system of Equation (1) can be rewritten as:

$$\mathbf{F}(\mathbf{x}) = 0 \tag{2}$$

which can be solved by the methods used in solving system of nonlinear equations.

As previously stated, Parikh and Lam [15] used a hybrid strategy for FKP analysis of parallel robots in which initial guesses for the Newton–Raphson algorithm are provided by ANNs. Similar to [15], in this paper ANNs are used to generate initial guesses. However, in order to further reduce the solution time, SCON structure is proposed for designing the neural networks while the Newton–Raphson method is replaced with a 3rd-order numerical technique. A graphical representation of the proposed method is shown in Figure 1.

The mathematics of the 3rd-order numerical algorithm and the SCON structure of the ANNs are described in Subsections 2.1 and 2.2, respectively.

2.1. *Numerical algorithm with 3rd-order convergence*

Using Adomian decomposition, Chun [18] has reviewed some numerical algorithms with different convergence orders. Amongst these methods, Darvishi and Barati [17] extended a 3rd-order algorithm which does not require second derivative of the equations, and applied it to system of nonlinear equations. In the present work, this algorithm is used to solve the system of nonlinear equations of FKP of parallel robots. Considering the equation (2), Darvishi and Barati [17] proposed the iterative relation of the equation (3) for solving this system of equations, and proved that this algorithm benefits from a 3rd-order convergence.

$$\mathbf{x}_{m+1} = \mathbf{x}_m - \mathbf{F}'(\mathbf{x}_m)^{-1}\left(\mathbf{F}(\mathbf{x}_m) + \mathbf{F}\left(\mathbf{x}^*_{m+1}\right)\right), \tag{3}$$

$$\mathbf{x}^*_{m+1} = \mathbf{x}_m - \mathbf{F}'(\mathbf{x}_m)^{-1}\mathbf{F}(\mathbf{x}_m). \tag{4}$$

In the above equations, $\mathbf{F}'$ is the $n \times n$ Jacobian matrix of the system of Equation (2) and is computed as:

$$(F')^{i,j} = \frac{\partial F^i}{\partial x^j} \tag{5}$$

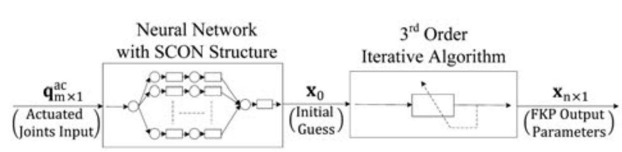


Figure 1. Graphical representation of the improved hybrid method.

in which $(F')^{i,j}$ is the (i,j)th array of the Jacobian matrix and F^i is the ith row of the nonlinear vector $\mathbf{F}$. This algorithm considers a vector $\mathbf{x}_0$ as an initial guess of the solution and then computes the answer with the desired level of accuracy by iterating Equations (3) and (4). The iteration stops when the condition of Equation (6) is satisfied,

$$||\mathbf{x}_{m+1} - \mathbf{x}_m||_\infty < \mathrm{E}_{\max} \quad (6)$$

where $E_{\max}$ is the maximum allowable error of the parameters and

$$||\mathbf{x}||_\infty = \max(|x^1|, \ldots, |x^n|). \quad (7)$$

As the initial guess gets closer to the exact answer, less number of iterations will be required to achieve the desired accuracy and the solution time will decrease. Therefore, in this paper ANNs are used to generate suitable initial guesses.

It is well known that velocity relation of parallel robots can be written as $\mathbf{J}_{\mathrm{inv}}\,\dot{\mathbf{q}}^{ac} = \mathbf{J}_{\mathrm{dir}}\dot{\mathbf{X}}_{\mathrm{e}}$, in which $\dot{\mathbf{X}}_{\mathrm{e}}$ is the velocity vector of end effector and $\dot{\mathbf{q}}^{ac}$ is the velocity vector of actuated joints. Forward kinematics singularities occur when the determinant of $\mathbf{J}_{\mathrm{dir}}$ becomes zero. Therefore, one concern may arise when the robot is getting close to singular configurations since the $\mathbf{F}'$ matrix defined in Equation (5) may be viewed as $\mathbf{J}_{\mathrm{dir}}$. However, they are not the same because the vector $\mathbf{x}$ defined in this paper includes passive joints parameters in addition to the end-effector ones. Thus, the $\mathbf{F}'$ matrix does not necessarily become singular in forward kinematics singular configurations.

Nevertheless, one should be aware that if there is a configuration of the robot where determinant of the $\mathbf{F}'$ matrix becomes zero (which is not necessarily a singular point of the robot), the proposed improved hybrid method (as well as the hybrid strategy [15] and the Newton–Raphson algorithm) may fail to converge.

2.2. *Artificial neural networks (ANNs)*

To create a model of the forward kinematics of parallel robots, the ANN should be designed such that it takes the displacements of active joints, $\mathbf{q}^{ac}$, as its inputs, and estimates unknown kinematic parameters, $\mathbf{x}$, as its outputs.

Before using an ANN it should be trained by adequate input–output sets. In the present work, inverse kinematics is used to obtain training data. To do this, the position and the orientation of the end effector are chosen somewhere in the workspace of the robot and active joints variables, as well as other kinematic parameters, are calculated by solving the IKP. Then, the network is trained by considering the active joints displacements as the inputs and other kinematic parameters as the corresponding outputs.

After preparing the training data, the structure of ANN should be chosen. If there is more than one output to be estimated, two different ACON and OCON approaches can be taken.[15] In the ACON approach a single neural network is used for estimating all the outputs, while in the OCON approach one individual network is used for each output variable. The ACON and OCON strategies are commonly compared to MIMO and MISO control systems, respectively.

Given that in each individual OCON network the weights should be adjusted to approximate only one output variable, the accuracy of estimation is somewhat increased in this type of neural networks. On the other hand, in the OCON approach generally there are additional number of weights and connections which may lead to an increase in training and simulation time as well as the amount of memory needed to store network data.

As both the accuracy and run time of the ANNs affect the overall speed of the hybrid method, in this paper a new structure called SCON (Same Class One Network) is proposed for designing the ANNs. In this approach, the outputs are divided to some classes of variables with the same range of variations. Then, for each class of variables, one individual neural network is trained to approximate the relations between input–output data-sets.

It is well known that when all the outputs of an ANN have the same range of variations, the accuracy of the network estimations will increase. Therefore, using the SCON structure will increase the ANNs accuracies when compared with the ACON approach. This will decrease the run time and resource use when compared with the OCON approach. In fact the SCON structure provides a balance between speed and accuracy and can be considered as an average of the ACON and the OCON structures.

Thus, by using the SCON structure, the numerical algorithm will be provided with fairly accurate and fast initial guesses and consequently, the overall speed of the improved hybrid method will be increased.

The three different ANN structures are depicted in Figure 2, where C_1, C_2, …, C_k, $k \le n$, represent different classes of output variables. Here, k is the number of output classes and n is the total number of output variables. It should be noted that the ACON, OCON, and SCON structures just differ in how the outputs are categorized and they are not related to the interior structure of ANNs such as learning algorithm, activation functions, hidden layers connections, and so on.

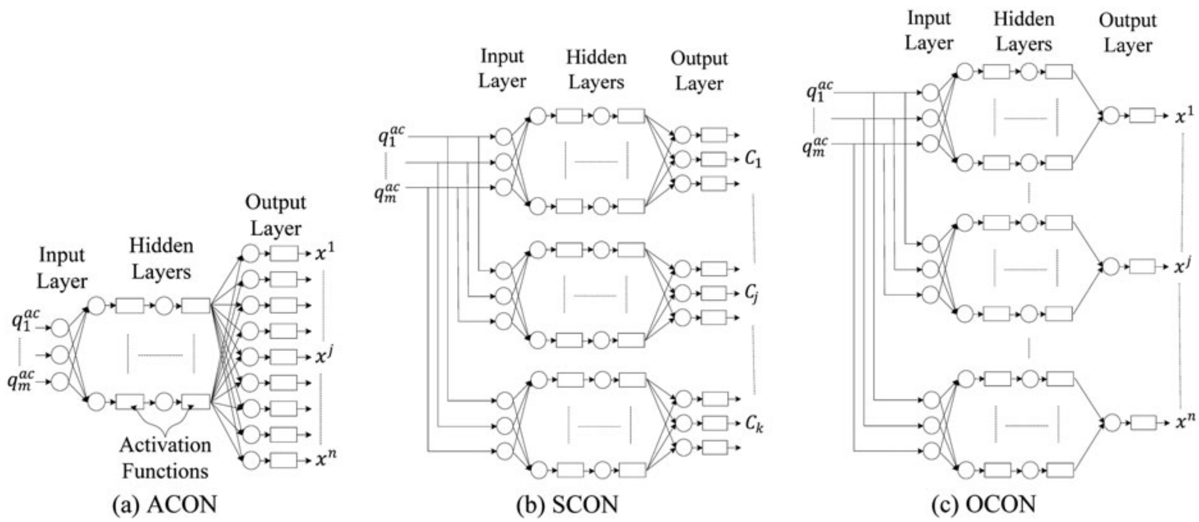


Figure 2. Schematic overview of (a) ACON, (b) SCON, and (c) OCON structures for ANNs.

3. Sample robots

In this section, FKPs of two representative parallel robots are introduced and are later used to evaluate effectiveness of the proposed method. According to Rolland [19], the majority of planar tripods can be modeled by the 3-RPR parallel manipulator. Therefore, the 3-RPR parallel robot is selected as a general case of planar manipulators. Moreover, the proposed method is applied to FKP of a 3-PSP parallel robot to cover the case of spatial manipulators.

3.1. Planar 3-RPR parallel robot

The planar 3-RPR manipulator is a parallel robot with three degrees-of-freedom: two in-plane translations and one rotation about an axis perpendicular to the plane. As depicted in Figure 3, structure of this robot consists of a moving platform which is connected to a fixed base through three identical branches (legs). Each leg includes two passive revolute joints (R-joints) and an active prismatic joint (P-joint).

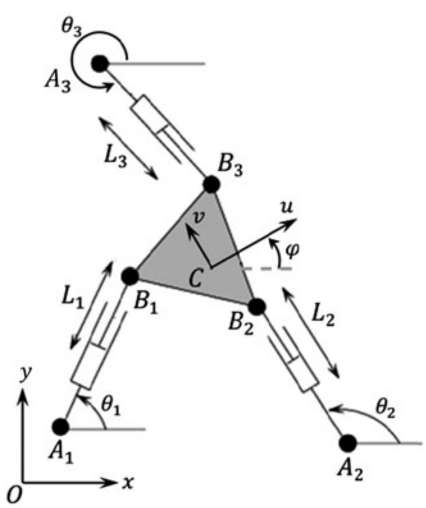


Figure 3. The 3-RPR parallel robot [20].

A special case of symmetric 3-RPR robot, with equilateral triangles as its moving and fixed platforms, is considered. The parameters and the coordinate systems required for the kinematic description of this robot are depicted in Figure 3 and Table 1. It should be noted that the symmetry assumption does not impose any limitation and can be easily removed. The coordinate system $\mathbf{O}\{x, y\}$ and $\mathbf{C}\{u, v\}$ are attached to the fixed and moving platforms, respectively.

There are three closed kinematic chains consisting of the points A_i, B_i, C, O, A_i. It is an easy task to verify that the vectorial equality of Equation (8) holds for each chain.[21]

$$ {}^{\mathbf{O}}\mathbf{A_iB_i} = {}^{\mathbf{O}}\mathbf{OC} + {}^{\mathbf{O}}_{\mathbf{C}}\mathbf{R}\ {}^{\mathbf{C}}\mathbf{CB_i} - {}^{\mathbf{O}}\mathbf{OA_i} \quad \text{for } i = 1,\ 2,\ 3 \tag{8} $$

in which superscript **O** and **C** indicate that the vectors are described in coordinate system **O** or **C**, respectively. ${}^{\mathbf{O}}_{\mathbf{C}}\mathbf{R}$ is the rotation matrix from coordinate system **O** to **C**,

Table 1. Physical configuration of the planar 3-RPR manipulator.

Base joints					
$A_1(x)$	$A_1(y)$	$A_2(x)$	$A_2(y)$	$A_3(x)$	$A_3(y)$
0 (m)	0 (m)	2 (m)	0 (m)	1 (m)	2sin(π/3) (m)
Moving platform joints					
$B_1(u)$	$B_1(v)$	$B_2(u)$	$B_2(v)$	$B_3(u)$	$B_3(v)$
0 (m)	0 (m)	1 (m)	0 (m)	0.5 (m)	sin(π/3) (m)

$$ {}^{\mathbf{O}}_{\mathbf{C}}\mathbf{R} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}, \tag{9} $$

where φ is the angle of rotation of the moving platform. It is also clear that

$$ {}^{\mathbf{O}}\mathbf{A_iB_i} = \begin{Bmatrix} L_i\cos\theta_i \\ L_i\sin\theta_i \end{Bmatrix} \quad \text{for} \quad i = 1,\ 2,\ 3 \tag{10} $$

$$ {}^{\mathbf{O}}\mathbf{OC} = \begin{Bmatrix} x_c \\ y_c \end{Bmatrix}. $$

The vectors ${}^{\mathbf{C}}\mathbf{CB_i}$ and ${}^{\mathbf{O}}\mathbf{OA_i}$ are parts of the robot structure and therefore have fixed values. Equation (8) describes three vectorial equations in a 2D space which totally form a system of six nonlinear equations with nine variables. In FKP, the actuated joints lengths $\left(L_i = q_i^{ac}\right)$ are known as inputs to the problem and other kinematic parameters should be computed by solving Equation (8). Therefore, the vector of unknown parameters for this case becomes,

$$ \mathbf{x} = [x_c,\ y_c,\ \varphi,\ \theta_1,\ \theta_2,\ \theta_3]. \tag{11} $$

Detailed information about FKP and IKP of the 3-RPR robot can be found in [19–21].

3.2. *Spatial 3-PSP parallel robot*

The 3-PSP robot is a fully parallel robot with three degrees-of-freedom in space. As shown in Figure 4, this robot has a star-like moving platform consisting of three branches with each branch making an angle of 120° to the other. The moving star is connected to the base through three identical legs, which are consisted of an active prismatic joint (P-joint), a passive spherical joint (S-joint) and a passive prismatic joint (P-joint). Therefore, each one of the three legs has a serial PSP structure and together form the parallel 3-PSP structure.

The parameters and the coordinate systems required for the kinematic description of the 3-PSP parallel robot are depicted in Figure 5 and Table 2. The coordinate system $\mathbf{B}\{x, y, z\}$ is attached to the base and located at the center of triangle $\Delta A_1A_2A_3$, while coordinate system $\mathbf{T}\{u, v, w\}$ is attached to the center of the star platform. Vector $\mathbf{a_i}$ locates the position of ith linear actuator with respect to the center of the coordinate system $\mathbf{B}$ and vector $\mathbf{h}$ locates the tool tip with respect to the center of the star platform. Both of these vectors are parts of the robot structure and therefore have fixed values.

Vector $\mathbf{q}_\mathbf{i}^{ac}$ represents displacement of ith actuated joint. Vectors $\mathbf{t}$ and $\mathbf{p}$ locate the position of center of star platform and tool tip relative to the coordinate system $\mathbf{B}$, respectively. Vectors $\mathbf{b_i}$ and $\mathbf{S_i}$ locate the position of ith spherical joint relative to center of coordinate systems $\mathbf{B}$ and $\mathbf{T}$, respectively.

Each leg of the robot is part of a closed kinematic chain which forms

$$ {}^{\mathbf{B}}\mathbf{a_i} + {}^{\mathbf{B}}\mathbf{q}_\mathbf{i}^{ac} = {}^{\mathbf{B}}\mathbf{b_i} + {}^{\mathbf{B}}\mathbf{t} \quad \text{for } i = 1,\ 2,\ 3 \tag{12} $$

Considering the relation between vectors in Figure 5, Equation (12) can be rewritten as:

$$ {}^{\mathbf{B}}\mathbf{a_i} + {}^{\mathbf{B}}\mathbf{q}_\mathbf{i}^{ac} = {}^{\mathbf{B}}_{\mathbf{T}}\mathbf{R}\left({}^{\mathbf{T}}\mathbf{b_i} - {}^{\mathbf{T}}\mathbf{h}\right) + {}^{\mathbf{B}}\mathbf{p} \quad \text{for } i = 1,\ 2,\ 3 \tag{13} $$

where ${}^{\mathbf{B}}_{\mathbf{T}}\mathbf{R}$ is the rotation matrix from coordinate system $\mathbf{B}$ to $\mathbf{T}$

$$ {}^{\mathbf{B}}_{\mathbf{T}}\mathbf{R} = \begin{bmatrix} c\lambda c\varphi & -s\lambda c\theta + c\lambda s\varphi s\theta & s\lambda s\theta + c\lambda s\varphi c\theta \\ s\lambda c\varphi & c\lambda c\theta + s\lambda s\varphi s\theta & -c\lambda s\theta + s\lambda s\varphi c\theta \\ -s\varphi & c\varphi s\theta & c\varphi c\theta \end{bmatrix}. \tag{14} $$

In the above matrix, θ, φ, and λ are Euler angles of the coordinate systems $\mathbf{T}$ about x, y, and z axis of the coordinate system $\mathbf{B}$, respectively. Here, c stands for cosine and s for sine. Equation (13) describes three vector equations which totally form a system of 12 variables (q_1^{ac}, q_2^{ac}, q_3^{ac}, x_p, y_p, z_p, θ, φ, λ, b_1, b_2, b_3) and nine equations. In the forward kinematics problem of 3-PSP parallel robot, the displacement of three active prismatic joints (q_1^{ac}, q_2^{ac}, q_3^{ac}) are known as inputs to the problem and the other variables should be computed by solving the system of Equation (13). Therefore, the unknowns of this problem are,

$$ \mathbf{x} = \left[x^1, x^2, \ldots, x^9\right]^{\mathrm{T}} = [x_T, y_T, z_T, \theta, \varphi, \lambda, b_1, b_2, b_3]^{\mathrm{T}}. \tag{15} $$

Detailed information about the forward kinematics problem of this robot is provided in [22,23].

4. Results

In this section, the improved hybrid method is used for analysis of FKP of the 3-RPR and 3-PSP parallel robots and its performance is evaluated in comparison with the Newton–Raphson algorithm as well as hybrid strategy.[15]

The proposed SCON structure is used for designing the ANNs. The kinematic parameters of the robots are categorized into classes of variables with the same nature. The selected classes of output parameters for SCON structures are described in Table 3.

To obtain the required data for the training of the ANNs, 50 random values are chosen for end-effector position and orientation ($\{x_c,\ y_c,\ \varphi\}$ for the 3-RPR planar manipulator and $\{x_p,\ y_p,\ z_p\}$ for the 3-PSP spatial robot) within the workspace of the robots. The corresponding kinematic parameters for each input set are computed by solving the IKP of the robots. Therefore,

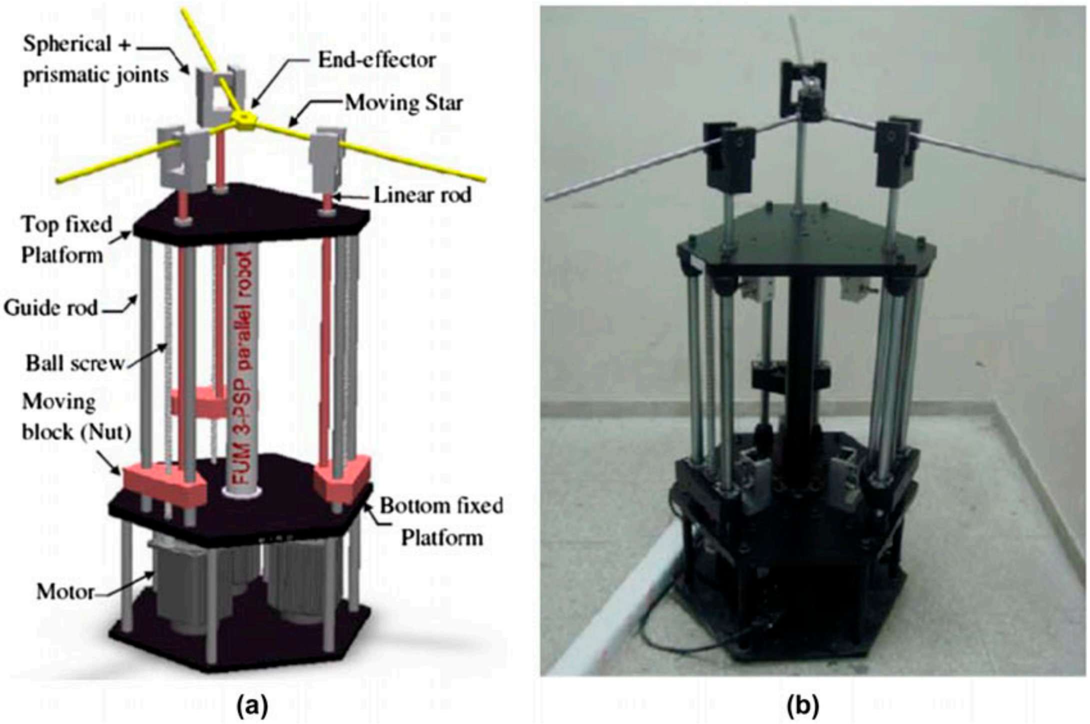


Figure 4. Spatial 3-PSP parallel robot: (a) CAD model and (b) physical model [22].

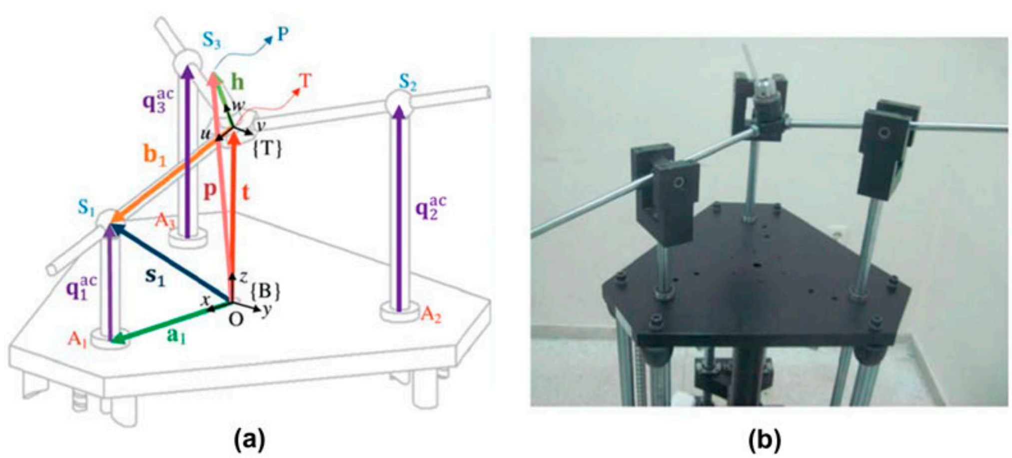


Figure 5. Required parameters for kinematic description of 3-PSP parallel robot [23].

Table 2. Physical configuration of the spatial 3-PSP robot.

Parameter	a	h
Value (m)	0.181	1.07

50 input–output pairs are obtained for the training of each ANN and Levenberg–Marquardt algorithm is applied to adjust the weights of the ANNs.

A large number of neural networks with different structures (various numbers of layers and neurons as well as different activation functions) are investigated to find suitable ANNs which model the FKP of the robots with short simulation time and low error. Finally, neural networks with one hidden layer are adopted for approximating the kinematic parameters of the two robots. The ANNs of the 3-RPR manipulator have three neurons in their hidden layers while there are five neurons in the hidden layers of the ANNs of the 3-PSP robot. Sigmoid

Table 3. Selected classes for output variables.

	3-RPR robot		3-PSP robot	
Class	Description	Variables	Description	Variables
C_1	End-effector position	$x_c,\ y_c$	End-effector position	$x_p,\ y_p,\ z_p$
C_2	End-effector orientation	φ	End-effector orientation	$\theta,\ \varphi,\ \lambda$
C_3	Revolute joint variables	$\theta_1,\ \theta_2,\ \theta_3$	Passive links lengths	$b_1,\ b_2,\ b_3$

functions are used as activation functions of all layers except the output layers which have linear activation functions.

In order to assess the performance of the improved hybrid method, outputs of this method with only one iteration of numerical algorithm are depicted in Figures 6–8. The input parameters of the 3-RPR manipulator, L_1, L_2, and L_3 are chosen in a way that the centroid of the moving platform moves on a circular path, while maintaining a constant platform orientation, $\varphi = 0$.

In the case of the 3-PSP robot, the input parameters q_1^{ac}, q_2^{ac}, and q_3^{ac} are chosen such that the center of the star platform moves on a circular path in the plane of $z = 0.5$. Note that, although the 3-PSP robot is assumed to follow a circular trajectory in a constant height, all of its three orientational degrees of freedom (θ, φ, λ) are actuated (see Figure 8).

The required input parameters are calculated by discretizing the desired path into 500 points and solving the IKP for each point. Detailed solutions of IKP of these robots are presented in [19–23].

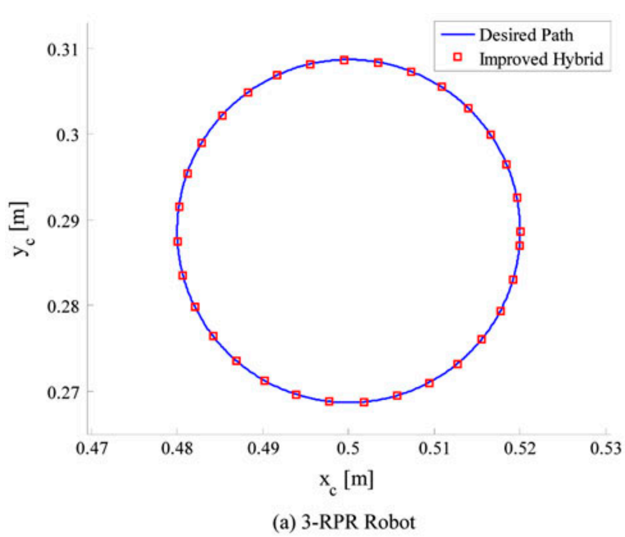
Figures 7 and 8 compare the outputs of the improved hybrid method with the desired ones which are acquired from solving the IKP. According to the dimensions of the 3RPR robot given in Table 1, when the moving triangle is located symmetrically at the center of the fixed platform, the kinematic parameters take the value of

$$\begin{cases} \theta_1 = \pi/6 \ [\text{rad}] \\ \theta_2 = 5\pi/6 \ [\text{rad}] \\ \theta_3 = 3\pi/2 = -\pi/2 \ [\text{rad}] \end{cases}, \quad \begin{cases} x_c = 0.5 \ [\text{m}] \\ y_c = 0.5 \times \tan(\pi/6) \ [\text{m}] \\ \varphi = 0 \ [\text{rad}] \end{cases} \tag{16}$$

which are used as biases in Figure 7 to make the curves almost symmetric about zero.

As it can be seen from Figures 7 and 8, absolute errors of the parameters are of the order of 10^{-5} units. The good agreement between calculated and desired path as well as the small errors, represent excellent performance of the proposed method which has been able to achieve such good accuracies with only one iteration.

To obtain higher accuracies, the numerical algorithm should be repeated. Tables 4 and 5, and Figure 9 compare the number of required iterations to achieve different levels of accuracy for the improved hybrid method, the hybrid strategy,[15] and the Newton–Raphson method. Hundred random values are considered for each input parameter and the number of required iterations is calculated for each sample and averaged over all samples.



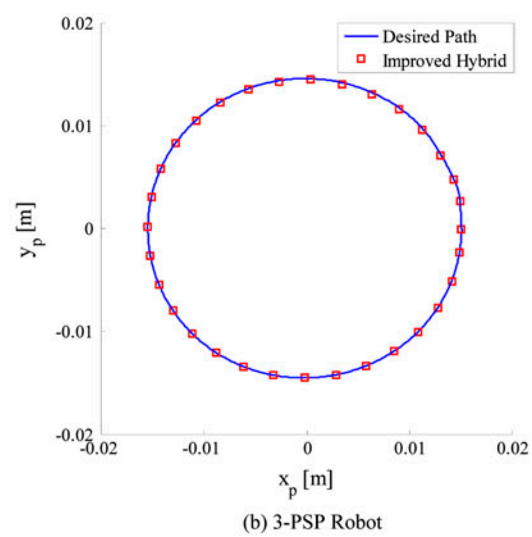


Figure 6. Evaluating the improved hybrid method in a circular path for (a) 3-RPR robot (b) 3-PSP robot.

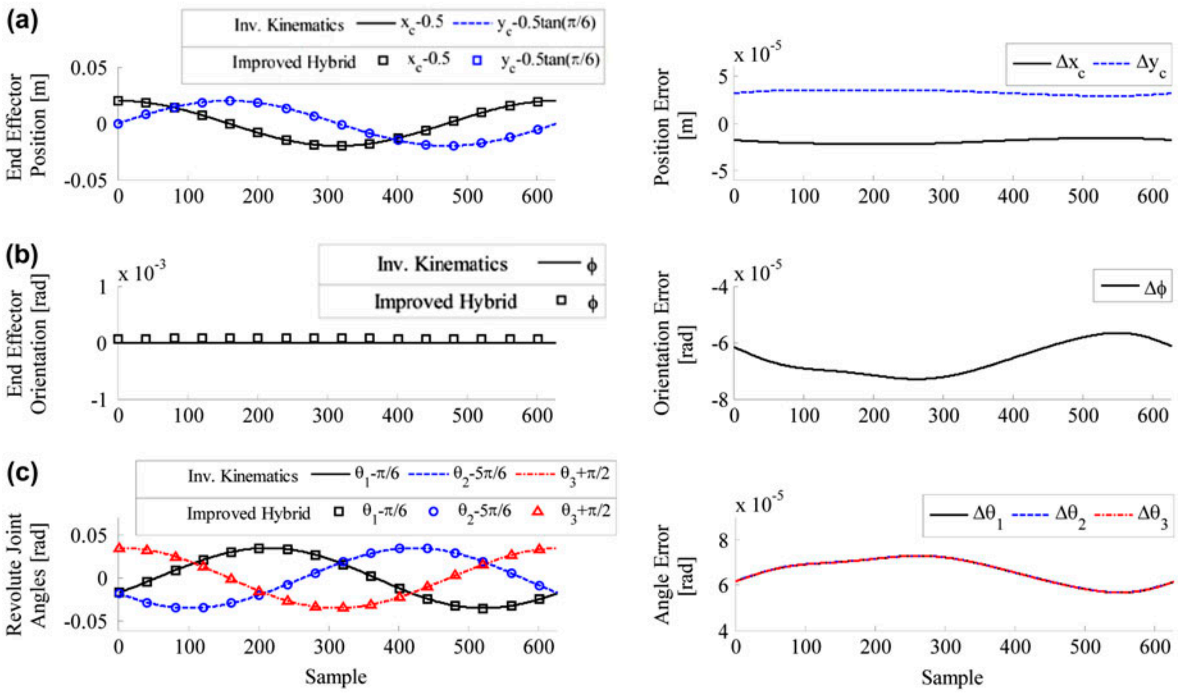


Figure 7. Improved hybrid method versus inverse kinematics solutions for 3-RPR robot: (a) Position of moving frame, (b) orientation of moving frame, and (c) base frame revolute joints variables.

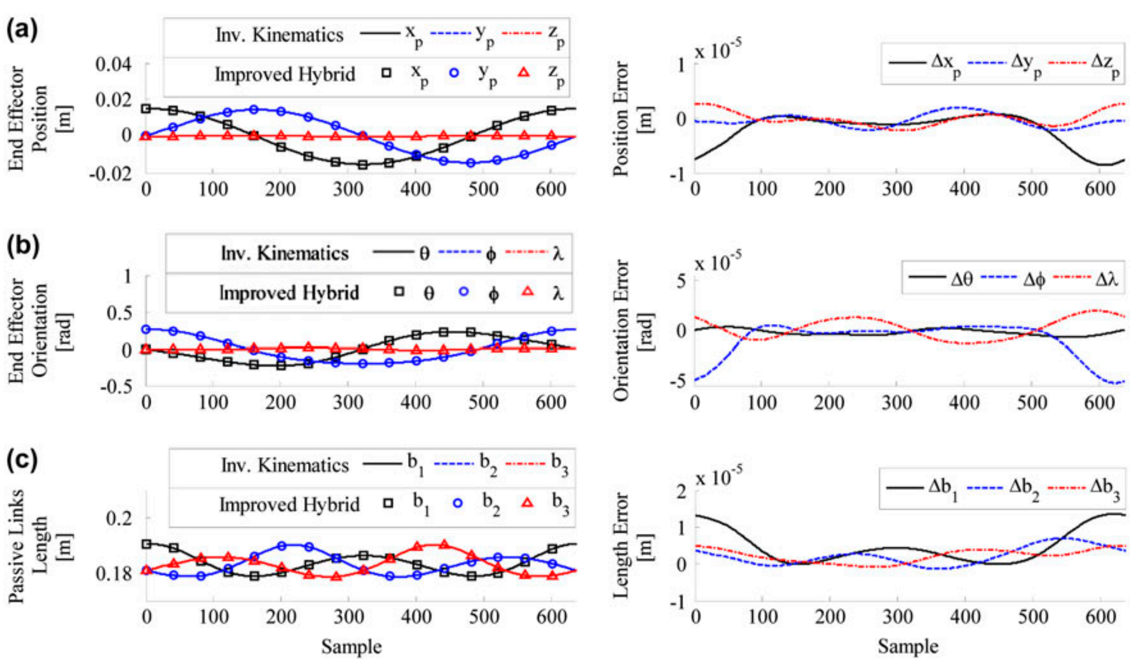


Figure 8. Improved hybrid method versus inverse kinematics solutions for 3-PSP robot. (a) Position of center of star platform, (b) orientation of star platform, and (c) passive links lengths.

It is clear that by increasing the required level of accuracy, the difference between the improved hybrid method and the hybrid strategy increases. On the other hand, the difference between the hybrid strategy [15] and the Newton–Raphson method always remains around a constant value of one iteration.

Table 4. Average number of iterations for the 3-RPR robot and percent of improvement of the proposed method.

Accuracy level	Stop criteria, E_{max}[a]	Average number of iterations			% of improvement	
		NR[b]	HS[c]	IHM[d]	Over NR	Over HS
1	10^{-3}	2.20	1.10	1.00	54.5	9.1
2	10^{-4}	3.02	1.93	1.00	66.9	48.2
3	10^{-5}	3.65	2.52	1.22	66.6	51.6
4	10^{-6}	4.15	3.05	1.62	61.0	46.9
5	10^{-7}	4.83	3.71	2.01	58.4	45.8

[a]E_{max} has the dimension of [m] for linear variables and [rad] for angular variables.
[b]Newton–Raphson.
[c]Hybrid strategy.
[d]Improved hybrid method.

Table 5. Average number of iterations for the 3-PSP robot and percent of improvement of the proposed method.

Accuracy level	Stop criteria, E_{max}	Average number of iterations			% of improvement	
		NR	HS	IHM	Over NR	Over HS
1	10^{-3}	2.05	1.00	1.00	51.2	00.0
2	10^{-4}	2.53	1.65	1.00	60.5	39.4
3	10^{-5}	3.27	2.35	1.13	65.4	51.9
4	10^{-6}	3.91	2.95	1.45	62.9	50.8
5	10^{-7}	4.60	3.52	1.93	58.0	45.2

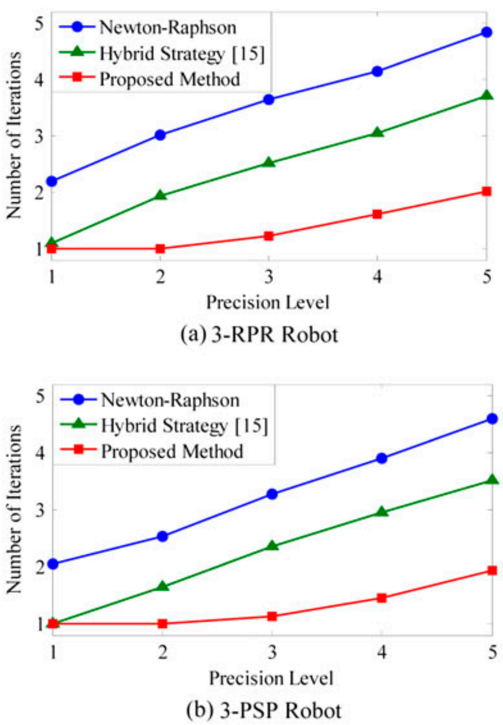


Figure 9. Average number of required iterations in different methods for (a) 3-RPR robot and (b) 3-PSP robot.

This observation can be justified by considering the order of convergence of the numerical algorithm used in each method. Both of the Newton–Raphson method and the hybrid strategy [15] use the Newton–Raphson algorithm with 2nd order convergence and therefore, the difference between these methods remains mostly constant. In contrast, the improved hybrid method uses a 3rd-order numerical algorithm and therefore, the difference between this method and other methods increases by increasing the level of accuracy.

It is noteworthy that for the 3-RPR robot the values of parameters when the moving platform is located concentric with the base and $\varphi = 0$, are considered as the initial guess for the Newton–Raphson algorithm. Likewise for the 3-PSP robot, the values of parameters when the star platform becomes parallel to the base are chosen as the initial guess. These selections have increased the convergence rate of the Newton–Raphson method such that the difference between the Newton–Raphson method and the hybrid strategy is decreased to one iteration, while in [15] this difference was about two iterations.

In the case of the 3-PSP robot each iteration of the Newton–Raphson algorithm takes about 6×10^{-4} s (0.6 ms), while each iteration of the 3rd-order algorithm takes about 8×10^{-4} s (0.8 ms). Also, for each input set, the ACON and SCON ANNs take about 3×10^{-4} s (0.3 ms) and 4×10^{-4} s (0.4 ms) to calculate the outputs, respectively. Table 6 compares the required time to reach the desired levels of accuracy. It can be seen that, by increasing the accuracy level, the improved hybrid method reduces the duration of solving the FKP up to

Table 6. Required time for solving the FKP of the 3-PSP robot and percent of improvement of the proposed method.

		Required time (ms)			% of improvement	
Accuracy level	Stop criteria E_{max}	NR	HS	IHM	Over NR	Over HS
1	10^{-3}	1.25	0.93	1.23	1.60	−32.26
2	10^{-4}	1.50	1.33	1.23	18.00	7.52
3	10^{-5}	1.99	1.71	1.40	29.65	18.13
4	10^{-6}	2.30	2.00	1.59	30.87	20.50
5	10^{-7}	2.85	2.47	1.98	30.53	19.84

Table 7. Effect of ANN structure on the duration of FKP analysis of the 3-PSP robot.

	Required time (ms)		
FKP analysis method	OCON	ACON	SCON
Hybrid strategy	2.35	2.47	2.24
Improved hybrid method	2.28	2.25	1.98

20% compared with the hybrid strategy [15] and up to 30% compared with the Newton–Raphson method. These results are obtained by running the simulations on a PC with a 2.4 GHz processor and six gigabytes of RAM.

The influence of the proposed SCON structure on the duration of the FKP analysis of the 3-PSP robot is demonstrated in Table 7. The architecture of ANNs of the improved hybrid method and hybrid strategy is changed between OCON, ACON, and SCON structures and the required time to reach the fifth level of accuracy is measured. The required time is calculated for 100 random samples and averaged over all samples.

The results in Table 7 show that using the SCON structure leads to the minimum solution time for both of the hybrid strategy and the improved hybrid methods. This observation confirms the main idea of proposing the SCON structure that keeping a balance between speed of providing the initial guesses and their accuracy will reduce the overall analysis time.

Regarding the obtained results, it can be concluded that in higher levels of accuracy the improved hybrid method achieves the desired accuracy with less iteration and in shorter time. This fact verifies that the proposed method successfully fulfills the mission of reducing the duration of solving forward kinematics problem of parallel robots.

5. Conclusion

In this paper, an improved hybrid method is proposed to reduce the duration of the forward kinematics analysis of parallel robots. In this method, an approximate solution of the FKP is produced by neural networks. This solution is next considered as an initial guess for the 3rd-order numerical technique which solves the nonlinear forward kinematics equations and obtains the answer with a desired level of accuracy. In order to further increase the solution speed, a new SCON structure is proposed for designing the neural network which categorizes outputs into classes of similar variables.

The efficiency of the proposed method is demonstrated for two representative parallel robots, a 3-RPR planar manipulator and a 3-PSP spatial robot. The results of applying different methods to FKP of the robots indicate that, compared with the Newton–Raphson method and the hybrid strategy,[15] the improved hybrid method achieves the desired accuracy with less iteration and in shorter time. The advantage of the proposed method becomes more evident by increasing the required level of accuracy. In the fifth level of accuracy, which is equivalent to an absolute error of 10^{-7} m for linear parameters and 10^{-7} radians for angular parameters, the improved hybrid method reduces the required calculation time up to 30% compared with the Newton–Raphson algorithm and up to 20% compared with the hybrid strategy.[15] Also the required iteration is reduced by 58 and 45% in comparison with the Newton–Raphson and the hybrid strategy,[15] respectively.

These achievements can improve the performance of parallel robots, especially in real-time applications in which obtaining high levels of accuracy in short time is essential. Additionally, by accelerating the forward kinematics analysis, the proposed method brings the possibility of practical implementation of more complex control algorithms. It should be also noted that the proposed method is general and may be applied to more complex robots.

Notes

1. All Class One Network.
2. One Class One Network.
3. Same Class One Network.

Notes on contributors

Iman Kardan received his BSc and MSc degree in Mechanical Engineering from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran in 2008 and 2011, respectively. He is currently a PhD student in Mechanical Engineering Department of Ferdowsi University of Mashhad, Mashhad, Iran. His research interest is in the area of robotics, microrobotics, control and smart materials.

Alireza Akbarzadeh received his PhD in Mechanical Engineering in 1997 from the University of New Mexico in USA. He worked at Motorola, USA, for 15 years where he led R&D as well as automation teams. He joined the Ferdowsi University of Mashhad in 2005 and is currently a full professor in the Mechanical Engineering Department. He has over 45 journal publications and over 60 conference papers. His areas of research include robotics (parallel robots, biologically inspired robots, bipedal robots, and rehabilitation robotics), dynamics, kinematics, control, automation, optimization as well as design and analysis of experiments. He is also a founding member of the Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP).

References

[1] Ottaviano E, Ceccarelli M. Application of a 3-DOF parallel manipulator for earthquake simulations. IEEE ASME Trans. Mechatron. 2006;11:241–246.
[2] Wang L, Wu J, Wang J, You Z. An experimental study of a redundantly actuated parallel manipulator for a 5-DOF hybrid machine tool. IEEE ASME Trans. Mechatron. 2009;14:72–81.
[3] Rezaei A, Akbarzadeh A. Position and stiffness analysis of a new asymmetric 2 PRR–PPR parallel CNC machine. Adv. Robot. 2013;27:133–145.
[4] Enferadi J, Akbarzadeh A. A novel approach for forward position analysis of a double-triangle spherical parallel manipulator. Eur. J. Mech. A Solids. 2010;29:348–355.
[5] Merlet JP. Direct kinematics of parallel manipulators. IEEE Trans. Rob. Autom. 1993;9:842–846.
[6] Ku DM. Direct displacement analysis of a Stewart platform mechanism. Mech. Mach. Theory. 1999;34:453–465.
[7] Sadjadian H, Taghirad HD. Numerical methods for computing the forward kinematics of a redundant parallel manipulator. In: IEEE Conference on Mechatronics and Robotics. Proceedings; 2004; Aachen (Germany). p. 557–562.
[8] Liu K, Fitzgerald JM, Lewis FL. Kinematic analysis of a Stewart platform manipulator. IEEE Trans. Ind. Electron. 1993;40:282–293.
[9] Li Y, Xu Q. Kinematic analysis of a 3-PRS parallel manipulator. Rob. Comput. Integr. Manuf. 2007;23:395–408.
[10] Cheng HH, Lee JJ, Penkar R. Kinematic analysis of a hybrid serial-and-parallel-driven redundant industrial manipulator. Int. J. Rob. Autom. 1995;10:159–166.
[11] McAree PR, Daniel RW. A fast, robust solution to the Stewart platform forward kinematics. J. Rob. Syst. 1996;13:407–427.
[12] Yang C, Huang Q, Ogbobe PO, Han J. Forward kinematics analysis of parallel robots using global Newton–Raphson method. In: Second International Conference on Intelligent Computation Technology and Automation. Proceedings; 2009 Oct 10–11; Changsha, Hunan (China). p. 407–410.
[13] Yang CF, Zheng ST, Jin J, Zhu SB, Han JW. Forward kinematics analysis of parallel manipulator using modified global Newton–Raphson method. J. Cent. South Univ. T. 2010;17:1264–1270.
[14] Guez A, Ahmad Z. Accelerated convergence in the inverse kinematics via multilayer feed forward networks. In: International Joint Conference on Neural Networks. Proceedings; 1989 Jun 18–22; Washington, DC (USA). p. 341–344.
[15] Parikh PJ, Lam SS. A hybrid strategy to solve the forward kinematics problem in parallel manipulators. IEEE Trans. Robot. 2005;21:18–25.
[16] Chapra SC, Canale R. Numerical methods for engineers. New York: McGraw-Hill; 2005.
[17] Darvishi MT, Barati A. A third-order Newton-type method to solve systems of nonlinear equations. Appl. Math. Comput. 2007;187:630–635.
[18] Chun C. A new iterative method for solving nonlinear equations. Appl. Math. Comput. 2006;178:415–422.
[19] Rolland L. Synthesis on forward kinematics problem algebraic modeling for the planar parallel manipulator: displacement-based equation systems. Adv. Robot. 2006;20:1035–1065.
[20] Merlet JP, Gosselin CM, Mouly N. Workspaces of planar parallel manipulators. Mech. Mach. Theory. 1998;33:7–20.
[21] Chandra R, Rolland L. On solving the forward kinematics of 3RPR planar parallel manipulator using hybrid metaheuristics. Appl. Math. Comput. 2011;217:8997–9008.
[22] Rezaei A. Start up and control of the 3-PSP parallel manipulator [MSc thesis]. Mashhad: Ferdowsi University of Mashhad; 2011 (Persian).
[23] Rezaei A, Akbarzadeh A, Nia PM, Akbarzadeh-T MR. Position, Jacobian and workspace analysis of a 3-PSP spatial parallel manipulator. Rob. Comput. Integr. Manuf. 2013;29:158–173.