

KINEMATIC CONTROL FOR NAVIGATION OF MOBILE PARALLEL ROBOTS APPLIED TO LARGE STRUCTURES.

**Rafael Aracil , Miguel Almonacid* , Roque J. Saltarén*
José M^a Sabater**, José M^a Azorín****

roque.saltaren@umh.es

*Polytechnic University of Madrid,
ETSII, DISAM, 28006, Madrid, Spain*

**Polytechnic University of Cartagena,
ETSII, 30203, Cartagena, Spain*

***Miguel Hernández University,
Dep. of Industrial Engineering, Campus of Elche,
03202, Elche (Alicante), Spain*

Abstract: In this article we propose a path planning strategy for a parallel climbing robot. The robot must be capable of displacing along an “a priori” unknown trajectory, i.e. tubes or beams. Firstly we show the characteristics of the proposed robot. Next we briefly review the theoretic tools of parallel platforms, the inverse and forward kinematic needed for the path planning of 6-6 parallel robot. Then a new navigation algorithm for path planning based on the auto-centring of the bases is presented. We also show some simulations of the functioning of the algorithm. Finally, some conclusions and future work is presented.

Keywords: Robotics, construction robots, path planning, navigation, control.

1 INTRODUCTION

The use of parallel platforms as mobile robots for the maintenance and inspection of cylindrical structures, i.e. tubes or beams, is motivated by the fact that this kind of robots has several advantages in front of robots with serial legs [1].

Based in a Stewart parallel platform of 6 DOF, the Miguel Hernández University is developing a climbing robot for the pruning and maintenance of palm trees. At figure 1 we show the first prototype of the robot. The robot is able to climb automatically using a path planning algorithm based on the kinematics of the platform [2][3]. In general the robot can climb along any cylindrical structure as beams, tubes and also the palm tree trunks.



Figure 1. Parallel Climbing Robot.

We have developed some computational tools based on the multibody dynamics that solve the kinematic and dynamic performance of the parallel robot [4][5]. Using these tools we have tested a path planning algorithm that allows the robot to evolve itself automatically along an unknown trajectory. We show several simulations that demonstrate the correct functioning of the algorithm. Nowadays the incorporation of this algorithm to the microcontrollers of the robot is being developed.

First of all the characteristics of the mobile parallel robot are detailed. Next we briefly review the theoretic tools of parallel platforms, the inverse and forward kinematic needed for the path planning of 6-6 parallel robot and we explain the strategy of the path planning. Finally we show some simulations of the functioning of the algorithm.

2 ROBOT STRUCTURE

The mobile robot proposed has a mechanic structure based on a 6-6 Stewart platform with 6 degrees of freedom [6], figure 2. The robot is made with two hexagonal rings that are linked with six linear power actuators. The actuators are linked to the base through universal joints, and to the final effector through spherical joints. The actuators give the robot six degrees of freedom that are needed to position and orientation itself in the workspace.

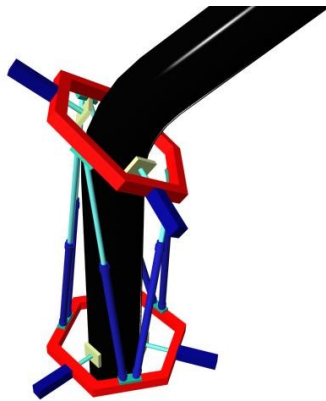


Figure 2. Robot Structure.

The parallel robot has a control architecture composed by a group of sensors, a data acquisition card and a control card. This control architecture allows the robot to move itself automatically.

3 ROBOT KINEMATICS

The study of the kinematics is based on the multibody dynamics [7][8]. The type of robots proposed in this paper are based on a parallel

platform with 6 degrees of freedom, with a **RRPS** kinematic chain (Rotational_Rotational Prismatic and Spherical), where the **RR** degrees of freedom belongs to an universal joint, **P** is a prismatic degree of freedom that belongs to the linear power actuator and **S** is the spherical joint connecting the linear actuator with the final effector.

3.1 Inverse Kinematics

The inverse kinematic solution that is calculated from the position and orientation of the final effector allow us to get the necessary command variables to fit with a programmed path planning.

The inverse geometric model of a RRPS platform implies establishing the values of the joint variables of the kinematic chain for a certain configuration of the final effector. The raising of the solution can be easily obtained from the next vectorial description on generalized coordinates, figure 3:

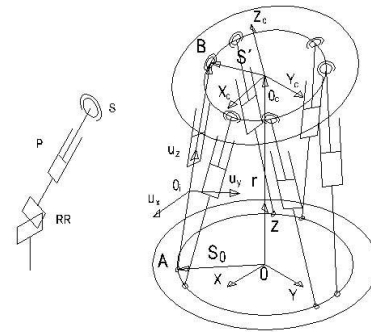


Figure 3: General structure of a 6 DOF parallel manipulator with RRPS kinematic chains.

$$\mathbf{r}_{ABi} = \mathbf{r} + \mathbf{A}_c \mathbf{s}_i - \mathbf{s}_0 \quad (1)$$

In other words, the inverse kinematics calculate the needed displacements of the linear actuators to get the position and orientation required and the generalized coordinates vector (in terms of Euler parameters) that indicates the final configuration of the robot. A more detailed explanation can be found in [4].

3.2 Forward Kinematics

The forward kinematics of a parallel platform studies the relationships between the command variables of the linear actuators and the resultant position of the final effector.

There are in specified literature several methods for the geometric calculation of direct kinematic of 6 DOF parallel platforms, some of them allow us to get

the possible solutions through the use of polynomials that result of the geometric modelling of the kinematic chains of the platform, for example in [9], the 16 possible solutions for a 6 DOF platform are calculated, [10] and [11] prove that the Stewart platform have 12 possible solutions.

We are using a numerical method based on the initial estimation of the generalized coordinates vector q_i will be exposed. In general, a 6 DOF RRPS parallel platform is formed by 12 links that constitute the linear actuators. Each couple of the previous links, are linked between them by a prismatic joint, and each one of the extremes are connected to the base and to the final effector through an universal joint and a spherical joint. Then the generalized coordinates vector will be represented as:

$q = [q_1, q_2, q_3, \dots, q_{13}]_{91 \times 1}^T$ where q_1 is the generalized coordinates system of the final effector and q_2, q_3, \dots, q_{13} correspond to the unit of generalized coordinates assigned to the couple of links that form the linear actuators. In general, each link is defined by a generalized coordinates system where: $q_i = [r_{ABi}, P_i]^T$ with $r_i = [x_i, y_i, z_i]^T$ and the Euler parameters: $P_i = [e_0^i, e_1^i, e_2^i, e_3^i]^T$

The description of the kinematic chain of a RRPS, is based on the constraints vector:

$$\phi(q, t) = \begin{bmatrix} \phi^k(q) \\ \phi^D(q, t) \\ \phi^P(q) \end{bmatrix}_{91 \times 1} = 0 \quad (2)$$

Where $\phi^k(q)_{72 \times 1} = 0$ is the vector of the 72 holonomic constraints imposed by the prismatic, spherical and universal joints. $\phi^D(q, t)_{6 \times 1} = 0$ is a vector of 6 constraints imposed by the actuators, that in this case they are function of the command joint variables for which direct kinematics will be calculated. $\phi^P(q)_{13 \times 1}$ is a vector of 13 constraints for the normalisation of the Euler parameters.

A more detailed explanation of how to obtain this constraints vector can be found in [4].

3.3 Numerical calculation of the kinematic solution.

As it was mentioned before, to calculate the forward kinematic solution we start from an approximated generalized coordinates vector q_i , and the command

variables. For these effects, it is commonly used the Newton-Raphson method:

$$\phi_q \Delta q^{(j)} = -\phi(q^{(j)}, t) \quad (3)$$

$$q^{(j+1)} = q^{(j)} + \Delta q^{(j)} \quad (4)$$

where ϕ_q is the Jacobean of the vector of constraints described in (2) y $q^{(j+1)}$ is the forward kinematic solution when $\Delta q^{(j)} \approx 0$.

4 PATH PLANNING

The path planning is based on an autoguided-algorithm. The needed information for the guided is obtained from ultrasonic sensors allocated on each ring. The developed algorithm consists on a process of detection and correction of the centring error. The robot can displace itself along a spatial curve in a continuous way. Depending on the expected curvature of the trajectory, we divide the path in a number of intervals. On each interval the ultrasonic sensors detect the centring error, and with these information and the inverse kinematics, the next displacements of the linear power actuators are calculated. The calculated displacements try to correct the error in the next interval. The initial and final positions of the effector or the base constitute a vector that allows us to calculate the orientation. Starting from the sensorial information and the algorithms implemented on the microcontrollers the robot can follow an unknown trajectory automatically, positioning and orienting itself according with the curvature of the path.

4.1 Description

Each one of the hexagonal rings has a rotational joint that allows the ring to be opened manually, figure 4. This permits the initial positioning of the robot round the structure to climb.



Figure 4. Superior View of the robot. The rotational joint permits the opening of the rings.

Besides, each ring has a press system made of three linear actuators, figure 5, that allow the subjecting of the robot to the structure. There are also protection springs that guarantee the robot to stay fastened to the structure if the power supply falls down.

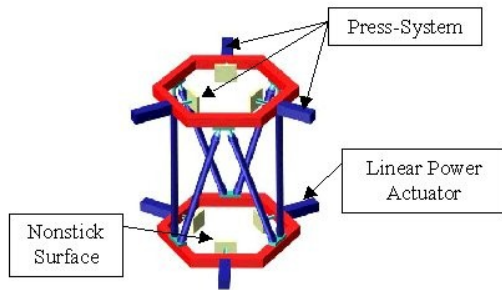


Figure 5. View of the press system on each ring.

4.2 Displacement strategy

First of all the robot is allocated on the initial position of the structure rounding it. The rings are opened manually and then, when the robot round the structure, the rings must be closed. Next, the robot is fixed to the structure by activating the press system of both rings. Figure 6 shows the initial position of the robot round a structure.

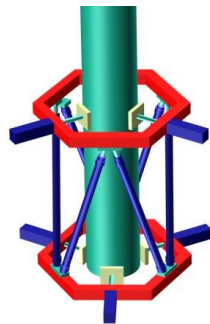


Figure 6. Initial position of the robot rounding a structure. Note that the press systems of both rings are activated.

The displacement of the robot along a straight structure take a sequence of five steps, figure 7:

1. Put off the press system of the final effector (the upper ring). Figure 7 (a).
2. Put on the displacement of the linear actuators until the top of the career, and the final effector moves up. Figure 7 (b).
3. First the press system of the final effector puts on, and then the base one's puts off. Figure 7 (c).
4. The linear actuators are contracted until the minimum career, and the lower base moves up. Figure 7 (d).
5. Put on the press system of the lower ring.

This sequence repeats until the desired position is obtained.

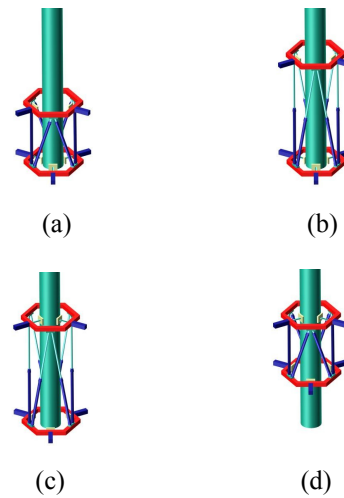


Figure 7. Sequence of steps that define the displacement of the robot along a straight structure. (a) shows the robot with the press system of the base off. (b) shows the displacement of the final effector. (c) puts the press system of the final effector on and the base's one off. (d) the base moves up.

At the descent of the robot, the sequence is the same but now the final effector is the lower ring and vice-versa.

4.3 Sensorial information

If the robot went always along straight structures, the former algorithm would be enough. Nevertheless, in general the robot will climb through unknown trajectories. To make the robot able to follow any trajectory it would need sensorial information. This information is provided by three ultrasonic sensors allocated on each ring and separated 120°, as it is shown in figure 8.

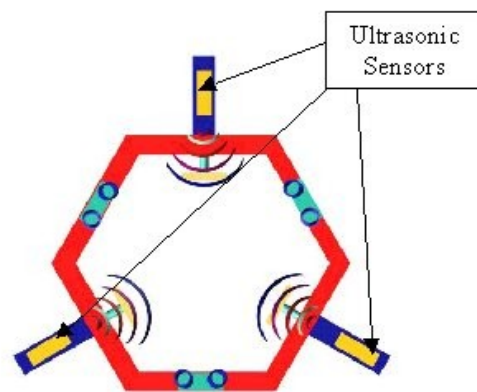


Figure 8. View of the position of the ultrasonic sensors on the rings.

The positions of the ultrasonic sensors define a plane in the space where the measures are taken. The sensorial information allows us to know the position of the centre of the structure on the plane of the ring. The intersection of this plane with the large structure, figure 9, allows us calculate the corrections needed for the centring of the ring. The distances measured by the sensors (d_1, d_2, d_3) give the sufficient information to determine the deviation ($\Delta x, \Delta y$) of the centre of the structure (c_c), in relation with the reference system of the ring (X_0, Y_0).

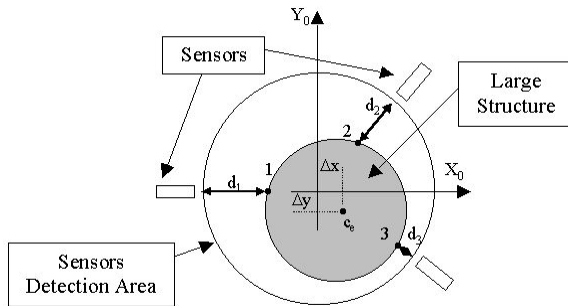


Figure 9. Transversal section on the ring plane. Calculus of the deviation ($\Delta x, \Delta y$) of the centre of the structure in relation with the reference system of the ring (X_0, Y_0).

The position of the sensors is known. Distances d_i ($i=1,2,3$) give the coordinates of the points 1, 2 and 3. These points define a circle whose centre matches with the structure one's. This way, to determine the deviation ($\Delta x, \Delta y$) is necessary to solve the next system of equations:

$$R = \sqrt{(x_i - \Delta x)^2 + (y_i - \Delta y)^2} \quad (5)$$

where (x_i, y_i) for $i = 1, 2, 3$ are the coordinates of the points and R is the radius of the structure circle.

4.4 Displacement Algorithm

The algorithm allows the robot to displace in an autonomous way. The main purpose of the algorithm is to get the final position in the shorter time possible. To get this is necessary:

- Divide the trajectory with intervals as much larger as possible.
- The robot must go from one position to other changing the position and orientation at the same time.

In general way, the algorithm uses the information sensorial obtained at one instant of time to correct the

position and orientation of the ring at the later instant, and so on. Next, we detail the steps of the algorithm, figure 10. The algorithm starts with an initial position obtained manually as it was explained before.

1. Put off the press system of the final effector.
2. Let $(X_{i-1}, Y_{i-1}, Z_{i-1})$ the reference system of the final effector ring before the displacement starts. With this information, the deviation ($\Delta x_i, \Delta y_i$) of the centre of the structure in relation with the reference system of the ring is solved, using the system of equations (5).
3. The final effector moves up a distance previously fixed (Δ), in a normal direction (Z_{i-1}). At the same time the position of the centre of the ring is corrected using the sensorial information ($\Delta x_i, \Delta y_i$). This way, the position of the new reference system (X_i, Y_i, Z_i) centred on the ring, corresponds with the position $(\Delta x_i, \Delta y_i, \Delta)$ in relation with the reference system $(X_{i-1}, Y_{i-1}, Z_{i-1})$. With the coordinates of the new reference system and the former one we obtain a vector that allows us to correct the orientation. The new orientation of the perpendicular of the ring (axis Z_i) will be aligned with this vector. To obtain the orientation of the ring we need to solve the following cross products:

$$X_i = Y_{i-1} \times Z_i \quad Y_i = Z_i \times X_i \quad (6)$$

4. Starting from the position and orientation of the base and the final effector, the command variables that activate the displacements of the actuators are calculated using the inverse kinematics.
5. Steps 2, 3 and 4 are repeated until one of the actuators reaches the contour condition.
6. First the press system of the final effector is putting on, and then the base's one is putting off.
7. Steps 2 and 3 are repeated but considering as final effector the lower ring.
8. Using the inverse kinematics and starting from the position and orientation of the base and the final effector, the command variables of the actuators are calculated.
9. Steps 7 and 8 are repeated until one of the actuators reaches the contour condition.
10. The press system of the base ring is putting on.
11. Steps 1-10 are repeated until the robot reach the desired position.

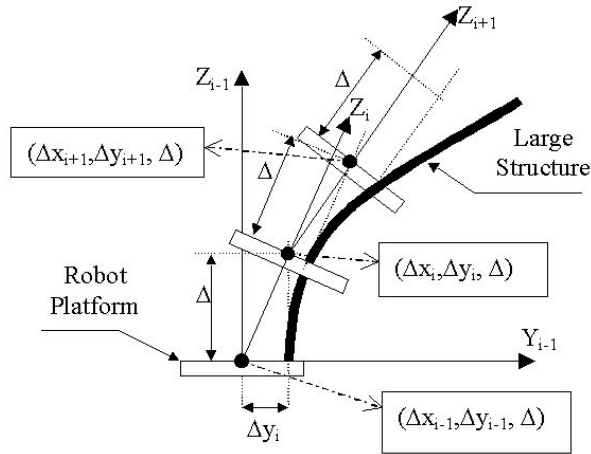


Figure 10. Path planning algorithm.

Considerations

- When solving the inverse kinematics, besides of obtaining the command variables of the actuators, the generalized coordinates vector are also obtained, as it was explained in the inverse kinematics section. That way avoids the use of the forward kinematics to get the final configuration of the robot. To test the vector obtained, we must check that the Jacobean of the constraints vector, ϕ , described in (2), is rank complete. The check of the rank is made to verify that the command variables can be applied to the actuators according with the mechanical constraints of the robot. This method implies a reduction of the computational cost, avoiding the use of Newton-Raphson to calculate the configuration of the robot when we send the command variables.
- The contour condition indicates the maximum and minimum displacement allowed for the linear actuators. This displacement is dependent of the curvature radius of the structure. If there is neither autocollision nor collision with the structure, this value could reach the maximum or minimum career of the cylinders. This value is not a fixed value, and it could vary according with the curvature at each instant.
- The displacement of the ring (Δ) is fixed in function of the curvature radius of the structure. Besides, it would be possible that this quantity varies along the trajectory, according with the curvature at each moment.
- To make the robot comes back to the initial position, it is not necessary to use the sensorial information, due to the robot has been storing the positions of each base along the trajectory. This way, when the robot comes back to the origin,

the command variables are calculated using the inverse kinematics.

5 SIMULATIONS.

We have developed computational tools in Matlab that allow us to test the former algorithm proposed. Using these tools we can define different kinds of structures, different displacement of ring Δ , and many other variables. Besides, these tools can visualise the functioning of the algorithm. Figure 11 show several instants of the simulation of the robot moving along a structure using this algorithm.

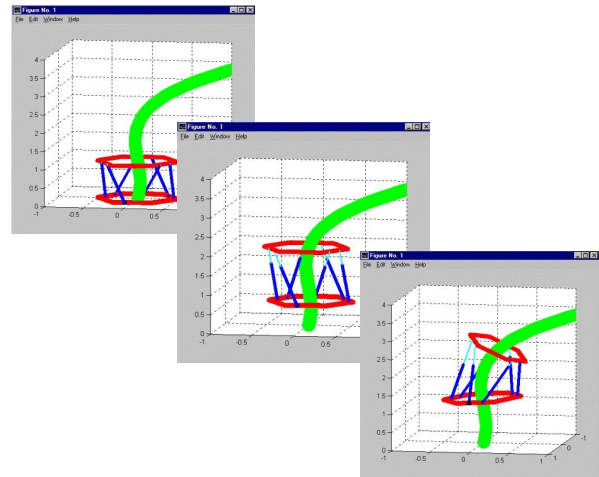


Figure 11. Different views of the simulation of the robot moving along a structure.

To have an idea of the computational cost of the path planning algorithm, figure 12 shows the number of float operations required at each interval of displacement that the robot makes. Note that the number of float operations is nearly equal, so that the computational cost will be greater as much as the number of intervals to cover a trajectory grows.

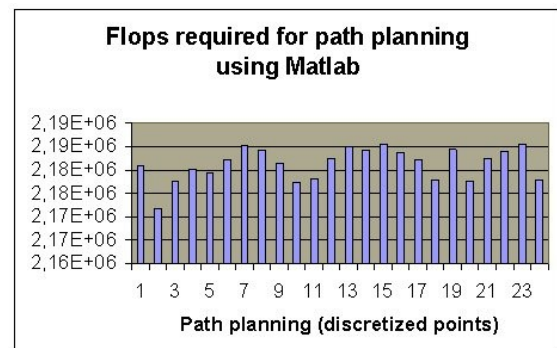


Figure 12. Number of float operations required.

6 CONCLUSIONS

This article presents a navigation algorithm that allows the parallel climbing robot to displace itself along an unknown trajectory. The parallel structure gives the robot a high stiffness and a high load capacity, useful for maintenance and inspection tasks.

The main advantage of the proposed algorithm is that it is not necessary to solve the forward kinematics to get the final configuration of the robot because we obtain it from the inverse kinematics, and we only check if it is in accordance with the mechanical constraints. This implies a reduction of computational costs.

Nowadays we are working in the implementation of the algorithm inside the control architecture so that we can make real experiments to verify the results of the simulations.

REFERENCES

- [1] R. Aracil, R. Saltarén, and J.M. Sabater, "TREPA, Parallel Climbing Robot for Maintenance of Palm Trees and Large Structures", in *Proc. 2nd International Workshop & Conference on CLIMBING & WALKING ROBOTS (CLAWAR)*, Sept. 1999, pp. 453-461.
- [2] J.P. Merlet, "Trajectory Verification of Parallel Manipulators in the Workspace", *IEEE International Conference on Robotics and Automation*, pp. 2166-2171, 1994.
- [3] I. Alizzade Rasim, and R. Tagiyev Nazim, "A Forward and Reverse Displacement Analysis of a 6 DOF in Parallel Manipulator", *Mechanisms and Machine Theory*, vol. 29, no. 1, pp. 115-124, 1994.
- [4] R. Saltarén, R. Aracil, J.M. Sabater, O. Reinoso, and L.M. Jiménez, "Modelling, Simulation and Conception of Parallel Climbing Robots for Construction and Service", in *Proc. 2nd International Workshop & Conference on CLIMBING & WALKING ROBOTS (CLAWAR)*, Sept. 1999, pp. 253-265.
- [5] M. Almonacid, R. Saltarén, O.Reinoso, N. García, and L.M. Jiménez, "Herramientas para la simulación de plataformas paralelas usadas como robots trepadores", in *XX Jornadas de Automática*, Sept. 1999, pp. 149-153.
- [6] J.P. Merlet, *Les Robots Parallèles*, Ed. Hermes, 1997.
- [7] E. J. Haug, *Computer-Aided Kinematics and Dynamics Of Mechanical Systems*, USA: Allyn and Bacon, 1989.
- [8] P. E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, USA, 1988.
- [9] J. P. Merlet, "Direct kinematics and assembly modes of parallel manipulators", *Int. Journal of Robotics Research*, 11(2), pp. 150-162, April 1992.
- [10] D. Lazard and J.P. Merlet, "Stewart platform has 12 configurations", *IEEE Int. Conf. On Robotics and Automation*, pp. 2160-2165, San Diego, May 8-13, 1994.
- [11] G. Danescu and M. Dahan, *Prolégomènes n° 6 (INRIA)*, January 1994.