

# Optimal Forward Kinematics Modeling of Stewart Manipulator Using Genetic Algorithms

A. Omran<sup>a,\*</sup>, M. Bayoumi<sup>b</sup>, A Kassem<sup>b</sup>, G. El-Bayoumi<sup>b</sup>

<sup>a</sup>Department of Aerospace Engineering, Old Dominion University, Norfolk, VA, USA

<sup>b</sup>Department of Aerospace Engineering, Cairo University, Giza, Egypt

## Abstract

In this paper, a new technique for modeling the forward kinematics of a Stewart manipulator is presented based on a predicted squared error cost function. Initially, the inverse kinematics model was introduced in detail for seeding the regression points. Genetic algorithms were then employed to search for optimal model structure from a set of candidate regression predictors. The proposed technique provides a significant reduction in the expenses of memory and computation compared to the traditional Taylor series expansion model. Such reduction increases the suitability of the model for a task space control application of the Stewart manipulator.

© 2009 Jordan Journal of Mechanical and Industrial Engineering. All rights reserved

Keywords: Stewart Forward Kinematics; Predicted Squared Error; Genetic Algorithm.

## 1. Introduction

Stewart manipulators have been used in numerous industrial applications. Their potential benefits include a high force to weight ratio, accuracy, and rigidity. The Stewart manipulator consists of a moving plate connected to a base plate by six legs. Each leg has an upper part sliding inside a lower part. This gives the upper plate six degrees of freedom: three translational motions (surge, sway, and heave) and three rotational motions (pitch, roll, and yaw). The acceleration forces of this moveable plate can emulate the physical feeling of piloting an aircraft in forward, backward, or turning motions. Such capability makes Stewart manipulators the best candidates for flight simulators. This perspective attracts many researchers to design a controller that can sustain a more realistic motion with minimum error. Two schemes are commonly used to control the Stewart manipulator: joint space control and task space control. The joint space scheme is developed using the information of joint displacement. Each leg of the manipulator is controlled as a single-input single-output (SISO) system as shown in Fig. 1-a. The joint space control involves the inverse kinematics model, which has a closed form. Although the joint space control scheme has been given great attention [1-5], it does not achieve high tracking performance because of the nonlinear coupling in the system.

Task space control, on the other hand, achieves high performance tracking by counting the system coupling, and it has been widely investigated [6-10]. In this scheme, the framework is multi-input multi-output (MIMO) as

shown in Fig. 1-b. Thus, the forward kinematics is involved instead of inverse kinematics. This scheme is exacerbated by the fact that the Stewart manipulator forward kinematics has no closed form solution. For example, Dietmaier showed that there are 40 such possible solutions [11].

Many studies have attempted different approaches to simplify the Stewart manipulator forward kinematics problem. Ilian *et al.* [8] presented a new closed-form solution of the problem, yet it required three extra sensors. Pratik *et al.* [7] and Sadjadian *et al.* [12-13] used the neural network approach. The accuracy of this approach is very sensitive to the structure of the neural network. For example, Sadjadian *et al.* [13] showed that different structures of neural networks lead to different accuracy in modeling the forward kinematics of Stewart manipulator. Liu *et al.* [14-15] proposed a numerical algorithm based on fundamental geometric operation with three nonlinear simultaneous algebraic equations, which is impractical for control process. Sadjadian and Taghirad [12] derived a fourth order Taylor series expansion using least square estimation (LSE). This technique has an over parameterization by 774 regression coefficients. Sheng *et al.* [16] presented a hybrid immune algorithm, genetic algorithm, and fuzzy system method for solving the forward kinematic problem of Stewart manipulator. The produced model is very complicated and has not yet been applied to any simulation or experimental test case. Tarokh [17] proposed an algorithm for solving the forward kinematics of Stewart manipulator. The algorithm consists of two phases: an on-line phase and an off-line phase. A database of the relationship between links and poses is decomposed into cells.

\* Corresponding author. aomra001@odu.edu.

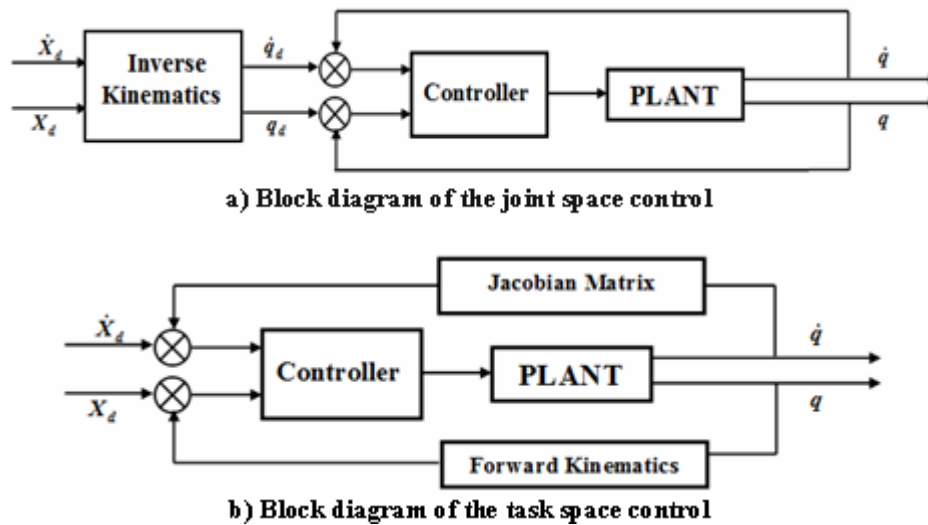


Figure 1. Comparison between the block diagram of the task space control and the block diagram of the joint space control:  $X$  refers to task space displacements and  $q$  refers to joint space displacements.

During the online phase, given link lengths, the appropriate cell is identified, and the poses are computed. Chen *et al.* [18] proposed a nonlinear observer based method to solve the forward kinematics. This observer uses the inverse kinematics as a baseline to update the model.

The current research offers an efficient technique for modeling the Stewart manipulator forward kinematics. This technique uses the predicted squared error (PSE) cost function to find an optimal structure for the forward kinematics model by reducing the number of terms in the model without affecting the required accuracy for control applications. Six optimization problems are considered to define the task space coordinates (surge, sway, heave, roll, pitch, and yaw) as functions of joint space coordinates (length of each leg). The candidate modeling predictors were selected from a third order polynomial's terms in six-dimensions (joint space). Then, the genetic algorithms (GA) was used to search for an optimal structure to minimize PSE cost function using the so-called "Switching Factor". The results of this research can be used to save the computation and memory costs. This paper is organized as follows: in section two, the description of inverse kinematics enables one to determine the link lengths in terms of desired/specified upper platform position and angular orientations. Section three includes a brief discussion of PSE regression model. Section four presents the use of genetic algorithm to search for optimal model structure. Section five offers the results of the simulation used to examine the proposed technique. Finally, section six is the conclusion.

**2. Inverse Kinematics Model**

There are two frames describing the motion of the moving plate: an inertia frame ( $X, Y, Z$ ) located at the center of the base plate and a body frame ( $x_B, y_B, z_B$ ) located at the center of the moving plate with the  $z_B$ -axis pointing outward. The angle between the local  $x_B$ -axis of the moving plate and the line of the joint  $J_{ui}$  is denoted by  $\beta_i$  as shown in Fig. (2).

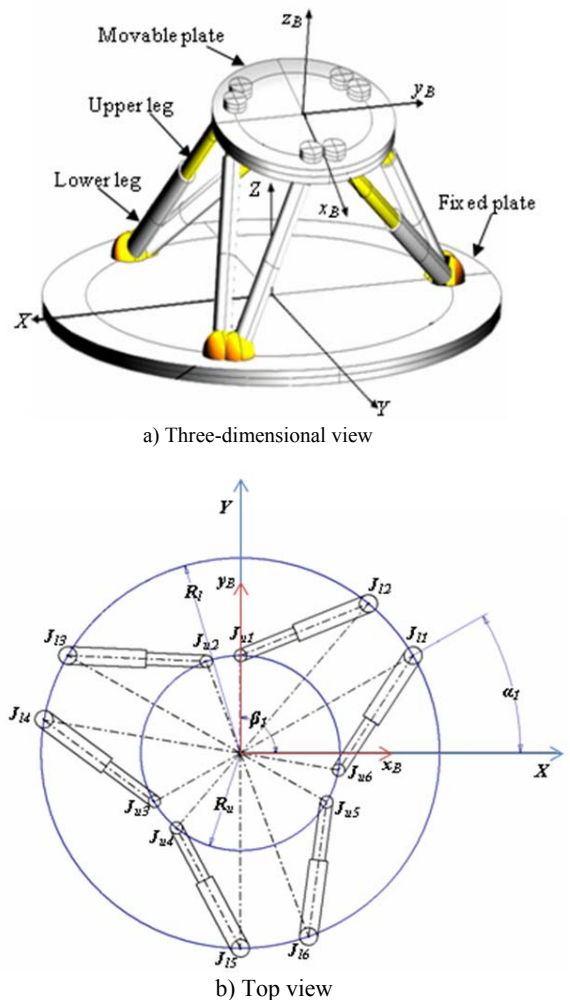


Figure 2. Stewart manipulator.

The position of the joint  $J_{ui}$  in the plate body frame is

$$P_{J_{ui}}^B = \begin{bmatrix} X_{ui}^B & Y_{ui}^B & Z_{ui}^B \end{bmatrix}^T = [R_u \cos(\beta_i) \quad R_u \sin(\beta_i) \quad 0]^T \quad i = 1, 2, \dots, 6 \quad (1)$$

In the same manner, an angle  $\alpha_i$  is defined between the inertia  $X$ -axis and the line of the joint  $J_{li}$ . The position of the joint  $J_{li}$  in the inertia frame is defined as:

$$P_{J_{li}}^I = \begin{bmatrix} X_{li}^I & Y_{li}^I & Z_{li}^I \end{bmatrix}^T = [R_l \cos(\alpha_i) \quad R_l \sin(\alpha_i) \quad 0]^T \quad i = 1, 2, \dots, 6 \quad (2)$$

The upper plate has a capability for 6-DOF motion (three rotational motions and three translational motions). The rotational motions of the plate are defined by Euler angles in sequence 1-2-3. Thus the transformation from the body frame  $(x_B, y_B, z_B)$  to the inertia frame  $(X, Y, Z)$  is given by the Matrix  $R_{plate}$ :

$$R_{plate} = R_z^T(\psi)R_y^T(\theta)R_x^T(\varphi) = \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\varphi & -S_\varphi \\ 0 & S_\varphi & C_\varphi \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} C_\theta C_\psi & S_\varphi S_\theta C_\psi - C_\varphi S_\psi & C_\varphi S_\theta C_\psi + S_\varphi S_\psi \\ C_\theta S_\psi & S_\varphi S_\theta S_\psi + C_\varphi C_\psi & C_\varphi S_\theta S_\psi - S_\varphi C_\psi \\ -S_\theta & S_\varphi C_\theta & C_\varphi C_\theta \end{bmatrix}$$

where  $C$  refers to angle cosine and  $S$  refers to angle sine. The angles  $\psi$ ,  $\theta$ , and  $\varphi$  are Euler angles. In addition to the rotation, one should consider the translation vector  $T_{plate}^I$  as:

$$T_{plate}^I = \begin{bmatrix} x(t) \\ y(t) \\ z(t) + h \end{bmatrix} \quad (4)$$

where  $h$  is the initial height of the upper plate's center. The trajectory of the upper plate's center is defined by  $x(t)$ ,  $y(t)$ , and  $z(t)$ . The position of the joint  $J_{ui}$  in inertial frame  $(X, Y, Z)$  is then calculated as:

$$P_{J_{ui}}^I = \begin{bmatrix} X_{ui}^I & Y_{ui}^I & Z_{ui}^I \end{bmatrix}^T = R_{plate} P_{J_{ui}}^B + T_{plate}^I \quad (5)$$

$$L_i^2 = \left( C_\theta C_\psi R_u \cos(\beta_i) + (S_\varphi S_\theta C_\psi - C_\varphi S_\psi) R_u \sin(\beta_i) + x(t) - R_l \cos(\alpha_i) \right)^2 + \left( C_\theta S_\psi R_u \cos(\beta_i) + (S_\varphi S_\theta S_\psi + C_\varphi C_\psi) R_u \sin(\beta_i) + y(t) - R_l \sin(\alpha_i) \right)^2 + \left( -S_\theta R_u \cos(\beta_i) + S_\varphi C_\theta R_u \sin(\beta_i) + z(t) + h \right)^2 \quad (7)$$

where  $i = 1, 2, \dots, 6$ . Based on Eq. (7), the inverse kinematics has a closed form. On the other hand, it is "difficult" to develop any closed form for the forward kinematics.

Each leg has three degrees of freedom: two rotational and one translational motion. Thus a spherical joint is employed to connect the upper part of each leg by the movable plate while the lower part is connected to the base plate by a universal joint as shown in Fig. (3). The length vector of the  $i^{\text{th}}$  leg  $L_i^I$  can then be computed from Eq. (2) and (5) as:

$$L_i^I = P_{J_{ui}}^I - P_{J_{li}}^I \quad i = 1, 2, \dots, 6 \quad (6)$$

By substituting Eq. (3) and Eq. (4) in Eq. (5), and considering the square value of vector  $L_i^I$  in Eq. (6), the relationship between the joint space variables and task space variables can be summarized as (Equation. 7):

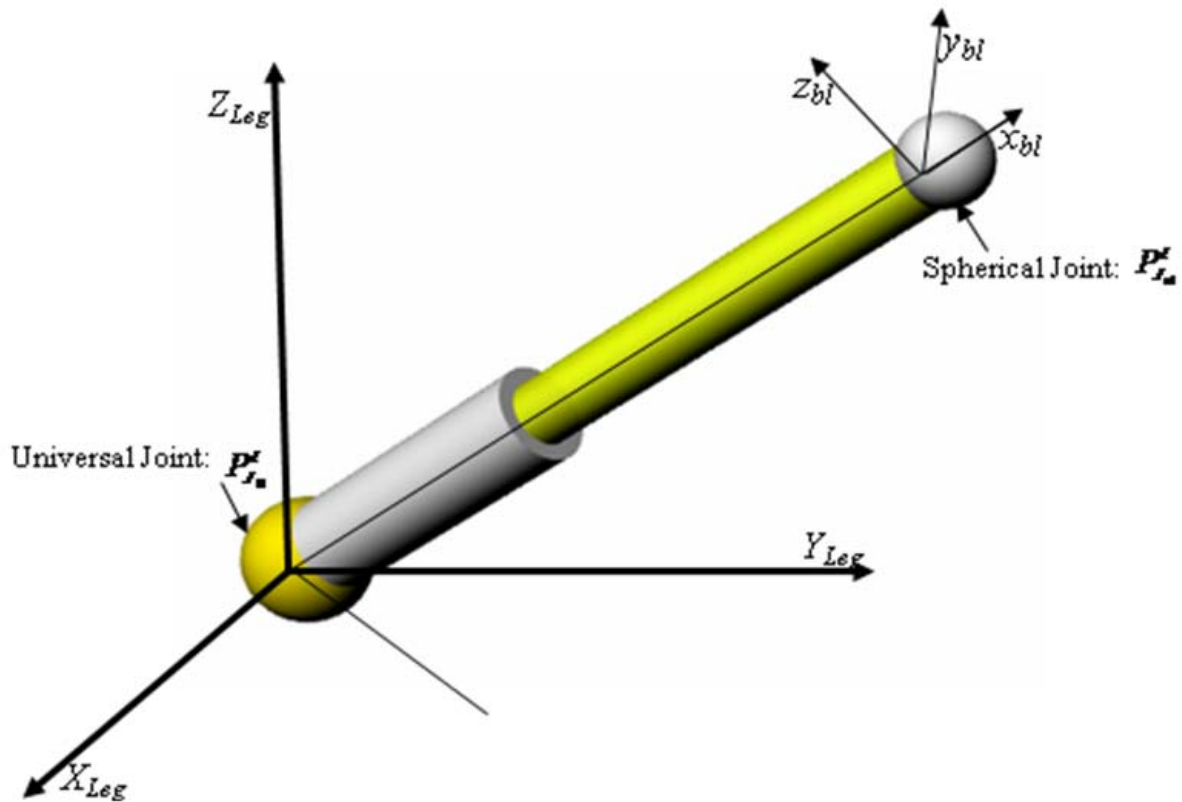


Figure 3. Leg mechanism of Stewart manipulator.

**3. Regression Model and Predicted Squared Error**

If an  $N$ -dimensional vector of independent responses  $Y = [y_1 \ y_2 \ \dots \ y_N]^T$  is modeled in terms of linear combinations of  $K$  modeling predictors  $Q = [q_1 \ q_2 \ \dots \ q_K]$ , then:

$$Y = \gamma_1 q_1 + \gamma_2 q_2 + \dots + \gamma_K q_K + \varepsilon = Q\gamma + \varepsilon \tag{8}$$

where  $\gamma = [\gamma_1 \ \gamma_2 \ \dots \ \gamma_K]$  is an  $K$ -dimensional constant model vector to be determined, and  $\varepsilon$  denotes the  $N$ -dimensional modeling residual vector. Each  $q_i$  is an  $N$ -dimensional vector function depending on the independent variables. In conventional least square estimation (LSE), the optimization goal is to minimize the sum of square errors between the estimated and measured values by defining a scalar function  $J$ .

$$J = \sum_{i=1}^N \varepsilon_i^2 = \varepsilon^T \varepsilon = (Y - Q\gamma)^T (Y - Q\gamma) \tag{9}$$

By minimizing  $J$ , the regression coefficients are calculated by:

$$\gamma = (Q^T Q)^{-1} Q^T Y \tag{10}$$

Although LSE minimizes the error between the measured and estimated data, this method cannot ensure an optimal regression structure. Thus, considering many regression predictors sometimes leads to over parameterization and noisy model. Many methodologies have been proposed to search for the optimal regression structure; such as, backward elimination, stepwise, and others. One of the most efficient techniques to find the minimum number of predictors with minimum error is the predicted squared error (PSE). This technique has an optimization problem because the structure of the model should be chosen to minimize  $F_{PSE}$  defined by [19]:

$$F_{PSE} = \frac{2J}{N} + \frac{K}{N} \sigma_{max}^2 \tag{11}$$

The constant  $\sigma_{max}^2$  is the prior upper-bound estimate of the square error between future data and the model. The  $F_{PSE}$  function depends on the mean squared fit error  $J/N$  and a term proportional to the number of terms in the model. Thus  $\sigma_{max}^2$  is considered as the residual variance estimate for a constant model structure equal to the mean of the measured response values by:

$$\sigma_{max}^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2 \tag{12}$$

where  $\bar{y}_i$  is the mean value of vector  $Y$ .

#### 4. Optimization Procedure for Modeling the Forward Kinematics

Genetic algorithm (GA) is now considered as one of the most popular optimization and search techniques. The first obvious application for the algorithm was traced back to 1962 when Holland introduced the algorithm in his work studying adaptive systems [20]. The algorithm then received an enormous exploration by Goldberg [21]. The main advantages of GA are its global optimization performance and the ease of distributing its calculations among several processors or computers as it operates on the population of solutions that can be evaluated concurrently. It is a very simple method, generally applicable, not inclined to local optimization problems that arise in a multimodal search space, and no needs for special mathematical treatment. Moreover the algorithm is more applicable for the discontinuous problem, as in the case of the present study, unlike the conventional gradient-based searching algorithms.

Basically, genetic algorithm works based on the mechanism of natural selection and evolutionary genetics. The algorithm starts by coding the variables to binary strings (chromosomes). Every chromosome has  $n$  genes. The gene is a binary bit by value zero or one. Three main operations control the procedure of the GA: reproduction, crossover, and mutation. Reproduction is processing to select the parent from a generation. The process is based

on survival of the fittest (highest performance index). In this way, the reproduction process guides the search for the best individuals (high performance index). After the individuals are selected, the crossover process is then used to swap between two chromosomes by specific probabilistic decision. The crossover process generates offspring carrying mixed information from swapped parents (chromosomes). Mutation is the mechanism to prevent the algorithm from local optimal points by adding some degree of randomness. The process is performed by alternation of the gene from zero to one or from one to zero with the mutation point determined uniformly at random. The mutation rate should be considered carefully since the higher mutation rate means more number of generations are required for algorithm convergence and a low mutation rate may lead to a convergence for a local minimum. The algorithm maintains a constant size of generation by selecting the fittest chromosomes from parents and offsprings. The algorithm iteratively operates to converge for schema matches by some tolerance. Further description of genetic algorithms can be found in Goldberg [21-22].

In the current research, the optimization procedure is developed by using a switching factor  $SF$ . This factor  $SF = [sf_1 \quad sf_2 \quad \dots \quad sf_K]$  is a  $K$ -dimensional vector. Each element ( $sf_k$ ) of this vector is a binary bit to disable or enable the vector  $q_i$  as revealed by:

$$SF = [ 1 \quad 1 \quad 0 \quad \dots \quad 0 \quad 0 \quad 1 \quad \dots \quad 0 \quad 1 ]$$

$$Q = \begin{bmatrix} q_{1,1} & q_{2,1} & q_{3,1} & \dots & q_{i-1,1} & q_{i,1} & q_{i+1,1} & \dots & q_{K-1,1} & q_{K,1} \\ q_{1,2} & q_{2,2} & q_{3,2} & \dots & q_{i-1,2} & q_{i,2} & q_{i+1,2} & \dots & q_{K-1,2} & q_{K,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{1,N} & q_{2,N} & q_{3,N} & \dots & q_{i-1,N} & q_{i,N} & q_{i+1,N} & \dots & q_{K-1,N} & q_{K,N} \end{bmatrix} \quad (13)$$

If the value of this bit is “1”, the equivalent predictor vector  $q_i$  is then counted in the structure of the regression model; if the value is “0”,  $q_i$  is omitted. Searching for the optimal  $SF$  or optimal structure model is a discontinued optimization problem. The main reason to use the GA in the current problem is that the most conventional optimization techniques depend on slope or curvature of the response surface which is not available in the current problem. The procedure can be summarized as:

1. Generate a binary vector  $SF$ .
2. Construct the  $Q$  matrix from Eq. (13).
3. Compute the regression constant model vector  $\gamma$  using LSE in Eq. (10).
4. Compute the  $F_{PSE}$  cost function in Eq. (11).

An example is listed here to clear the idea behind the switching factor:

If a set of  $N$  data  $X = [x_1 \quad x_2 \quad \dots \quad x_N]^T$  and  $y = [x_1 \quad x_2 \quad \dots \quad x_N]^T$  is collected.

Assume, there are five functions considered to be candidates for modeling  $y = f(x)$ , these functions are compile in a vector  $Q$  as

$$Q = \left[ 1 \quad x^2 \quad \cos(x) \quad \log(x) \quad \frac{1}{x^2 + 1} \right] \quad (14)$$

In this example, there are  $32 = 2^5$  possibilities to build the regression model. The proposed switching factor  $SF$  is used to code these possibilities. Thus, the switching factor is a 5-dimensional vector. Each element is either “1” or “0”. If the  $SF = [1 \ 0 \ 1 \ 0 \ 0]$ , then the first and third candidates are only passed to the LSE or the selected regression structure is  $y = \gamma_1 + \gamma_2 \cos(x)$ . Based on this regression structure the coefficients  $\gamma_1$  and  $\gamma_2$  are computed and the PSE cost function is evaluated as a function of number of terms  $K = 2$  and the error between the estimated and the measured vector  $y$ . Then, GA to qualify the performance of each  $SF$ , where 32 possibilities are available to construct the structure of the regression model for minimum  $F_{PSE}$ , repeats this procedure. Note: the model is based on the assumption that manipulator has a rigid links or no uncertainty source.

## 5. Simulation Results and Discussion

The proposed scheme is applied to the Stewart platform with parameters given in Table.1 . A population of 15000 samples was seeded from the inverse kinematics model. The regression predictors were generated from a third order polynomial in the sense that any analytical function can be expanded in polynomial form. Each term of this polynomial is a six-dimensional function given by:

$$q_i = \prod_{i=1}^{i=6} L_i^{j_i} \quad \text{and} \quad \sum_{i=1}^6 j_i \leq 3 \quad i = 1, 2, \dots, 6 \quad (15)$$

where  $j_i = 1, 2, \text{ and } 3$  is a power index. The number of predictors generated by Eq. (15) is 84. Six optimization problems are then addressed to model the translational  $\{x(t), y(t), z(t)\}$  and rotational  $\{\varphi(t), \theta(t), \psi(t)\}$  motions in terms of joint space variables  $\{L_i(t), i = 1, 2, \dots, 6\}$ . Genetic algorithms then propagate the search for optimal  $SF$  for each motion. Since  $2^{84}$  possible structures are assigned to be candidates for the optimization search, the length of each chromosome is coded by 84 bits. For the six optimization problems, the mutation rate is 20%. Each generation has a fixed population size 20 or no generation overlap. The optimization code was run many times with different starting guess (initial solution) and it converges to the same solution every time. The algorithm is highly convergent for all the optimization problems. The average number of generations for convergence is 3500. The resultant optimal regression structure is listed in Appendix. A.

Table 1. Manipulator parameters used in the simulation.

Variable	Description	Value	Unit
$L_u$	Length of upper leg	0.95	$m$
$L_l$	Length of lower leg	0.95	$m$
$R_u$	Radius of the movable plate	0.4	$m$
$R_l$	Radius of base plate	1	$m$
$\bar{\alpha}$	Joint angles of base plate	[-50, 50, 70, 170, -170, -70]	$deg$
$\bar{\beta}$	Joint angles of upper plate	[-2, 2, 118, 122 -122 -118]	$deg$

Table 2 shows the computational expenses of the model listed in Appendix A. The computational expense of each expression is computed as the number of terms and the number of multiplication operations. For example, the yaw expression has 30 terms: 21 cubic/tri-linear terms, 8 quadratic/bi-linear terms, and one linear term. The total number of the multiplication in this expression is 80 ( $21 \times 3 + 8 \times 2 + 1$ ) operations. For the full Taylor expansion [12], the total number of terms is 84: 56 cubic/tri-linear terms, 21 quadratic/bi-linear terms, 6 linear terms, and 1 constant term. The total number of multiplication is then 216 operations. Such comparison proves that the proposed technique has the capability to deliver an optimal forward kinematics model for Stewart manipulator. Herein, the

optimality is defined in terms of saving memory and computation expenses. The yaw motion example indicates that there is 65% saving in term of terms and 63% saving in number of multiplications. This example is the worst in the overall model (see Table. 2).

Table 2. Computational expenses of the proposed model in the form of number of terms and number of multiplications for each expression

	Surge	Sway	Heave	Roll	Pitch	Yaw
<b>Number of terms</b>	24	24	25	26	22	30
<b>Number of multiplications</b>	62	66	67	65	48	80

Two tracks are selected to test the model adequacy. The first track is a smooth movement from an initial point  $\{x = 10 \text{ cm}, y = -10 \text{ cm}, z = -8 \text{ cm}, \varphi = 10 \text{ deg}, \theta = -10 \text{ deg}, \text{ and } \psi = -10 \text{ deg}\}$  to a final point  $\{x = -10 \text{ cm}, y = 10 \text{ cm}, z = 8 \text{ cm}, \varphi = -10 \text{ deg}, \theta = 10 \text{ deg}, \text{ and } \psi = 10 \text{ deg}\}$ . The time horizon of this track is 20 sec. This track is assigned to have zero velocities and accelerations for the initial and final points. After the track is generated, the inverse kinematics is then used to find the equivalent joint displacements  $\{L_i(t), i = 1, 2, \dots, 6\}$  as shown in Fig. (4). The second track is a sinusoidal movement. The task displacement amplitudes of these sinusoidal waves are  $A_x = 15 \text{ cm}, A_y = 15 \text{ cm}, A_z = 10 \text{ cm}, A_\varphi = 10 \text{ deg}, A_\theta = 10 \text{ deg}, \text{ and } A_\psi = 5 \text{ deg}$ . The resultant joint displacements of the second track are shown in Fig. (5). The two tracks have been selected to cover a wide range of displacements and velocities inside the workspace of the manipulator. The task space displacements  $\{x(t), y(t), z(t), \varphi(t), \theta(t), \text{ and } \psi(t)\}$  for the first and second tracks are now reproduced by the model listed in Appendix A. Figures (6) and (7) show a comparison between the original track and the simulation results using the proposed model.

Figure (8) shows the variation of the percentage error for the two tracks with time. The results indicate that the percentage error is within  $\pm 4\%$ . The pitch motion has the worst error signal in the first track while the roll motion has the worst error signal in the second track. However, both of them are still inside the acceptable level of the error (less than 5%). The influence of the amplitude and the frequency variations on the error signal for the second track is shown in figure (9). The simulation results show that the roll motion has the highest maximum error for all amplitudes and frequencies. The percentage of the maximum error (ME) for position and velocity roll motion is used as an indicator for the error. There is insignificant variation of ME % with frequency. Such result is expected since the model is a kinematic one. the model delivers the task space displacements as functions of the joint space displacements without any derivatives (see Appendix A). Increasing the amplitude increases the percentage of the maximum error with a quadratic shape.

Table (3) summarizes the statistics of errors from the approximate model in the two test cases. MSE is the mean squared error. ME is the maximum error during the track. The accuracy of the model is tested for both position and velocity since the accuracy in the velocity should be considered for a controller designing purpose (PD controller). The maximum position error in the first track

for sway motion is 0.28 cm and for the pitch motion is 0.41 deg. On the other hand, the maximum velocity error that appeared in sway motion is 0.23 cm/s and in the pitch motion is 0.34 deg/sec. This is sufficiently adequate for building and designing a task space control for the Stewart manipulator. Moreover, the model has a simple computational manipulation.

The authors previously developed the workspace for the same manipulator parameters in reference [4]. The workspace indicates that the heave motion limits are 50 cm

downward and 65 cm upward, if all the rest of motions have zero values. A new trajectory operating close to these boundaries (singularity-nearby) is generated in Fig. 10-a. This test case investigates the singularity-nearby performance. Figure 10-b shows that the accuracy level is within the accepted range, however as the manipulator heads to the limit values as the error increases. All these observations emphasize the applicability of the proposed model to be imbedded in the control loop more efficiently than other approaches.

Table 3. Statistical Comparison between the first and the second tracks: MSE refers to the mean squared error during the track and ME refers to the maximum error during the track

	Position MSE		Velocity MSE		Position ME		Velocity ME	
	Track 1	Track 2	Track 1	Track 2	Track 1	Track 2	Track 1	Track 2
Surge	1.03 $cm^2$	0.81 $cm^2$	0.04 $cm^2/s^2$	0.05 $cm^2/s^2$	0.13 $cm$	0.17 $cm$	0.09 $cm/s$	0.08 $cm/s$
Sway	1.12 $cm^2$	1.45 $cm^2$	0.15 $cm^2/s^2$	0.52 $cm^2/s^2$	0.28 $cm$	0.27 $cm$	0.04 $cm/s$	0.23 $cm/s$
Heave	1.25 $cm^2$	0.86 $cm^2$	0.22 $cm^2/s^2$	0.56 $cm^2/s^2$	0.11 $cm$	0.04 $cm$	0.03 $cm/s$	0.20 $cm/s$
Roll	0.21 $deg^2$	1.09 $deg^2$	0.19 $deg^2/s^2$	0.78 $deg^2/s^2$	0.11 $deg$	0.39 $deg$	0.11 $deg/s$	0.28 $deg/s$
Pitch	2.06 $deg^2$	2.00 $deg^2$	0.96 $deg^2/s^2$	1.89 $deg^2/s^2$	0.41 $deg$	0.33 $deg$	0.12 $deg/s$	0.34 $deg/s$
Yaw	1.05 $deg^2$	0.46 $deg^2$	0.45 $deg^2/s^2$	0.32 $deg^2/s^2$	0.30 $deg$	0.14 $deg$	0.06 $deg/s$	0.13 $deg/s$

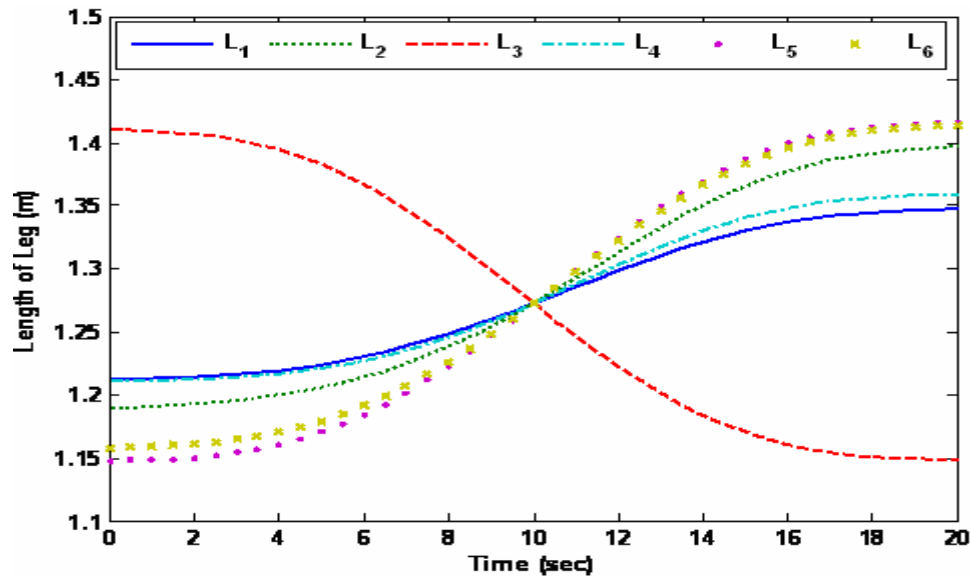


Figure 4. The time history of the joint displacements for the first track.

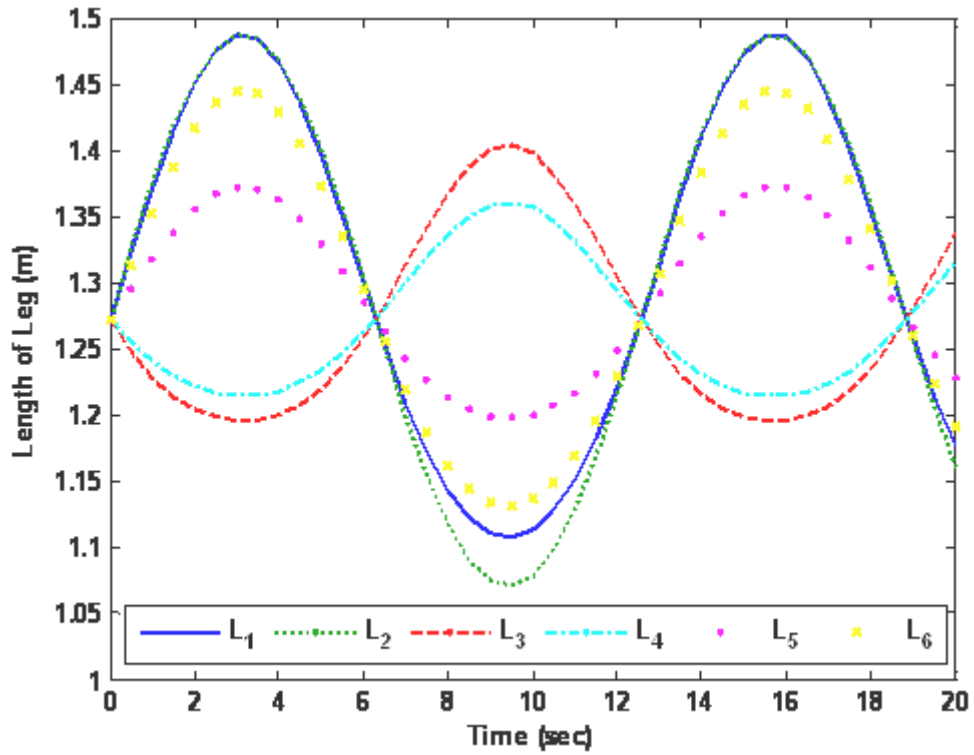
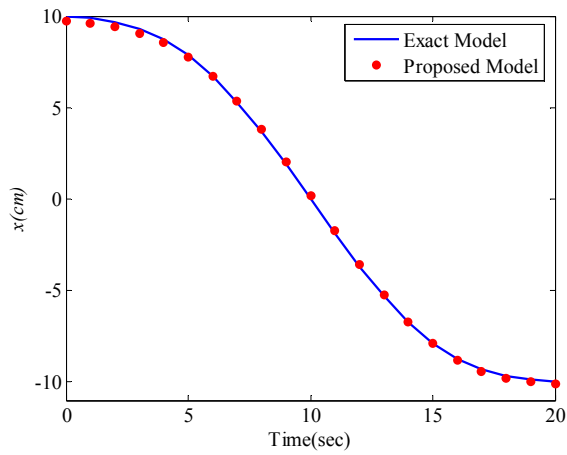
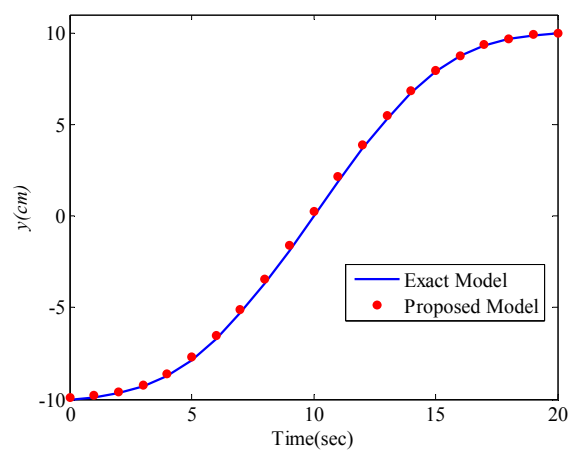


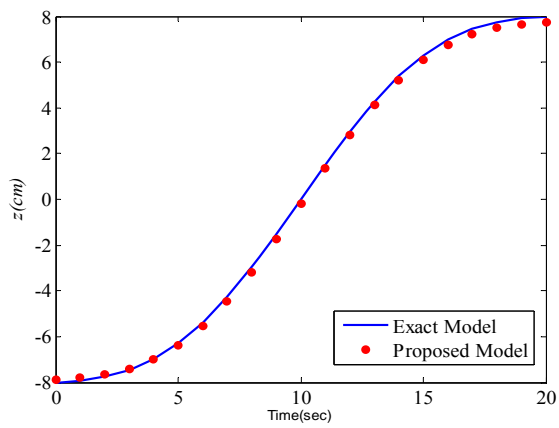
Figure 5. The time history of the joint space displacements for the second track.



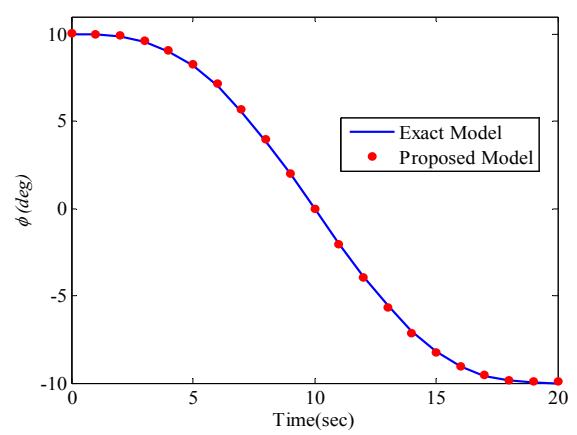
a) Surge motion of the first track



b) Sway motion of the first track



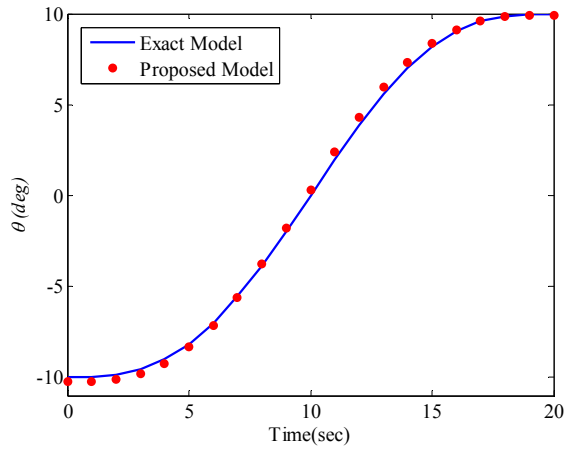
c) Heave motion of the first track



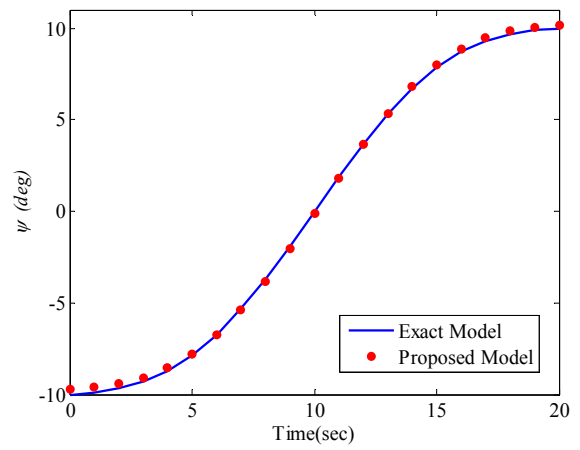
d) Roll motion of the first track

Figure 6. continues next page.....



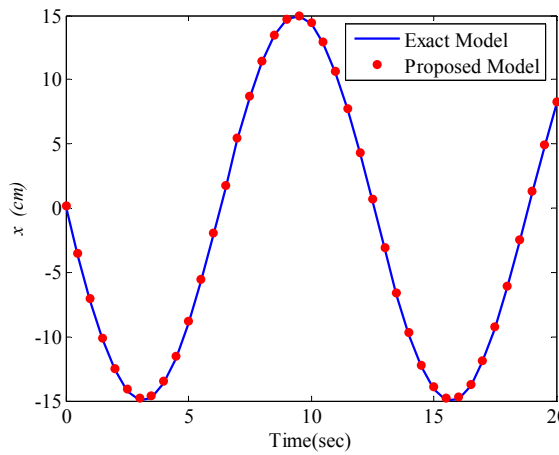


e) Pitch motion of the first track

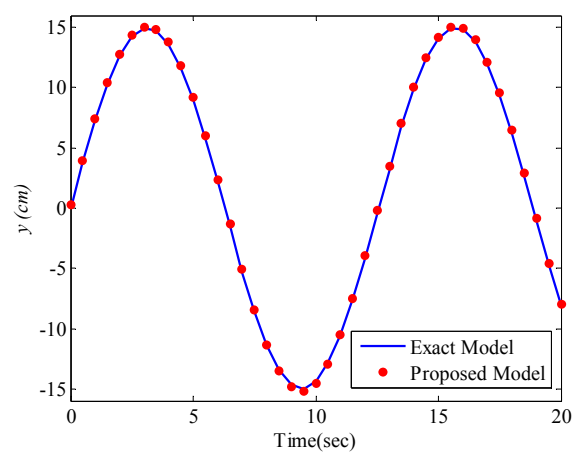


f) Yaw motion of the first track

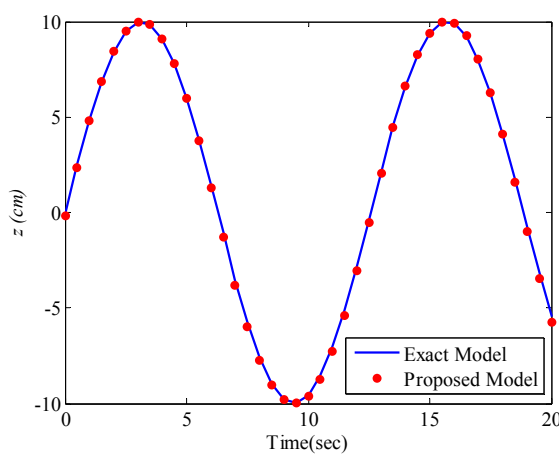
Figure 6. Comparison between the exact model and the results from the proposed approximate model for the first track: smooth motion between two different points.



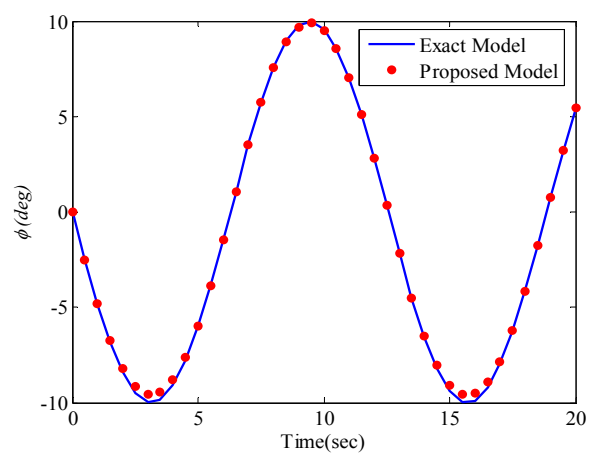
a) Surge motion of the second track



b) Sway motion of the second track

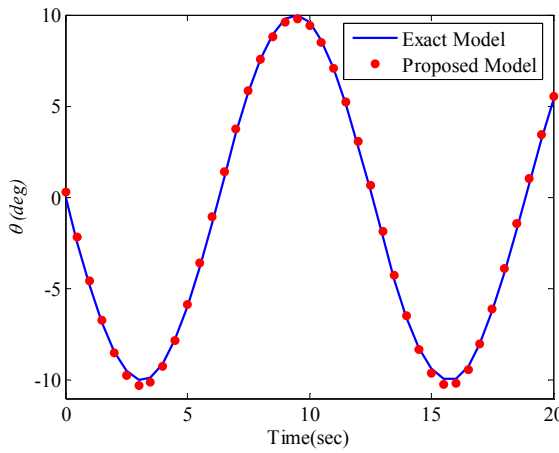


c) Heave motion of the second track

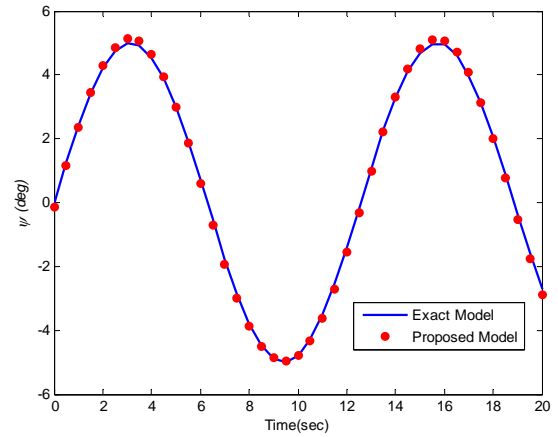


d) Roll motion of the second track

Figure 7. continues next page...

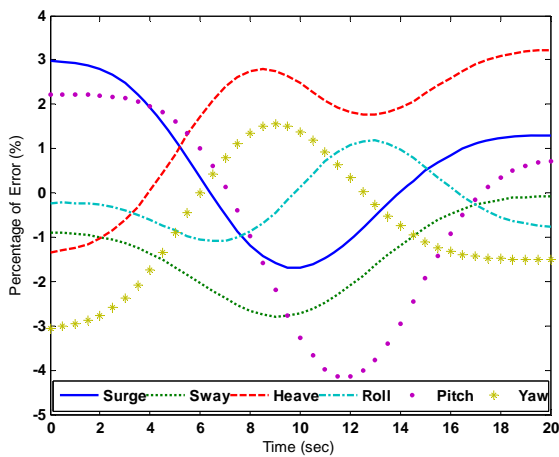


e) Pitch motion of the second track

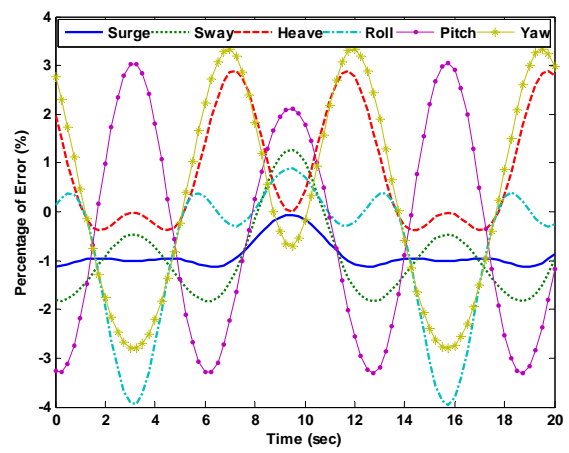


f) Yaw motion of the second track

Figure 7. Comparison between the exact model and the results from the proposed approximate model for the second track: sinusoidal motion with frequency 0.5 rad/s.

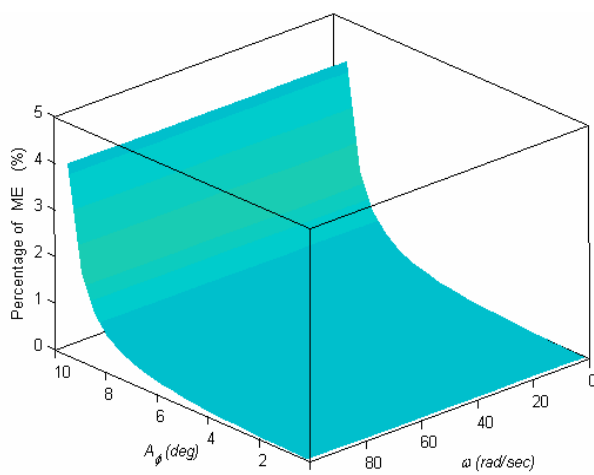


a) First track

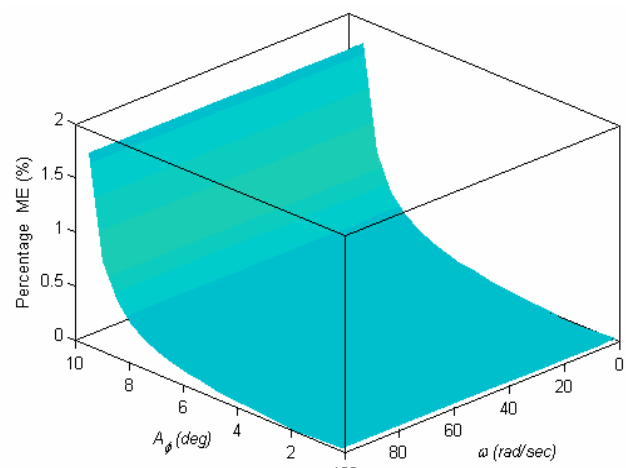


b) Second track

Figure 8. The variation of the error between the exact and the proposed approximate model .



a) Position Error



b) Velocity Error

Figure 9. The variation of the percentage maximum error (ME) for the roll motion and velocity with amplitude and frequency.

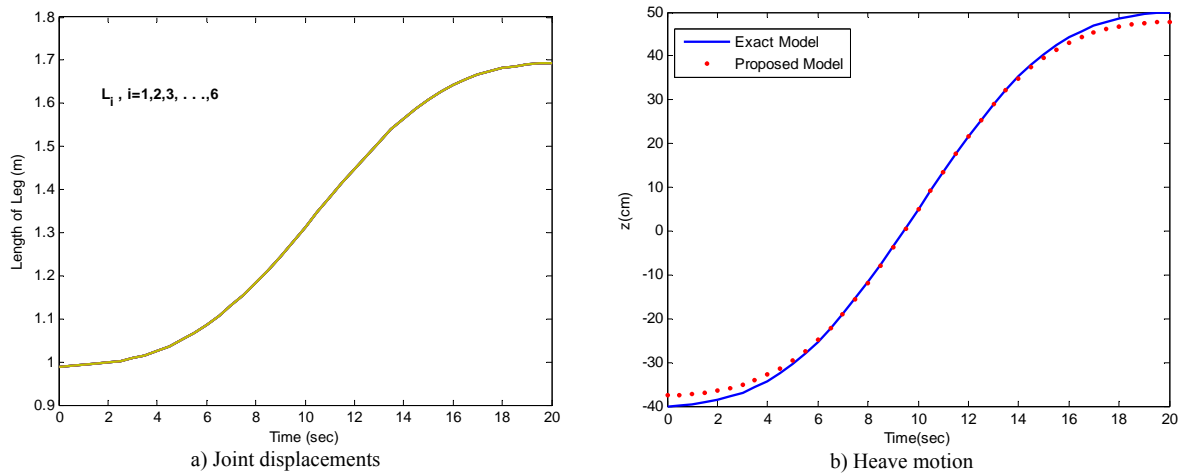


Figure 10. The performance of the proposed model in the singularity-nearby vicinity.

## 6. Conclusion

In this paper, we propose a new procedure to model the Stewart manipulator forward kinematics. The modeling procedure is based on predicted squared error cost function. This cost function is a trade-off between a minimum error and a minimum number of terms in the regression model. To demonstrate the technique feasibility, a simulation by numerical example has been developed.  $2^{84}$  candidate regression structures were selected to model the forward kinematics. The structures' predictors were generated from a third order polynomial in a six-dimensional space. Using a genetic algorithm as a global optimization technique has successfully solved the six optimization problems. In this way, a simple structure of forward kinematics for the Stewart manipulator was developed. The model has the capability to reduce the computational expense by at least 65% from the traditional Taylor expansion previously investigated. Such reduction increases the suitability for the model in the real time control. Thus, most of previous in literature models seek the computational expenses, which restrict their practical ability. The model was tested by two generic tracks providing high precision results with maximum percentage error less than 5%. All these observations emphasize that the proposed model is quite applicable for a task space control.

## References

- [1] I. Nag, W. Chong, "High speed tracking control of stewart platform manipulator via enhanced sliding mode control". IEEE International Conference on Robotics & Automation, Leuven, Belgium, 1998, 2716-2721.
- [2] C. Fang, C. Hung-Hsiang, L. Chin-Teng, "Fuzzy control of a six-degree motion platform with stability analysis". IEEE SMC Conference, Vol.11, 1999, 325-330.
- [3] Y. Su, Y. Duan, C. Zheng, Y. Zhang, G. Chen, J. Mi, "Disturbance-rejection high-precision motion control of a Stewart platform," IEEE transactions on control systems technology, Vol. 12, 2004, 364-374.
- [4] A. Omran, G. El-Bayoumi, M. Bayoumi, A. Kassem, "Genetic algorithm-based optimal control for a 6-DOF non redundant stewart manipulator". International Journal of Mechanical Systems Science and Engineering, Vol. 2, No. 2, 2008, 73-79.
- [5] A. Omran, A. Kassem, G. El-Bayoumi, M. Bayoumi, "Mission-based optimal control of stewart manipulator". Aircraft Engineering and Aerospace Technology Journal, Vol. 81, No. 3, 2009, 226-233.
- [6] P. Nanua, W. Kenneth, M. Vasudeva, "Direct kinematics solution of a stewart platform". IEEE Trans. Robot. Automat. Vol. 6, 1990, 438-444.
- [7] J. Pratik, Y. Sarah, "A hybrid strategy to solve the forward kinematics problem in parallel manipulators". IEEE Trans. Robot. Automat. Vol. 21, 2005, 18-25.
- [8] B. Ilian, R. Jaha, K. Sung-Gaun, L. Sun-Kyu, "A closed-form solution to the direct kinematics of nearly general parallel manipulators with optimally located three linear extra sensors". IEEE Trans. Robot. Automat, Vol. 17, 2001, 148-156.
- [9] D. Li, S. Salcudean, "Modeling, simulation, and control of a hydraulic stewart platform". IEEE Int. Conf on Robotics and Automation, Albuquerque, 1997, 3360-3366.
- [10] S. Fu, Y. Yao, "Comments on "A Newton-Euler formulation for the inverse dynamics of the stewart platform manipulator," Mech. Mach. Theory, Vol. 8, 2006, 1-3.
- [11] P. Dietmaier, "The Stewart-Gough platform of general geometry can have 40 real postures". In Advances in Robot Kinematics: Analysis and Control, J. Lenarcic and M. L. Hust, Eds. Norwell, MA: Kluwer, 1998, 7-16.
- [12] H. Sadjadian, H. Taghirad, "Comparison of different methods for computing the forward kinematics of a redundant parallel manipulator". Journal of Intelligent and Robotic Systems, Vol. 44, No. 3, 2005, 225-246.
- [13] H. Sadjadian, H. Taghirad, A. Fatehi, "Neural networks approaches for computing the forward kinematics of a redundant parallel manipulator". International Journal of Computational Intelligence, Vol. 2, No. 1, 2006, 40-47.
- [14] K. Liu, M. Fitzgerald, F. Lewis, "Kinematic analysis of a Stewart platform manipulator". IEEE Trans. Industrial Electronics, Vol. 40, No. 2, 1993, 282-293.
- [15] K. Liu, F. Lewis, F., M. Fitzgerald, "Solution of nonlinear kinematics of a parallel-link constrained Stewart platform manipulator". Circuits, Systems, and Signal Proc., Special Issue on "Implicit and Robust Systems", Vol. 13, No. 2 1994, 167-183.

- [16] L. Sheng, L. Wan-long, D. Yan-Chun, F. Liang, "Forward kinematics of the Stewart platform using hybrid immune genetic algorithm". Proceedings of the IEEE: International Conference on Mechatronics and Automation, 2006, 2230-2235.
- [17] M. Tarokh, "Real time forward kinematics solution for general Stewart manipulator". IEEE International Conference on Robotics and Automation, 2007, 901-906.
- [18] S. Chen, C. Huang, L. Fu, "Applying a nonlinear observer to solve forward kinematics of a Stewart platform". 17<sup>th</sup> IEEE International Conference on Control Applications, 2008, 1183-1188.
- [19] Barron R. Predicted squared error: A criterion for automatic model selection. Self-Organization Methods in Modeling arrow, S.J., ED., Marcel Dekker, Inc., New York; 1984.
- [20] Holland J. Adaptation in natural and artificial systems. The University of Michigan Press; 1975.
- [21] Goldberg D, David E. The design of innovation: lessons from and for competent genetic algorithms. Boston, MA: Kluwer Academic Publishers; 2002.
- [22] J. John, "Optimization of control parameters for genetic algorithms". IEEE Trans on System, Man, and Cybernetics, Vol. 16, No. 1, 1986, 566-574.

## APPENDIX A: Resultant Model

$$\begin{aligned}
 x(L_1, L_2, L_3, L_4, L_5, L_6) = & \\
 & 0.229L_3L_6^2 - 0.049L_4L_6^2 - 0.167L_2L_6^2 - 0.341L_1L_6^2 - 0.048L_1L_4 - 0.233L_1^2L_3 + 0.303L_3L_6 - 0.281L_3^2L_6 \\
 & + 0.357L_1L_3^2 - 0.317L_1L_2 - 0.074L_2^2L_3 - 0.032L_1L_6 - 0.003L_2L_4^2 + 0.046L_1^2L_5 - 0.014L_2L_4L_5 - 0.003L_2^3 \\
 & + 0.534L_1L_2L_6 - 0.003L_2L_6 - 0.049L_1L_2L_5 + 0.063L_6 + 0.089L_6^3 - 0.0719L_5 + 0.044L_3^2 - 0.047L_1L_3L_4
 \end{aligned}$$

$$\begin{aligned}
 y(L_1, L_2, L_3, L_4, L_5, L_6) = & \\
 & 0.345L_2L_4 + 0.006L_1L_2^2 + 0.125L_3^2L_4 + 0.459L_1L_6 + 0.282L_1^2L_3 + 0.143L_3L_6^2 + 0.381L_3L_6 + 0.309L_1L_2L_5 \\
 & - 0.685L_1L_4 + 0.051L_1L_4^2 - 0.316L_6 + 0.076L_5^3 - 0.005L_4L_5L_6 - 0.256L_2L_3L_6 - 0.454L_1L_2L_3 + 0.362L_1L_2L_6 \\
 & - 0.187L_2L_4L_6 + 0.175L_3L_3L_5 - 0.545L_1L_3L_6 + 0.194L_1L_2L_4 - 0.049L_1^3 - 0.194L_1^2L_2 - 0.104L_4^3 - 0.124L_3^3
 \end{aligned}$$

$$\begin{aligned}
 z(L_1, L_2, L_3, L_4, L_5, L_6) = & \\
 & -1.210 + 0.175L_2L_4^2 - 0.151L_2^2L_4 - 0.405L_1^2L_6 - 0.065L_3L_6^2 - 0.407L_2L_3^2 + 0.072L_3^2L_4 + 0.282L_3L_6 + 0.002L_4^3 \\
 & + 0.413L_4L_5 - 0.229L_3^2L_6 + 0.076L_1L_6^2 - 0.294L_1L_5^2 + 0.049L_1L_6^2 - 0.222L_2L_6^2 + 0.505L_5^2L_6 + 0.252L_2L_3L_6 \\
 & + 0.290L_2L_6 - 0.436L_3L_6^2 + 0.739L_1L_2L_5 - 0.352L_1^2L_2 - 0.106L_1L_4L_5 + 0.039L_1L_3L_6 + 0.212L_1^3 + 0.025L_4^2
 \end{aligned}$$

$$\begin{aligned}
 \varphi(L_1, L_2, L_3, L_4, L_5, L_6) = & \\
 & -0.187L_2^4L_6 - 0.125L_1^2L_5 + 0.741L_1L_4L_6^2 + 0.582L_1^2L_6 - 0.204L_2^2L_3 - 0.048L_2L_6^2 + 0.435L_3L_5 + 1.46L_3L_4 \\
 & - 0.293L_3L_6^2 + 0.040L_1L_6^2 - 0.374L_1L_3^2 - 1.893L_1L_6 - 0.013L_2L_5^2 - 0.088L_1^2L_4 + 0.170L_4L_5^2 - 0.267L_4L_5L_6 \\
 & - 0.289L_3L_5L_6 - 0.541L_3L_4L_5 + 0.278L_2L_3L_5 - 0.563L_1L_4L_6 + 0.434L_1L_4L_5 + 0.863L_1L_3L_6 + 1.433L_2 \\
 & - 0.216L_1^2 - 0.027L_6^3 - 1.298L_4
 \end{aligned}$$

$$\begin{aligned}
 \theta(L_1, L_2, L_3, L_4, L_5, L_6) = & \\
 & 0.119L_2L_6^2 + 0.201L_2L_5 - 0.093L_4L_6^2 - 0.068L_1L_5^2 - 0.006L_3L_6 + 0.328L_3^2L_5 + 0.027L_4L_5^2 - 0.843L_1L_4^2 \\
 & - 0.155L_1L_2^2 - 0.268L_3L_4 - 0.201L_1L_3 + 1.445L_1L_4 - 0.982L_2L_3 + 0.135L_5^2L_6 - 1.063L_6 + 2.091L_5 \\
 & + 0.564L_1L_3L_4 - 0.538L_5^2 - 1.131L_1 - 0.088L_3^2 + 0.616L_4^2 - 0.083L_3L_3L_5
 \end{aligned}$$

$$\begin{aligned}
 \psi(L_1, L_2, L_3, L_4, L_5, L_6) = & \\
 & -0.398L_2L_3^2 + 0.249L_2L_4 + 0.199L_1^2L_3 + 0.042L_2L_4^2 - 0.065L_1^2L_5 - 1.252L_1L_2^2 + 0.928L_2L_6 + 0.267L_1^2 \\
 & + 0.160L_4^3 - 0.598L_1L_3 - 0.102L_2L_5 + 0.513L_1L_6^2 + 1.400L_1^2L_2 + 0.212L_1L_2L_4 - 0.767L_1^2L_6 - 0.454L_1L_4^2 \\
 & + 0.545L_2^3 + 0.162L_4^2 - 0.089L_1^2L_4 - 0.506L_1L_2 - 0.406L_3L_6 + 0.525L_2L_4L_5 - 0.336L_2^2L_6 - 0.150L_4L_5^2 \\
 & + 0.113L_3L_6 - 0.556L_4 + 0.350L_3L_1^2 - 0.505L_2L_4L_6 + 0.264L_2L_3L_6 - 0.439L_1L_5L_6
 \end{aligned}$$

