A PARALLEL ROBOT WITH THREE TRANSLATIONAL DEGREES OF FREEDOM FOR MACHINING OPERATIONS

CRUZ, Daniel Fontan Maia da[1]; LEPIKSON, Herman Augusto[1]; WEINGAERTNER, Walter Lindolfo[2]

[1]Mechatronics Graduate Program, *Universidade Federal da Bahia* - Brazil
[2]Precision Engineering Laboratory (LMP), *Universidade Federal de Santa Catarina* - Brazil

ABSTRACT

This paper proposes a new machine tool structure design for machining based on a Delta Robot architecture as object of study, known as Tsai's Manipulator, patented by Tsai in 1997. The new architecture employs only rotary joints instead of spherical ones as it is usual for Delta Robots. The rotary joints allow its mount under pretension to eliminate clearances or backlash, avoiding the use of expensive spherical joints. All the other mechanical parts are standard components, which makes the solution attractive in terms of cost. A simplified inverse kinematic solution over the one proposed by Tsai is presented, based on a practical approach. The simplified solution reduces the number of steps to solve the inverse kinematics without any loss of performance. In addition, a solution to deal with the coupled axis without parasitic motion is presented. To increase the accuracy and the stiffness of the robot, a special attention was given to the rotary joints using preload rotational ball bearing joints. The structure parts were manufactured mostly by laser cutting with almost no complementary machining processes. In order to evaluate the proposed solutions, a prototype was built and a dedicated control software was developed for this particular robot. Workpieces were milled with the robot to demonstrate its capability and the advantages regarding the proposed machine architecture.

Index Terms - *Machining Robot, Parallel kinematics, Tsai Manipulator.*

1. INTRODUCTION

The market of industrial robots is traditionally related to handling, pick and place, assembling, painting, dispensing and welding operations [1]. The use of robots for machining operations is considered non-conventional, and represents only a small part of the market share. However, according to the report "Industrial Robotics Market" [2], the segment is expected to grow at a significant pace in the near future with, for example, applications involving the pre-machining of hard material parts at low tolerances performed by robots. The robotic machining can be an alternative to computer numerical control (CNC) machining for non-metallic materials and metals, depending on material hardness, required surface finishing and part complexity [3]. Many research works have been developed in this field [3, 4, 5, 6, 7, 8]. Klimchik [4] evaluated in his work several industrial serial robots performing machining operations, and pointed out the dificulties related to the low stiffness and the cumulative errors on the end-effector. Parallel robot is considered much more suitable to perform machining operations due to the advantages of parallel kinematic mechanisms (PKMs) over serial ones. The advantage list includes compact structure, low moving inertia, high load to weight ratio, high dynamic performance, high accuracy potential, and suitability for the manufacture of complex and typical curved parts or free form surfaces [6, 7].

A historical and great review about the origin of PKMs was written by Ilian Bonev [9], clarifying the contribution of the pioneers Pollard [10], Gough [11], Stewart [12] and Cappel [13]. For nearly two decades, parallel robots attracted very little attention, but, in the early 1980s, their popularity suddenly started to grow and has never stopped [14]. Since then, various novel designs have been proposed, and almost as many applications filed [15]. Many of these parallel architectures were truly innovative, but there was one that became by far the most successful parallel robot for industrial application: the Delta Robot [16]. The Delta Robot is a spatial translational 3-DOFs parallel-kinematic robot. This mechanism can provide pure 3D translational motion to the moving platform. It has been marketed and used industrially for pick and place applications. A sketch of this mechanism is shown in Figure 1. Many variations of the Delta mechanism have been proposed and implemented. One of these is the patented Tsai's manipulator [17], which provides pure 3D translational motion to its moving platform (Figure 2).
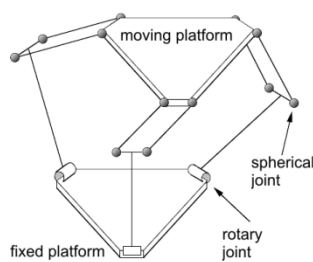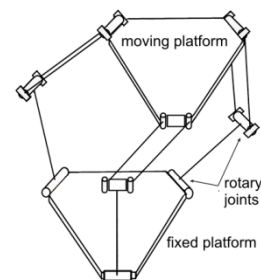

Figure 1 – Delta Manipulator

Figure 2 - Tsai's Manipulator

Several theoretical works investigated the Tsai's Manipulator [18, 19, 20, 21]. Additional to the theoretical approach, Stamper [18] and Hodgins [21] designed and constructed a prototype to proof the feasibility of the concept. See Figure 3 and 4.
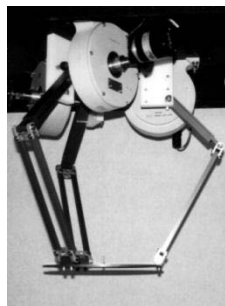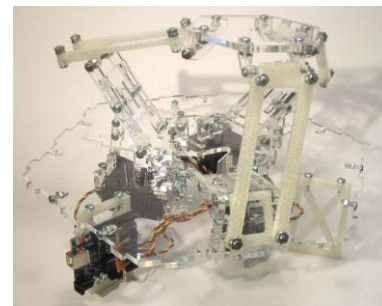

Figure 3 - Stamper/Tsai's Prototype [18]

Figure 4 - Hodgins's Prototype [21]

The main advantage of Tsai's Manipulator is due to the use of rotary joints instead of spherical ones, as it is usual for Delta Robots. The rotary joints are mounted under preload to eliminate clearances or backlash, dispensing the use of expensive spherical joints. All the other parts are standard components, which makes the solution attractive in terms of cost and effectiveness. Tsai's patent has expired and it is time to give more attention to this particular robot and explore all of its potential.

The focus of this work is to develop a new robot design with sufficient rigidity to perform machining operations. A simplified inverse kinematic solution is presented, based on a practical approach. In addition, a solution to deal with the coupled axis based on an interpolation method is presented. In order to evaluate the solutions, a prototype was built, together with a dedicated software to control the robot. 3D workpieces were milled using the robot to demonstrate its potential. The robot prototype structure was built using mainly laser cut steel plates, and using preloaded rotational ball bearing joints.

2. INVERSE KINEMATIC

A schematic representation of the manipulator is shown in Figure 5, in which the reference frame (XYZ) is attached to the fixed platform at the point O, located at the center of the fixed platform. Three identical limbs connect the moving platform to the fixed platform. Each limb consists of an input link and an upper arm. Each upper arm is a planar four-bar parallelogram. All links and platforms are considered rigid bodies.
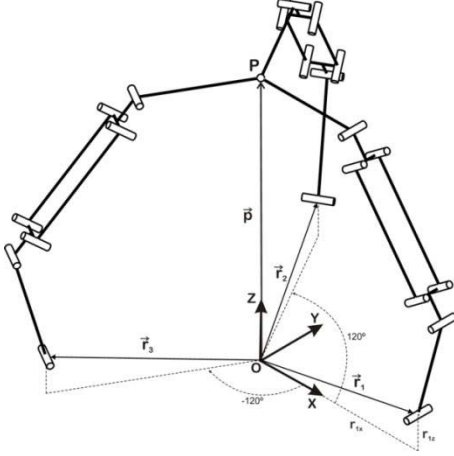

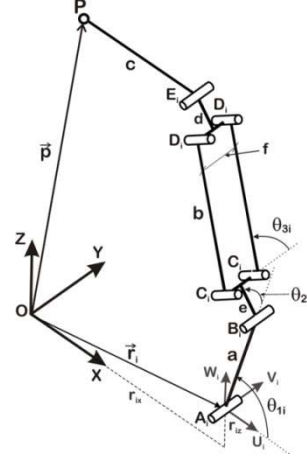Figure 5 – Schematic of the three-DOF manipulator

Figure 6 – Joint angles and link lengths for leg i.

The ith leg of the manipulator is show in Figure 6. The vector $\vec{p}$ is the position vector of point P in the (XYZ) coordinate frame, where P is attached at the center of the moving platform. The actuated joint of leg-1 is the A_1 joint, where the vector $\vec{r_1} = [r_{1x} \quad 0 \quad r_{1z}]$ represents its position in the (XYZ) coordinate frame. Another coordinate system $(U_1 V_1 W_1)$, with the same orientation of (XYZ), is attached to the fixed platform at A_1 joint, such that the u_1-axis is perpendicular to the axis of rotation of the joint at A_1 and the v_1-axis is along the joint axis of A_1. The actuated angle of leg-1 θ_{11} is measured between U_1 and $\overline{AB}$ (Figure 6).

According to Figure-6, the vector $\vec{R}$ between the origins of $(U_1 V_1 W_1)$ coordinate system and E_1 joint can be calculated as follows:

$$\vec{R} = -\vec{r_1} + \vec{p} + c\vec{k} \tag{1}$$

where $\vec{r_1}$ is the vector from O to the joint A_1, $\vec{p}$ is the desired position of the moving platform, c is the length between P and E, and $\vec{k}$ is the unit vector along $\overline{PE}$ (see details in Figure 7). Expressions for R_{u1}, R_{v1} and R_{w1} are given by:

$$R_{u1} = a\cos(\theta_{11}) + [e + d + b\sin(\theta_{31})]\cos(\theta_{21}) \tag{2}$$

$$R_{v1} = b\cos(\theta_{31}) \tag{3}$$

$$R_{w1} = a\sin(\theta_{11}) + [e + d + b\sin(\theta_{31})]\sin(\theta_{21}) \tag{4}$$

Two solutions for θ_{31} calculated from Eq. (3) are given by:

$$\theta_{31} = \pm\arccos\left(\frac{R_{v1}}{b}\right) \tag{5}$$

The lateral displacement of the four-bar mechanism (Figure 8) represents exactly the p_y component for the leg-1, and it can be written as:

$$p_y = R_{v1} = b\cos(\theta_{31}) \tag{6}$$

Based on Figure 8 and taking leg-1 as example, the following statements can be made: for $p_y = 0$, $\theta_{31} = \pi/2$; for $p_y > 0$, $\theta_{31} < \pi/2$; and for $p_y < 0$, $\theta_{31} > \pi/2$. The two solutions from Eq. (5) only represent this situation. In fact, the value of θ_{31} is not necessary to solve the problem. In practical, the lateral displacement of the four-bar mechanism will affect only the length L, which can be expressed as:

$$L = e + d + \sqrt{b^2 - (p_y)^2} \tag{7}$$

where the signal of p_y does not matter in Eq. (7). With L known, the problem will be reduced to a solution of a two-link planar arm [22].
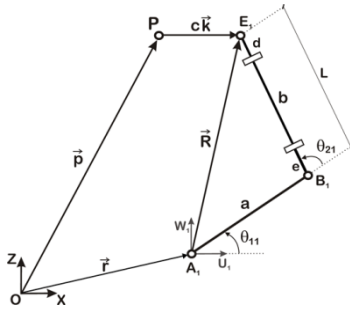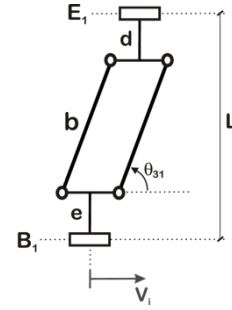

Figure 7 – Lateral view for leg 1. Figure 8 – Front view of the upper arm link for leg 1.

The application of the cosine theorem to the triangle formed by the segments $\boldsymbol{a}$, $\boldsymbol{L}$ and $\boldsymbol{R}$ at U_1W_1 plane (Figure 7), gives

$$R_{u1}{}^2 + R_{w1}{}^2 = a^2 + L^2 + 2aL\cos(\pi - \theta_{21}) \tag{8}$$

The two admissible configurations of the triangle are shown in Figure 9. Observing that $\cos(\pi - \theta_{21}) = -\cos(\theta_{21})$ leads to

$$\theta_{21} = \pm\arccos\left(\frac{(R_{u1})^2 + (R_{w1})^2 - a^2 - L^2}{2aL}\right) \tag{9}$$

The elbow-right posture is obtained for $\theta_{21} \in (0, \pi)$ while the elbow-left posture is obtained for $\theta_{21} \in (-\pi, 0)$, see Figure 9. However, the conception of the structural links of the manipulator restricts $\theta_{21} \in (0, \pi)$, and from that the negative result from Eq. (9) can be discarded.
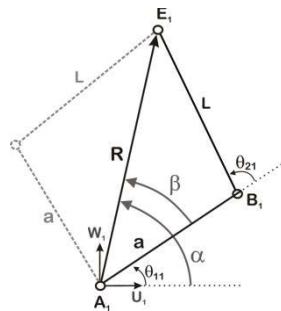

Figure 9 – Admissible postures for the two links.

To find θ_{11}, one can consider the angles α and β in Figure 9. Since the determination of α depends on the sign of R_{u1} and R_{w1}, α shall be computed by:

$$\alpha = Atan2(R_{w1}, R_{u1}) \tag{10}$$

To compute β, the application of the cosine theorem yields

$$\cos\beta \sqrt{(R_{u1})^2 + (R_{w1})^2} = a + L\cos(\theta_{21}) \tag{11}$$

and using the expression of $\cos(\theta_{21})$ given in Eq. (9) leads to

$$\beta = arcos\left(\frac{(R_{u1})^2 + (R_{w1})^2 + a^2 - L^2}{2a\sqrt{(R_{u1})^2 + (R_{w1})^2}}\right) \tag{12}$$

with $\beta \in (0, \pi)$, and $\theta_{21} > 0$, the θ_{11} can be computed as

$$\theta_{11} = \alpha - \beta \tag{13}$$

Table-1 summarizes the steps to solve the inverse kinematic of leg-1.

Table 1- Steps to solve the inverse kinematics of leg-1

Step	Procedure	Equation
1	Get the desired position (vector $\vec{p}$)	-
2	Compute $\vec{R}$	(1)
3	Compute L	(7)
4	Compute α	(10)
5	Compute β	(12)
6	Compute de θ_{11}	(13)

To solve the other two legs in an easy way, the same steps of table-1 can be used, since the $\vec{p}$ vector is reoriented. The rotation of $\vec{p}$ vector by an angle of -120° around z-axis gives

$$R_z(-120°)\vec{p} = \begin{bmatrix} \cos(-120°) & -\sin(-120°) & 0 \\ \sin(-120°) & \cos(-120°) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \vec{p}_{leg2} \tag{14}$$

Solving the inverse kinematic for $\vec{p}_{leg2}$ using the steps of leg-1 is equivalent to solve the inverse kinematics of leg-2. The result will be θ_{12}.

In the same way, applying a rotation on $\vec{p}$ vector by an angle of 120° around z-axis, gives

$$R_z(120°)\vec{p} = \begin{bmatrix} \cos(120°) & -\sin(120°) & 0 \\ \sin(120°) & \cos(120°) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \vec{p}_{leg3} \tag{15}$$

Solving the inverse kinematic for $\vec{p}_{leg3}$ using the steps of leg-1 is equivalent to solve the inverse kinematics of leg-3. The result will be θ_{13}.

For the cases that $r_y \neq 0$, two more steps must be previously performed. The first step is to compute the angle of rotation of leg-1 relative to z-axis, as follows:

$$\varphi = \arcsin\left(\frac{r_y}{r_x}\right) \tag{16}$$

The second step is to apply a rotation on $\vec{p}$ vector by an angle of φ around z-axis.

$$R_z(\varphi)\vec{p} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \vec{p}_{leg1} \tag{17}$$

Once we have computed $\vec{p}_{leg1}$, proceed with the steps of table-1. The vectors $\vec{p}_{leg2}$ and $\vec{p}_{leg3}$ are obtained rotating $\vec{p}_{leg1}$ around z-axis by an angle of -120º and 120º, respectively.

3. PATH INTERPOLATION

To perform machining operations, the robot should be able to follow trajectories in 3D space with a desired accuracy. To achieve this, two points are fundamental: the motors have to move in a synchronous way, without any delay with respect to one another, and it is necessary to use an interpolation method between the trajectory control points to guarantee the correct motion, without parasitic motion.

Since the axes of the robot are coupled, all the three arms have to be actuated simultaneously to perform a motion in only one direction inside the workspace (x-axis for example). The following example will demonstrate this situation.

It is desired that the robot performs a rectilinear motion inside the workspace (only x-axis). The moving platform has to go from initial point P_i to the final point P_f in 1 second (see Figure 10). Considering that there is a milling tool attached to the moving plataform, this motion defines a feed rate of 50 mm/s. For an actual machining operation, the 50 mm/s feed rate can be a lot, but for demonstration purposes, this will highlight the errors and the parasitic motion of the robot.
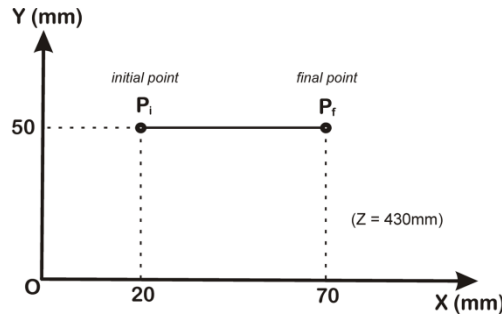

Figure 10 – Cartesian path.

In this example are going to be used the dimensions and characteristics of the actual manufactured robot, as follows:

$$r_x = 300.01mm \quad r_y = -5.04mm \quad r_z = 59.76$$

$$a = 180mm \quad b = 200mm \quad c = 175.044mm$$

$$e = d = 63.08mm \quad f = 43.8mm$$

For the desired positions on Figure 10, the inverse kinematics results in the following joint positions:

$$P_i = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 20 \\ 50 \\ 430 \end{bmatrix} \rightarrow \theta_{1i} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix} = \begin{bmatrix} 50.4249 \\ 46.5507 \\ 65.6397 \end{bmatrix}$$

$$P_f = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 70 \\ 50 \\ 430 \end{bmatrix} \rightarrow \theta_{1f} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix} = \begin{bmatrix} 40.1219 \\ 56.6527 \\ 73.3127 \end{bmatrix}$$

The average angular velocity can be computed as

$$\omega_{avg} = \frac{\Delta\theta}{\Delta t} = \begin{bmatrix} (40.1219 - 50.4249)/1 \\ (56.6527 - 46.5507)/1 \\ (73.3127 - 65.6397)/1 \end{bmatrix} = \begin{bmatrix} -10.303 \\ 10.102 \\ 7.673 \end{bmatrix} (°/s) \tag{18}$$

The motion controller uses the desired final position, maximum target velocity, and acceleration values to determine how much time it spends in the three primary move segments (which include acceleration, constant velocity, and deceleration).

For the acceleration segment of a typical trapezoidal profile, motion begins from a stopped position and follows a prescribed acceleration ramp until the speed reaches the target velocity for the move. Motion continues at the target velocity for a prescribed period until the controller determines that it is time to begin the deceleration segment and slows the motion to a stop exactly at the desired final position (Figure 11). If a move is short enough that the deceleration beginning point occurs before the acceleration has completed, the profile appears triangular instead of trapezoidal and the actual attained velocity may be smaller than the desired target velocity (Figure 12).
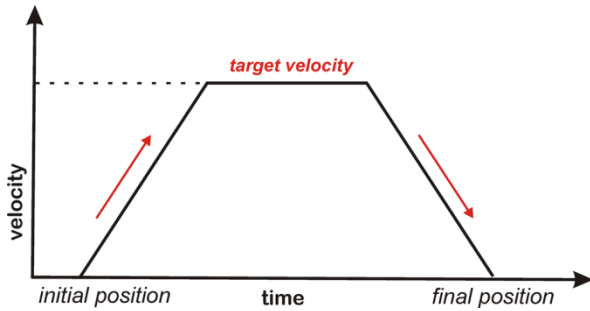

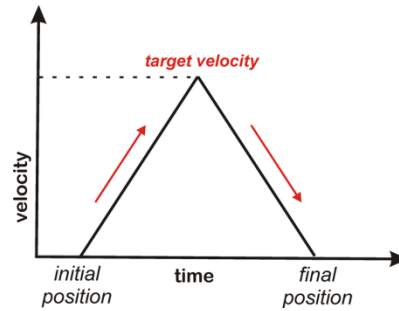
Figure 11 – A typical trapezoidal velocity profile.

Figure 12 – A typical triangular velocity profile.

The basic premise of a triangular move profile is to accelerate to a maximum speed, then immediately decelerate, with acceleration and deceleration being equal in terms of both time and distance. To reach the average speed computed in Eq. (18), the target velocity should be twice the average speed (Figure 13).
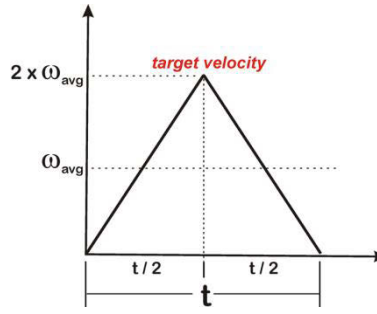
Figure 13 – Target velocity for triangular profile.

For a better understanding, three different situations will be considered to achieve the desired motion from P_i to P_f (Figure 10):

- First Situation   - one new joint position per second will be sent to the motion controller;
- Second Situation   - two new joint positions per second will be sent to the motion controller;
- Third Situation   - sixty new joint positions per second will be sent to the motion controller;

For each new joint position, each motion controller will execute a triangular profile, as showed in Figure 13. By applying this to the angular displacement of the three joints for the **first situation**, it is possible to plot the velocity and position graph during the time interval of the movement (Figures 14 and 15). The joint positions (Figure 15) are calculated by integrating the velocity curve over the time (Figure 14). In the interval of 1 second, 10 increments of 0.1 second were used.
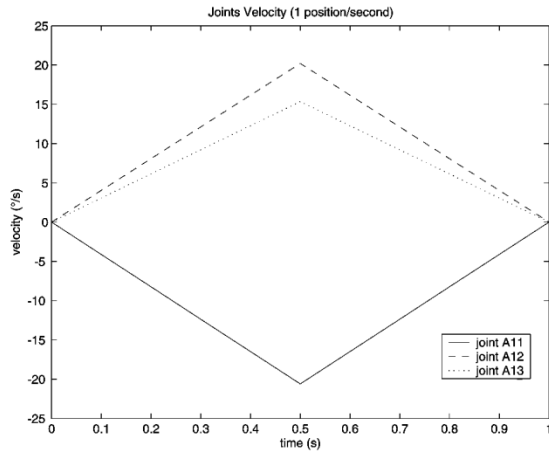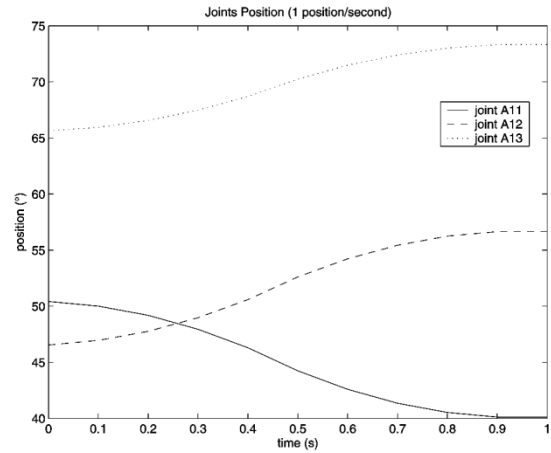

Figure 14 – Joints velocity (1 position/s).

Figure 15 – Joints position (1 position/s).

For the **second situation**, there will be an intermediate position P_m between points P_i and P_f (see Figure 10). Applying the inverse kinematics for this new point, gives

$$P_i = \begin{bmatrix} 20 \\ 50 \\ 430 \end{bmatrix}, \ \theta_{1i} = \begin{bmatrix} 50.4249 \\ 46.5507 \\ 65.6397 \end{bmatrix}, \ P_m = \begin{bmatrix} 45 \\ 50 \\ 430 \end{bmatrix}, \ \theta_{1m} = \begin{bmatrix} 45.0980 \\ 51.0937 \\ 69.0379 \end{bmatrix}, \ P_f = \begin{bmatrix} 70 \\ 50 \\ 430 \end{bmatrix}, \ \theta_{1f} = \begin{bmatrix} 40.1219 \\ 56.6527 \\ 73.3127 \end{bmatrix}$$

The velocity and position graphs of the joints for the second situation are shown in Figures 16 and 17.
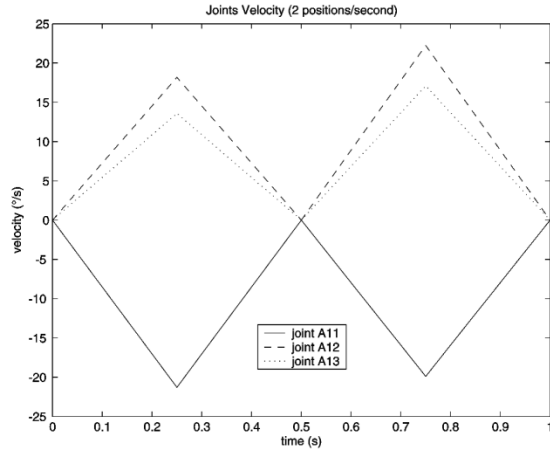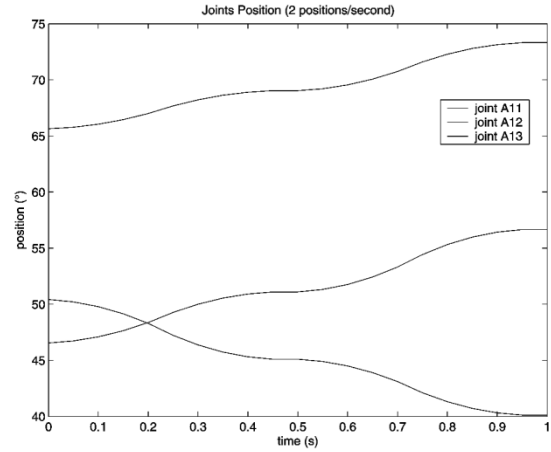

Figure 16 – Joints velocity (2 position/s).

Figure 17 – Joints position (2 position/s).

In both situations, the robot reaches the desired final position. What changes in the second situation is that at t = 0.5 seconds the robot reaches the exact position P_m. If more intermediate positions are sent during this time period, higher position accuracy can be achieved. The parasitic motion problem occurs during the joint motion transition from one position to another.

In this work an iterative algorithm that calculates the forward kinematics of the robot by means of a scan of positions in the workspace was implemented. Using the inverse kinematics solution and the iterative algorithm, it is possible to find the vector $\vec{p}$ from the desired joint positions of the three legs.

For each joint position on Figures 15 and 17, the equivalent position in the workspace was found. The results are shown in Figures 18 to 22.
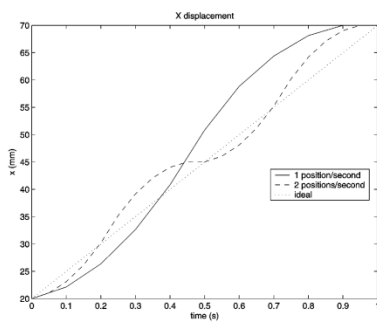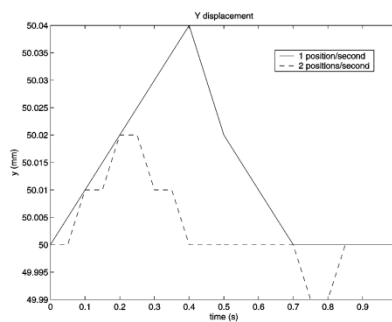

Figure 18 – X displacement.
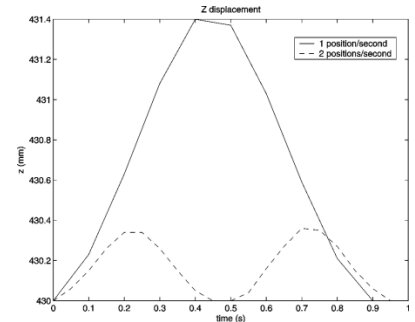

Figure 19 – Y displacement.
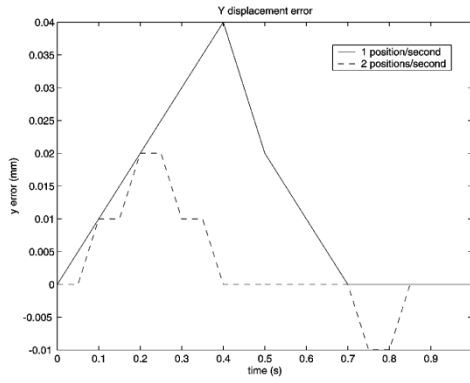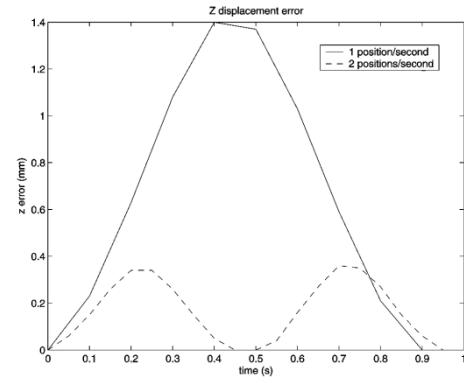

Figure 20 – Z displacement.

Figure 21 – Y error.

Figure 22 – Z error.

There is no displacement error on x-axis (Figure 18). The difference related to the ideal displacement represents the velocity error. On y-axis and z-axis (Figures 19 and 20) there are displacements errors. This undesired motion on y and z represents the parasitic motion. The magnitude of the error for the two situations can be seen in Figures 21 and 22.

The **third situation** represents the real one. The manufactured robot can send 60 positions per second to the motion controller. Since the number of positions to be computed with the iterative algorithm is significantly high, only the first tenth of a second of motion from the third situation was chosen to be plotted, which is sufficient to show the magnitude of the displacement error. The results are shown in figures 23 to 27.
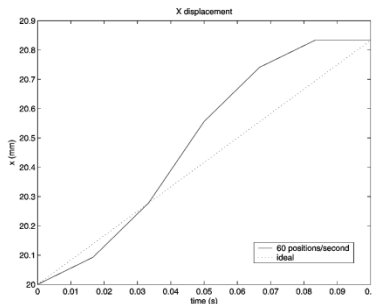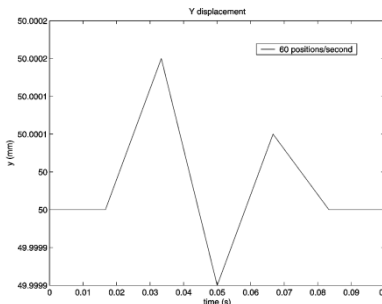

Figure 23 – X displacement.
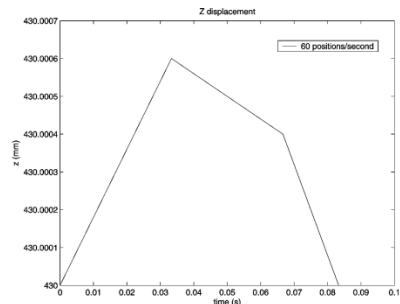

Figure 24 – Y displacement.
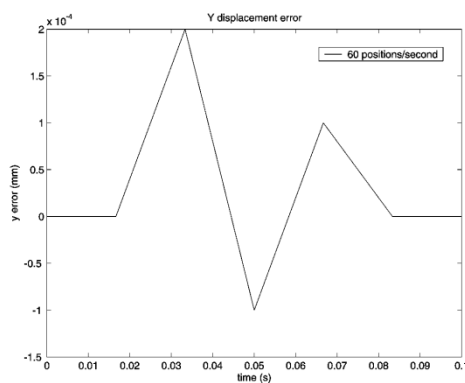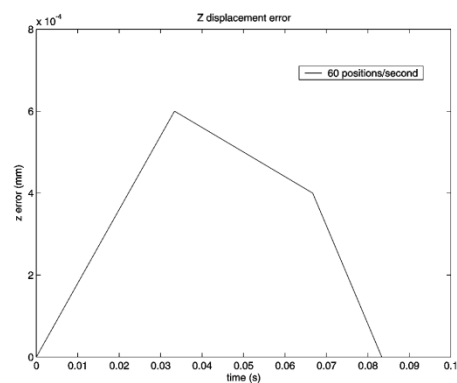

Figure 25 – Z displacement.

Figure 26 – Y error.

Figure 27 – Z error.

For the third situation, the displacement pitch for each position was 50 mm / 60 positions = 0.8333 mm / position, resulting in a maximum error of 0.0006 mm. For a feed rate of 1 mm / s, for example, the displacement pitch would be 1 mm / 60 positions = 0.0167 mm / position. The positioning error in this case would reach negligible values for several applications.

The method used does not eliminate 100% of the parasitic motion error from the robot, however, the error magnitude can be considered negligible for some machining operations.

4. ROBOT DESIGN AND PROTOTYPE

One of the main goals of this work was to evaluate if the robot would be able to perform machining operations with satisfactory resulting workpiece dimension accuracy. Another important goal was to design a workable, low-cost, and easy-to-manufacture solution, since there was a strong budget restriction.

Based on that, the robot's links were designed using flat laser-cut steel plates. The robot was designed using a CAD software and with interlocking pieces to facilitate the assembly (Figures 28, 29, 30 and 31). Carbon steel plates with thicknesses of: 1, 4.76, 6.35 and 7.94 mm were used.
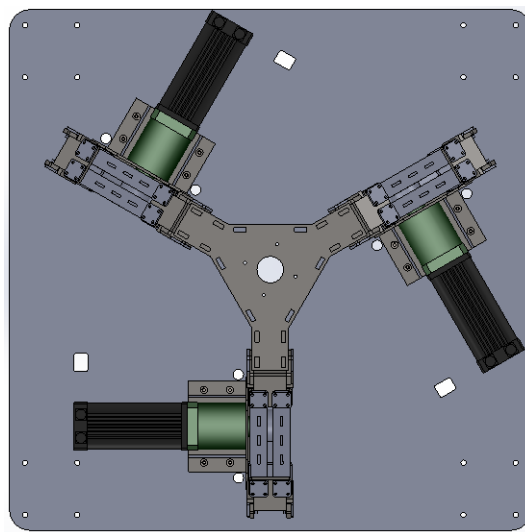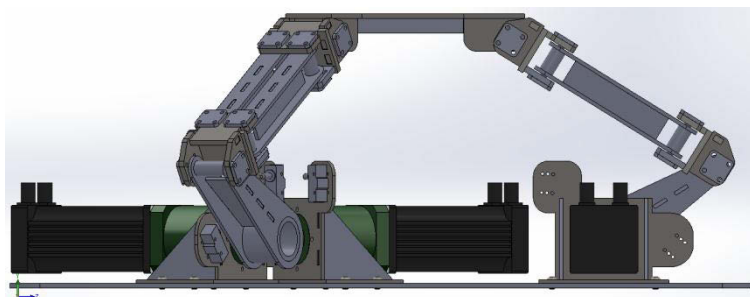

Figure 28 – Top view of robot model.

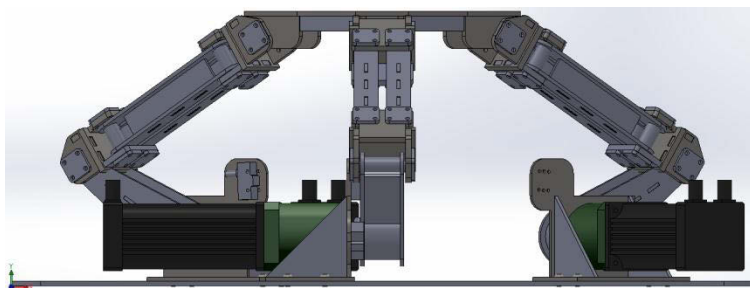Figure 29 – Side view of robot model.

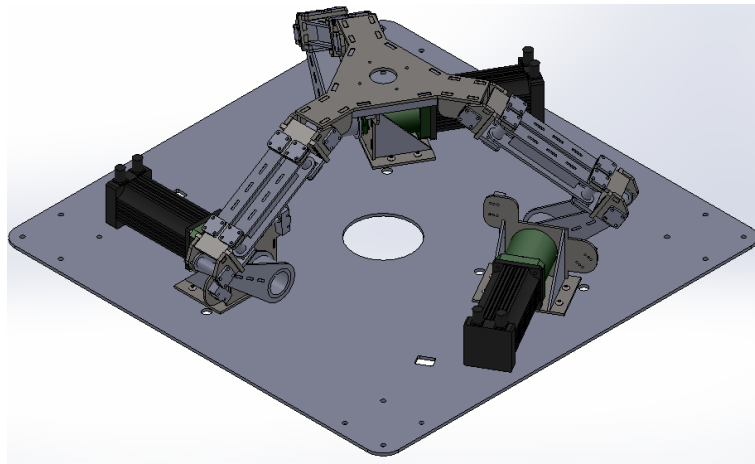Figure 30 – Front view of robot model.

Figure 31 – Isometric view of robot model.

During the welding of robot's parts, welding templates were used to ensure parallelism, orthogonality and to compensate any dimensional error of the laser cutting process. This was sufficient for ensure the desired size for all pieces. Figures 32 and 33 are shown some pieces after the manufacturing process.

Figure 32 – Robot pieces (1).

Figure 33 – Robot pieces (2).

Three 4 N/m servo motors and 50:1 planetary gear units were used, which represented the highest cost share of the robot's manufacturing. As an interface between the computer and the motion controllers, a RS-485 serial communication was used. An USB / RS-485 conversion hardware was also designed and manufactured [23] (Figure 34).
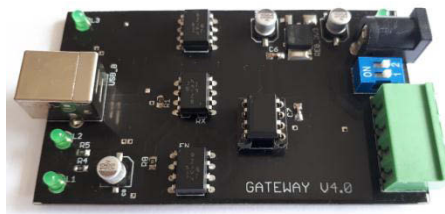

Figure 34 – Gateway USB/RS-485.

A special attention was given to the rotary joint in order to eliminate clearances or backlash. The solution found was to mount the bearings under an external load applied by an external plate. A spacer ring was used for it, so that the fine adjustment of the external load can be done by the lateral screws.

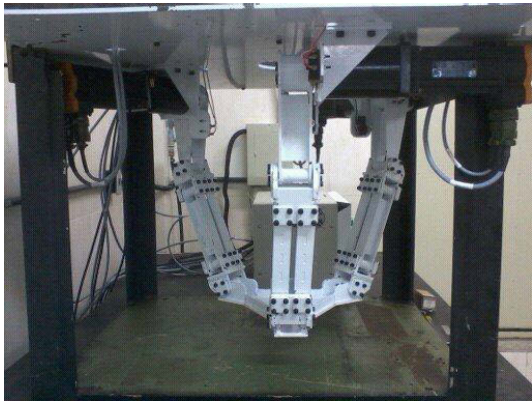In Figures 34 and 35 is presented the robot after final assembly.

Figure 34 – Final robot assemble.

Figure 35 – Moving platform.

After painting, a high speed rotating tool for milling was fixed to the moving platform of the robot (Figure 36). A workpiece fixture for machining was also manufactured (Figure 37).
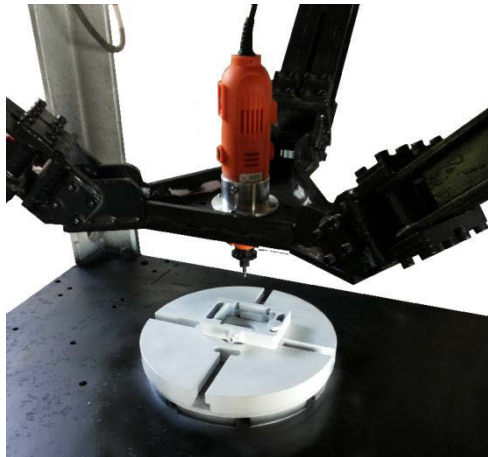

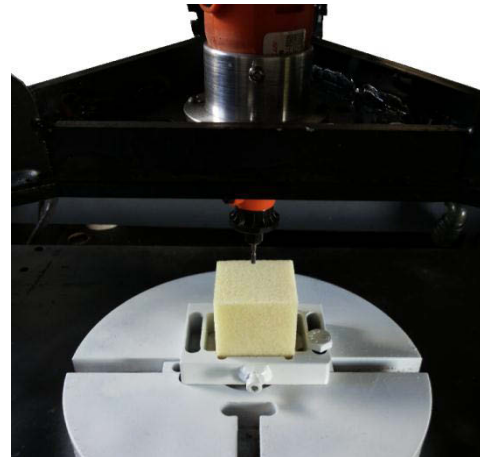
Figure 36 – Robot with rotary tool.

Figure 37 – The workpiece fixture.

As this particular architecture is not available in commercial robot control softwares, it was necessary to develop a complete control software. The software should be capable of interpretating a generic G code and generating all required numerical interpolations in the workvolume. The software also computes the inverse kinematics for each point and sends it to the servo converters (60 positions per second) in a controlled manner. The final result was a DNC control software specific for this architecture. The solution also controls the motors in a synchronized manner, without delay with respect to one another.

To validate the robot, a workpiece created in CAD software was milled. The workpiece was subdivided into layers and, for each layer, a specific G code was generated. The software was able to import and process all layers of G code and generate the trajectory to be performed by the robot. Figures 38 and 39 illustrate the final result of the machining.

Figure 38 – Milled workpiece (top view).

Figure 39 – Milled workpiece.

5. CONCLUSIONS

A simplified inverse kinematic solution was presented. The step by step to solve the inverse kinematics was shown, simplifying the solution implementation for this particular architecture. The need for path interpolation was presented, highlighting its importance for the reduction of the parasitic motion. Proper control of speed and acceleration had be taken into account. It has been shown that the feed speed of the tool is directly related to the magnitude of error for a fixed position sending rate. The robot uses a sending rate of 60 positions per second, which is sufficient for the desired applications. There is no restrictions on increasing this position send rate to increase the positioning accuracy. The low cost method used to manufacture the robot proved to be feasible and efficient. The joints' backlash is neglegible to the whole structure, fact that ensures a very good accuracy to the robot. The dimensions of the milled workpiece reached a tolerance of less than 0.01 mm. Although it is a low cost solution with a common rotary tool, the solution proved to be feasible and quite efficient. Many industrial applications can benefit from this architecture. With small adjustments and improvements, it is possible to make this robot an industrial solution in a short time. Some potential business partners are in contact to exploit this solution.

REFERENCES

[1]   http://www.metalworkingworldmagazine.com/growth-forecast-for-robotics-market-to-2020. 2017.

[2]   https://www.alliedmarketresearch.com/industrial-robotics-market. 2017.

[3]   Y. Chen and F. Dong, "Robot machining: Recent development and future research issues," The international Journal of Advanced Manufacturing Technology, 66(9-12), 1489-1497, 2013.

[4]   A. Klimchik, A. Ambiehl, S. Garnier, B. Furet and A. Pashkevich, "Efficiency evaluation of robots in machining applications using industrial performance measure," Robotics and Computer-Integrated Manufacturing, Volume 48, Pages 12-29, ISSN 0736-5845, 2017.

[5]   D. Zhang, L. Wang and E. Esmailzadeh. "PKM capabilities and applications exploration in a collaborative virtual environment," Robotics and Computer-Integrated Manufacturing, Volume 22, Issue 4, Pages 384-395, ISSN 0736-5845, 2006.

[6]   F. Xie, X. Liu and J. Wang, "A 3-DOF parallel manufacturing module and its kinematic optimization," Robotics and Computer-Integrated Manufacturing, Volume 28, Issue 3, Pages 334-343, ISSN 0736-5845, 2012.

[7]   Y. Patel and P. George, "Parallel Manipulators Applications - A Survey," Modern Mechanical Engineering, Vol. 2, No. 3, pp. 57-64, 2012.

[8]   J. Marie, "On Manufacturing Technology as an Enabler of Flexibility – Affordable Reconfigurable Tooling and Force-Controlled Robotics," Linköping Studies in Science and Technology, Dissertation Thesis no. 1501, ISBN 978-91-7519-691-6, 2013.

[9]   I. Bonev. "The True Origins of Parallel Robots," http://www.parallemic.org/Reviews/Review007.html, 2003.

[10]   W. L. Pollard, "Position Controlling Apparatus," US Patent No. 2286571, 1942.

[11]   V. E. Gough and S. G. Whitehall, "Universal Tyre Test Machine," Proceedings of 9th International Congress FISITA, pp. 117-137, 1962.

[12]   D. Stewart, "A Platform with Six Degrees of Freedom," Proceedings of the Institution of Mechanical Engineers, Vol. 180, No. 1, pp. 371-386, 1965.

[13]   K. L. Cappel, "Motion simulator," US Patent No. 3,295,224, 1967.

[14]   E. F. Fichter, "A Stewart platform-based manipulator: General theory and practical construction," The International Journal of Robotics Research, Vol. 5, No. 2, pp. 157-182, 1986.

[15]   J. P. Merlet, "Parallel Robots," Springer, 2nd edition, 2006.

[16]   R. CLAVEL, "Delta, a Fast Robot With Parallel Geometry," Proc. Of 18th International Symposium on Industrial Robots, Lausanne, Switzerland, pp. 91-100, 1988.

[17]   L. W. Tsai and R. Stamper, "A Parallel Manipulator with Only Translational Degrees of Freedom," ASME Design Engineering Technical Conferences, 1996.

[18]   R. STAMPER, "A Three Degree of Freedom Parallel Manipulator with Only Translational Degrees of Freedom," Ph.D. Dissertation, Department of Mechanical Engineering, University of Maryland, College Park, MD, 1997.

[19]   V. P. Castelli, R. D. Gregorio and F. Bubani, "Workspace and optimal design of a pure translation parallel manipulator," Meccanica,Vol. 35 (3), 203-214, 2000.

[20]   R. D. Gregorio and V. P. Castelli, "A Translational 3-DOF parallel manipulator;" In: J. Lenarcic and M. L. Husty (Eds.), Advances in Robot Kinematics: Analysis and Control, Kluwer Academic Publishers, pp. 49–58, 1998.

[21]   J. Hodgins and D. Zhang, "Improvement, Optimization, and Prototyping of a Three Translational Degree of Freedom Parallel Robot," ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 4: 36th Mechanisms and Robotics Conference, Parts A and B:525-534, 2012.

[22]   L. SCIAVICCO and B. Siciliano, "Modelling and control of robot manipulators," Springer, Advanced textbooks in control and signal processing, 2004.

[23]   D. F. M. CRUZ, "Isolated USB/RS485 gateway converter," Sennze Automation, V. 4, 2015.