

THE FORWARD AND INVERSE KINEMATICS PROBLEMS FOR STEWART PARALLEL MECHANISMS

Domagoj Jakobovic
Leonardo Jelenkovic

Mr.sc. Domagoj Jakobovic, University of Zagreb, FEE&C, Unska 3, 10000 Zagreb

Mr.sc. Leonardo Jelenkovic, University of Zagreb, FEE&C, Unska 3, 10000 Zagreb

Keywords: hexapod, forward kinematics, inverse kinematics, iterative algorithms, canonical formulation, workspace area

ABSTRACT

The aim of this work is to combine different mathematical representations of the forward kinematics problem with various optimization algorithms and 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 kinematics problem in real-time conditions with very high precision. In addition inverse kinematics problem is presented and its usage is demonstrated on workspace area calculation.

1. INTRODUCTION

The forward kinematics of a parallel manipulator is 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). This procedure would be invaluable in controlling the force-feedback loop of the manipulator. It would also provide new application possibilities for a hexapod mechanism, such as a force-torque sensor, position-orientation sensor etc. In this work several mathematical representations of the forward kinematics 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 kinematics solving in real-time conditions.

In the last two sections inverse kinematics is briefly described and a method for workspace calculation is presented.

2. THE FORWARD KINEMATICS PROBLEM

The forward kinematics 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 kinematics 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} :

$$\vec{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (1)$$

and the three rotational angles are here defined as *roll-pitch-yaw* angles \mathbf{a} , \mathbf{b} and \mathbf{g} . The angle values represent the consecutive rotation about the x, y and z axis, respectively. 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:

$$\vec{b}_i = \begin{bmatrix} b_{ix} \\ b_{iy} \\ 0 \end{bmatrix}, \quad \vec{p}_i = \begin{bmatrix} p_{ix} \\ p_{iy} \\ 0 \end{bmatrix}, \quad i = 1, \dots, 6. \quad (2)$$

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

$$\vec{l}_i = -\vec{b}_i + \vec{t} + \underline{R} \cdot \vec{p}_i, \quad i = 1, \dots, 6, \quad (3)$$

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

$$q_i = D(\vec{b}_i, \vec{t} + \underline{R} \cdot \vec{p}_i), \quad i = 1, \dots, 6, \quad (4)$$

where D represents the Euclidean distance between the vector pairs. For an arbitrary solution to a forward kinematics 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

$$F_1 = \sum_{i=1}^6 \left[D(\vec{b}_i, \vec{t} + \underline{R} \cdot \vec{p}_i)^2 - q_i^2 \right]^2. \quad (5)$$

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

B. The canonical formulation of the forward kinematics

The idea behind this approach [1.] is to use the elements of the rotation matrix, rather than the angle values, to represent orientation:

$${}^B \underline{R} = [\vec{n} \quad \vec{o} \quad \vec{a}] = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}. \quad (6)$$

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. 1, which gives us the following parameter values:

$$b_{1x} = b_{1y} = p_{1x} = p_{1y} = b_{2y} = p_{2y} = 0. \quad (7)$$

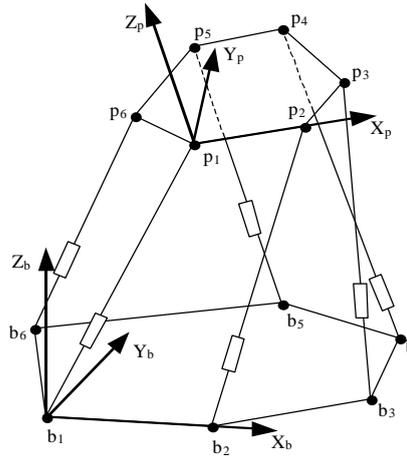


Fig. 1 Positioning of coordinate systems for base and mobile platform

After extensive simplifications, the forward kinematics can be expressed as a system of 9 equations with 9 unknowns, which allows us to define the optimization function F_2 and the related variables vector as

$$\begin{aligned}
 F_2 = & (n_x^2 + n_y^2 + n_z^2 - 1)^2 + (o_x^2 + o_y^2 + o_z^2 - 1)^2 + (t_x^2 + t_y^2 + t_z^2 - l_1^2)^2 \\
 & + (n_x o_x + n_y o_y + n_z o_z)^2 + (n_x t_x + n_y t_y + n_z t_z - A_1 t_x - A_2 n_x - A)^2 \\
 & + (o_x t_x + o_y t_y + o_z t_z - B_1 t_x - B_2 n_x - B_3 o_x - B_4 t_y - B_5 n_y - B_6 o_y - B)^2 \\
 & + (C_{41} t_x + C_{42} n_x + C_{43} o_x + C_{44} t_y + C_{45} n_y + C_{46} o_y - C_4)^2 \\
 & + (C_{51} t_x + C_{52} n_x + C_{53} o_x + C_{54} t_y + C_{55} n_y + C_{56} o_y - C_5)^2 \\
 & + (C_{61} t_x + C_{62} n_x + C_{63} o_x + C_{64} t_y + C_{65} n_y + C_{66} o_y - C_6)^2,
 \end{aligned} \tag{8}$$

$$\vec{X}_2 = [n_x \quad n_y \quad n_z \quad o_x \quad o_y \quad o_z \quad t_x \quad t_y \quad t_z]^T,$$

and the constants' values can be found in [1].

C. Reduced canonical formulation

Three equations used in canonical formulation 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. For instance, if we choose to eliminate the following variables

$$\begin{aligned}
 t_x &= D_{11} n_x + D_{12} o_x + D_{13} o_y + D_1, \\
 t_y &= D_{21} n_x + D_{22} o_x + D_{23} o_y + D_2, \\
 n_y &= D_{31} n_x + D_{32} o_x + D_{33} o_y + D_3,
 \end{aligned} \tag{9}$$

we can define another target function as

$$\begin{aligned}
 F_3 = & (n_x^2 + n_y^2 + n_z^2 - 1)^2 + (o_x^2 + o_y^2 + o_z^2 - 1)^2 + (t_x^2 + t_y^2 + t_z^2 - l_1^2)^2 \\
 & + (n_x o_x + n_y o_y + n_z o_z)^2 + (n_x t_x + n_y t_y + n_z t_z - A_1 t_x - A_2 n_x - A)^2 \\
 & + (o_x t_x + o_y t_y + o_z t_z - B_1 t_x - B_2 n_x - B_3 o_x - B_4 t_y - B_5 n_y - B_6 o_y - B)^2
 \end{aligned} \tag{10}$$

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

D. Other optimization functions

It is possible to further reduce the number of unknowns to only three, but with inevitable complexity increase. Furthermore, the orientation of the mobile platform can be described with rotation vector, which allows us to define another optimization function. Those functions (F_4 , F_5) did not, however, show any advantages over the previous defined three, so they are omitted here. More information can be found in [13.].

3. THE OPTIMIZATION ALGORITHMS

The forward kinematics problem is presented as five 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.

4. EXPERIMENTAL RESULTS

Solving of forward kinematics 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' algorithm with function F_1 and Fletcher-Powell method with functions F_2 and F_3 , but the most successful optimization method was Newton-Raphson's algorithm applied to function F_3 .

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

$$\begin{aligned} x(t) &= 2 \cdot \sin\left(t \frac{\mathbf{p}}{2}\right), y(t) = 2.2 \cdot \cos\left(t \frac{\mathbf{p}}{2}\right), z(t) = 7 + 1.5 \cdot \sin(2t), \\ \mathbf{a}(t) &= 25 \cdot \sin(1.8t), \mathbf{b}(t) = 20 \cdot \sin\left(\frac{t}{2}\right) + 5 \cdot \cos(4t), \mathbf{g}(t) = 15 \cdot \arctan(2t - 4), \\ 0 &\leq t \leq 4. \end{aligned} \tag{11}$$

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 2 ms, which equals to a 500 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 *F3* are shown in Fig. 2 and Fig. 3.

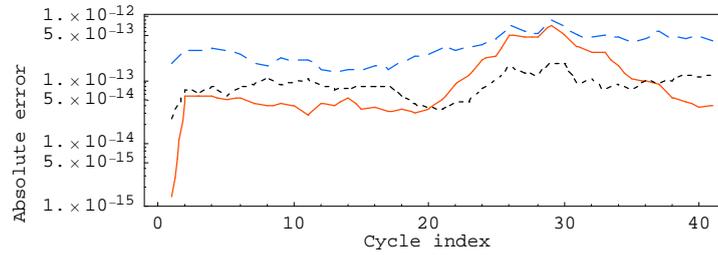


Fig. 2 Absolute angle error ($a = \text{—}$, $b = \text{- - -}$, $g = \text{...}$), NR algorithm with F_3 , movement *A*

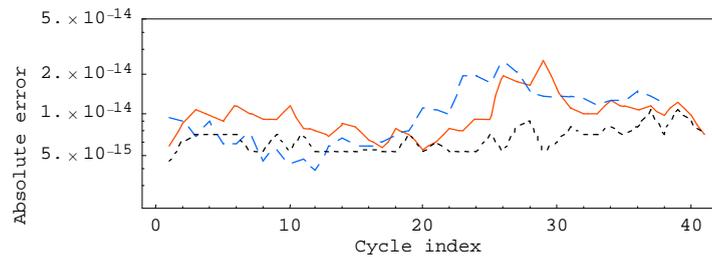


Fig. 3 Absolute coordinate error ($a = \text{—}$, $b = \text{- - -}$, $g = \text{...}$), NR algorithm with F_3 , movement *A*

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 (11), which is denoted as movement *B* (the altered values are in boldface):

$$\begin{aligned}
 x(t) &= 2 \cdot \sin\left(t \frac{\mathbf{p}}{2}\right), y(t) = 2.2 \cdot \cos\left(t \frac{\mathbf{p}}{2}\right), z(t) = \mathbf{8} + \mathbf{3} \cdot \sin(2t), \\
 \mathbf{a}(t) &= \mathbf{55} \cdot \sin(1.8t), \mathbf{b}(t) = \mathbf{30} \cdot \sin\left(\frac{t}{2}\right) + 5 \cdot \cos(4t), \mathbf{g}(t) = 15 \cdot \arctan(2t - 4), \\
 0 &\leq t \leq 4.
 \end{aligned}
 \tag{12}$$

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 (4., 5., 6.) that there may exist up to 40 distinctive solutions for forward kinematics problem for Stewart platform with planar base and mobile platform. The existence of multiple solutions for the same set of strut lengths may prove as a problem for the solving method.

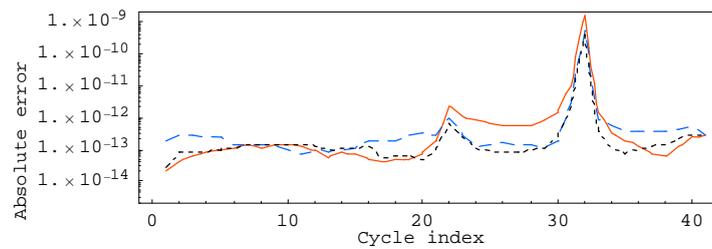


Fig. 4 Absolute angle error ($a = \text{—}$, $b = \text{- - -}$, $g = \text{...}$), NR algorithm with F_3 , movement *B*

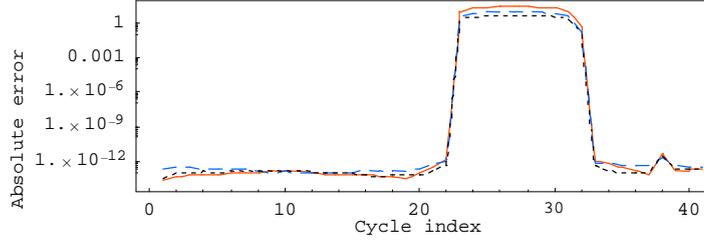


Fig. 5 Absolute angle error ($a = \text{---}$, $b = \text{- - -}$, $g = \text{...}$), NR algorithm with F_3 , movement B - division

Let us suppose that in one hexapod configuration there exists no other forward kinematics 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 kinematics 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, than the absolute error looks like in Fig. 5.

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 kinematics problem. The problem is, only one of them represents the actual hexapod configuration in each point of time. The error between junction points in Fig. 5 actually shows the difference between the two equivalent trajectories.

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 kinematics 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 2 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:

$$\vec{X}_0 = 2 \cdot \vec{P}_0 - \vec{P}_1, \quad (13)$$

$$\vec{X}_0 = 1.5 \cdot \vec{P}_0 - 0.5 \cdot \vec{P}_2, \quad (14)$$

$$\left. \begin{aligned} \vec{T}_1 &= 0.5 \cdot (\vec{P}_0 + \vec{P}_1), \\ \vec{T}_2 &= 0.5 \cdot (\vec{P}_1 + \vec{P}_2), \\ \vec{X}_0 &= 2.5 \cdot \vec{T}_1 - 1.5 \cdot \vec{T}_2 \end{aligned} \right\}. \quad (15)$$

The above methods were tested in conjunction with NR algorithm and function F_3 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 times 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 algorithms behaviour, simulated movement B was accelerated by factor 2, 4 and 8, while maintaining the same cycle duration of 2 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.

5. THE INVERSE KINEMATICS PROBLEM

The inverse kinematics problem is almost trivial for parallel manipulator such as hexapod and is extensively used in many methods.

Inverse kinematics will be presented for two different hexapod structures: standard Stewart platform based manipulator as shown in Fig.6 and discussed in previous sections, and hexapod shown on Fig.7.

First hexapod model 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 (\mathbf{j} , \mathbf{q}) and joint moving area (assuming cone with angle γ). From those values are then calculated values for \vec{a}_i and \vec{b}_i which are used in calculations.

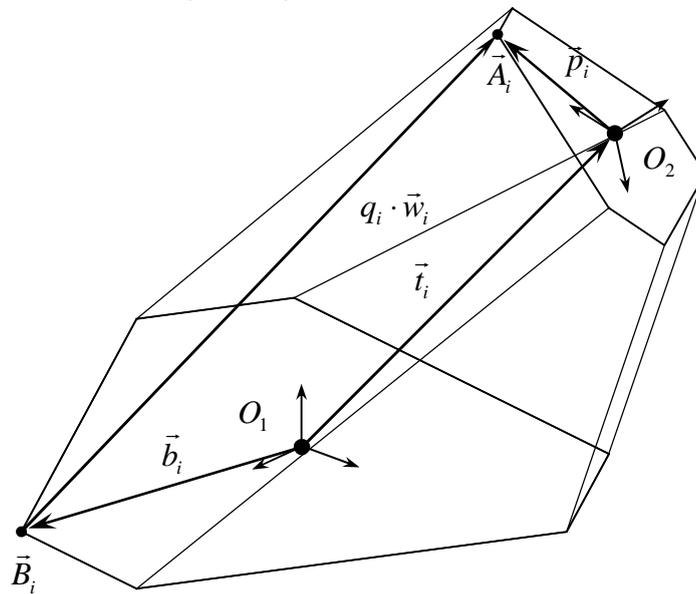


Fig. 6 Stewart Platform manipulator

Inverse kinematics can be described with equations:

$$\vec{A}_i = \vec{t} + \underline{R} \cdot \vec{a}_i, \quad (16)$$

$$q_i = d(\vec{A}_i, \vec{B}_i), \quad (17)$$

where \vec{A}_i and \vec{B}_i are joint position vectors on base and mobile platform, \vec{a}_i are joint position vectors of mobile platform in local coordinate system, \vec{t} is translation vector between base and mobile systems, \underline{R} is orientation matrix of mobile platform, q_i are strut lengths calculated with inverse kinematics and $d()$ is distance between two joints, at the beginning and end of struts – the actual strut lengths.

The second observed hexapod model, shown in Fig.7, 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 where guideways are placed: $\vec{B}_{k,i}$ 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 where pairs of guideways are parallel, those vectors can be defined using four parameters: d as distance between closes parallel guide ways, r_{11} and r_{12} as radii of circles where guide ways ends are placed with height difference h .

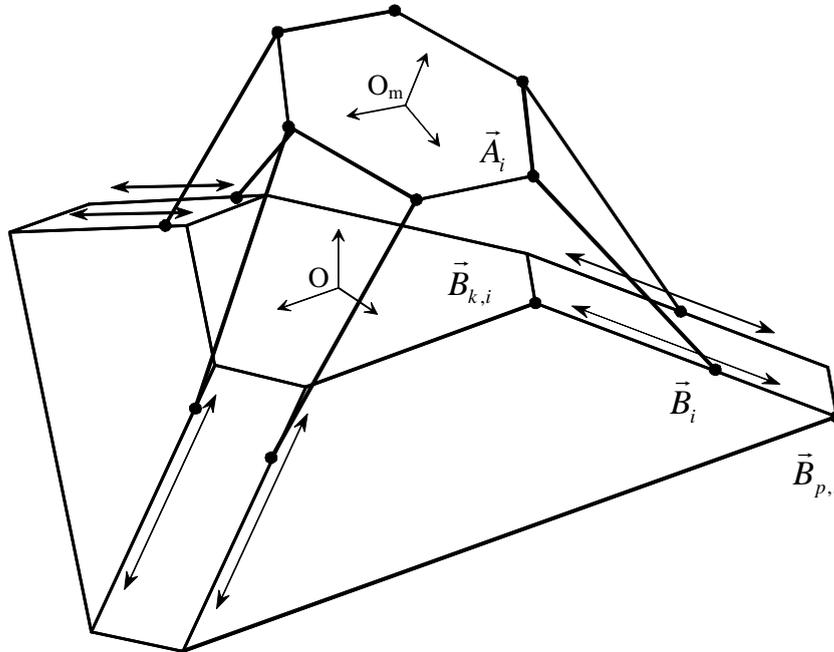


Fig. 7 Hexapod with fixed strut lengths

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

$$\begin{aligned}\vec{A}_i &= \vec{T} + \underline{R} \cdot \vec{a}_i, \\ l &= d(\vec{A}_i, \vec{B}_i), \\ \vec{B}_i &= \vec{B}_{p,i} + s_i \cdot (\vec{B}_{k,i} - \vec{B}_{p,i})\end{aligned}\tag{18}$$

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.

End effector (tool) is placed on mobile platform above the geometrical center of joints placed on that platform by height l_{tool} . Therefore, origin of local coordinate system of

mobile platform is placed in that point. Subsequently vectors \vec{a}_i are calculated for that origin.

6. WORKSPACE AREA CALCULATION

For given end effector (tool) position and orientation defined with translation vector \vec{t} and rotation matrix \underline{R} , joint positions on mobile platform \vec{A}_i can be calculated using (16). Using inverse kinematics strut lengths q_i and directions \vec{w}_i for first model and joint positions s_i and directions \vec{w}_i for second model can be calculated. With these values it is possible to check if that hexapod is able to put its mobile platform to required position verifying several constraints. First, strut lengths must be within given ranges for 1st model and joints can be placed on guideways for 2nd model.

Second, joint constraints must be met. We assumed spherical joints whose restrictions can be defined with some initial direction and a maximal angle which strut can have regarding this direction. Joint constraints' checking is done by comparing those angles on base and mobile platform joints. Initial joint direction on mobile platforms must first be rotated as whole platform before checking is made.

Third and last constraint we checked is struts collisions. Since struts have some thickness it is possible that collision between any two struts occur. It is required that minimal distance between every two struts d_{\min} are calculated as shown on Fig. 8.

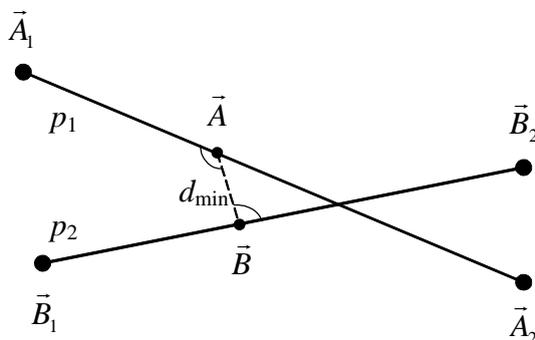


Fig. 8 Distance between two struts

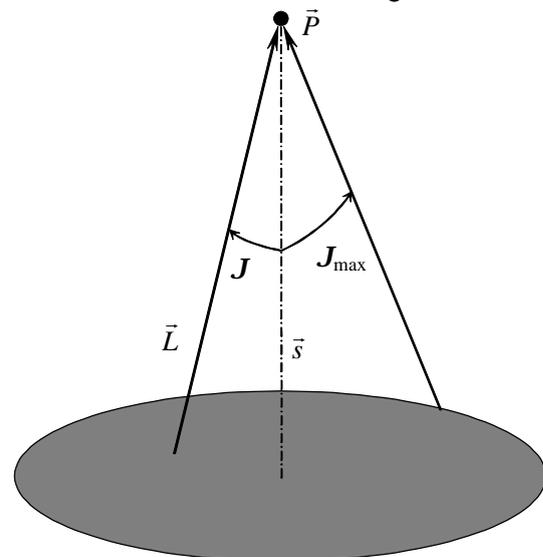


Fig. 9 Orientations used in calculations

For second hexapod model it is also checked for strut collisions with base platform. It is assumed that for first model joint constraints will be violated before collision between struts and platforms will occur and therefore are not checked.

If all constraint for given end effector are satisfied then given position is possible for observed hexapod. With a fixed end effector orientation a predefined area can be checked and workspace with given orientation found.

Assuming that manipulator is used for machining free surface items, working area can be better defined as area where manipulator can work for any required orientation. Required orientations which give optimal surface characteristics usually can be defined

with vectors within a cone with defined angle as in Fig. 9. Working area calculated using this definition gives superior visual and numeric description of manipulator. In our implementation such cone is approximated with a dozen of vectors for each of several different angles J smaller than or equal to J_{\max} , starting with the smallest ones. In this way the result isn't just twofold, and if point isn't a part of workspace we can still obtain information for which J_{\max} it will eventually be.

When dealing with 6-DOF hexapod manipulators, which on its end effector have tool on spindle, inverse kinematics can't generally give unique result. This gives us 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.

Using described methods workspace area for one first and one second hexapod model are calculated and presented both numerical and graphical. Length unit isn't directly specified because values can be easily scaled with any factor while proportions would remain same.

Table 1 show parameters for first hexapod model and volume V calculated for it. Volume for second model is more than three times larger but that hexapod itself is much larger – its base platform is three times wider.

TABLE 1. First hexapod model

param.	value	param.	value
r_1	50	j	30°
r_2	25	q	30°
l_{\max}	90	y	45°
l_{\min}	50	J_{\max}	20°
l_{tool}	10	V	30099

TABLE 2. Second hexapod model

param.	value	param.	Value
r_{11}	75	l_{tool}	10
r_{12}	10	q	30°
h	45	y	45°
d	10	J_{\max}	20°
r_2	10		
l	70	V	108060

Fig. 10 shows workspace area intersections with vertical plane $x=0$.



Fig. 10 Workspace in intersection with $x=0$ for first and second model

Darker the point is the bigger J_{\max} is satisfied. The brightest gray points show workspace with $J_{\max}=0^\circ$, workspace which every point is reachable but with vertical end effector direction. Workspaces as shown in Fig. 10 are not symmetric in intersection with single plane. Intersection would be symmetric if two planes rotated by 120° were used.

Fig. 11 and Fig. 12 shows workspace area shape and hexapod models with its end effector placed in lowest workspace points.



Fig. 11 Workspace and model (1)



Fig. 12 Workspace and model (2)

7. CONCLUSION

Combining several representations of the forward kinematics problem with optimization techniques, an efficient method for solving the problem 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^{-12} of the minimum hexapod dimension) and with 500 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. The solving algorithm may, under some circumstances, skip to an equivalent trajectory at certain division points. 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.

The inverse kinematics problem as opposed to direct kinematics problem is much simple and therefore applicable in different methods regarding parallel structures. Its usage to working area calculation is demonstrated on two parallel structures: first is standard Stewart platform based manipulator and second is model with fixed struts lengths but instead of base platform it has guide ways where joints are placed.

8. 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.

9. LITERATURE

1. B. Dasgupta, T.S. Mruthyunjaya, "A Canonical Formulation of the Direct Position Kinematics Problem for a General 6-6 Stewart Platform", *Mech. Mach. Theory*, Vol. 29, No. 6, pp. 819-827, 1994.
2. J. -P. Merlet, "Direct Kinematics of Parallel Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, pp. 842-845, 1993.
3. R. P. Paul, *Robot Manipulators*, The MIT Press, Cambridge, 1981.
4. M. Raghavan, "The Stewart Platform of General Geometry has 40 Configurations", *Journal of Mechanical Design*, Vol. 115, pp. 277-282, June 1993.
5. 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.
6. 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.
7. J. Tlustý, "High-Speed Machining" *Annals of CIRP*, vol. 42/2, pp. 56-59, 1993.
8. J.P. Merlet, "Designing a parallel manipulator for a specific workspace" Research report 2527, INRIA, 1995.
9. C. Luh, F. Adkins, E. Haung and C. Qiu, "Working Capability Analysis of Stewart Platforms", *ASME Journal of Mechanical Design*, Vol.118, 220-227, 1996.
10. L. Molinari Tosatti, G. Bianchi, I. Fassi, C.R. Boer, F. Jovane, "An Integrated Methodology for the Design of Parallel Kinematic Machines (PKM)" *Annals of the CIRP*, vol.46/2, pp. 341-345, 1997.
11. Q.Y. Wang, et al, "Design and Kinematics of a Parallel Manipulator for Manufacturing", *Annals of CIRP*, 46(1):297-300, 1997.
12. H.J. Warnecke, R. Neugebauer, F. Wieland, "Development of Hexapod Based Machine Tool" *Annals of CIRP*, vol. 47/1, pp. 337-340, 1998.
13. 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.
14. L. Jelenkovic, *The Evaluation and Analysis of Stewart Parallel Mechanisms*, MSc thesis (in Croatian), <http://www.zemris.fer.hr/~leonj>, 2001.