MODELING TWO CLASSES OF STEWART-GOUGH PLATFORMS [1]

Pedro Cruz [*] **Ricardo Ferreira** [**]
João Silva Sequeira [***]

[*] *Institute for Systems and Robotics*
[**] *Institute for Systems and Robotics*
[***] *Instituto Superior Técnico,*
Institute for Systems and Robotics
IST, Av. Rovisco Pais 1
1049-001 Lisboa, Portugal

Abstract
This paper describes the kinematics of two classes of Stewart-Gough manipulators. The forward kinematics solutions are obtained from the solutions of an optimization problem numerically solved using a steepest descent method with adaptive update steps. The paper presents simulation results for the two classes of robots.

Keywords: Parallel robot, Direct kinematics, Inverse kinematics

1. INTRODUCTION

The use of parallel robots undergoes a renewed interest nowadays due to the variety of possible practical applications. This paper describes a preliminary study on the application of a numerical method to solve the kinematics of two classes of Stewart-Gough robots. The aim of this work is to assess the use of numerical algorithms for a real time application to the control of these classes of robots.

Parallel robot structures can be designed which are specially tailored to handle heavy loads with accurate positioning. Roughly, positioning errors in serial kinematic chains tends to propagate additively throughout the chain links. This is not the case with parallel manipulators, which are consequently capable of performing positioning tasks to a high degree of accuracy. Furthermore, the parallel structure inherently distributes the forces/torques by the actuators giving this class of robots high bandwidth dynamic characteristics. Typical applications include flight simulators, seismic tables, support structures for the accuracy positioning of instrumentation, medical instrumentation and even entertainment devices (see for instance (Merlet, 2000)).

The general Stewart-Gough platform is composed of two rigid bodies connected through a number of prismatic actuators. Usually six actuators are used, pairing arbitrary points in the two bodies. The linkages between the actuators and the bodies are made through universal or ball-in-socket joint.

In Figure 1 is presented a physical model of a Stewart-Gough platform with six prismatic actuators (6-6), constructed with Lego.

This paper describes the kinematic modeling of both the direct and inverse kinematics of two subclasses of the general Stewart-Gough platform, denoted 6-6 and 6-3/\ [3] configurations (the notation in (Merlet, 2000) is followed here). Figure 2 illustrates a geometrical model of the Stewart-Gough
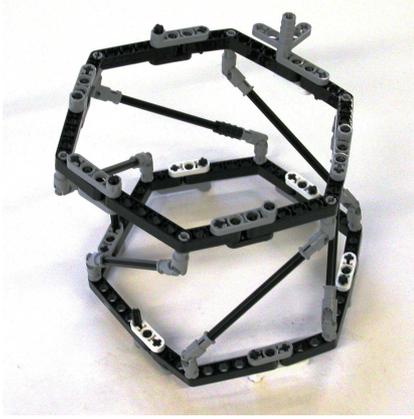
Figure 1. A Stewart-Gough platform model constructed using Lego

robot considered in this paper. The two bodies are denoted by B and P. The polygonal lines joining the actuator anchor points in each body, $\{\mathbf{b}_1, \ldots, \mathbf{b}_6\}$ in B and $\{\mathbf{p}_1, \ldots, \mathbf{p}_6\}$ in P, form simple polygons. The joint variables are denoted by l_i, $i = 1, \ldots, 6$. The i-th actuator connects the points $\mathbf{b}_i$ and $\mathbf{p}_i$ and has as length l_i. For the sake of clarity, body B will be assumed rigidly connected to the ground. The body P moves according to the joint values and the kinematic constraints.
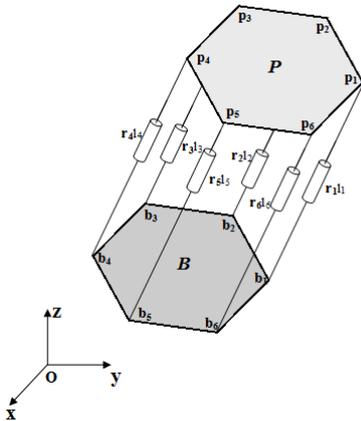

Figure 2. A generic Stewart-Gough platform

Unlike serial manipulators, the direct kinematics of generic parallel manipulators can not be easily written in closed form. Numerical methods are often used to determine the position and attitude of the P body from the set of joint variables (the lengths of the actuators). The direct kinematics of a generic Stewart-Gough platform has often multiple solutions. For instance, the general

Stewart-Gough platform can have 40 real solutions, (Dietmaier, 1998). Practical applications of direct kinematics, e.g, the control of the robot, often require a one to one correspondence between subsets in the spaces of joint variables and positions and attitudes. Therefore, a relevant aspect in the study of the kinematics of parallel robots is related with the ability of the kinematics solutions methods to converge to a particular solution.

Various approaches to the computation of the kinematics of Stewart-Gough robots have been presented in the literature. In (Jakobović and Budin, 2002) the direct kinematics problem is addressed by solving six optimization problems, one for each actuator. Algorithms like Powell's method, Hooke-Jeeves', steepest descent with constant update steps and Fletcher-Powell's were used to solve those problems. The direct kinematics is also addressed as an optimization problem in (Hopkins, 2002), solved using a Newton-Raphson method. The work in (Hopkins, 2002) is integrated in a control architecture. A hybrid strategy using neural networks and Newton-Raphson techniques is proposed in (Parikh and Lam, 2005). In this strategy the neural network stage is used to obtain the initial estimate for the Newton-Raphson method. Dynamic modeling, a fundamental aspect for high performance control, has been addressed in (Khalil and S., 2004).

This paper presents an algorithm based on a steepest descent procedure with adaptive update steps to the computation of the direct kinematics solutions of Stewart-Gough platforms. The direct kinematics problem is formalized as an optimization problem and simulation results obtained from the two classes of robots considered are presented.

The paper is organized as follows. Section 2 describes the direct and inverse kinematics. This section is mainly devoted to the direct kinematics problem, as the inverse kinematics is trivial, formalizing the direct kinematics optimization problem after the Kuhn-Tucker conditions. Section 3 describes a set of simulation experiments. Section 4 presents the conclusions of the paper and points future work.

2. ROBOT KINEMATICS

The direct kinematics of the platform in Figure 2 are completely described by the set of equations

$$\mathbf{p}_i = \mathbf{b}_i + l_i \mathbf{r}_i, \quad i = 1, \ldots, 6 \tag{1}$$

with

$$\|\mathbf{p}_i - \mathbf{p}_j\|^2 - h_{ij} = 0, \tag{2}$$
$$i, j = 1, \ldots, 6$$
$$i \neq j$$

and

$$((\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_5 - \mathbf{p}_1)) \cdot (\mathbf{p}_2 - \mathbf{p}_1) = 0 \quad (3)$$
$$((\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_5 - \mathbf{p}_1)) \cdot (\mathbf{p}_4 - \mathbf{p}_1) = 0$$
$$((\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_5 - \mathbf{p}_1)) \cdot (\mathbf{p}_6 - \mathbf{p}_1) = 0$$
$$((\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_6 - \mathbf{p}_2)) \cdot (\mathbf{p}_1 - \mathbf{p}_2) = 0$$
$$((\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_6 - \mathbf{p}_2)) \cdot (\mathbf{p}_3 - \mathbf{p}_2) = 0$$
$$((\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_6 - \mathbf{p}_2)) \cdot (\mathbf{p}_5 - \mathbf{p}_2) = 0$$

where the l_i are the joint variables, the $\mathbf{r}_i$ are unit vectors oriented along the actuators axis and the h_{ij} are known constants as P is a rigid body.

The inverse kinematics is easily obtained from (1) as

$$l_i = \|\mathbf{p}_i - \mathbf{b}_i\|, \quad i = 1, \ldots, 6 \quad (4)$$

The actuators connect the two rigid bodies B and P respectively at points $\{\mathbf{b}_i\}$ and the $\{\mathbf{p}_i\}$. It is worth to mention that the connection joints are implicitly accounted for in expression (1) by the $\mathbf{r}_i$ vectors.

The input data to the direct kinematics problem is composed by the set of lengths, l_i, of the prismatic actuators connecting B and P and hence the $\mathbf{r}_i$ are undefined. This means that (1) can not be solved directly. A common approach to solve this problem is to transform the direct kinematics into an optimization problem. Defining the cost function,

$$J = \sum_{i=1}^{6} \left(l_i^2 - \|\mathbf{p}_i - \mathbf{b}_i\|^2 \right)^2 \quad (5)$$

where l_i stands for the desired length for the i-th actuator and the $\mathbf{p}_i$ are the unknowns, the solution for the direct kinematics is given by

$$P^\star = \arg_P \min J \quad (6)$$

s.a.

$$g_i \equiv \|\mathbf{p}_i - \mathbf{p}_j\|^2 - h_{ij} = 0, \quad (7)$$
$$i, j = 1, \ldots, 6$$
$$i \neq j$$
$$g_{16} \equiv ((\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_5 - \mathbf{p}_1)) \cdot (\mathbf{p}_2 - \mathbf{p}_1) = 0$$
$$g_{17} \equiv ((\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_5 - \mathbf{p}_1)) \cdot (\mathbf{p}_4 - \mathbf{p}_1) = 0$$
$$g_{18} \equiv ((\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_5 - \mathbf{p}_1)) \cdot (\mathbf{p}_6 - \mathbf{p}_1) = 0$$
$$g_{19} \equiv ((\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_6 - \mathbf{p}_2)) \cdot (\mathbf{p}_1 - \mathbf{p}_2) = 0$$
$$g_{20} \equiv ((\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_6 - \mathbf{p}_2)) \cdot (\mathbf{p}_3 - \mathbf{p}_2) = 0$$
$$g_{21} \equiv ((\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_6 - \mathbf{p}_2)) \cdot (\mathbf{p}_5 - \mathbf{p}_2) = 0$$

It is worth to note that (6) could be written as a strict quadratic cost function. However, this form is easy to work with.

The constrained cost function is the Lagrangian,

$$J_\lambda = J + \sum_{i=1}^{21} \lambda_i g_i \quad (8)$$

where the λ_i are Lagrange multipliers. The solutions can be easily obtained from the Kuhn-Tucker conditions, (Luenberger, 1973), namely

$$\nabla J_\lambda = 0$$
$$\nabla^2 J_\lambda \text{ is positive definite} \quad (9)$$

As aforementioned, a variety of methods to solve (9) in the context of parallel robots has been reported in the literature. In this paper a steepest descent method with adaptive steps is considered. It is worth to point on that the problem has a total of $6 \cdot 3 + 21$ variables and hence solution techniques requiring the explicit inverse of $\nabla^2 J_\lambda$, such as standard Newton methods, may be innapropriate in what concerns real time applications. The step updating method is based on that proposed in (Almeida and Silva, 1991), in the context of neural networks, where the update rate depends on the change of the sign in the gradient coordinate being updated. The criterion proposed in this paper differs from the aforementioned one as the updating is identical in all gradient directions.

Defining the vector of unknowns at instant k as $x_k = [\mathbf{p}_1, \ldots, \mathbf{p}_6, \lambda_1, \ldots, \lambda_{21}]_k$ the step update algorithm is written,

$$x_{k+1} = x_k - \eta_k \nabla J_{\lambda_k} \quad (10)$$

where the update of the adaptive step is given by

$$\eta_{k+1} = \begin{cases} \eta_k \mu \text{ if } \nabla J_{\lambda_k} \cdot \nabla J_{\lambda_{k-1}} \geq 0 \\ \eta_k d \text{ if } \nabla J_{\lambda_k} \cdot \nabla J_{\lambda_{k-1}} < 0 \end{cases} \quad (11)$$

with $\mu > 1$ and $0 < d < 1$. This method decreases the steepest update rate whenever the cost function is having difficulties in finding a convergence direction, i.e., ∇J_{λ_k} and $\nabla J_{\lambda_{k-1}}$ point to very different directions.

It is worth to note that the algorithm presented to solve the direct kinematics problem also converges using the technique described in (Almeida and Silva, 1991). However, using the step update technique in (11) was empirically found to converge in a lesser number of iterations than that using the updating technique in (Almeida and Silva, 1991).

3. SIMULATION RESULTS

This section presents two experiments to illustrate the solution of the direct kinematics problem for the two classes of Stewart-Gough robots considered.

For both experiments the initial step was set to $\eta_0 = 0.0001$ and the update steps rates were set to d $= 0.01$ and $\mu = 1.05$. The initial Lagrange multipliers were set to $\lambda_{1,\ldots,21} = 0.3$ Note that the method does not require that the rigid body constraints be verified at the start of the simulations.

In the first experiment the points p_i and b_i were chosen in positions yielding regular hexagonal polygons. This kinematic configuration, often named a 6-6 robot, is of the type depicted in Figure 2. The final robot configuration obtained from the simulation is presented in Figure 3.
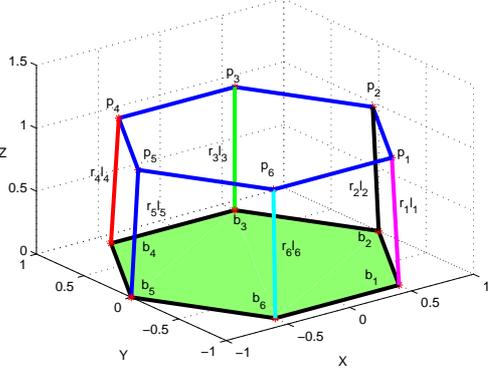

Figure 3. Solution for the 6-6 robot

Table 1 presents the values used as initial conditions by the algorithm. For this experiment, the optimal actuator values are $l_{1,...,6} = 1$.

Act.	l_i	$\mathbf{p}_i(x,y,z)$	$\mathbf{b}_i(x,y,z)$	$\mathbf{r}_i(x,y,z)$
1	2.5495	(0.7500 -0.4330 2.5000)	(0.5000 -0.8660 0.0000)	(0.0981 0.1698 0.9806)
2	2.5495	0.7500 -0.4330 2.5000)	(1.0000 0.0000 0.0000)	(-0.0981 -0.1698 0.9806)
3	2.5495	(0.0000 0.8660 2.5000)	(0.5000 0.8660 0.0000)	(-0.1961 0.0000 0.9806)
4	2.5495	(0.0000 0.8660 2.5000)	(-0.5000 0.8660 0.0000)	(0.1961 0.0000 0.9806)
5	2.5495	(-0.7500 -0.4330 2.5000)	(-1.0000 0.0000 0.0000)	(0.0981 -0.1698 0.9806)
6	2.5495	(-0.7500 -0.4330 2.5000)	(-0.5000 -0.8660 0.0000)	(-0.0981 0.1698 0.9806)

Table 1. Experiment 1a - Initial conditions for the 6-6 robot

The algorithm was stopped after 116 iterations, with a final quadratic error of 0.9×10^{-5}. Table 2 displays the lengths of the actuator rods, along with the corresponding positions of the connecting points in the P body.

Figure 4 displays the evolution of the quadratic error. In this particular experiment, the error converges in few iterations.

Figure 5 displays the quadratic error presented in Figure 4 using a logarithmic scale. The quadratic error decreases quickly in the first 60 iterations.

Act.	l_i	$\mathbf{p}_i(x,y,z)$	$\mathbf{b}_i(x,y,z)$	$\mathbf{r}_i(x,y,z)$
1	1.0006	(0.4890 -0.8447 1.0003)	(0.5000 -0.8660 0.0000)	(-0.0110 0.0214 0.9997)
2	1.0006	0.9760 -0.0011 1.0003)	(1.0000 0.0000 0.0000)	(-0.0240 -0.0011 0.9997)
3	1.0006	(0.4870 0.8458 1.0003)	(0.5000 0.8660 0.0000)	(-0.0130 -0.0202 0.9997)
4	1.0006	(-0.4870 0.8458 1.0003)	(-0.5000 0.8660 0.0000)	(0.0130 -0.0202 0.9997)
5	1.0006	(-0.9760 -0.0011 1.0003)	(-1.0000 0.0000 0.0000)	(0.0240 -0.0011 0.9997)
6	1.0006	(-0.4890 -0.8447 1.0003)	(-0.5000 -0.8660 0.0000)	(0.0110 0.0214 0.9997)

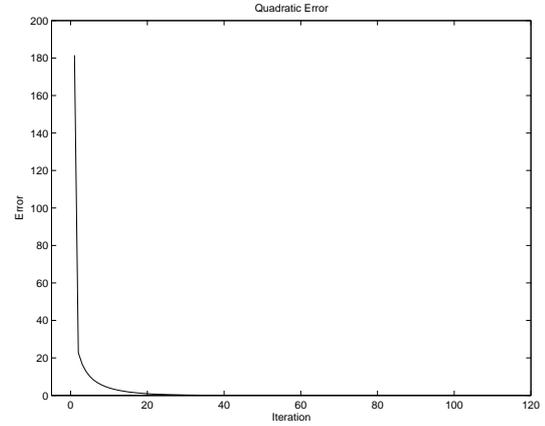Table 2. Experiment 1a - Final results for the 6-6 robot

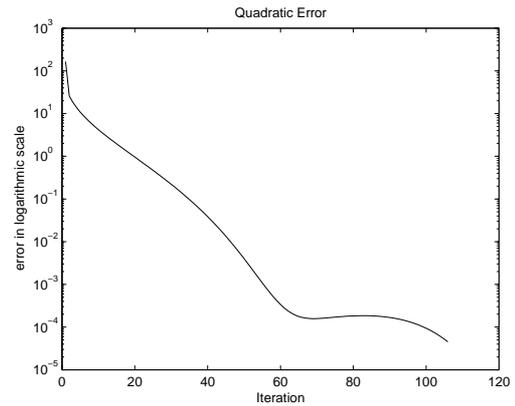Figure 4. Temporal evolution of the quadratic error for the 6-6 configuration

Figure 5. Logarithmic representation of the quadratic error for the 6-6 robot

The proposed method shows a fast convergence to a solution. Figure 6 shows a comparison between the proposed and the classical steepest descent methods in a single experiment. Step adjustment exhibits a faster decrease of the quadratic error.

The initial convergence of both methods is comparable. However the constant update step steepest descent requires around 5000 iterations to reach a quadratic error of 1.2×10^{-5} whereas the adaptive method requires some 120 iterations.
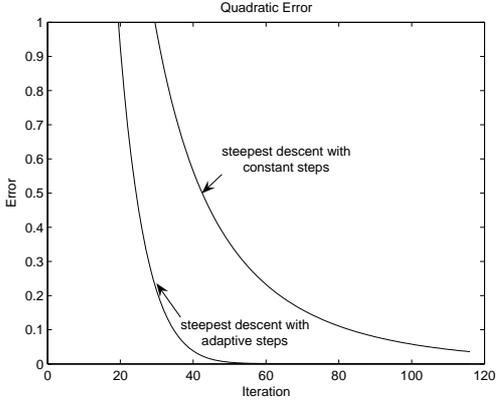

Figure 6. Temporal evolution of the quadratic error of steepest descent experiments with and without adaptive update steps

The solution obtained corresponds to having the robot in the upward position (see Figure 3). The 6-6 configuration has however multiple solutions. For instance, starting from the position illustrated in Figure 3 it is possible to maintain the same actuator lengths just by lowering the P body while applying the proper rotation. Figure 7 displays the evolution of the quadratic error when the initial configuration of the P body is rotated around the z axis.
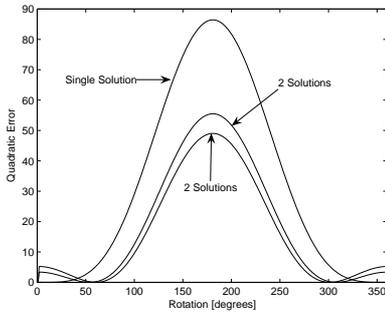

Figure 7. Multiple solutions in the 6-6 configuration

The solution in Table 2 is denoted solution 1 and corresponds to the 0° rotation. For the lowered P body, each $z < 1$ coordinate generates two different solutions differing by a rotation except in multiples of π rad. This is clearly visible in the two minima exhibited by the two curves indicated in the plot. If the initial condition of the P body is rotated the algorithm will tend to converge to one of the two solutions shown.

Table 3 shows the initial conditions for the second run of the simulation with the 6-6 robot. The P body is rotated around the z axis and the actuator values are kept at $l_i = 1, \; i = 1, \ldots, 6$. Figure 10 shows the final pose of the robot.

Act.	l_i	$\mathbf{p}_i(x, y, z)$	$\mathbf{b}_i(x, y, z)$	$\mathbf{r}_i(x, y, z)$
1	2.0107	(0.5848 0.8764 1.0000)	(0.5000 -0.8660 0.0000)	(0.0422 0.8665 0.4973)
2	2.0075	(-0.5150 0.8572 1.0000)	(1.0000 0.0000 0.0000)	(-0.7547 0.4270 0.4981)
3	2.0075	(-0.9998 -0.0175 1.0000)	(0.5000 0.8660 0.0000)	(-0.7471 -0.4401 0.4981)
4	2.0075	(-0.4848 -0.8746 1.0000)	(-0.5000 0.8660 0.0000)	(0.0076 -0.8671 0.4981)
5	2.0075	(0.5150 -0.8572 1.0000)	(-1.0000 0.0000 0.0000)	(0.7547 -0.4270 0.4981)
6	2.0075	(0.9998 0.0175 1.0000)	(-0.5000 -0.8660 0.0000)	(0.7471 0.4401 0.4981)

Table 3. Experiment 1b: Initial conditions for the 6-6 robot

In this case, the simulation was stopped after 435 iterations, with a final quadratic error of 4.8×10^{-5}. Table 4 displays the lengths of the actuators rods, along with the corresponding positions of the connecting points.

Act.	l_i	$\mathbf{p}_i(x, y, z)$	$\mathbf{b}_i(x, y, z)$	$\mathbf{r}_i(x, y, z)$
1	0.9978	(0.9958 -0.1143 0.4297)	(0.5000 -0.8660 0.0000)	(0.4969 0.7534 0.4306)
2	0.9995	(0.5961 0.8029 0.4372)	(1.0000 0.0000 0.0000)	(-0.4041 0.8033 0.4375)
3	1.0003	(-0.3963 0.9209 0.4407)	(0.5000 0.8660 0.0000)	(-0.8960 0.0548 0.4406)
4	0.9974	(-0.9875 0.1141 0.4380)	(-0.5000 0.8660 0.0000)	(-0.4888 -0.7539 0.4391)
5	0.9999	(-0.5902 -0.8036 0.4312)	(-1.0000 0.0000 0.0000)	(0.4099 -0.8037 0.4313)
6	0.9998	(0.4029 -0.9193 0.4261)	(-0.5000 -0.8660 0.0000)	(0.9031 -0.0533 0.4262)

Table 4. Experiment 1b: Final results for the 6-6 robot

Figure 8 shows the behavior of the quadratic error for this second run with the 6-6 robot. The small peak around iteration 120 is due to a large variation in the direction of cost gradient.

Figure 9 illustrates the quadratic error evolution presented in Figure 8 using a vertical logarithmic scale. In this experiment, the quadratic error evolution presents some noise because the algorithm uses a adaptive steps technic.
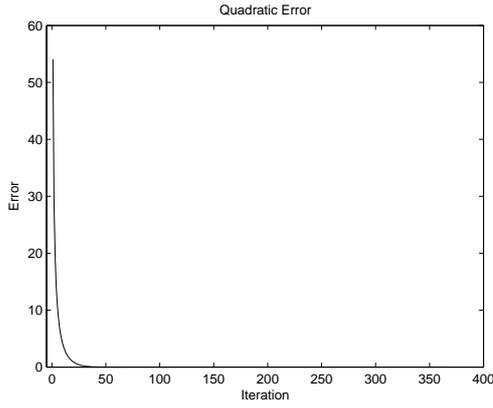
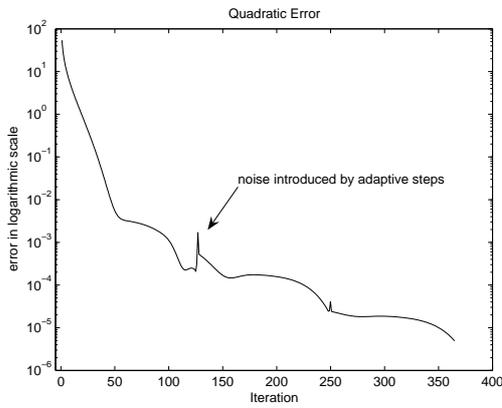Figure 8. Temporal evolution of the quadratic error the for 6-6 robot

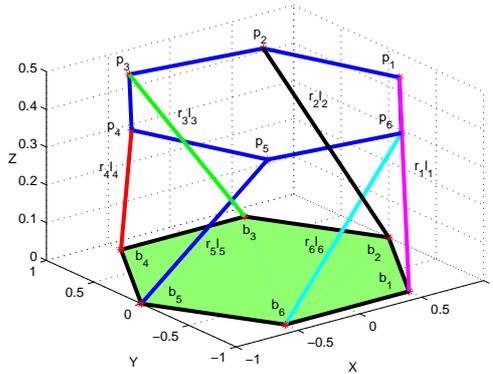Figure 9. Logarithmic representation of the quadratic error for the 6-6 robot

Figure 10. Alternative solution for the 6-6 robot

3.2 The 6-3/\³ robot

The second experiment presented considers a 6-3 /\³ configuration, in which the actuators are pairwise connected to just 3 different points in the P body.

Figure 11 shows the solution obtained.

Table 5 presents the initial conditions. For this experiment, the desired actuator lengths were also set to $l_{1,...,6} = 1$.
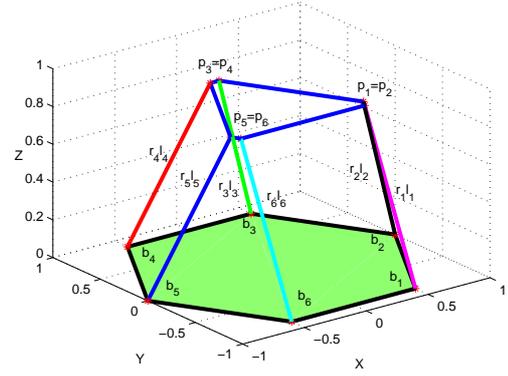

Figure 11. 3D view of 6-3/\³ robot

Act.	l_i	$\mathbf{p}_i(x,y,z)$	$\mathbf{b}_i(x,y,z)$	$\mathbf{r}_i(x,y,z)$
1	2.5495	(0.7500 -0.4330 2.5000)	(0.5000 -0.8660 0.0000)	(0.0981 0.1698 0.9806)
2	2.5495	0.7500 -0.4330 2.5000)	(1.0000 0.0000 0.0000)	(-0.0981 -0.1698 0.9806)
3	2.5495	(0.0000 0.8660 2.5000)	(0.5000 0.8660 0.0000)	(-0.1961 0.0000 0.9806)
4	2.5495	(0.0000 0.8660 2.5000)	(-0.5000 0.8660 0.0000)	(0.1961 0.0000 0.9806)
5	2.5495	(-0.7500 -0.4330 2.5000)	(-1.0000 0.0000 0.0000)	(0.0981 -0.1698 0.9806)
6	2.5495	(-0.7500 -0.4330 2.5000)	(-0.5000 -0.8660 0.0000)	(-0.0981 0.1698 0.9806)

Table 5. Experiment 2 - Initial conditions for the 6-3/\³ robot

The presented algorithm was stopped after 685 iterations, with a final quadratic error of 1×10^{-5}. The final numerical results are shown in Table 6.

The quadratic error temporal behavior is presented in Figure 12. Similarly to the first experiment, the initial evolution of the error decays rapidly.

Figure 13 shows the temporal evolution of the quadratic error presented in Figure 12 using a logarithmic vertical scale. In this experiment, the adaptive update steps technic also introduces some noise in the quadratic error evolution.

The general 6-3/\³ configuration has a more restricted workspace than the 6-6 configuration of the first experiment and the multiple solutions, in a maximum of 16, (Merlet, 2000), may not be easy to visualize. The procedure used with the 6-6 robot, rotating the P body around the z axis does not work with the 6-3 robot. Figure 14 shows the variation of the quadratic error with the rotation of the P body for different displacements along the z axis.

Act.	l_i	$\mathbf{p}_i(x,y,z)$	$\mathbf{b}_i(x,y,z)$	$\mathbf{r}_i(x,y,z)$
1	1.0006	(0.5003 -0.3012 0.8260)	(0.5000 -0.8660 0.0000)	(0.0003 0.5645 0.8255)
2	1.0006	0.5110 -0.2826 0.8260)	(1.0000 0.0000 0.0000)	(-0.4887 -0.2825 0.8255)
3	1.0006	(0.0107 0.5839 0.8260)	(0.5000 0.8660 0.0000)	(-0.4890 -0.2820 0.8255)
4	1.0006	(-0.0107 0.5838 0.8260)	(-0.5000 0.8660 0.0000)	(0.4890 -0.2820 0.8255)
5	1.0006	(-0.5110 -0.2827 0.8260)	(-1.0000 0.0000 0.0000)	(0.4887 -0.2825 0.8255)
6	1.0006	(-0.5003 -0.3012 0.8260)	(-0.5000 -0.8660 0.0000)	(-0.0003 0.5645 0.8255)

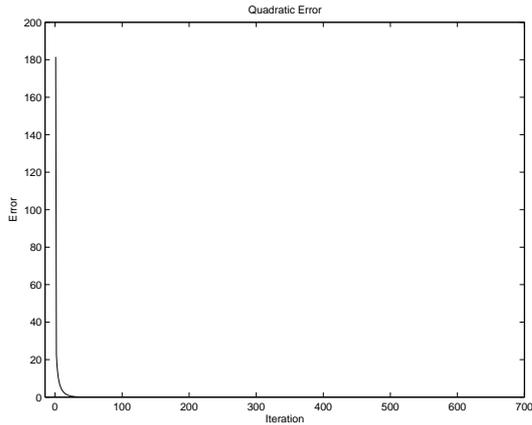Table 6. Experiment 2 - Final results for the 6-3/\$^3$ robot

Figure 12. Temporal evolution of the quadratic error for the 6-3/\$^3$ configuration

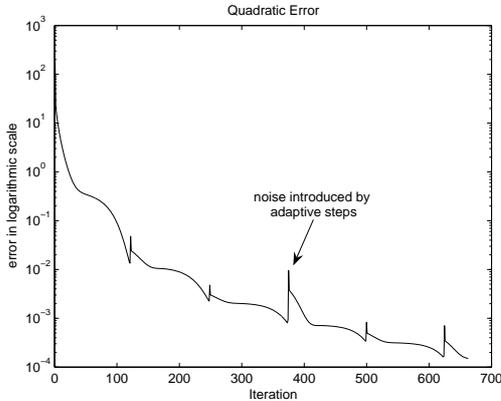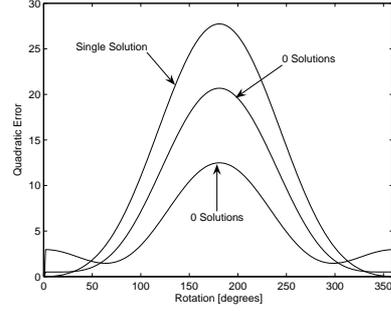Figure 13. Logarithmic representation of the quadratic error for the 6-3/\$^3$ configuration

Figure 14. Single solution in the 6-3/\$^3$ robot

The number of solutions of the direct kinematics optimization problem defined by (5) is bounded by the number of zeros of J_λ. The rigid nature of the P body indicates that any two solutions must differ at most by a rigid body motion transformation and hence the problem of finding the multiple solutions is transformed into the problem of finding the rigid body motion transforms in the 3D space that when applied to the $\mathbf{p}_i$ yields the minima of (5), that is the translations and rotations for the $\mathbf{p}_i$ that still preserve the rigidity constraints.

The 6-3 /\$^3$ configuration in this experiment considers that the $\mathbf{b}_i$ form a regular hexagon and the $\mathbf{p}_i$ form an equilateral triangle. The 16 possible solutions are easy to visualize when the l_i are identical. The symmetry of the robot relative to the plane formed by the B body indicates that 8 of these solutions are mirrors (relative to the plane formed by the $\mathbf{b}_i$) of the remaining 8.

The group of motion generators is composed by translations and rotations. The translations lead to 4 solutions, obtained by mirroring relative to the $\mathbf{b}_i$ plane and by flipping the triangle formed by the $\mathbf{p}_i$ around an horizontal axis. The remaining solutions can be obtained through rotations of P around an axis passing through each of the vertices of the triangle formed by the $\mathbf{p}_i$.

The method proposed in this paper will often find only 1 solution. The reason for this behavior lies in the fact that during the optimization process the $\mathbf{p}_i$ are computed using a gradient descent technique. If the initial condition is close to one solution then the method will naturally converge to this solution as the step update algorithm does not enforce the changes in the search direction defined by J_λ.

4. CONCLUSIONS

The paper presents a solution for the direct kinematics of two classes of Stewart-Gough robots supported on a steepest descent method using adaptive update steps.

The formulation of the direct kinematics as a constrained optimization problem, solved using standard techniques (Kuhn-Tucker conditions and a steepest descent method) was able to rapidly converge to a solution.

The mathematical proof of convergence of this method is based on the steepest descent method properties (see (Luenberger, 1973)).

Future work includes (i) the formal assessment of the method aiming at a real time control application, including the study of alternative strategies for the adjustment of the descent step and noise robustness issues, and (ii) the formal analysis of the motion between the symmetries for the 6-6 and 6-3 robots when using gradient descent methods.

REFERENCES

Almeida, L. B. and F. Silva (1991). Speeding-up backpropagation by data orthonormalization. *Artificial Neural Networks* **2**, 56–149.

Dietmaier, P. (1998). The stewart-gough platform of general geometry can have 40 real postures. *Advances in Robot Kinematics: Analysis and Control* pp. 1–10.

Hopkins, Brian R., Williams II. Robert L. (2002). Modified 6-psu platform. *Industrial Robot: An International Journal* **29**(5), 443–451.

Jakobović, Domagoj and Leo Budin (2002). Forward kinematics of a stewart parallel mechanism. pp. 149–154. Opatija, May 26-28.

Khalil, W. and Guegan S. (2004). Inverse and direct dynamic modeling of gough-stewart robots. *IEEE Trans. on Robotics* **20**(4), 754–762.

Luenberger, D. G. (1973). *Introduction to Linear and Nonlinear Programming*. Addison-Wesley.

Merlet, Jean-Pierre (2000). *Parallel Robots*. Kluwer Academic.

Parikh, P. J. and S. S. Y. Lam (2005). A hybrid strategy to solve the forward kinematics problem in parallel manipulators. *IEEE Transactions on Robotics* **21**(1), 18–25.