

Kinematic evaluation and forward kinematic problem for Stewart platform based manipulators

Domagoj Jakobovic

Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb
Croatia
domagoj.jakobovic@fer.hr

Leonardo Jelenkovic

Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb
Croatia
leonardo.jelenkovic@fer.hr

Abstract – In this paper two methods are presented. The first is a kinematic evaluation method for two different hexapod structures: standard Stewart platform manipulator with extensible struts and second structure with fixed strut lengths but sliding guideways on fixed platform. The second method addresses the forward kinematic problem where different mathematical representations are combined with various optimization algorithms to find a suitable combination that may be utilized in real-time environment. Additionally, we note the existence of equivalent trajectories of the mobile platform and suggest an adaptation to the solving method that, having satisfied certain assumptions, is able to successfully solve the forward kinematic problem in real-time conditions with very high precision.

I. INTRODUCTION

Parallel kinematic manipulators (PKM) have been rediscovered in the last decade as microprocessor's power finally satisfies computing force required for their control. Its great payload capacity, stiffness and accuracy characteristic as result of their parallel structure make them superior to serial manipulators in many fields.

One of the most accepted PKM is Stewart platform based manipulator, also known as hexapod or Gough platform. Hexapod, originally, consists of two platforms, one fixed on the floor or ceiling and one mobile, connected together via six extensible struts by spherical or other types of joints. That construction gives mobile platform 6DOF (degree of freedom). Hexapod movement and control is achieved only through strut lengths changes. One variation to this structure, also observed here, is when struts are fixed in length but one of their ends is placed on guideway. Control is then obtained only by moving those joints on guideways. Although in this model the forces acting on struts aren't just along the axis of the struts, like with the original Stewart's design, practically attainable sliding characteristics of guideways make it very considerable structure for manipulators.

One of the qualities we want from a manipulator is its good kinematic characteristics. Those characteristics have direct impact on manipulability and working speed of a manipulator. One part of this paper presents a method for calculating several kinematic parameters. The method can be used to optimize hexapod structure for better kinematic characteristics or combined with other methods were kinematic can be just one measure in optimization process.

Second part of this paper deals with forward kinematics. The forward kinematic [7] of a parallel manipulator is the problem of finding the position and orientation of the mobile platform when the strut lengths are known. This problem has no known closed form solution for the most general 6-6 form of hexapod manipulator (with six joints on the base and six on the mobile platform). In this work several mathematical representations of the forward

kinematic problem, in the form of optimization functions, are combined with various optimization algorithms and adaptation methods in order to find an efficient procedure that would allow for precise forward kinematic solving in real-time conditions.

II. KINEMATIC EQUATIONS

Standard Stewart Platform based manipulator as shown in Fig. 1 can be defined in many ways but most common set of parameters are: minimal and maximal struts length (l_{min} , l_{max}), radii of fixed and mobile platforms (r_1 , r_2), joint placement defined with angle between closest joints for both platforms (a , b) and joint moving area (assuming cone with angle g).

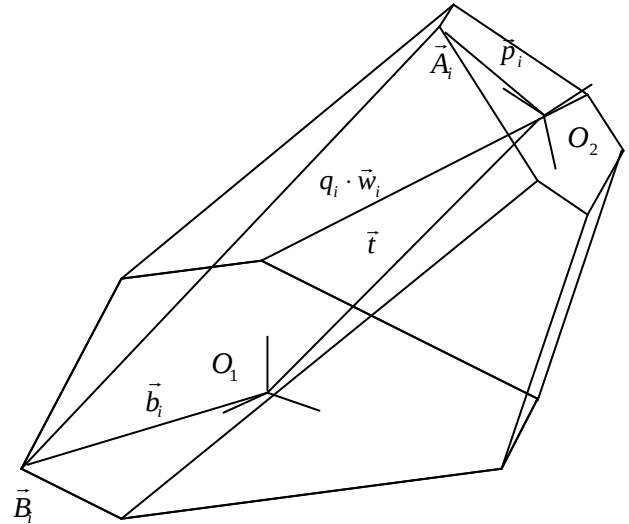


Fig. 1. Stewart Platform manipulator

For kinematic evaluation we need relation between actuators speed and end effector speed. Observing one vector chain through i^{th} strut (Fig. 1), the following equation can be deducted:

$$\vec{b}_i + q_i \cdot \vec{w}_i = \vec{t} + \vec{p}_i \quad (1)$$

Since $\vec{w}_i$ is unity vector and $\frac{\partial \vec{b}_i}{\partial t} = \vec{0}$ derivation of eq.

(1) yields eq. (2), where $\vec{v}$ and $\vec{w}$ are linear and angular end effector velocities.

$$\dot{q}_i \cdot \vec{w}_i + q_i \cdot \vec{w} \times \vec{w}_i = \vec{v} + \vec{w} \times \vec{p}_i \quad (2)$$

Eq. (2) can be easily transformed in form of eq. (3) and then finally in matrix form as on eq. (4). We have a kinematic equation, where relation between end effector velocity and actuator velocity (strut lengths changes) is given.

$$\dot{q}_i = \vec{v} \cdot \vec{w}_i + (\vec{w} \times \vec{p}_i) \cdot \vec{w}_i \quad (3)$$

$$\vec{q} = \begin{bmatrix} \vec{w}_1^T & (\vec{p}_1 \times \vec{w}_1)^T \\ \vdots & \vdots \\ \vec{w}_6^T & (\vec{p}_6 \times \vec{w}_6)^T \end{bmatrix} \cdot \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix} \Leftrightarrow \vec{q} = \underline{J} \cdot \dot{\vec{x}} \quad (4)$$

The second observed hexapod model, shown in Fig. 2, differs from standard Stewart manipulator at base platform and struts. Strut lengths are constant and same for all struts but their joints on one side are placed on sliding guideways where actuators are placed. Parameters which describe this model differ only for base platform: $\vec{B}_{k,i}$ and $\vec{B}_{p,i}$ define i^{th} guide way and t_i as value between $[0, 1]$ identify actual joint position. If we observe models like on Fig. 2, those vectors can be defined using four parameters: d as distance between closer parallel guide ways, r_{11} and r_{12} as radii of circles where guide ways end are placed with height difference h , as shown on Fig. 3.

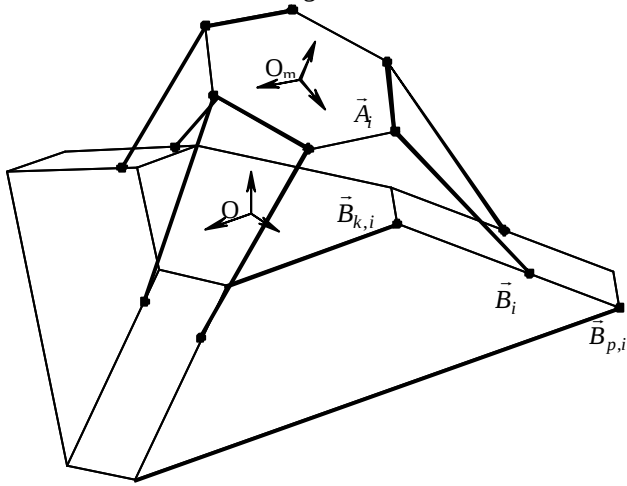


Fig. 2. Hexapod with fixed strut lengths

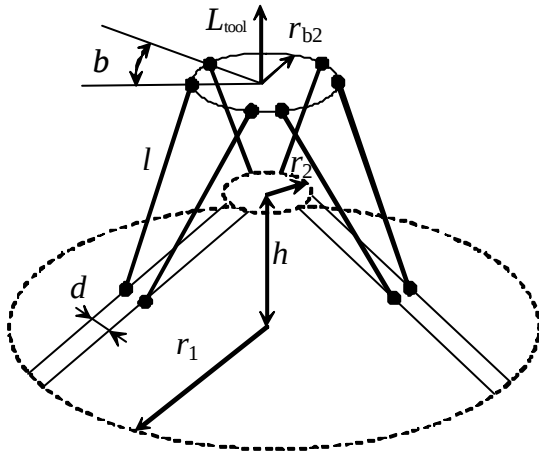


Fig. 3. Parameters that define hexapod structure

Inverse kinematic for this model is slightly more complex than standard hexapod and can be computed using equations:

$$\begin{aligned} \vec{A}_i &= \vec{t} + \vec{p}_i, l = d(\vec{A}_i, \vec{B}_i) \\ \vec{B}_i &= \vec{B}_{p,i} + s_i \cdot (\vec{B}_{k,i} - \vec{B}_{p,i}) \end{aligned} \quad (5)$$

s_i is calculated from quadratic equation and therefore can give two possible joint position on same guide way. This problem must be solved in control procedures.

From Fig. 2, for one vector chain through i^{th} strut, the following equation can be deducted:

$$\vec{b}_i + s_i \cdot \vec{l}_i + q \cdot \vec{w}_i = \vec{t} + \vec{p}_i \quad (6)$$

Derivation of eq. (6) yields eq. (7), and with little more mathematical operations we get kinematic equation (8) very similar to first hexapod model.

$$\dot{s}_i \cdot \vec{l}_i + q \cdot \vec{w} \times \vec{w}_i = \vec{v} + \vec{w} \times \vec{p}_i \quad (7)$$

$$\begin{bmatrix} \vec{w}_1 \cdot \vec{l}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \vec{w}_6 \cdot \vec{l}_6 \end{bmatrix} \begin{bmatrix} \dot{s}_1 \\ \vdots \\ \dot{s}_6 \end{bmatrix} = \begin{bmatrix} \vec{w}_1^T & (\vec{p}_1 \times \vec{w}_1)^T \\ \vdots & \vdots \\ \vec{w}_6^T & (\vec{p}_6 \times \vec{w}_6)^T \end{bmatrix} \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix} \quad (8)$$

$$\underline{L} \cdot \vec{s} = \underline{K} \cdot \dot{\vec{x}} \Rightarrow \vec{s} = \underline{L}^{-1} \cdot \underline{K} \cdot \dot{\vec{x}} = \underline{J} \cdot \dot{\vec{x}}$$

End effector (tool) is placed on mobile platform with height l_{tool} . Therefore, origin of local coordinate system of mobile platform is put in that point.

Using inverse kinematic for any point and tool orientation, it is possible to compute mobile platform position and orientation. Struts lengths for first, or joint positions for second model can then be calculated. If strut lengths are within given ranges, or joints can be placed on guideways for 2nd model, and other constraints are fulfilled, as joint angle constraint and no collision between struts, than hexapod is capable of putting its end effector in given point with given orientation. In this way area reachable with given orientation – the working area, can be found by finding all points which satisfies all constraints.

Assuming that manipulator is used for machining free surface pieces, working area can be better defined as area where manipulator can work for not just one but any required orientation. Required orientations which give optimal surface characteristics can usually be defined with vectors within a cone with defined angle as in Fig. 4. Working area calculated using this definition gives superior visual and numeric description of manipulator.

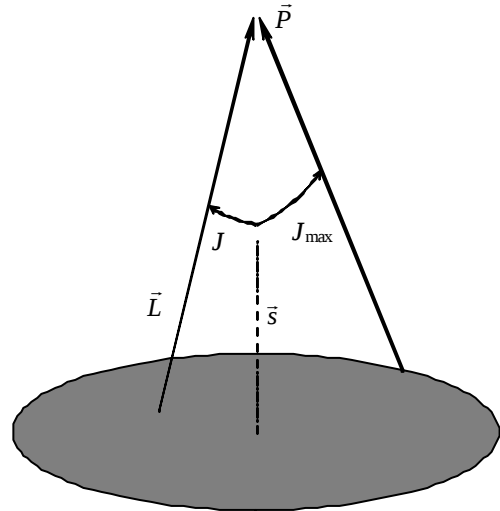


Fig. 4. Orientations used in calculations

When dealing with 6-DOF hexapod manipulators, which on its end effector have tool on spindle, inverse kinematic can't generally give unique result. This gives freedom to apriori choose rotation angle of moving platform as the 6th DOF. For simplicity, no rotation angle was used whenever such orientation was feasible.

III. KINEMATIC PARAMETERS

As equations (4) and (8) show, relation between end effector velocities and strut changes is given by a matrix commonly called jacobian. Kinematic characteristics must therefore be extracted from that matrix. Commonly used values for kinematic evaluation of manipulator are singular values of jacobian [1], [6]. Singular values of matrix $\underline{A}$ are calculated by formula shown in eq. (9), where s_i is i -th eigenvalue of $\underline{A}$.

$$s_i = \sqrt{l_i(\underline{A}^T \cdot \underline{A})}, \quad i \in \{1, 2, \dots, 6\} \quad (9)$$

Geometrical meaning of singular values can be viewed through equations (10) and (11).

$$\|\underline{A}\|_2 = \max_{\|\vec{x}\|_2=1} \|\underline{A} \cdot \vec{x}\| = s_{\max} \quad (10)$$

$$s_{\min} \leq \|\underline{A} \cdot \vec{x}\| \leq s_{\max}, \quad \|\vec{x}\| = 1 \quad (11)$$

If $\vec{x}$ is unity vector then $\underline{A} \cdot \vec{x}$ is a hyperellipsoid with singular values as axis length values. If $\underline{A}$ is jacobian matrix and $\vec{x}$ is velocity, hyperellipsoid represent ability to generate end effector's velocities in given directions. Hyperellipsoid volume is proportional to determinant of a jacobian. Ratio between maximum and minimum singular value is measure for homogeneity. The smaller that ratio is, the ability of generating speed is less dependent of direction.

Practically, the greater the singular value is, the greater strut change is needed for achieving same end effector movement for that particular direction, and vice versa, the smaller singular value means less actuator activity is needed for end effector movement. Very small singular values can cause big problems. If very small actuator movement is enough to move end effector, than even errors as a result of imperfect material, temperature and pressure dilatation, can have big influence on end effector path. In other words, end effector can't be controlled well enough.

Three parameters based on singular values are usually called for kinematic evaluation:

1. condition number: $k = s_{\max}/s_{\min}$ – better smaller values
2. minimal singular value: $s_{\min}$ – better larger values
3. manipulability: $|\det(\underline{J})| = \prod s_i$ – better larger values.

The method we propose to evaluate manipulator from a kinematic aspect is to calculate those three parameters through whole workspace of the manipulator or just on some part of it. For every point where calculations are to be performed, those three parameters are calculated not only for one end effector orientation but for all orientations as shown on Fig. 4. The value for particular kinematic parameter is then calculated as average value.

It is sometimes appropriate to calculate those values on entire volume or just in cross-section with vertical or

horizontal plane to find a spot where kinematic characteristics are better. One such example is on Fig. 5.

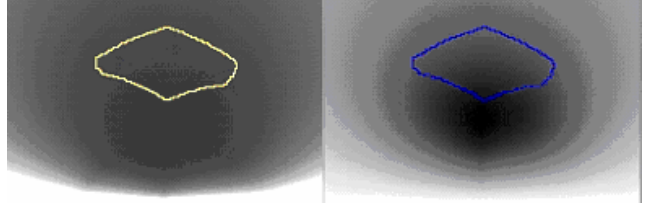


Fig. 5 Conditional number (left) and minimal singular number (right)

Two hexapod models, first as on Fig. 1 and second as on Fig. 2, with parameters shown in TABLE I are evaluated.

TABLE I
Hexapod model parameters

1 st model		2 nd model	
parameter	value	parameter	value
$l_{\max}$	85	l	70
$l_{\min}$	45	r_1	75
r_1	50	r_2	10
r_2	25	h	20
l_{alat}	0	d	3.5
a	0°	r_{b2}	30
β	0°	β	0°
$f_{\max}$	45°	l_{alat}	0
		$f_{\max}$	45°
Working Area Volume			16677

Kinematic parameters are calculated over working area and average values are presented in TABLE II.

TABLE II
Average kinematic parameter values

	1 st model	2 nd model
k	1.835	2.655
$s_{\min}$	1.022	1.225
$ \det(\underline{J}) $	5.842	51.177

For comparison only, models proposed in [1], [3] and [4] are evaluated also and presented in [5]. Some of models have better parameters than 1st model shown in TABLE II, but with at least halved working area volume.

IV. THE FORWARD KINEMATIC PROBLEM

The forward kinematic relations for a hexapod machine can be mathematically formulated in several ways. Every representation of the problem can have its advantages and disadvantages which become emphasized when a different optimization algorithm is applied.

A. The position and orientation of the mobile platform

In order to define a forward kinematic problem we have to represent the actual hexapod configuration, i.e. the actual position and orientation of the mobile platform. The most common approach utilizes the three positional coordinates of the center of the mobile platform and three

angles that define its orientation. The coordinates are represented by vector $\vec{t}$:

$$\vec{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (12)$$

and the three rotational angles are here defined as *roll-pitch-yaw* angles a , b and g . The angle values represent the consecutive rotation about the x , y and z axis, respectively [8]. The hexapod geometry is defined with six vectors for base and six vectors for mobile platform, which define the six joint coordinates on each platform:

$$\vec{b}_i = \begin{bmatrix} b_{ix} \\ b_{iy} \\ 0 \end{bmatrix}, \quad \vec{p}_i = \begin{bmatrix} p_{ix} \\ p_{iy} \\ 0 \end{bmatrix}, \quad i = 1, \dots, 6. \quad (13)$$

The above vectors are represented in local coordinate systems of the base and mobile platform and are of constant value. The base and mobile platform are presumed to be planar, which can be perceived from the z coordinate of the joint vectors. The strut vectors $\vec{l}_i$ can then be expressed as

$$\vec{l}_i = -\vec{b}_i + \vec{t} + \underline{R} \cdot \vec{p}_i, \quad i = 1, \dots, 6, \quad (14)$$

where $\underline{R}$ is the rotational matrix, calculated from three rotational angles. If the position and orientation of the mobile platform is known, the length of each strut is

$$q_i = D(\vec{b}_i, \vec{t} + \underline{R} \cdot \vec{p}_i), \quad i = 1, \dots, 6, \quad (15)$$

where D represents the Euclidean distance between the vector pairs. For an arbitrary solution to a forward kinematic problem, i.e. arbitrary position and orientation of the mobile, the error can be expressed as the sum of squares of differences between the calculated and actual length values. Having stated the above relations, we can define the first optimization function and the related unknowns as

$$F_1 = \sum_{i=1}^6 \left[D(\vec{b}_i, \vec{t} + \underline{R} \cdot \vec{p}_i)^2 - q_i^2 \right]^2, \quad (16)$$

$$\vec{X}_1 = [t_x \quad t_y \quad t_z \quad a \quad b \quad g]^T$$

B. The canonical formulation of the forward kinematics

The idea behind this approach [6] is to use the elements of the rotation matrix, rather than the angle values, to represent orientation:

$$\underline{R} = [\vec{n} \quad \vec{o} \quad \vec{a}] = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}. \quad (17)$$

Without loss of generality we can position the origins of the local coordinate systems of the base and mobile platform at the strut joints with index one, as shown in Fig. 6, which gives us the following parameter values:

$$b_{1x} = b_{1y} = p_{1x} = p_{1y} = b_{2y} = p_{2y} = 0. \quad (18)$$

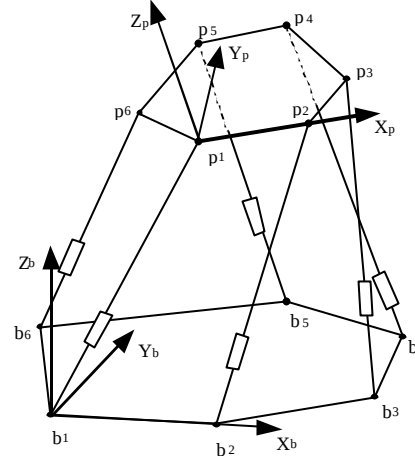


Fig. 6 Positioning of coordinate systems for base and mobile platform

After extensive simplifications, the forward kinematic can be expressed as a system of 9 equations with 9 unknowns. Three of those 9 equations are of linear form, which can be used to reduce the number of variables without introducing additional complexity in the system. Three of the six variables t_x , n_x , o_x , t_y , n_y and o_y can be replaced with linear combinations of the other three, which leaves us with only six unknowns. We used that approach to define another target function as

$$F_2 = (n_x^2 + n_y^2 + n_z^2 - 1)^2 + (o_x^2 + o_y^2 + o_z^2 - 1)^2 + (t_x^2 + t_y^2 + t_z^2 - l_1^2)^2 + (n_x o_x + n_y o_y + n_z o_z)^2 + (n_x t_x + n_y t_y + n_z t_z - A_1 t_x - A_2 n_x - A)^2 + \left(o_x t_x + o_y t_y + o_z t_z - B_1 t_x - B_2 n_x - B_3 o_x - B_4 t_y - B_5 n_y - B_6 o_y - B \right)^2$$

$$\vec{X}_3 = [n_x \quad n_z \quad o_x \quad o_y \quad o_z \quad t_z]^T.$$

and the constants' values can be found in [6] or [12].

In the scope of this work some other problem formulations have been used. Those formulations did not, however, show any advantages over the previous defined two, so they are omitted here. More information can be found in [12].

V. EXPERIMENTAL RESULTS

The forward kinematic problem is presented as five (two of which are shown here) optimization functions for which the optimization algorithm has to find the minimum, the value of the functions being the error of the estimated solution. Several optimization methods have been applied to each of the functions in order to find an effective combination which would allow for real-time application. The algorithms applied in this work are Powell's method, Hooke-Jeeves', steepest descent search, Newton-Raphson's (NR) method, NR method with constant Jacobian and Fletcher-Powell algorithm.

Solving of forward kinematic was simulated in static and dynamic conditions. The goal was to find the combination which would yield the best results considering the convergence, speed and accuracy. The most promising combinations were tested in dynamic conditions, where the

algorithm had to track a preset trajectory of the mobile platform with as small error and as large sampling frequency as possible. Those combinations include Hooke-Jeeves' and Fletcher-Powell algorithm with function F_1 , but the most successful optimization method was Newton-Raphson's algorithm applied to function F_2 .

In dynamic simulation, the starting hexapod configuration is known and serves as an initial solution. During the sampling period T the algorithm has to find the new solution, which will become the initial solution in the next cycle. Several hexapod movements were defined as time dependant functions of the position and orientation of mobile platform. One of those trajectories, hereafter denoted as *A*, is defined with

$$\begin{aligned} x(t) &= 2 \cdot \sin\left(t \frac{p}{2}\right), y(t) = 2.2 \cdot \cos\left(t \frac{p}{2}\right) \\ z(t) &= 7 + 1.5 \cdot \sin(2t), a(t) = 25 \cdot \sin(1.8t), \\ b(t) &= 20 \cdot \sin\left(\frac{t}{2}\right) + 5 \cdot \cos(4t), \\ g(t) &= 15 \cdot \arctan(2t - 4), \quad 0 \leq t \leq 4. \end{aligned} \quad (20)$$

The results of the dynamic simulation are presented in the form of a graph where errors in the three rotation angles and three position coordinates of the mobile are pictured. The sampling period T was set to 1 ms, which equals to a 1000 Hz sampling frequency. The errors shown represent the absolute difference between the calculated and the actual hexapod configuration. Due to the large number of cycles, the error is defined as the biggest absolute error value in the last 100 ms, so the graphs in each point show the *worst* case in the last 100 ms of simulation. The errors are presented separately for angles, in degrees, and position coordinates. The errors for movement *A* and Newton-Raphson algorithm with function F_2 are shown in Fig. 2 and Fig. 3.

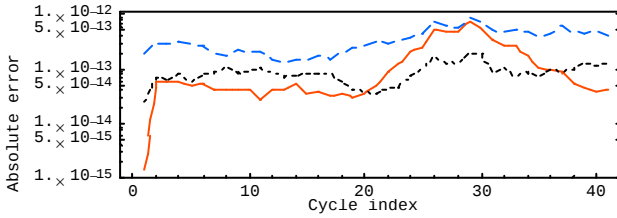


Fig. 7 Absolute angle error ($\alpha =$ —, $\beta =$ ---, $\gamma =$ ···), NR algorithm with F_2 , movement A

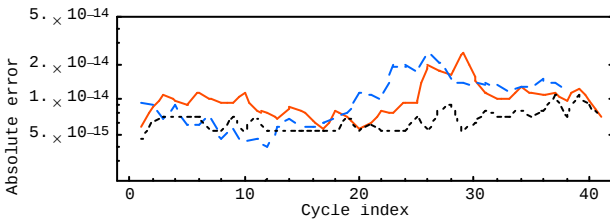


Fig. 8 Absolute coordinate error ($x =$ —, $y =$ ---, $z =$ ···), NR algorithm with F_2 , movement A

The achieved level of accuracy is very high as the absolute error does not exceed 10^{-12} both for angles and coordinates. Another trajectory is derived from the described one by enlarging some of the amplitudes in (20), which is denoted as movement *B* (the altered values are in boldface):

$$\begin{aligned} x(t) &= 2 \cdot \sin\left(t \frac{p}{2}\right), y(t) = 2.2 \cdot \cos\left(t \frac{p}{2}\right) \\ z(t) &= \mathbf{8} + \mathbf{3} \cdot \sin(2t), a(t) = \mathbf{55} \cdot \sin(1.8t), \\ b(t) &= \mathbf{30} \cdot \sin\left(\frac{t}{2}\right) + 5 \cdot \cos(4t), \\ g(t) &= 15 \cdot \arctan(2t - 4), \quad 0 \leq t \leq 4. \end{aligned} \quad (21)$$

The movement *B* errors are shown in Fig. 4. While still low, the error for movement *B* has two distinctive peaks at certain points in simulated motion. What is the cause of those peaks? Mathematical analysis has shown ([9], [10], [11]) that there may exist up to 40 distinctive solutions for forward kinematic problem for Stewart platform with planar base and mobile platform for the *same* set of strut lengths.

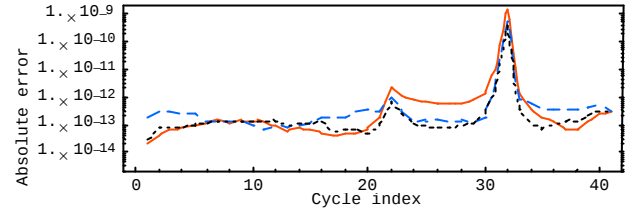


Fig. 9 Absolute angle error ($\alpha =$ —, $\beta =$ ---, $\gamma =$ ···), NR algorithm with F_2 , movement B

Let us suppose that in one hexapod configuration there exists no other forward kinematic solution for actual set of strut lengths, but that in some other configuration there exist several of them. If hexapod in its movement passes through those two configurations, then at a certain point in between there has to be a division point where the number of solutions increases. In those division points the solving algorithm may, unfortunately, begin to follow any of the possible paths, because any of them represents a valid forward kinematic solution! That is exactly the problem that occurs in movement *B*: the algorithm may or may not follow the correct trajectory. If the latter is the case, then the absolute error looks like in Fig. 5.

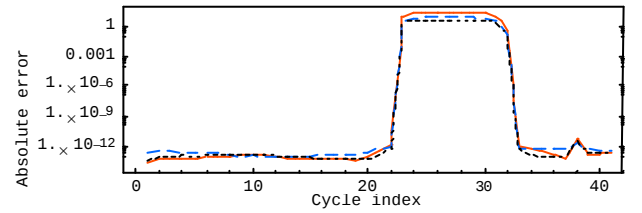


Fig. 10 Absolute angle error ($\alpha =$ —, $\beta =$ ---, $\gamma =$ ···), NR algorithm with F_2 , movement B - division

The algorithm will randomly follow either the correct trajectory or the equivalent one. It is important to note that in both cases the optimization function remains very low (app. 10^{-30} to 10^{-20}) during the whole process because both trajectories depict a valid solution to the forward kinematic problem. The problem is, only one of them represents the actual hexapod configuration in each point of time.

Without any additional information about the hexapod configuration, such as may be obtained from extra transitional displacement sensors, there is unfortunately no way to determine which of the existent solutions to the forward kinematic problem for the same set of strut lengths describes the actual hexapod configuration. Nevertheless, with some assumptions we may devise a strategy that

should keep the solving method on the right track. If the change of the direction of movement is relatively small during a single period, which is in this case only 1 ms, then we can try to predict the position of the mobile platform in the next cycle. We can use the solutions from the past cycles to construct a straight line and estimate the initial solution in the next iteration. Let the solution in the current iteration be $\vec{P}_0$ and the solutions from the last two cycles $\vec{P}_1$ and $\vec{P}_2$. Then we can calculate the new initial solution using one of the following methods:

$$\vec{X}_0 = 2 \cdot \vec{P}_0 - \vec{P}_1, \quad (22)$$

$$\vec{X}_0 = 1.5 \cdot \vec{P}_0 - 0.5 \cdot \vec{P}_2, \quad (23)$$

$$\left. \begin{aligned} \vec{T}_1 &= 0.5 \cdot (\vec{P}_0 + \vec{P}_1), \\ \vec{T}_2 &= 0.5 \cdot (\vec{P}_1 + \vec{P}_2), \\ \vec{X}_0 &= 2.5 \cdot \vec{T}_1 - 1.5 \cdot \vec{T}_2 \end{aligned} \right\}. \quad (24)$$

The above methods were tested in conjunction with NR algorithm and function F_2 for all the simulated trajectories. The results are very good: the solving method was now able to track the correct solution during the whole simulation process for all three estimation methods. The number of conducted experiments was several hundred and every time the algorithm's error margin was below 10^{-11} both for angles and coordinates. However, the described algorithm adaptation will only be successful if the assumption of a small direction change during a few iterations is valid. To test the algorithm's behaviour, simulated movement B was accelerated by factor 2, 4 and 8, while maintaining the same cycle duration of 1 ms. Only by reaching the 8-fold acceleration, when the total movement time equals a very unrealistic half a second, did the algorithm produce significant errors, while still holding to the correct solution.

VI. CONCLUSION

Proposed kinematic evaluation method is pure computational and heavy time consuming. Model must be first defined and then evaluated. However, in regard to most other methods this method gives more realistic kinematic parameter values, because it uses not just one end effector orientation but most orientations that can be asked for in manufacturing. Method can be easily combined with others hexapod evaluations such as working area or/and error analysis giving more powerful hexapod design tool.

Combining several representations of the forward kinematic problem with optimization techniques, an efficient method for solving the forward kinematic was found. The solving method was able to determine the exact position and orientation of the mobile platform within insignificant error margins (less than 10 to the power of -12 of the minimum hexapod dimension) and with 1000 Hz sampling frequency.

The problem of equivalent trajectories was noted: because of the existence of multiple solutions to forward kinematics, there may exist more than one path that mobile platform can follow while having exactly the same strut lengths in every point of the way. It has to be said that every such path represents an equal correct solution of the forward kinematics, but only one of them represents the

true mobile platform trajectory. An empirical algorithm was devised which can increase the probability of finding the right solution, and it proved itself successful in every test case. Unfortunately, it cannot be proven that it will do so in every imaginable movement of the mobile platform. However, the solving method will always find the right solution if the change in the position or moving direction of the mobile platform is relatively small during a few sampling periods.

VII. ACKNOWLEDGMENT

This work was carried out within the research project "Problem-Solving Environments in Engineering", supported by the Ministry of Science and Technology of the Republic of Croatia.

VIII. REFERENCES

- [1] R.S. Stoughton, T. Arai, "A Modified Stewart Platform Manipulator with Improved Dexterity" *IEEE Trans. on Robotics and Automation*, vol. 9, no. 2, pp. 166-173, 1993.
- [2] M. Uchiyama, et al., "Performance Evaluation of manipulators using the Jacobian and its applications to trajectory planning", in *Proc. 2nd Int. Symp. Robotics Res.*, Kyoto, Japan, 1984.
- [3] K.H. Pittens, R.P. Podhorodeski, "A Family of Stewart platforms with Optimal Dexterity", *J. of Robotic Systems*, 10(4):463-479, 1993.
- [4] T. Huang, D.J. Whitehouse, J. Wang, "The Local Dexterity, Optimal Architecture and Design Criteria of Parallel Machine Tools" *Annals of the CIRP*, vol. 47/1, pp.347-351 1998.
- [5] L. Jelenkovic, The Evaluation And Analysis Of Stewart Parallel Mechanisms, MSc thesis (in Croatian), <http://www.zemris.fer.hr/~leonj/>, 2001.
- [6] B. Dasgupta, T.S. Mruthyunjaya, "A Canonical Formulation of the Direct Position Kinematic Problem for a General 6-6 Stewart Platform", *Mech. Mach. Theory*, Vol. 29, No. 6, pp. 819-827, 1994.
- [7] J. - P. Merlet, "Direct Kinematic of Parallel Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, pp. 842-845, 1993.
- [8] R. P. Paul, *Robot Manipulators*, The MIT Press, Cambridge, 1981.
- [9] M. Raghavan, "The Stewart Platform of General Geometry has 40 Configurations", *Journal of Mechanical Design*, Vol. 115, pp. 277-282, June 1993.
- [10] M. Husty, "An Algorithm for Solving the Direct Kinematic Of Stewart-Gough-Type Platforms", <ftp://ftp.mrcim.mcgill.edu/pub/techrep/1994/CIM-94-01.pdf>, 1994.
- [11] F. Wen, C. Liang, "Displacement Analysis of the 6-6 Stewart Platform Mechanisms", *Mechanism and Machine Theory*, Vol. 29, No. 4, pp. 547-557, 1994.
- [12] D. Jakobovic, The Evaluation of Problem Solving Methods for Stewart Parallel Mechanism Kinematics, MSc thesis (in Croatian), http://www.zemris.fer.hr/~yeti/magisterij/MSc_thesis.pdf, 2001.