16*th.* Annual (International) Conference on Mechanical Engineering-ISME2008

May 14-16, 2008, Shahid Bahonar University, Kerman, Iran

A GA Optimized Bi-Level Tuning Fuzzy Controller for a Planar 3-RRR Parallel Manipulator

K. Kamali
M. Sc. Student-Department of Mechanical Engineering, Ferdowsi University of Mashhad
kaveh.kamali@gmail.com

P. Mahmoodinia
M. Sc. Student-Department of Mechanical Engineering, Ferdowsi University of Mashhad
p.mahmoodinia@gmail.com

A. Akbarzadeh-T.
Assistant professor-Department of Mechanical Engineering, Ferdowsi University of Mashhad
ali_akbarzadeh_t@yahoo.com

M.-R. Akbarzadeh-T.
Associate professor- *Department of Electrical Engineering, Ferdowsi University of Mashhad*
akbarzadeh@ieee.com

Abstract
This paper illustrates an application of intelligent control for a planar 3-RRR parallel manipulator. Unlike control of serial manipulators that has been vastly addressed in scientific literature, control of parallel manipulators has been only addressed by few. A GA optimized bi-level tuning fuzzy PD controller is designed here to control the manipulator. In order to consider the maximum allowable torque applied to motors, the maximum torque is assumed to be the same for both controllers. A bi-level tuning method is used for tuning the fuzzy controller. In the first level, the fuzzy PD controller's normalizing parameters are determined similar to a linear PD controller. In the second level, other parameters of the fuzzy controller are tuned using genetic algorithms. This fuzzy PD controller is compared by a simple linear PD controller. natural orthogonal compliment (NOC) method is used to simulate dynamics of the manipulator. Results indicate that the fuzzy PD controller has better performance over linear PD controller.
Keywords: 3-RRR parallel manipulator, fuzzy control, genetic algorithms, Model-free control

Introduction

Parallel manipulators are customarily used for tasks requiring high payloads, high speed and high accuracy due to their close-loop kinematic chain architecture [1]. As a result of these advantages, since the 1990s, parallel manipulators have received significant attention within the literature and have been utilized in various industrial areas [1]. The early design of the parallel manipulator was a six-linear jack system devised as a tire-testing machine proposed by Gough and Whitehall [2]. Stewart [3] designed a general six-legged platform manipulator as an airplane simulator. Hunt [4] suggested the use of the Stewart platform mechanism as a robot manipulator. Since then, parallel mechanisms have been studied extensively by numerous researchers.

Unlike control of serial manipulators that has been vastly addressed in scientific literature, control of parallel manipulators has only been addressed by few. This is due to the increased complexity in the dynamics and higher interaction between system components. A comprehensive summary on such techniques for parallel manipulators, in general, has recently appeared in [5]. In many industrial applications, such as some assembly and machining operations, parallel manipulators with fewer degrees of freedom than six are successfully used [6].

In this paper, control of a 3-RRR planar three-degrees-of-freedom parallel manipulator is studied. To the best of author's knowledge intelligent control methods, has not been applied to 3-RRR parallel robots. A bi-level tuned PD fuzzy controller is developed. In the first level, a linear PD controller is independently applied to each actuator. Next, fuzzy rules are developed to design a fuzzy PD controller. Fuzzy controller normalizing parameters are regulated according to maximum PD control errors. This level of tuning is named *linear tuning*. Linear tuned fuzzy controller has similar properties to the PD controller which is named *linear like fuzzy logic controller* (LLFLC) [7]. In the second level, named *nonlinear tuning*, other parameters of the fuzzy controller are tuned using genetic algorithms. This step allows increasing the performance of fuzzy controller to have better performance in tracing a desired trajectory without any increase in maximum torque applied to manipulator. In order to compare performance of linear PD and fuzzy PD controllers, NOC method[8] is used to simulate the manipulator's dynamics. At the end, a case study is performed to illustrate the performance of purposed controller against PD controller.

The manipulator's structure

The manipulator is shown in "Figure 1". The platform is connected with three legs to the base. Each leg has one active and two passive revolute joints and the three motors M_1, M_2 and M_3 are fixed and placed on the vertices of an equilateral triangle. This manipulator consists of a kinematic chain with three closed loops, namely M_1DABEM_2 , M_2EBCFM_3 and M_3FCADM_1. The gripper is rigidly attached to the moving base, triangle ABC. The manipulator is supposed to be symmetric for simplification.

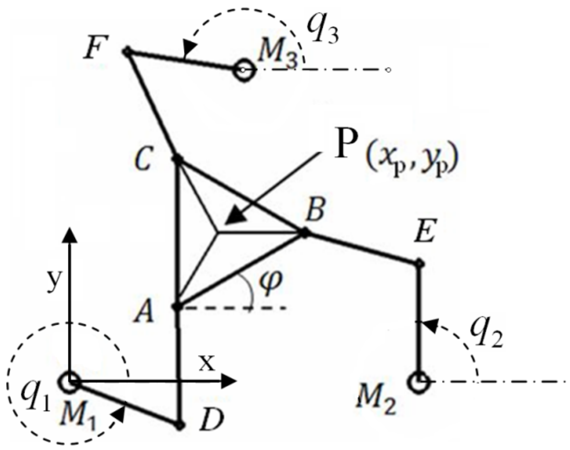


Figure 1: general form of a parallel 3-RRR manipulator

Kinematics analysis

The analysis of manipulator kinematics consists of *inverse kinematics* and *direct kinematics* problems. In direct kinematics problem the platform position and orientation, $[x_p, y_p, \varphi]^T$, are obtained from active joint angles, $[q_1, q_2, q_3]^T$, (see "Figure 1"). Direct kinematics analysis is an essential part of control and simulation in parallel manipulators. In inverse kinematics problem the active joint angles are obtained from platform position and orientation. The inverse kinematics problem is also important since manipulation tasks are naturally formulated in terms of the expected end effector position and orientation while the desired trajectory needs to be expressed in terms of the motors angles for control applications.

Inverse kinematics

There are straight forward inverse kinematics closed solutions for this parallel manipulator. For simplification it is assumed that the triangle $M_1M_2M_3$ is equilateral and the distance between any two motors is set to unity, for normalization purposes. As is shown in "Figure 1", origin is placed on point M_1. Solutions are studied in [6]. Following relations gives angles of the active joints:

$$\theta_i = \alpha_i \pm \psi_i \qquad for\ i = 1,2,3 \tag{1}$$

$$\alpha_i = atan2(x_{2i}, y_{2i}) \tag{2}$$

Where ψ_i is defined as following:

$$\psi_i = cos^{-1}\left[\frac{l_1^2 - l_2^2 + x_{2i}^2 + y_{2i}^2}{2l_1\sqrt{x_{2i}^2 + y_{2i}^2}}\right] \quad 0 \preccurlyeq \psi_i \preccurlyeq \pi \tag{3}$$

Defining $\{\Phi_i\}_1^3, \{x_{oi}\}_1^3$, $\{y_{oi}\}_1^3$, by Equations (4)-(8), x_{2i} and y_{2i} could be obtained from Equations (9) and (10):

$$\Phi_1 = \Phi + \pi/6 \tag{4}$$

$$\Phi_2 = \Phi + 5\pi/6 \tag{5}$$

$$\Phi_3 = \Phi - \pi/2 \tag{6}$$

$$\{x_{oi}\} = \{0,1,1/2\} \tag{7}$$

$$\{y_{oi}\} = \{0,0,\sqrt{3}/2\} \tag{8}$$

$$x_{2i} = x_p - l_3 cos\Phi_i - x_{oi} \tag{9}$$

$$y_{2i} = y_p - l_3 sin\Phi_i - y_{oi} \tag{10}$$

Direct kinematics

The degree of difficulty involved in finding a solution to the direct kinematics problem of parallel manipulators is higher than the corresponding serial manipulators. Direct kinematics problem for 3-RRR parallel manipulator is studied in [6]. Generally a closed form solution for the direct kinematics is impossible to obtain [6]. Therefore, the solution for the 3-RRR manipulator requires utilization of a numerical method which in general is not unique. The problem leads to a maximum of 6 solutions [9]. Derivative methods such as Secant method [10] are usually suggested to solve these problems. A disadvantage of these methods is that the computation is time consuming. Additionally, these methods provide only one of the solutions which depends on the initial guess. As pointed out earlier, the direct kinematics problem has multiple solutions. However, due to trajectory following only one of the solutions fits the path and is the correct solution. In this study, the previous position of platform is used as the initial guess. By using this initial condition we have a better chance to find the correct solution.

Dynamic analysis

Using NOC method introduced by Ma and Angeles [8], dynamic model of the 3-RRR parallel manipulator is expressed in the following compact form:

$$M(q)\ddot{q}^a(t) + C(q,\dot{q})\dot{q}^a(t) = \tau^a \tag{11}$$

Where $q^a = [q_1, q_2, q_3]^T$ is the generalized coordinates that $q_i, 1 \leq i \leq 3$ denotes the angles of active joints (motors) and $q = [q_1, q_2, \dots, q_9]^T$ where $q_i, 4 \leq i \leq 9$ denotes relative angels of passive joints. In this equation $M(q) \in \Re^{3\times3}$ is the inertia matrix, $C(q,\dot{q}) \in \Re^{3\times3}$ is the coefficient matrix of Coriolis and centrifugal forces and $\tau^a \in \Re^{3\times1}$ denotes the actuation torques of the motors.

If the value of $q, \dot{q}$ and $\ddot{q}^a$ are known one can easily find τ^a by Equation (11). This procedure is named inverse dynamics. Direct dynamic problem is defined as finding the $\ddot{q}^a$, when we know values of $q, \dot{q}$ and τ^a. In this study direct dynamics is used to simulate the manipulator. When we don't have a real robot, simulation helps us to calculate feedback values. But in simulating process we just have τ^a, angles of active joints (q_i^a) and their angular velocities ($\dot{q}_i^a$). Therefore direct kinematics is utilized to calculate q and $\dot{q}$. In direct dynamic problem Equation (11) should be solved as a differential equation, therefore, a numerical method is needed. The Runge-Kutta-Fehlberg method with variable time steps is implemented in this study.

Control methods

Two control approaches are studied. First, a total of three independent linear PD controllers, one for each actuator, is considered. Second approach uses three independent fuzzy logic PD controllers. In both approaches, the positional error and their derivatives are defined as follows:

$$e_i = q^a_{i\ desired} - q^a_i \qquad (12)$$

$$\dot{e_i} = \frac{d}{dt}\left(q^a_{i\ desired} - q^a_i\right) \qquad (13)$$

Where $i = 1,2,3$ denotes the actuator number. $e = [e_1, e_2, e_3]^T$ and $\dot{e} = [\dot{e}_1, \dot{e}_2, \dot{e}_3]^T$ will be used as the controller inputs. As it was mentioned, the desired active joints angles, $q^a_{i\ desired}$, are obtained by inverse kinematics and $\dot{q}^a_{i\ desired}$ is derived from $q^a_{i\ desired}$. "Figure 2" describes a linear PD or fuzzy PD controller when they are implemented in the direct action (DA) form [11].

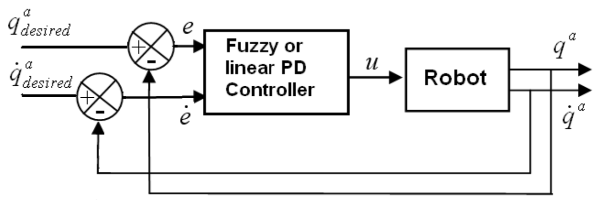


Figure 2: linear PD or fuzzy PD controller when they are implemented in the direct action (DA) form

Linear PD control

The linear PD control, allows one to use simple PD regulators for each actuator. The control signal, u_i, produced by a PD regulator for i^{th} actuator is given by:

$$u_i = K_{P_i} e_i + K_{D_i} \dot{e_i} \qquad (14)$$

Where K_{P_i} and K_{D_i} are proportional and derivative gains for i^{th} actuator.

Direct action fuzzy logic control

Fuzzy logic control (FLC) has been successful in many engineering applications since its early introduction by Mamdani in 1974 [12]. These knowledge based control formulations allow control to be performed in a nonlinear fashion and therefore FLC has the ability to provide improved performance against conventional control. FLC is based on expert rules and requires no additional system model for implementation. As it is indicated, 3-RRR parallel manipulator has complex forward kinematics which makes implementation of conventional model based control algorithms difficult. Under these circumstances, FLC has the ability to provide improved control while requiring no additional model estimates. For this work a fuzzy controller with two inputs and one output is selected.

We will use suitable scale factors S_e and S_{ce} for inputs and S_u for output, for normalization purposes. The error and its derivative are normalized by using the following conditions:

$$\hat{e} = \max\left(1, \min(1, S_e e)\right) \qquad (15)$$

$$\hat{\dot{e}} = \max\left(1, \min(1, S_{ce}\dot{e})\right) \qquad (16)$$

These two normalized parameters will be used as controller inputs. The controller output after the fuzzy inference is denoted by $\hat{u}$. Similarly the FLC output is denormalized using the condition $u = S_u\hat{u}$. "Figure 3" illustrates the process of normalizing inputs and denormalizing outputs for fuzzy controller in a control loop.

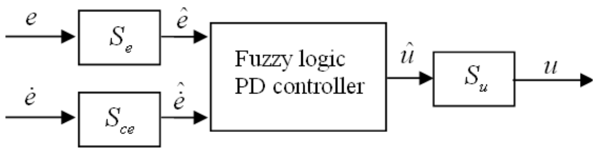
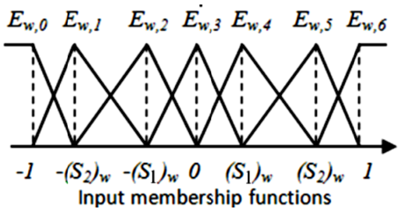


Figure 3: Normalization of inputs and outputs of fuzzy controller

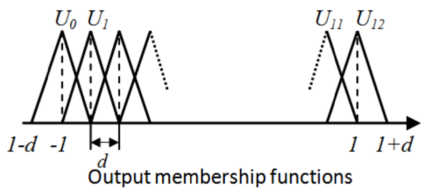
Using feedback errors, $\hat{e}$, and its derivatives, $\hat{\dot{e}}$, as inputs, a two-dimensional type fuzzy rule base is developed. Seven membership functions for each of two inputs are assumed in triangle form. Values of $(S_i)_1$ for first input of controller (e) and $(S_i)_1$ for second input of controller ($\dot{e}$) can change the shape of triangles. Where for each value of $(S_i)_1$ and $(S_i)_2$, we have i= 1, 2 and j= 1, 2. Twelve triangle membership functions are used for output (See "Figure 4b"). For the two inputs and one output configuration in "Figure 4", a complete rule matrix of size 7 by 7 is defined as:

$$If\ \hat{e}\ is\ \ E_{wi}\ and\ \hat{\dot{e}}\ is\ \ E_{wi}\ then\ \hat{u}\ is\ U_{i+j} \qquad (17)$$

Where E_{wi} is input membership function used for fuzzy controller where indices for input membership function are indicated as the number of inputs (w) and the number of the membership functions (i). U_i is output membership function for fuzzy controller where i indicates the number of the membership functions. Therefore a complete rule base would consist of 7×7=49 rules.



a



b

Figure 4: Input membership functions (a) and output membership functions (b) for fuzzy logic controller

Tuning the fuzzy controller

A two level tuning method introduced by Mann, Hu and Gosine [13] used to tune fuzzy system parameters. In the first level, called *linear tuning*, fuzzy controller input and output normalizing parameters (S_e, S_{ce} and S_u) are regulated which makes fuzzy controller to act like a linear PD controller. In order to increase the performance of the controller without any increase in maximum torque applied to manipulator, the second level of tuning, named *nonlinear tuning*, is performed. In this level, four nonlinear tuning parameters $(S_i)_w$ and $(S_i)_w$, $i = 1, 2$ and $w = 1, 2,$ of FLC (see "Figure 2") should be adjusted. A genetic algorithms method is used for finding the optimum values of the nonlinear tuning parameters.

Step 1: linear tuning

The linear tuning is defined as the determination of linear PD gains based on linear controller performance. When a fuzzy system is set to produce a linear function, the FLC will become a linear type PD controller and is defined as an equivalent linear controller (ELC) [7]. Using the ELC output the linear PD output can be arranged in the following form:

$$u = K_P . e + K_D . \dot{e} \tag{18}$$

Where K_P and K_D are defined as the ALG terms of the FLC system. A FLC having linear rule base and uniform partition of universe of discourse of all variables is named as a linear-like fuzzy logic controller (LLFLC) [7]. The ELC defined for the LLFLC is used for deriving the linear tuning variables. Therefore, for all fuzzy PD types in the paper, the ELC systems are defined by considering uniformly partitioned fuzzy subsets for all the fuzzy variables as follows. The relations between normalizing parameters, S_e, S_{ce}, S_u, and K_P and K_D are as following [7]:

$$S_u = 2 \times K_P \tag{19}$$

$$S_{ce} = 4 \times K_P \times K_D \tag{20}$$

$$S_e = 1/error_{max} \tag{21}$$

Where, $error_{max}$ is the maximum absolute error when an actuator is controlled by a simple linear PD controller.

Step 2: nonlinear tuning

The nonlinear tuning is defined as the determination of nonlinearity tuning parameters for obtaining the desired normalized FLC output. These parameters are $(S_1)_w$ and $(S_2)_w$ where $w = 1,2$. An alternative instruction for tuning these parameters is suggested by [7]. However, here, a genetic algorithms optimizing method is applied to find optimum values of $(S_1)_w$ and $(S_2)_w$.

Simulation results

NOC method was used for dynamic simulating of manipulator. The designed linear PD and fuzzy PD controllers were tested during simulation. A specific Cartesian trajectory in terms of time is chosen for center of platform. For simplification orientation of platform assumed to be zero along path. Using inverse kinematics, angles of actuators as function of time were obtained. The calculated angles of actuators were supplied to both PD and fuzzy PD close loop systems as desired input. Magnitude of K_P and K_D are selected 10 and 1 respectively using the trial and error approach. By substitution K_P and K_D in Equations (19-21) the normalizing parameters will be obtained. The S_{1w} and S_{2w} parameters were optimized by GA algorithms while the fitness function is the maximum absolute error of following the trajectory for center of platform in x-y coordinates. For each input of FLC S_{1w} should less than S_{2w} as it is shown in "Figure 4a". It should be noted that the used GA method finds the minimum fitness. Properties of the used genetic algorithms method are given in "Table 1":

Table 1

Option	Value
Crossover function	Heuristic
Crossover fraction	0.8
Elite number	2
Initial penalty	10
Mutation function	Adaptive feasible
Penalty factor	100
Population initial range	[-1,1]
Population size	100
Population type	Bit string
Selection function	Stochastic uniform

All considered parameters for FLC are listed in "Table 2" and "Table 3". Parameters of all three controllers of the three actuators are chosen the same, for simplification.

Table 2: FLC linear tuning parameters

Parameter	S_e	S_{ce}	S_u
value	10	1	2

Table 3: FLC nonlinear tuning parameters

Parameter	$(S_1)_1$	$(S_2)_1$	$(S_1)_2$	$(S_2)_2$
value	0.0625	0.4375	0.3594	0.3750

Results are shown in “Figure 3” which indicates the fuzzy PD controller performs with less error than simple PD controller.

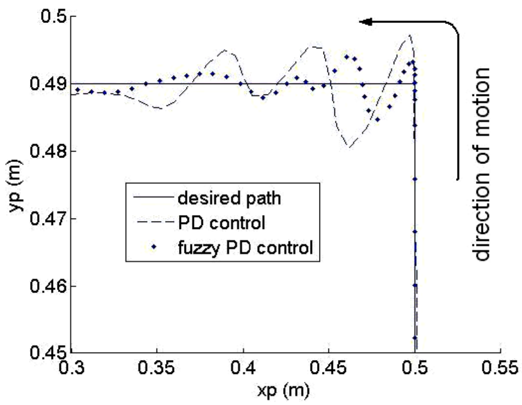
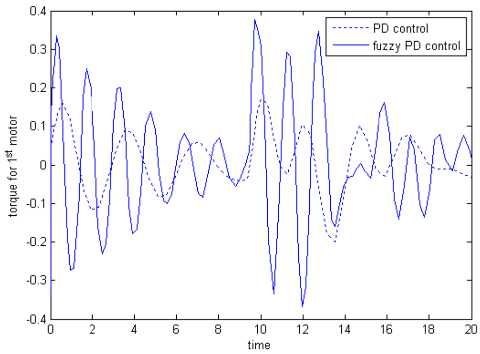
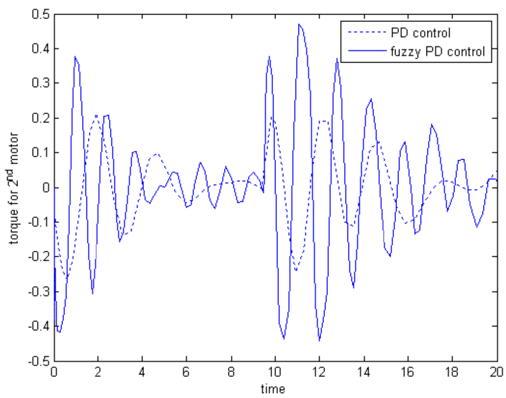


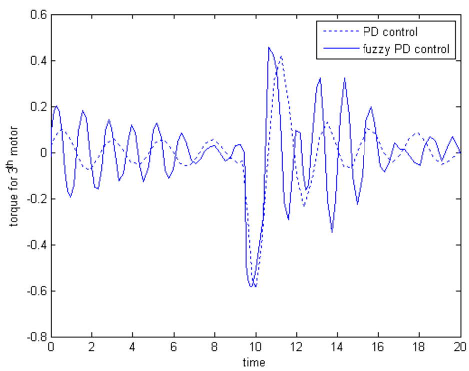
Figure 5: Comparison of PD and fuzzy controllers results for following a specific trajectory



a



b



c

Figure 6: Comparing maximum torques for PD and fuzzy optimized PD controller

Table 4: maximum absolute torques

	Maximum absolute torque of motor 1	Maximum absolute torque of motor 2	Maximum absolute torque of motor 3
Linear PD	0.21	0.53	0.58
Fuzzy PD	0.36	0.26	0.58

“Figure 6” shows the torques which are applied to each motor by the two controllers. In “Table 4” the maximum absolute torques for linear PD controller and fuzzy PD controller are compared. Results show that the maximum torque for linear PD and fuzzy PD are somewhat the same. Therefore, it can be concluded that fuzzy PD controller has better performance than linear PD controller in a same condition.

Conclusion

In this paper a GA optimized Bi-level tuning fuzzy PD controller is applied to a 3-RRR planar parallel manipulator. Two control approaches are studied. First, a total of three independent linear PD controllers, one for each actuator, are considered. Second approach uses three independent fuzzy logic PD controllers. A two level tuning method is used to tune fuzzy system parameters. In first level, named linear tuning, a method is implemented to fined normalizing parameters of fuzzy controller from gains of a specified linear PD controller. In next level, named nonlinear tuning, optimum values of other parameters of FLC are obtained using genetic algorithms method. The fitness function for the genetic algorithms method was selected the maximum absolute error for following the trajectory. The performance of the optimized FLC was compared against the PD controller for a specific trajectory. Results indicate that the fuzzy PD controller has a better performance in tracking a desired trajectory. Additionally, the maximum torque applied by the fuzzy controller does not exceed that applied by the PD controller.

List of symbols

$C(q,\dot{q})$ The coefficient matrix of coriolis and centrifugal forces

E_{wi} The i^{th} membership function for w^{th} input of fuzzy controller

e_i Control error vector of ith actuator

$\dot{e}_i$ Derivative of control error of ith actuator

$error_{max}$ The maximum absolute error

K_{P_i} Proportional gain of ith controller

K_{D_i} Derivative gain of ith controller

l_1 Length of the first links in each leg

l_2 Length of the second links in each leg

l_3 The distance between moving platform’s apex to it’s centre of surface

$M(q)$ The inertia matrix

$q^a_{desired}(t)$ Vector of desired active joint angles

$q^a(t)$ Vector of active joint angles

$\dot{q}^a(t)$	Vector of active joint angular velocities
$\ddot{q}^a(t)$	Vector of active joint angular accelerations
q^u	Vector of active joint angles
S_e	Scale factor of first input of fuzzy controller
S_{ce}	Scale factor of first input of fuzzy controller
S_u	Scale factor of output of fuzzy controller
u_i	Applied torque to the i^{th} actuator
$\hat{u}_i$	Normalized applied torque of i^{th} actuator
U_i	The i^{th} membership function for output of fuzzy controller
x_{2i}	The x coordinate of the third joint on each leg
y_{2i}	The y coordinate of the third joint on each leg
$\{x_{oi}\}$	The x coordinate of i^{th} active joint position
$\{y_{oi}\}$	The y coordinate of i^{th} active joint position
x_p	The x coordinate of moving platform position
y_p	The y coordinate of moving platform position
θ_i	Defined in text
α_i	Defined in text
τ^a	Vector of actuation torques of the motors
φ	Moving platform pure rotation about z direction
$\{\Phi_i\}$	Defined in text
ψ_i	Defined in text

References

[1] Su, Y., Sun, D., Ren, L., and Mills, J. K., 2006, "Integration of Saturated PI Synchronous Control and PD Feedback for Control of Parallel Manipulators", IEEE Transactions on Robotics, vol. 22, pp. 202-207.

[2] V.E. Gough, S.G. Whitehall, "Universal tire test machine", in: Proceedings of 9th International Technical Congress,F.I.S.I.T.A., 1962, pp. 177.

[3] D. Stewart, "A platform with six degrees of freedom, in: Proceedings of the Institute of Mechanical Engineering", vol.180(5), 1965, pp. 371–386.

[4] K.H. Hunt, Kinematic Geometry of Mechanism, Clarendon Press, Oxford, 1978.

[5] He, J.F., Jiang, H.Z., Cong, D.C., Ye, Z.M. and Han, J.W., 2007, "A Survey On Control Of Parallel Manipulator", Key Engineering Materials vol. 339, pp. 307-313.

[6] C. Gosselin, J.Angeles, Kinematics of parallel manipulators, McGill University Montreal, Quebec, Canada, December 1989.

[7] G. K. I. Mann, B.-G. Hu, and R. G. Gosine, "Two-Level Tuning of Fuzzy PID Controllers", IEEE Transactions on systems, man, and cybernetics-part B: cybernetics, April 2001, vol. 31, no. 2.

[8] Ma, O. and Angeles, J., 1989, "Direct Kinematics and Dynamics of a Planar 3-DOf Parallel manipulator", Advances in Design Automation, Proc. Of ASME Design and Automation Conference, 3, pp. 313-320.

[9] K.H. Hunt, "Structural kinematics of in-parallel-actuated robot arms", ASME Journal of Mechanisms, Transmission and Automation in Design 105, 1983,pp. 705–712.

[10] J. H. Mathews, Numerical Methods for Mathematics Science and Engineering, Prentice Hall, 1992.

[11] M. -S. Tsai, T.-N Shiau a, Y.-J. Tsai and T.-H. Chang, "Direct kinematic analysis of a 3-PRS parallel mechanism", Mechanism and Machine Theory 38, 2003, pp. 71–83.

[12] E. H. Mamdani, "Application of fuzzy algorithms for control simple dynamic plant", Proc IEE in Control and Science, 121, 1974, pp.1585-1588.

[13] Mann, G. K. I., Hu, B.-G. and Gosine, R. G., 2001, "Two-Level Tuning of Fuzzy PID Controllers", IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics, 31, no. 2.