GEOMETRICAL ERRORS OF PARALLEL ROBOTS

Szabolcs SZATMÁRI

Department of Manufacture Engineering
Technical University of Budapest,
H–1521 Budapest, Hungary
Tel.: +36-1-463-2517, Fax: +36-1-463-3176,
E-mail: szsz@manuf.bme.hu

Received: June 5, 1999

Abstract

The goal of this paper is to determine and make a graphical representation of the geometrical errors occurred when commanding a parallel manipulator. It is also proposed to establish a relation between errors and position of the robot in the workspace and to create an error map. There is a complex knowledge available about the drive system and precision in the joints. An imposed position of the tool centre point (TCP) is compared with the real value which is a result of solving the inverse geometrical problem, rounding the length of the legs, and then the direct problem for the new values. Certainly, there are a lot of other errors (for example: imprecision of the joints, elastic deformation of the structure, etc.). These errors will form subject of other papers.

Keywords: hexapod, error map, resolution.

1. Introduction

Parallel manipulators have been under increasing development over the last few years from a technical viewpoint as well as for technical applications [3]. By definition a parallel manipulator is a closed-loop mechanism in which the end-effector is connected to the base by at least two independent kinematics chains [8].

Although the excellent load/weight ratio may be useful, parallel link mechanisms also present other interesting features [12]. A parallel manipulator was first used in a robotics assembly cell by McCallion in 1979 mostly because the position of the end-effector of a parallel manipulator is much less sensitive to the error on the articular sensors than for serial link robots [14]. Furthermore, their high stiffness insures that the deformations of the links will be minimal and this feature greatly contributes to the high positioning accuracy of the manipulator [14].

Another important feature of parallel manipulators is the possibility of using them as a 6-component force-sensor. Indeed, it can be shown that the measurement of the traction-compression stress in the links enables to calculate the forces and torques acting on the mobile platform.

That kind of manipulators is introducing new opportunities in industrial applications which involves high precision and high stiffness [13]. Nevertheless, for some processes when an extra precision is required, we have to take the errors occurred at the-end effector's position into consideration [4].

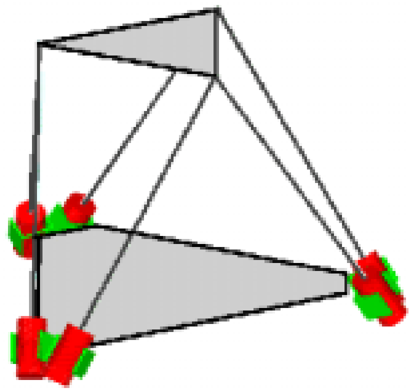

Fig. 1. Six-leg robot

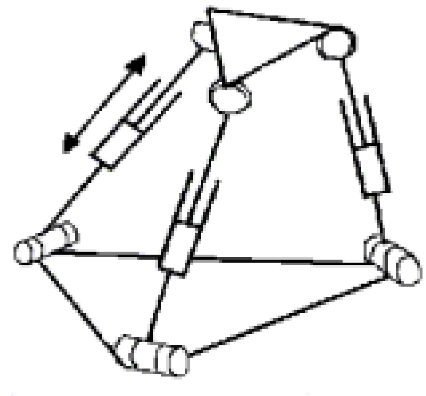

Fig. 2. Three-leg robot

2. Design of Parallel Robots

Many different designs of parallel manipulators are possible and the scientific literature on this topic is very rich (see *Fig. 1* and *2*) [13]. All have in common their low cost since most of the components are standard although the assembly of the manipulator has to be done with care. The design is important as some features may be upgraded by an appropriate choice.

In contrast to common serial link mechanisms with three intersecting wrist joint axes the workspace of a parallel manipulator cannot be decoupled in two 3D workspaces characterising the possible translation and orientation motions [9]. Therefore the workspace is completely imbedded, and there is no human readable way to represent it. However, some projections of the full workspace can be drawn. For example, it is usual to represent the possible translations of the robot in a plane for a fixed orientation and altitude of the mobile platform, either by using a discretization procedure or, more efficiently, a geometrical algorithm which can take into account the limited range of the actuators, the mechanical limits of the passive joints and links interference. It is also possible to assume that one point of the mobile plate is fixed and the possible rotations of the mobile plate around this point can be illustrated.

As the architecture of parallel manipulators is very different from the one used for serial-link manipulators, most of the theoretical problems have to be reconsidered. In fact, there is a strange duality between parallel link mechanisms and serial link mechanisms: a difficult (simple) problem for one kind is easily solved (with difficulty) for the other kind [2]. This duality has yet to be explained satisfactorily, although some attempts have already been made [10].

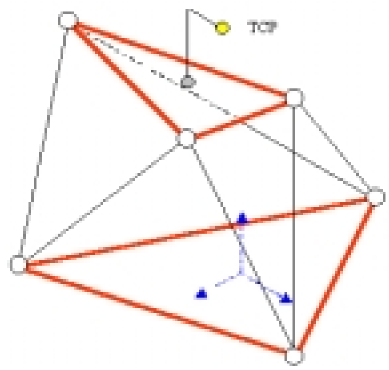

Fig. 3. TCP attached to the robot

3. Kinematics

Two problems can be distinguished for the kinematics aspects: inverse kinematics and direct kinematics. The inverse kinematics problem, i.e. finding the link lengths for a given posture of the mobile platform (a difficult problem for serial-link mechanisms [7]) is straightforward for parallel manipulators [6]. Thus, their control is usually very simple [5]. On the other hand, the direct kinematics problem is much more difficult. In general, this problem has more than one solution.

4

Let us consider the mechanism shown in *Fig. 1*. The upper plate (end- effector) is connected to the base through 6 articulated links. Linear actuators allow to change the link length which in turn permit to control the position and orientation of the upper-plate. At the extremities of the links we find a universal joint (with centre) and a ball-and-socket joint (with centre). A reference frame is attached to the base and a mobile frame is attached to the moving platform. The posture of the moving platform is defined by the co-ordinates in the reference frame together with the rotation matrix defining the rotation between the reference frame and the mobile frame [3].

4. Creating an Error Map

We will force the TCP to go through a net of pre-defined points situated in a horizontal plain at given value of Z [4]. *Fig. 4* shows the flowchart of the process.

The inverse kinematics of the robot supposes the distance between the corresponding points of the upper and the lower platform [6].

For each step the lengths of the legs (joints) are computed. That is rounded because the length can take discrete values. Using the direct kinematics we are computing back the position of the TCP. This time the new values of the leg's

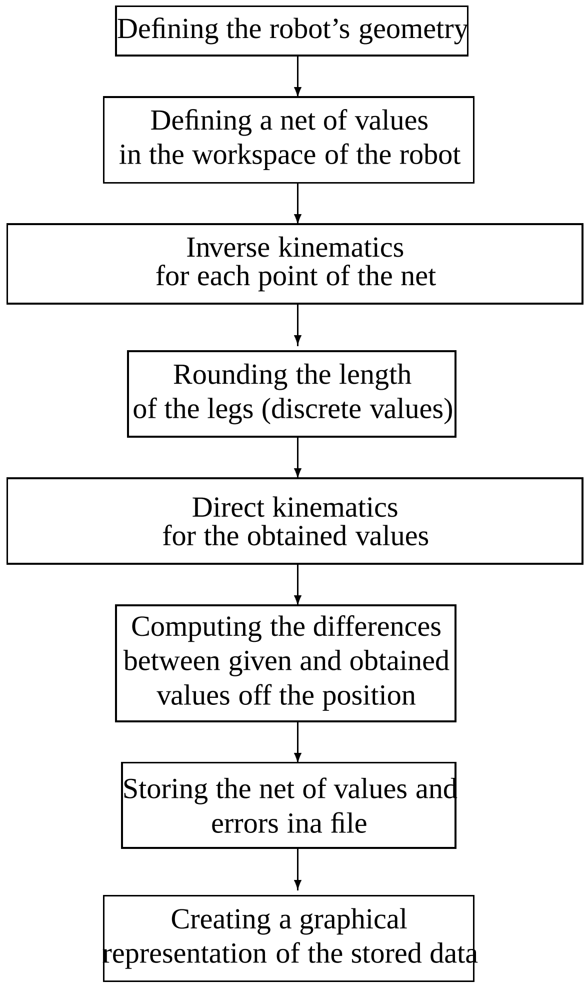

Fig. 4. Flowchart of the process

lengths are known. We are taking into consideration that an edge of the working platform is situated on a sphere with a radius given by the length of the leg. Distances between edges are constrained by a corresponding side of the triangle.

The error is defined as a difference between given and obtained values of the TCP position. The values are stored and prepared for a graphical representation.

The representation is made by another MathCad program and it uses the stored values.

A simple MathConnex graphical program gives the net of values. For each step of values the MathCad is used to solve the complex problem (inverse and direct

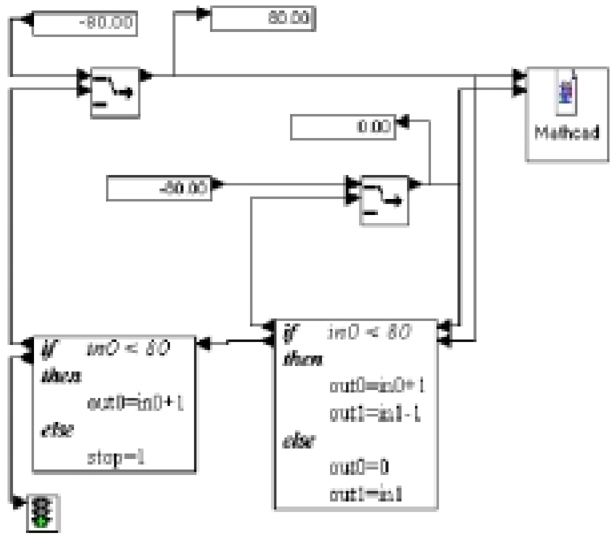

Fig. 5. Using MathConnex to define the net of values in the workspace

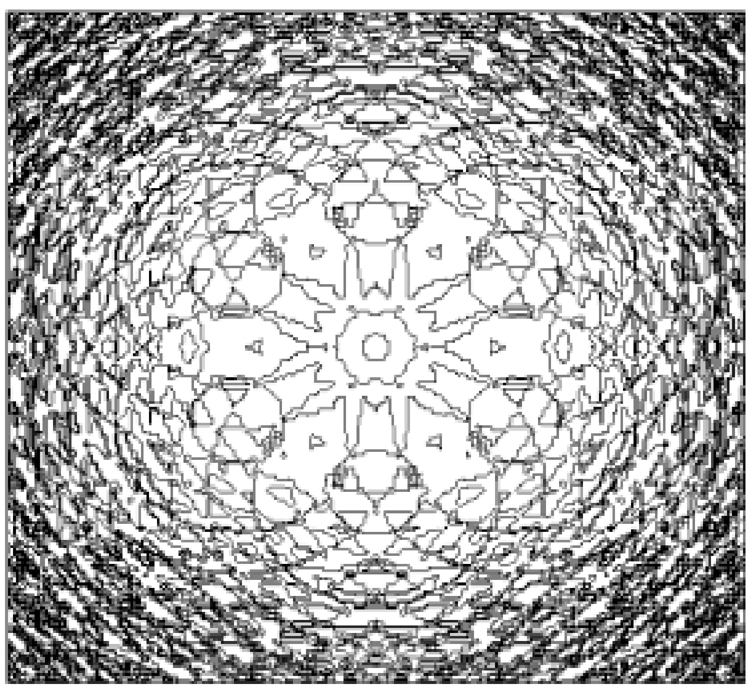

Fig. 6. Error map of a hexapod robot (errors along Z axis)

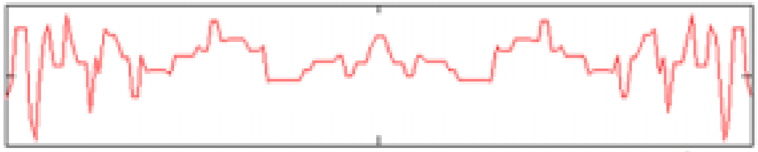

Fig. 7. A slice of the error map

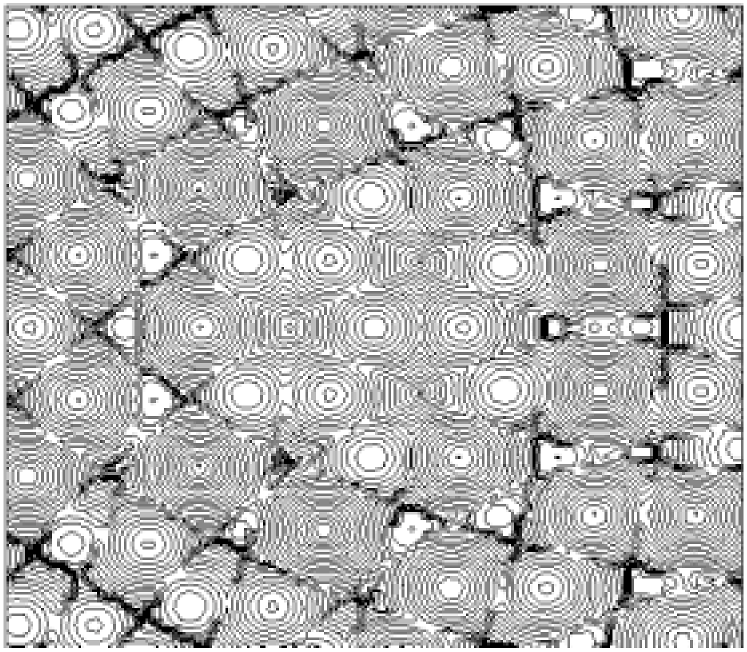

Fig. 8. Error map of a hexapod robot (absolute errors)

kinematics). In *Fig. 5* we can see the MathConnex program.

Watching the error map presented in *Fig. 6* we can observe that the errors along Z axes have low values around the centre of the analysed area. That means the robot's precision is better around this area, where the robot has the nest (origin point). Going further in the direction of the map's border the density of the contour lines demonstrates that the errors are much more around there. That can be seen as well in a slice of the error map which is going through the nest position of the robot. The errors are the differences between given and obtained values of the TCP position. They can be positive or negative depending on the analysed point's relative position to the nearest point of the pre-defined net of values. These errors give the resolution of the analysed robot: better resolution for less errors.

For a clear vision of the errors the error map shows a more larger area than a real robot can reach. The step of the leg's movement is also exaggerated. For

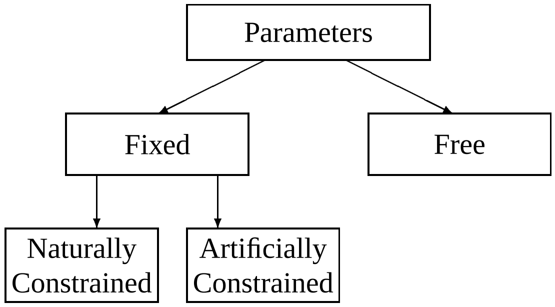

Fig. 9. Parameters of the error map

a real robot we have to determine not only one error map but more of them taking more parameters into consideration. The resolution of the robot depends on the following factors:

- position in the workspace
- orientation in those points
- step of the leg's movement
- size of the platforms

Due to the fact that we cannot represent all of these parameters in a single 3D diagram we have to make constraints to eliminate some of them. We may fix, for example, the orientation, the movement along Z axis. Other factors can be considered to be constructive constants of the robot (the platform's size and shape, step in the joints movement). In this way, the parameters can be naturally fixed (constrained), artificially constrained and free (see *Fig. 9*). So, we have to represent the error map occurred when changing just those free parameters. In *Fig. 8* we can see another error map [4]. That map represents the absolute error (quadratic sum of the errors along the X, Y and Z axes). It is useful when the sign of the errors separately along the axes does not matter. In some cases it is not allowed to work with this map because we must know exactly mostly the Z error. That is the case when the robot is used as a milling machine, for example.

5. Conclusions

We had realised a proper error map, that can be stored in the database of the command software and for each accessed point we could realise a tool correction that allows much more precision of the manufacturing process. To realise an error map, we have to know exactly what kind of purpose the robot has, what the parameters are (for example, degrees of freedom) that can be eventually neglected (by this point of view).

6. Research Trends and Further Problems

Some other problems are expected to be realised referring to the error map of the parallel manipulators. For example, we could watch how these errors are changing taking into consideration the temperature of the legs one by one and the stiffness of the full structure.

References

[1] SOMLÓ, J. – LANTOS, B. – CAT, P.T.: Advanced Robot Control, Budapest, Akadémiai Kiadó 1997.

[2] SZATMÁRI, SZ.: Velocity and Acceleration Constrained Motion of Parallel Type Robots, *9. DAAAM Symposium (DAAAM'98)*, 1998, Cluj-Napoca, Romania, pp. 455-456.

[3] SZATMÁRI, SZ.: Bases of Programming Parallel Robots, *Robotics'98 National Symposium*, 1998, Brasov, Romania, pp. 309-312.

[4] SZATMÁRI, SZ.: Resolution Map of Multipod Robots, *MicroCAD '99*, 1999, Miskolc, Hungary.

[5] ZENTAY, P. – ZOLLER, Z.: Time Optimal Trajectory Planning for Robots in LabView Programming System, *MicroCAD '99*, 1999, Miskolc, Hungary

[6] SZATMÁRI, SZ.: Labview Based Control of Parallel Type Robots, *FMTU'99*, 1999, Cluj-Napoca, Romania, pp. 65–68.

[7] ZOLLER, Z. – ZENTAY, P.: Constant Kinetic Energy Robot Trajectory Planning, *Periodica Polytechnica*, 1999. (to be published).

[8] ARAI, T. – CLEARY, K. et al.: Design, Analysis and Construction of a Prototype Parallel Link Manipulator. *IEEE Int. Workshop on Intelligent Robots and Systems (IROS)*, **1**, pp. 205–212, Ibaraki, Japan, 3–6 July 1990.

[9] ARAI, T. et al.: Development of a Parallel Link Manipulator. *ICAR*, p. 839–844, Pise, 19–22 June, 1991.

[10] DO, W. Q. D.– YANG, D. C. H.: Inverse Dynamic Analysis and Simulation of a Platform Type of Robot. *J. of Robotic Systems*, **1**, 5(3), pp. 209–227, May, 1988.

[11] FICHTER, E. F.: A Stewart Platform Based Manipulator: General Theory and Practical Construction. *Int. J. of Robotics Research*, **1**, 5(2), pp. 157–181, December 1986.

[12] INOUE, H. – TSUSAKA, Y. – FUKUIZUMI, T.: Parallel Manipulator. *Proc. 3rd ISRR* pp. 321–327, Gouvieux, France, 7–11, Octobre 1985.

[13] KOHLI, D. – LEE, S-H – TSAI, K-Y – SÁNDOR, G. N.: Manipulator Configurations Based on Rotary-Linear (R-L) Actuators and Their Direct and Inverse Kinematics. *J. of Mechanisms, Transmissions and Automation in Design*, **110**, pp. 397–404, December 1988.

[14] WANG, J. – MASORY, O.: On the Accuracy of a Stewart Platform-part I: The Effect of Manufacturing Tolerances. *IEEE Int. Conf. on Robotics and Automation*, pp. 114–120, Atlanta, 2–6 May 1993.