HEXAPOD STRUCTURE EVALUATION AS WEB SERVICE

Leonardo Jelenkovic, Domagoj Jakobovic, Leo Budin
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
Email: leonardo.jelenkovic@fer.hr, domagoj.jakobovic@fer.hr, leo.budin@fer.hr

Keywords: Stewart platform, workspace, error analysis, kinematic parameters, forward kinematics, web services

Abstract: This paper describes several methods for evaluation of kinematic parameters of a Stewart platform. One of those methods is the calculation of workspace area both in numerical and graphical form. The second method allows us to analyze and estimate inherent mechanism errors that occur due to actuator errors, elastic and thermal deformations and other error sources. Furthermore, another procedure is presented which calculates certain kinematic parameters throughout the workspace area of the model and outputs them as numerical and graphical data. Finally, a forward kinematics algorithm designed for use in real-time conditions and its adaptation is presented. The described algorithms are implemented and made available as web services on the project web site (http://hexapod.zemris.fer.hr/).

1 INTRODUCTION

Parallel kinematic manipulators (PKMs) have been rediscovered in the last decade as microprocessor's power finally satisfies computing force required for their control. Their 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 PKMs 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 with spherical or other types of joints. That construction gives mobile platform 6-DOF (degrees 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 a guideway. Control is then obtained only by moving the joints positioned on guideways. Although in this model the forces acting on struts are not just along the axis of the strut, as in the original design, practically attainable sliding characteristics of guideways make it a very considerable structure for manipulators.

One of the qualities we want from a manipulator is its good kinematics behavior. The kinematic characteristics have direct impact on manipulability and working speed of a manipulator. In this paper we present several methods for calculating various kinematic parameters of Stewart platform. These methods can be used to optimize hexapod structure for better kinematic characteristics or combined with other procedures were kinematic can be just one measure in optimization process.

The forward kinematic (Merlet, 1993) 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 paper we present a method for numerical solving of forward kinematics, which is derived from our previous work where several mathematical representations of the forward kinematic problem, in the form of optimization functions, were 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.

2 THE INVERSE KINEMATICS PROBLEM

The inverse kinematics problem compared to forward kinematics is almost trivial for parallel manipulator such as hexapod. Inverse kinematics will be presented here for two different hexapod structures: standard Stewart platform based manipulator as shown in Figure 1 and hexapod shown on Figure 2.

Standard Stewart Platform based manipulator as shown in Figure 1 can be defined with: 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).

Inverse kinematics can be described with equations:

$$\vec{A}_i = \vec{r} + \underline{R} \cdot {}^P\vec{a}_i, \qquad (1)$$

$$q_i = d\left(\vec{A}_i, \vec{B}_i\right), \qquad (2)$$

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, ${}^P\vec{a}_i$ are joint position vectors in local coordinate system of mobile platform, $\vec{r}$ is translation vector between base and mobile systems, $\underline{R}$ is orientation matrix of mobile platform, $d()$ is distance funtion and q_i are strut lengths calculated with inverse kinematics.

The second observed hexapod model, shown in Figure 1, differs from standard Stewart manipulator at base platform and struts. Strut lengths are constant but their joints on one side are placed on sliding guide-ways where actuators are placed. Parameters which describe this model differ only for base platform: $\vec{B}_{k,i}$ and $\vec{B}_{p,i}$ define ith guide way and s_i as value between [0, 1] identify actual joint position.

Inverse kinematics for this model can be defined using equations:

$$\begin{aligned} \vec{A}_i &= \vec{r} + \underline{R} \cdot {}^P\vec{a}_i, \quad l = d\left(\vec{A}_i, \vec{B}_i\right), \\ \vec{B}_i &= \vec{B}_{p,i} + s_i \cdot \left(\vec{B}_{k,i} - \vec{B}_{p,i}\right). \end{aligned} \qquad (3)$$

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

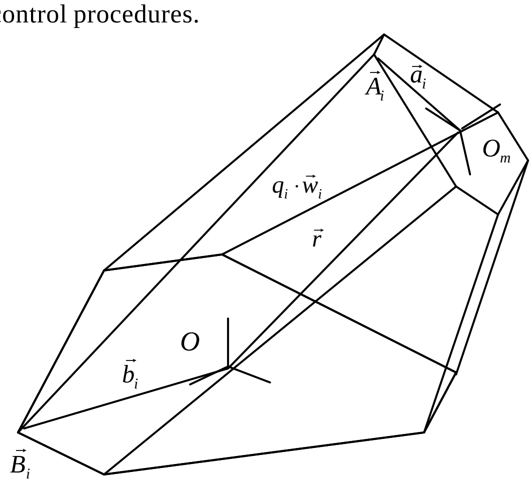

Figure 1: Stewart Platform manipulator

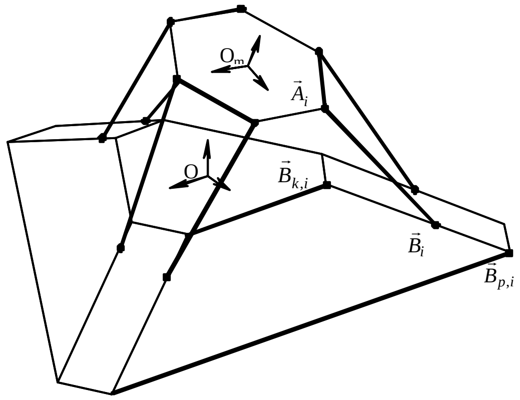

Figure 2: Hexapod with fixed strut lengths

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 $\vec{a}_i$ and ${}^P\vec{a}_i$ are calculated for that origin.

In our work inverse kinematics is used for calculating three hexapod characteristics: workspace volume, error study and kinematics evaluation.

3 WORKSPACE CALCULATION

For given end effector (tool) position and orientation, defined with translation vector $\vec{r}$ and rotation matrix $\underline{R}$, joint positions on mobile platform $\vec{A}_i$ can be calculated. Using inverse kinematics strut lengths q_i and directions $\vec{w}_i$ can be calculated for first model, and joint positions s_i and directions $\vec{w}_i$, for second model. With these values it is possible to check if hexapod is able to put its mobile platform to required position verifying several constraints.

Firstly, strut lengths must be within given ranges for first model or joints must lay on guide ways for second model.

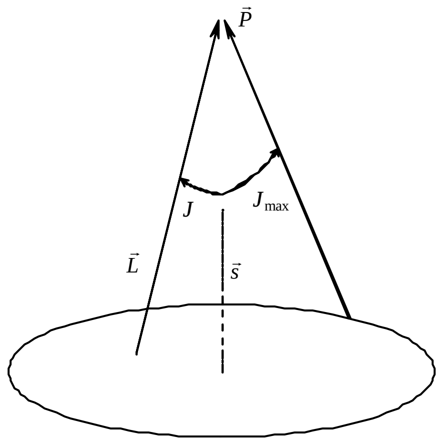

Figure 3: Orientations used in calculations

Secondly, joint constraints must be met. Spherical joints are used in modeling and its restrictions checked.

The third and the last constraints which are checked are struts collisions. Since struts have some thickness it is possible that collisions between any two struts occur. For second hexapod model strut collisions with base platform are also checked.

If all constraint for given end effector are satisfied then given position is reachable. With a fixed end effector orientation a predefined volume 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 were manipulator can work for (almost) any required orientation. Required orientations which give optimal surface characteristics are normals to surface itself. Usually they can be defined with vectors within a cone with defined angle as shown in Figure 3. Working area calculated using this definition gives superior visual and numeric description of manipulator. For performance reasons such cone is approximated with a dozen vectors for each of several different angles J smaller than or equal to $J_{\max}$. In this way the result isn't just twofold, and if point isn't a part of workspace, information for which $J_{\max}$ it will eventually be can still be obtained.

Using the described methods, workspace area for first and second hexapod model can be calculated. Detailed description can be found in (Jakobovic, 2002).

4 ERROR ANALYSIS

Control of hexapod manipulator is based on inverse kinematics. However, that was valid only for models. In reality there must be a feedback through some kind of sensors that measure actual strut lengths and end-effector position. Because of unpredictable environment some hexapod elements may have values different from nominal. This can be due to the assembly errors, elastic and thermal deformations, actuator errors and others error sources (Wang, 1995). Model that includes all sources of errors is hardly possible to implement because of nonlinear dependent error sources and most of error elements can't even be calculated or measured. What is shown in this paper is to give an approximate value for error at end effector if error sources are given as approximate values (tolerances), just quantities, not directions.

From Figure 1, for one vector chain through ith strut, the following equation can be deducted:

$$\vec{b}_i + q_i \cdot \vec{w}_i = \vec{r} + \underline{R} \cdot {}^P\vec{a}_i \,. \tag{4}$$

Differentiating this equation yields:

$$d\vec{b}_i + dq_i \cdot \vec{w}_i + q_i \cdot d\vec{w}_i = d\vec{r} + d\,\underline{R} \cdot {}^P\vec{a}_i + \underline{R} \cdot d\,{}^P\vec{a}_i \,, \tag{5}$$

which can be interpreted as relations between errors in joint positions $d\vec{b}_i, d\,{}^P\vec{a}_i$ and actuator errors dq_i with errors at end effector position $d\vec{r}$ and orientation $d\,\underline{R}$. Furthermore, two more error elements are added to (5), errors in joint centre position (Patel, 1997), both on mobile and fixed platform:

$$\begin{aligned} &d\vec{b}_i + \vec{c}_i + dq_i \cdot \vec{w}_i + q_i \cdot d\vec{w}_i = \\ &\quad d\vec{r} + d\,\underline{R} \cdot {}^P\vec{a}_i + \underline{R} \cdot \left(d\,{}^P\vec{a}_i + {}^P\vec{d}_i \right) \end{aligned} \tag{6}$$

Multiplying (6) with $\vec{w}_i^T$, than replacing $d\,\underline{R} = d\tilde{\vec{\Omega}} \cdot \underline{R}$, where $d\vec{\Omega}$ is orientation error vector, and with simple vector and mathematics transformations (6) becomes (7).

$$\begin{aligned} dq_i &= d\vec{r} \cdot \vec{w}_i + d\vec{\Omega} \cdot \left(\vec{a}_i \times \vec{w}_i \right) + \\ &+ \vec{w}_i \cdot \underline{R} \cdot \left(d\,{}^P\vec{a}_i + {}^P\vec{d}_i \right) - \vec{w}_i \cdot \left(d\vec{b}_i + \vec{c}_i \right) \end{aligned} \tag{7}$$

Equation (7) can be generalized and used in matrix form:

$$\begin{aligned} d\vec{\Lambda} &= \underline{J} \cdot d\vec{\Pi} + \underline{N} \cdot d\vec{A}, \\ d\vec{\Pi} &= \underline{J}^{-1} \cdot \left(d\vec{\Lambda} - \underline{N} \cdot d\vec{A} \right), \end{aligned} \tag{8}$$

where

$$d\vec{\Lambda} = \left[dq_1 \; dq_2 \; \ldots \; dq_6 \right]^T , \tag{9}$$

$$d\vec{\Pi} = \left[d\vec{r}^T \; d\vec{\Omega}^T \right]^T = \left[dr_x \; dr_y \; dr_z \; w_x \; w_y \; w_z \right]^T , \tag{10}$$

$$\underline{J} = \begin{bmatrix} \vec{w}_1^T & \left(\vec{a}_1 \times \vec{w}_1 \right)^T \\ \vdots & \vdots \\ \vec{w}_6^T & \left(\vec{a}_6 \times \vec{w}_6 \right)^T \end{bmatrix} d\vec{A} = \begin{bmatrix} {}^P d\vec{a}_1 + {}^P\vec{d}_1 \\ d\vec{b}_1 + \vec{c}_1 \\ \vdots \\ {}^P d\vec{a}_6 + {}^P\vec{d}_6 \\ d\vec{b}_6 + \vec{c}_6 \end{bmatrix}, \tag{11}$$

$$\underline{N} = \begin{bmatrix} \vec{w}_1^T \cdot \underline{R} & -\vec{w}_1^T & \cdots & \vec{0}^T & \vec{0}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vec{0}^T & \vec{0}^T & \cdots & \vec{w}_6^T \cdot \underline{R} & -\vec{w}_6^T \end{bmatrix}. \tag{12}$$

With formula (8) error in position and orientation at end effector can be calculated if all errors are known or at least presumed.

Formulas for the second hexapod model can be achieved following the same procedure, thus yielding formulas (13) to (16).

$$\begin{aligned} \underline{S} \cdot d\vec{\Psi} &= \underline{J} \cdot d\vec{\Pi} + \underline{N} \cdot d\vec{A} + \vec{B} \\ d\vec{\Pi} &= \underline{J}^{-1} \cdot \left(\underline{S} \cdot d\vec{\Psi} - \underline{N} \cdot d\vec{A} - \vec{B} \right) \end{aligned} \tag{13}$$

Equation (13) is an equivalent for (8) for model with fixed strut lengths. But exact values for each error element must be known to calculate errors at end effector.

$$d\vec{\Psi} = \begin{bmatrix} ds_1 \\ \vdots \\ ds_6 \end{bmatrix} \quad \underline{J} = \begin{bmatrix} \vec{w}_1^T & (\vec{a}_1 \times \vec{w}_1)^T \\ \vdots & \vdots \\ \vec{w}_6^T & (\vec{a}_6 \times \vec{w}_6)^T \end{bmatrix} \tag{14}$$

$$\underline{N} = \begin{bmatrix} \vec{w}_1^T \cdot \underline{R} & \cdots & \vec{0}^T \\ \vdots & \ddots & \vdots \\ \vec{0}^T & \cdots & \vec{w}_6^T \cdot \underline{R} \end{bmatrix} \quad \underline{S} = \begin{bmatrix} \vec{w}_1^T \cdot \vec{l}_1 & \cdots & \vec{0}^T \\ \vdots & \ddots & \vdots \\ \vec{0}^T & \cdots & \vec{w}_6^T \cdot \vec{l}_6 \end{bmatrix} \tag{15}$$

$$d\vec{A} = \begin{bmatrix} {}^P d\vec{a}_1 + {}^P \vec{d}_1 \\ \vdots \\ {}^P d\vec{a}_6 + {}^P \vec{d}_6 \end{bmatrix} \quad d\vec{B} = \begin{bmatrix} dq_1 + \vec{w}_1 \cdot \vec{c}_1 \\ \vdots \\ dq_6 + \vec{w}_6 \cdot \vec{c}_6 \end{bmatrix} \tag{16}$$

What can be done if errors can be only approximated with some border values? Using worst case method and formulas (8) or (13) a maximal error can be found searching through all possible input error values. This method is used in analysis. Because of large search space an approximate iterative numerical method very similar to coordinate axis search is used to find global maximum.

5 KINEMATIC ANALYSIS

For kinematics evaluation, the relation between actuators speed and end effector speed is required. Observing one vector chain through ith strut for the first model, the following equation can be written:

$$\vec{b}_i + q_i \cdot \vec{w}_i = \vec{r} + \vec{a}_i \tag{17}$$

Since $\vec{w}_i$ is unity vector and $\partial\vec{b}_i / \partial r = \vec{0}$ derivation of eq. (17) yields eq. (18), where $\vec{v}$ and $\vec{w}$ are linear and angular end effector velocities.

$$\dot{q}_i \cdot \vec{w}_i + q_i \cdot \vec{w} \times \vec{w}_i = \vec{v} + \vec{w} \times \vec{a}_i \tag{18}$$

Eq. (18) can be easily transformed in form of eq. (19) and then finally in matrix form as on eq. (20). This is a final kinematics equation, where relation between end effector velocity and actuator velocity - strut lengths changes, is given.

$$\dot{q}_i = \vec{v} \cdot \vec{w}_i + (\vec{w} \times \vec{a}_i) \cdot \vec{w}_i \tag{19}$$

$$\dot{\vec{q}} = \begin{bmatrix} \vec{w}_1^T & (\vec{a}_1 \times \vec{w}_1)^T \\ \vdots & \vdots \\ \vec{w}_6^T & (\vec{a}_6 \times \vec{w}_6)^T \end{bmatrix} \cdot \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix} \Leftrightarrow \dot{\vec{q}} = \underline{J} \cdot \dot{\vec{x}} \tag{20}$$

For second model shown on Figure 2, for one vector chain through ith strut, the following equation can be deducted:

$$\vec{b}_i + s_i \cdot \vec{l}_i + q \cdot \vec{w}_i = \vec{r} + \vec{a}_i \tag{21}$$

Derivation of eq. (21) yields eq. (22), and with little more mathematical operations we get kinematics equation (23) very similar to first hexapod model.

$$\dot{s}_i \cdot \vec{l}_i + q \cdot \vec{w} \times \vec{w}_i = \vec{v} + \vec{w} \times \vec{a}_i \tag{22}$$

$$\begin{bmatrix} \vec{w}_1 \cdot \vec{l}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \vec{w}_6 \cdot \vec{l}_6 \end{bmatrix} \begin{bmatrix} \dot{s}_1 \\ \vdots \\ \dot{s}_6 \end{bmatrix} = \begin{bmatrix} \vec{w}_1^T & (\vec{a}_1 \times \vec{w}_1)^T \\ \vdots & \vdots \\ \vec{w}_6^T & (\vec{a}_6 \times \vec{w}_6)^T \end{bmatrix} \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix} \tag{23}$$

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

As equations (20) and (23) show, relation between end effector velocities and strut changes is given by a matrix commonly called jacobian. Kinematics characteristics must therefore be extracted from that matrix. Commonly used values for kinematics evaluation of manipulator are singular values of jacobian (Stoughton, 1993, Pittens, 1993, Huang, 1998).

Three parameters based on singular values are usually used for kinematics evaluation:

1. condition number: $k = s_{\max} / s_{\min}$
2. minimal singular value: $s_{\min}$
3. manipulability: $|\det(\underline{J}^{-1})| = ?\ s_i$.

Proposed method used to evaluate manipulator from a kinematic aspect is to calculate those three parameters through whole workspace of the manipulator or just in 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 Figure 3. The value for particular kinematics parameter is then calculated as average value.

6 FORWARD KINEMATICS

In our work we have combined several mathematical representations of the forward kinematics problem with various optimization algorithms. The algorithms applied in this work were 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. The most successful combination was Newton-Raphson's algorithm applied to canonical representaion of the problem, of which more information can be found in (Jakobovic, 2002) and (Dasgupta, 1994).

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 is defined with

$$
\begin{aligned}
&x(t) = 2 \cdot \sin\left(t\frac{p}{2}\right), y(t) = 2.2 \cdot \cos\left(t\frac{p}{2}\right), \\
&z(t) = 8 + 3 \cdot \sin(2t), a(t) = 55 \cdot \sin(1.8t), \\
&b(t) = 30 \cdot \sin\left(\frac{t}{2}\right) + 5 \cdot \cos(4t), \\
&g(t) = 15 \cdot \arctan(2t - 4), 0 \le t \le 4 .
\end{aligned} \quad (24)
$$

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 drawn. 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 are shown in Figure 3 and Figure 4.

The achieved level of accuracy is very high as the absolute error does not exceed 10^{-12} both for angles and coordinates.

Mathematical analysis has shown (Raghavan, 1993, Wen, 1994) 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. 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.

If that is the case, the algorithm may either follow 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. The problem is, only one of them represents the actual hexapod configuration in each point of time.

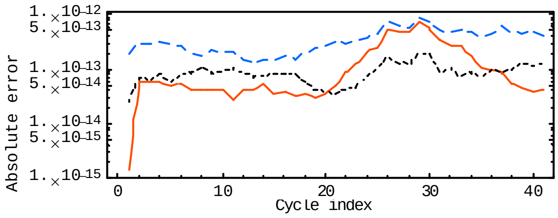

Figure 4: Absolute angle error

(a = , b = , g =)

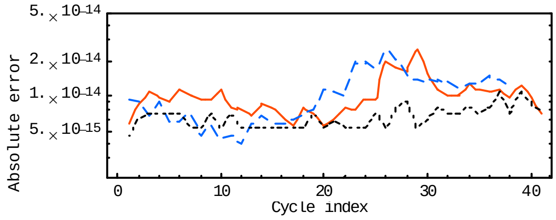

Figure 5: Absolute coordinate error

(x = , y = , z =)

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:

$$\vec{X}_0 = 2 \cdot \vec{P}_0 - \vec{P}_1 , \quad (25)$$

$$\vec{X}_0 = 1.5 \cdot \vec{P}_0 - 0.5 \cdot \vec{P}_2 , \quad (26)$$

$$
\left.
\begin{aligned}
\vec{T}_1 &= 0.5 \cdot (\vec{P}_0 + \vec{P}_1), \\
\vec{T}_2 &= 0.5 \cdot (\vec{P}_1 + \vec{P}_2), \\
\vec{X}_0 &= 2.5 \cdot \vec{T}_1 - 1.5 \cdot \vec{T}_2
\end{aligned}
\right\} . \quad (27)
$$

The above methods were tested for all the simulated trajectories. The results were very good: the solving method was 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 behavior, simulated movement 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 more significant errors, while still holding to the correct solution.

7 HEXAPOD ANALYSIS AS WEB SERVICE

The described methods of hexapod analysis have been implemented as Web services at (http://hexapod.zemris.fer.hr/). Hexapod structure can be defined through Web interface and then a particular operation is performed. Workspace volume can be calculated as a number representing volume in cubic units, or drawn as VRML shape or cross-section with horizontal or vertical plane. Error values and kinematics values can be calculated as overall values through all workspace or just in cross-section with a plane.

Implementation is done through CGI programs, PHP scripting language for Apache Web server, currently running on a two processor Windows 2000 Server. CGI is chosen because of performance issues since analysis methods are computationally intensive. PHP scripts are used to collect hexapod parameters from users and temporary store them in session variables. Before calling CGI programs, PHP script writes parameters to a file on a server. CGI then reads those files, performs calculations and produces results. Depending on demanded calculations, results can be in HTML form, images or VRML files. VRML format is used for displaying hexapod models and its workspace. An implicit surface triangulation method is used for generating workspace. Improving and optimizing process of this triangulation method is in progress.

To utilize multiprocessor system a multithreaded version of program is written since computations can be easily parallelized. Additional 15-20 percent speedup is achieved using hyper-threading technology of Intel Xeon processors.

Regarding speed, workspace calculation can take up to a few minutes to complete. Kinematics is little more time demanding, depending on chosen operation and precision. Error analysis, in spite of enormous effort in optimizing, is still extremely slow and time consuming, and it can take 15 to 20 minutes or even more to compute.

8 CONCLUSION

Methods for hexapod analysis are shown, starting with workspace calculation, error sensitivity analysis and kinematics evaluation. A forward kinematics algorithm designed for use in real-time environment is presented. These methods are prepared and implemented in a functional Web based system.

9 ACKNOWLEDGMENT

This work was carried out within the research project "Distributed Embedded Computing Systems", supported by the Ministry of Science and Technology of the Republic of Croatia.

REFERENCES

Patel, A.J., Ehmann, K.F., 1997. Volumetric Error Analysis of a Stewart Platform-Based Machine Tool, *Annals of the CIRP*, vol. 47/1, pp. 287-290.

Wang, S.M., Ehmann, K.F., 1995. Error Model and Accuracy Analysis of a Six-DOF Stewart Platform, *Manufacturing Science and Eng.*, 2-1, pp. 519-530.

Jakobovic, D., Jelenkovic, L., 2002. The Forward and Inverse Kinematics Problems For Stewart Parallel Mechanisms, *8th Int. Sci. Conf. Production Eng.: CIM2002*, Brijuni, 2002, pp. II-001- II-012.

Huang, T., Whitehouse, D.J., Wang, J., 1998. The Local Dexterity, Optimal Architecture and Design Criteria of Parallel Machine Tools, *Annals of the CIRP*, vol. 47/1, pp.347-351.

Stoughton, R.S., Arai, T., 1993. A Modified Stewart Platform Manipulator with Improved Dexterity, *IEEE Trans. on Robotics and Automation*, vol. 9, no. 2, pp. 166-173.

Merlet, J.-P., 1993. Direct Kinematic of Parallel Manipulators, *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, pp. 842-845

Jakobovic, D., Budin, L., 2002. Forward Kinematics of a Stewart Parallel Mechanism, *Proc. 6th Int. Conf. on Intelligent Engineering Systems INES 2002*, Opatija, May 26-28., pp.149-154

Dasgupta, B., Mruthyunjaya, T.S., 1994. 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,

Raghavan, M., 1993. The Stewart Platform of General Geometry has 40 Configurations, *Journal of Mechanical Design*, Vol. 115, pp. 277-282

Wen, F., Liang, C., 1994. Displacement Analysis of the 6-6 Stewart Platform Mechanisms, *Mechanism and Machine Theory*, Vol. 29, No. 4, pp. 547-557