Numerical and Analytical Analysis of a 3UPS-2RPRRR Parallel Robot

Julio Vargas, PhD student[1], Óscar A Varela, MSc[2], Ángel Valera, PhD[3]
[1,3]Universitat Politènica de València, España, [2]Universidad de los Llanos, Colombia
[1]julio_h_vargas_r@ieee.org, [2]oscar.agudelo@unillanos.edu.co [3]giuprog@isa.upv.es

***Abstract*– This work presents a 3UPS-2RPRRR parallel robot and its geometric analysis. The robot has three rotational degrees of freedom and two translational degrees of freedom in the XZ plane. The inverse geometric model was obtained by closed-loop analysis. The direct geometric model was obtained analytically by the representation of the geometrical varieties for each end limb. The intersection of varieties results in a multivariate nonlinear equation system. The nonlinear system was solved founding four assembly modes. We use the free package SageMath. Two solution methods were addressed: numerical one solution, and Gröbner basis with Toy Buchberger algorithm. Finally, example calculations of inverse and forward geometrical model were analyzed and plotted.**

***Keywords*-- parallel robots, inverse kinematics, direct kinematics, nonlinear systems, Gröbner basis, Buchberger algorithm.**

I. INTRODUCTION

A parallel robot is a mobile platform attached to a fixed base with several limbs or simple kinematic chains.

Parallel robots are studied because they have advantages like stiffness and speed over serial robots. An important thesis about geometric analysis [1].

Low mobility parallel robots, are known too as constrained mechanisms or sub-six degrees of freedom. These mechanisms have gained special interest because some tasks do not require six degrees of freedom and use less actuators. These kinds of parallel robots are classified by the number and type of degrees of freedom, e.g. rotation or translation.

In the 3UPS-2RPRRR robot, the platform coordinate system can rotate three degrees of freedom and translate two degrees of freedom with respect to a fixed base coordinate system.

The number and type of limbs, with the active joints underlined are represented by several references [2]–[8]. In such representation, the 3UPS-2RPRRR robot has 3 universal-prismatic-spherical and two rotative-prismatic-universal kinematic chains. The prismatic joints are active and the remaining joints are passive.

A literature review about this type of robots reveals that: planar mechanisms, 4 DOF, spherical and *Schönflies* motions are classified as low mobility parallel robots [9]–[27].

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2019.1.1.307
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

Related work on type synthesis of 5 DOF is in [25], but the limbs are symmetrical.

Several parallel robots are constrained by a central mast. Have passive joints with constraints, letting only pure rotation or translation by using guides. These kinds of robots are known as shoulder, wrist or knee. Because they are similar to animal or human body biomechanisms with central support rounded by actuators.

The use of central mast gives more power at the center of the platform letting higher loads and less stress forces applied to the structure.

This work focuses on the finding of the position and orientation between the end effector and the fixed base known as geometric analysis. The literature [28] gives disambiguation between geometric and kinematic analysis. Kinematics analysis involves linear and angular speeds and accelerations, Jacobians, self-motions and singularity analysis. All these concepts are derived from the robot geometry.

The geometric analysis consists of the inverse and the forward problems. The main objective of the inverse geometric analysis is to find the active limbs values given a desired position and orientation. Complex kinematic chains probably have different working modes.

The aim of the forward geometric problem is to find the position and orientation of the end effector, given the active joints values. Solving this problem, several assembly modes can be found.

The inverse problem is straightforward and only requires closed loop equations. The forward analysis is complex because implies nonlinear systems, algebraic manipulation, and numeric approximations.

In this work, our contribution is the solution of the inverse and forward geometric problems in analytical approach.

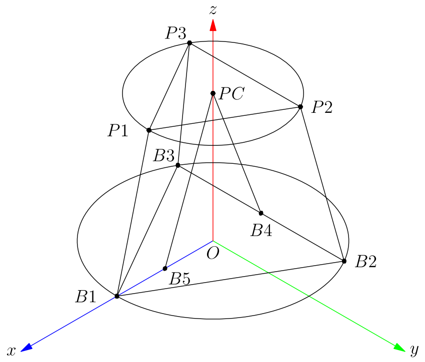
II. METHODOLOGY

The closed loop equations are directly defined from the robot schematic.

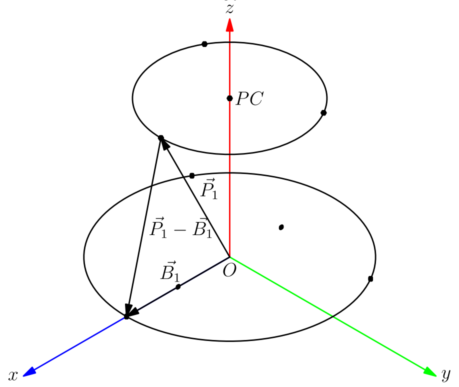
We realize that orthogonal passive joints are best analyzed via screw theory as shown in many sources before cited. The advantages are the easy geometric visualization of twist and wrench concepts. The reciprocity principle is useful for the study of constraints and mobility of the mechanism. In this work, we use the pitch, roll and yaw convention.

A. *Inverse Geometric Analysis*

To illustrate the concept, the Fig. 1 shows the schematic in a, and the vector difference in b.



a.



b.

Fig. 1 Schematic of a. Robot, and b. Loop vector

The vectors with endpoints P1, P2, P3, and PC represent the mobile platform. The vectors with endpoints B1, B2, B3, B4, B5 represent the fixed base. The origin O is the start point of all the vectors. The vectors are represented with the corresponding subscript; e.g.:
$\vec{P}_1 = \overrightarrow{OP1}$, $\vec{B}_1 = \overrightarrow{OB1}$, $\vec{P}_2 = \overrightarrow{OP2}$, $\vec{B}_2 = \overrightarrow{OB2}$, $\vec{P}_3 = \overrightarrow{OP3}$, $\vec{B}_3 = \overrightarrow{OB3}$, $\vec{P}_C = \overrightarrow{OPC}$, $\vec{B}_4 = \overrightarrow{OB4}$, $\vec{B}_5 = \overrightarrow{OB5}$,
The rotation matrix of the platform is:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{1}$$

And the position vector at the platform center is:

$$\vec{P}_C = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{2}$$

We use the *roll-pitch-yaw* convention, also known as Euler ZYX, that is:

$$R = R_z(\phi)R_y(\theta)R_x(\psi) \tag{3}$$

then, the homogeneous transformation matrix is:

$$T = \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi + s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi & x_c \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & y_c \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Here, $c_\gamma = \cos(\gamma)$ and $c_\gamma = \sin(\gamma)$ is a compact notation, where γ is ψ, θ or ϕ.

The points of the base and the platform corresponds to two equilateral circumscribed triangles. The radius r_b and r_p are proportional to D:

$$r_b = \frac{D}{2} \tag{5}$$

$$r_p = \frac{D}{3} \tag{6}$$

The vectors at the base are:

$$\vec{B}_1 = r_b(1,0,0)^T \tag{7}$$

$$\vec{B}_2 = r_b\left(c_\beta, s_\beta, 0\right)^T \tag{8}$$

$$\vec{B}_3 = r_b\left(c_{-\beta}, s_{-\beta}, 0\right)^T \tag{9}$$

$$\vec{B}_4 = r_b\left(-\frac{1}{2}, 0,0\right)^T \tag{10}$$

$$\vec{B}_5 = r_b\left(\frac{1}{2}, 0,0\right)^T \tag{11}$$

The platform vectors at the base are:

$$\vec{P}_1^b = r_p(1,0,0)^T \tag{12}$$

$$\vec{P}_2^b = r_p\left(c_\beta, s_\beta, 0\right)^T \tag{13}$$

$$\vec{P}_3^b = r_p\left(c_{-\beta}, s_{-\beta}, 0\right)^T \tag{14}$$

Where:

$$\beta = \frac{2}{3}\pi \tag{15}$$

Applying the transformation to the platform points:

$$\vec{P}_i = T\vec{P}_i^b \tag{16}$$

For $i = 1, 2, 3$, gives the relative position of the platform respect to the base.
The closed-loop vector equations are:

$$\vec{L}_i = \vec{P}_i - \vec{B}_i \tag{17}$$

For $i = 1, 2, 3$; and:

$$\vec{L}_j = \vec{P}_C - \vec{B}_j \tag{18}$$

for $j = 4,5$.
Then, the solutions of the square root of the self-dot products are:

$$q_m = \sqrt{L_m^T \cdot L_m} \tag{19}$$

Where m=1…5 and q_m are the lengths of all the limbs.

B. Forward Geometric Analysis

The forward geometric analysis is often more complex, in this work we follow the geometric approach. The values of the points in the base were defined in section A. All are proportional to D.

The Fig. 2 shows the intersections of the circles C_4 with center in B_4 and radius q_4, and C_5 with center in B_5 and radius q_5 are located in the points PC_1 and PC_2.

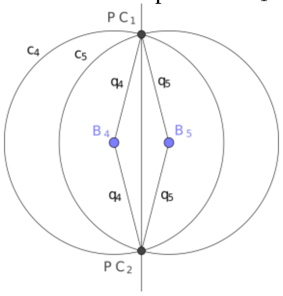


Fig. 2 Plattform central point positions.

Analytically, the central point can be calculated from the intersection between two circles in the XZ plane.

$$(x - B4_x)^2 + (z - B4_z)^2 - q_4^2 = 0 \tag{20}$$

$$(x - B5_x)^2 + (z - B5_z)^2 - q_5^2 = 0 \tag{21}$$

Solving the two-equation system, we got two points PC_1 and PC_2. The points are symmetrical to the XY plane, we choose the point with positive z. Renaming as point $PC = (x_c, 0, z_c)$, with coordinates:

$$x_c = \frac{q_4^2 - q_5^2}{D} \tag{22}$$

$$z_c = \frac{\sqrt{-D^4 - 16q_4^4 + 8D^2q_5^2 - 16q_5^4 + 8(D^2 + 4q_5^2)q_4^2}}{4D} \tag{23}$$

The Fig. 3 shows the intersection of two spheres. The central sphere with center in PC and radius r_p, and the sphere with center in B1 and radius q_1.

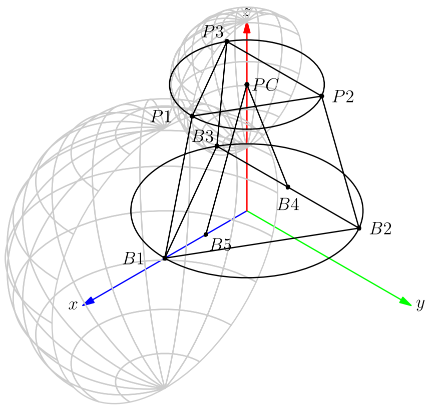


Fig. 3 Intersection of two spheres.

For each intersection between the spheres with center in B1, B2, B3 and radius q1, q2, and q3, respectively and the central sphere with center in PC and radius r_p. The equations are defined by:

$$(x_i - Bi_x)^2 + \left(y_i - Bi_y\right)^2 + (z_i - Bi_z)^2 - q_i^2 = 0 \tag{24}$$

$$(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 - r_p^2 = 0 \tag{25}$$

For i = 1,2 and 3. The points $Bi = (Bi_x, Bi_y, Bi_z)$ are the vertexes of the base triangle. The points $Pi = (x_i, y_i, z_i)$ are the vertexes of the platform, and $PC = (x_c, y_c, z_c)$ is the center point of the platform.

The difference between these two groups of equations gives:

$$-\frac{D^2 - 2q_4^2 + 2q_5^2}{D} \cdot x_1 \pm \frac{\gamma}{2D} \cdot z_1 + \frac{61}{144}D^2 - q_1^2 - \frac{1}{2}q_4^2 - \frac{1}{2}q_5^2 = 0 \tag{26}$$

$$\frac{D^2 + 4q_4^2 - 4q_5^2}{2D} \cdot x_2 - \frac{1}{2}\sqrt{3}D \cdot y_2 - \frac{\gamma}{2D} \cdot z_2 + \frac{61}{144}D^2 - q_2^2 - \frac{1}{2}q_4^2 - \frac{1}{2}q_5^2 = 0 \tag{27}$$

$$\frac{D^2 + 4q_4^2 - 4q_5^2}{2D} \cdot x_3 + \frac{1}{2}\sqrt{3}D \cdot y_3 - \frac{\gamma}{2D} \cdot z_3 + \frac{61}{144}D^2 - q_3^2 - \frac{1}{2}q_4^2 - \frac{1}{2}q_5^2 = 0 \tag{28}$$

Where:

$$\gamma = \sqrt{-D^4 + 8D^2q_4^2 - 16q_4^4 - 16q_5^4 + 8(D^2 + 4q_4^2)q_5^2} \tag{29}$$

The coordinates of the points $Pi = (x_i, y_i, z_i)$ for i = 1,2 and 3, are the nine unknowns.

The center of an equilateral triangle is PC, then the average of the three points is the center of the platform:

$$P1 + P2 + P3 = 3 \cdot PC \tag{30}$$

This property gives three more equations:

$$x_1 + x_2 + x_3 - 3 \cdot x_c = 0 \tag{31}$$
$$y_1 + y_2 + y_3 - 3 \cdot y_c = 0 \tag{32}$$
$$z_1 + z_2 + z_3 - 3 \cdot z_c = 0 \tag{33}$$

The Fig. 4 shows the distances between the platform points P1, P2, P3 and PC.

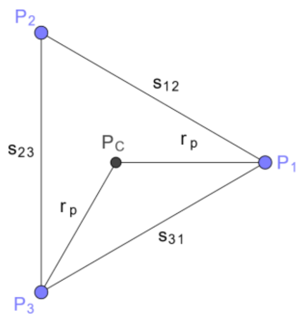


Fig. 4. Distances between platform points.

The sides of the equilateral triangle of the platform are equal to s:

$$s_{12} = s_{23} = s_{31} = s \tag{34}$$

By cosines law:

$$s^2 = 2r^2 - 2r^2 cos\left(\frac{2}{3}\pi\right) = 2r^2\left(1 + \frac{1}{2}\right) = 3r^2 \tag{35}$$

The relation between the distance s and the points coordinates are:

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - s^2 = 0 \tag{36}$$
$$(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 - s^2 = 0 \tag{37}$$
$$(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 - s^2 = 0 \tag{38}$$

Then, six first-order equations: 26-28, 31-33, and three second-order equations: 36-38. Nine equations in total.

III. Materials And Tools

In this work we solve the inverse and direct geometric methods by using SageMath 8.5 [29] installed in Windows 10. Running in an Intel core i7 at 2.5Mhz with 16Gb RAM.

The SageMath scripts and results are showed in the Appendix. The scripts are implementation of the geometrical analysis.

The solution methods applied are part of the libraries in SageMath.

IV. Results

In this section, we use the methods of section III with example values. The base and platform dimensions are proportional to D, we use a scale base of D = 1. This means that the robot can be scaled to any size proportional to D.

The x values are in the interval [-D/2, D/2], the y value is zero, and z values are in the interval [-D/4, D]. The orientation angles ψ, θ, and φ are in the interval (-π/3, π/3).

Solving the inverse geometric is easy and with a different position and orientation values of the Table I.

TABLE I.
Inverse Geometric Initial Values

x_c	z_c	Ψ	Θ	φ
-1/2	1/4	- π/3	0	0
-1/4	3/8	- π/4	π/6	π/6
0	1/2	- π/6	π/4	π/4
1/4	5/8	0	π/3	π/3
3/8	3/4	π/6	- π/3	- π/3
5/8	7/8	π/4	- π/4	- π/4
1/2	1	π/3	- π/6	- π/6

The values of the output for each pose are in the TABLE II.

TABLE II.
Lengths Of The Limbs

q1	q2	q3	q4	q5
0.712	0.506	0.712	0.353	0.790
0.560	0.568	0.68	0.375	0.625
0.456	0.665	0.866	0.559	0.559
0.402	0.874	1.067	0.800	0.625
1.049	1.009	0.736	0.976	0.760
1.160	1.232	1.004	1.237	0.951
1.201	1.288	1.013	1.250	1.030

The platform points for $xc = -\frac{D}{2}$, $zc = \frac{D}{4}$, $ps = -\frac{pi}{3}$ and $th = ph = 0$ are in the Table III.

TABLE III.
Points Of The Platform

Point	X	y	z
P1	−0.166	0.0	0.25
P2	−0.666	0.1447	0.0
P3	−0.666	−0.144	0.5

The values of $\mathbf{q_n}$ are in the first file at the Table II. The computations of the forward geometrical model found the solution by numerical approximation. Eight roots were found, six complex and two reals in the Table IV and Table V.

TABLE IV.
Real Solution 1

Point	X	y	z
P1	-0.194	-0.073	0.138
P2	-0.696	0.205	0.075
P3	-0.608	-0.131	0.536

TABLE V.
Real Solution 2

Point	x	y	z
P1	-0.166	0	0.25
P2	-0.666	0.144	0
P3	-0.666	-0.144	0.5

The Toy Buchberger algorithm and polynomial root found one solution (See the Table VI). This solution is coincident with the first numerical real solution.

TABLE VI.
Buchberger Algorithm And Polynomial Roots

Point	X	y	z
P1	−0.194	−0.073	0.138
P2	−0.696	0.205	0.075
P3	−0.608	−0.131	0.536

The Fig. 5 shows the plot of the inverse geometric problem with the initial data, when the direct numerical method diverges.

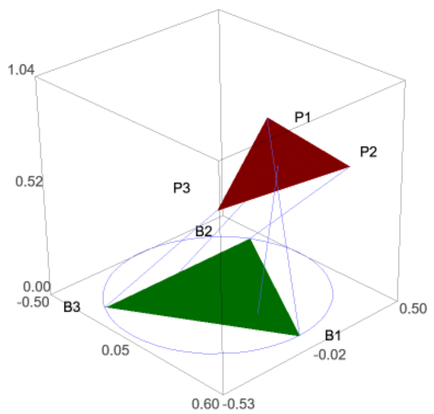


Fig. 5 Inverse geometric plot

The Fig. 6 shows two positive solutions. The red and green visualization of the platform explains why sometimes the direct numerical method diverges.

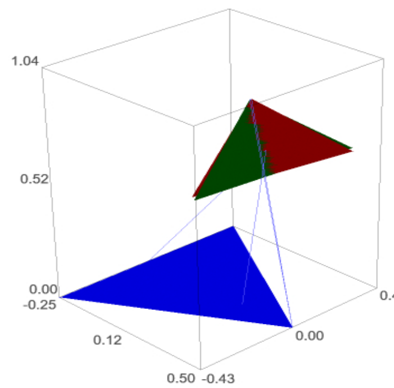


Fig. 6 two nearly direct geometric solutions

When the direct numerical method does not diverge, the initial conditions must be closest to the solution searched.

For example, in the Fig. 7 an inverse solution is found by direct numerical method and Gröbner basis method.

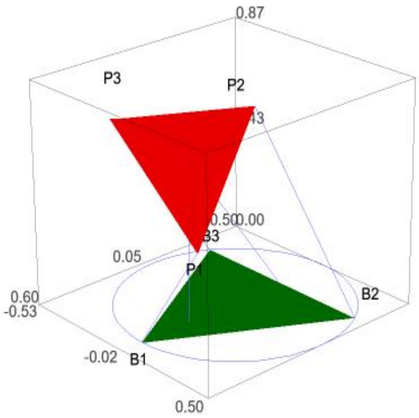


Fig. 7 Inverse solution found by two methods

The two corresponding z positive solutions, in red and green are showed in the Fig. 8.

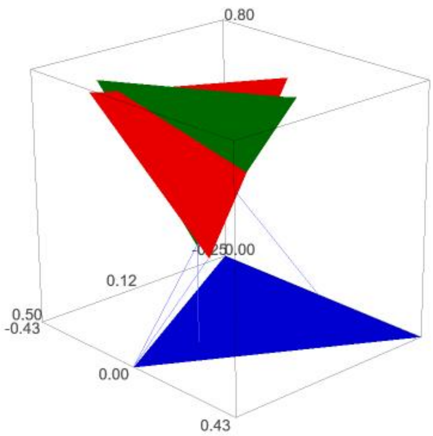


Fig. 8 two z positive direct solutions

Another two solutions, corresponding to the z negative are showed in the Fig. 9.

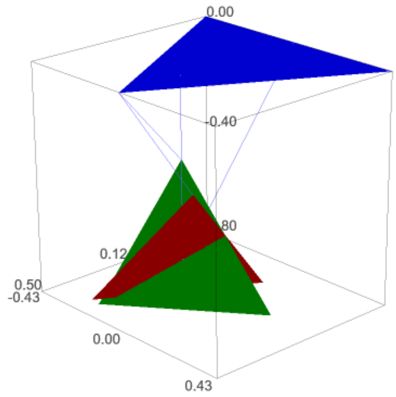


Fig. 9 two z negative solutions

CONCLUSIONS AND RECOMMENDATIONS

The inverse geometric analysis of the 3UPS-2RPRRR robot was solved by using the transformation matrix, no working modes were found because the limbs are simple prismatic joints.

SageMath is a free open tool for solving nonlinear systems and algebraic manipulation.

The accuracy solution of the forward geometric analysis can be found by using Gröbner basis and Toy Buchberger algorithm.

The 3UPS-2RPRR parallel robot has four assembly modes. Near to the origin the Newton Raphson method can oscillate between close values.

Sometimes the limbs cross and has not physical meaning. The analysis was made for all the possible values in the space of the numerical solution.

Is recommended to avoid the poses when the position oscillates. This issue must be addressed in the control of the robot. The *yaw* may be in positive values far from zero.

The execution time for inverse analysis must be implemented in real time because is important in motion planning.

The 3UPS-2RPRR robot can be applied in simulators of forward and backward displacements like cars, boats or spaceships.

APPENDIX

The scripts in SageMath.

A. *Inverse Geometric Model*

The script for initial values is:

```
D=2
rb=D/2
rp=D/3
xc=D/4
yc=0
zc=D/3
ps=pi/12
th=0
ph=0
```

```
B1=vector([rb, 0, 0])
B2=vector([rb*cos(2/3*pi), rb*sin(2/3*pi), 0])
B3=vector([rb*cos(-2/3*pi), rb*sin(-2/3*pi), 0])
B4=vector([-rb/2,0,0])
B5=vector([rb/2,0,0])
PB1=vector([rp,0,0])
PB2=vector([rp*cos(2/3*pi), rp*sin(2/3*pi), 0])
PB3=vector([rp*cos(-2/3*pi), rp*sin(-2/3*pi), 0])
PC=vector([xc,yc,zc])
```

The script defining the transformation matrix, the points and the differences is:

```
sx=cos(ph)*cos(th)
sy=sin(ph)*cos(th)
sz=-sin(th)
nx=cos(ph)*sin(th)*sin(ps)-sin(ph)*cos(ps)
ny=sin(ph)*sin(th)*sin(ps)+cos(ph)*cos(ps)
nz=cos(th)*sin(ps)
ax=cos(ph)*sin(th)*cos(ps)+sin(ph)*sin(ps)
ay=sin(ph)*sin(th)*cos(ps)-cos(ph)*sin(ps)
az=cos(th)*cos(ps)
R=matrix([[sx, nx, ax], [sy, ny, ay], [sz, nz, az]])
P1=R*PB1+PC
P2=R*PB2+PC
P3=R*PB3+PC
L1=P1-B1
L2=P2-B2
L3=P3-B3
L4=PC-B4
L5=PC-B5
```

The script for calculation and plotting the schematic is:

```
q1=sqrt(L1.dot_product(L1))
q2=sqrt(L2.dot_product(L2))
q3=sqrt(L3.dot_product(L3))
q4=sqrt(L4.dot_product(L4))
q5=sqrt(L5.dot_product(L5))

n(vector([q1,q2,q3,q4,q5]))
plt=polygon([P1, P2, P3],color='red')
bas=polygon([B1, B2, B3], color='green')
lb1=line([P1,B1])
lb2=line([P2,B2])
lb3=line([P3,B3])
lb4=line([PC,B4])
lb5=line([PC,B5])
crb=circle([0,0,0],rb)
show(crb+plt+bas+lb1+lb2+lb3+lb4+lb5)
```

B. Forward Geometric Model

In the equations 22-23, we show that there are two solutions sets; in the positive and negative z. We take first the positive, and then apply the same method for the negative one.

$$x_2 = -\frac{\begin{pmatrix} 13\sqrt{3}D^2 - 18\sqrt{3}q_1^2 - 18\sqrt{3}q_2^2 - \\ 18\sqrt{3}Dx_1 + 36\sqrt{3}x_1x_c - 36\sqrt{3}x_c^2 - \\ 36\sqrt{3}z_3z_c + 72\sqrt{3}z_c^2 - 27Dy_2 \end{pmatrix}}{9\left(\sqrt{3}D + 4\sqrt{3}x_c\right)} \quad (39)$$

$$x_3 = \frac{\begin{pmatrix} 13\sqrt{3}D^2 - 18\sqrt{3}q_1^2 - 18\sqrt{3}q_2^2 - \\ 27\sqrt{3}Dx_1 + 27\sqrt{3}Dx_c + 72\sqrt{3}x_c^2 - \\ 36\sqrt{3}z_3z_c + 72\sqrt{3}{z_c}^2 - 27Dy_2 \end{pmatrix}}{9\left(\sqrt{3}D + 4\sqrt{3}x_c\right)} \quad (40)$$

$$y_1 = -\frac{\sqrt{3\begin{pmatrix} 12\sqrt{3}Dy_2 - 13D^2 + 12q_1^2 + \\ 12q_2^2 + 12q_3^2 + 18Dx_1 - \\ 18Dx_c - 36x_c^2 - 36z_c^2 \end{pmatrix}}}{18D} \quad (41)$$

$$y_3 = \frac{\sqrt{3\begin{pmatrix} 6\sqrt{3}Dy_2 - 13D^2 + 12q_1^2 + \\ 12q_2^2 + 12q_3^2 + 18Dx_1 - \\ 18Dx_c - 36x_c^2 - 36z_c^2 \end{pmatrix}}}{18D} \quad (42)$$

$$z_1 = -\frac{\begin{pmatrix} 13D^2 - 36q_1^2 - 36Dx_1 + \\ 72x_1x_c - 36x_c^2 - 36z_c^2 \end{pmatrix}}{72z_c} \quad (43)$$

$$z_2 = \frac{\begin{pmatrix} 13D^2 - 36q_1^2 - 36Dx_1 + \\ 72x_1x_c - 36x_c^2 - 72z_3z_c + 180z_c^2 \end{pmatrix}}{72z_c} \quad (44)$$

The six equations system identified as eq1s, eq2s, eq3s, eq4s, eq5s, and eq6s, solved and stored the results in sx1y2z3 equation:

```
sx1y2z3=solve([eq1s, eq2s, eq3s, eq4s, eq5s, eq6s], x2, x3, y1, y3, z1, z2)
```

The resulting equations are in terms of x1, y2, and z3.

The identifiers of the equations 36-38 are eqp1, eqp2, and eqp3. Substituting the solutions in equations eqp11, eqp21, eqp31, respectively:

```
eqp11= eqp1.substitute(x2s,x3s,y1s,y3s,z1s,z2
eqp21= eqp2.substitute(x2s,x3s,y1s,y3s,z1s,z2
eqp31= eqp3.substitute(x2s,x3s,y1s,y3s,z1s,z2
```

We got three nonlinear equations in terms of x_1, y_2 and z_3.
Two solution methods in SageMath were applied: Numerical and the Toy Buchberger algorithm for Gröbner basis.

The numerical method is simple, but sometimes the solution diverges.

```
solt=solve([eqp11,eqp21,eqp31],x1,y2,z3)
```

Using the Gröbner basis method, we take the left side expression and redefine the variables as a polynomial in x, y, and z:

```
R1, (x, y, z) = PolynomialRing(RationalField(),3, 'xyz').objgens()
eqp11s=eqp11.lhs().subs(x_1=x,y_2=y,z_3=z)
eqp21s=eqp21.lhs().subs(x_1=x,y_2=y,z_3=z)
eqp31s=eqp31.lhs().subs(x_1=x,y_2=y,z_3=z)
```

Then, define the ideals with Gröbner fan function, we got three functions:

```
I = (eqp11s, eqp21s, eqp31s)*R1
g = I.groebner_fan()
F=g.buchberger()
fun0=F[0]
fun1=F[1]
fun2=F[2]
```

The first function is in z variable terms, we redefine as a polynomial in a zv variable and numerically find the roots. Finally, we replace in the two solutions, in terms of xv, and y:

```
R.<zv> = PolynomialRing(QQ)
fun0zv=fun0.subs(z=zv)
solgz1=find_root(fun0zv,0,z_c+D)
```

```
solgy1=n(-fun1.subs(z=solgz1)+y)
solgx1=n(-fun2.subs(z=solgz1)+x)
```

Then, we got the values of x_1, y_2 and z_3. Finally, the system is solved by substitution of these values in the equations 39-44.

ACKNOWLEDGMENT

This work was made thanks to the support of exterior doctorate program from Colfuturo and Colciencias, and the Faculty of Basic Sciences and Engineering of Universidad de los Llanos, Villavicencio, Meta Colombia.

REFERENCES

[1] I. A. Bonev, *Geometric analysis of parallel mechanisms*. Citeseer, 2002.

[2] S. Briot, I. Bonev, D. Chablat, P. Wenger, y V. Arakelian, “Self-Motions of General 3-RPR Planar Parallel Robots”, *The International Journal of Robotics Research*, vol. 27, núm. 7, pp. 855–866, jul. 2008.

[3] G. Chen y J. Kang, “Development of a MATLAB Toolbox for 3-PRS Parallel Robot.”, *International Journal of Hybrid Information Technology*, vol. 7, núm. 5, 2014.

[4] Z. Chen, K. Huang, Y. Zhang, Z. Huang, y H. Ding, “Actuation singularity analysis of a 3-RPS rotational parallel mechanism”, en *2015 IFToMM World Congress*, 2015.

[5] M. H. Fatehi, A. R. Vali, M. Eghtesad, y A. A. Fatehi, “Modeling and Control of 3-PRS parallel robot and simulation based on SimMechanics in MATLAB”, en *2011 2nd International Conference on Control, Instrumentation and Automation (ICCIA)*, 2011, pp. 785–790.

[6] C. Gang y G. Shi-rong, “Kinematic Analysis of 3-UCR Parallel Robot Leg”, en *Robot Manipulators New Achievements*, InTech, 2010.

[7] M.-S. Tsai, T.-N. Shiau, Y.-J. Tsai, y T.-H. Chang, “Direct kinematic analysis of a 3-PRS parallel mechanism”, *Mechanism and Machine Theory*, vol. 38, núm. 1, pp. 71–83, ene. 2003.

[8] D. R. Walter, M. L. Husty, y M. Pfurner, “A complete kinematic analysis of the SNU 3-UPU parallel robot”, *Contemporary Mathematics*, vol. 496, p. 331, 2009.

[9] O. Altuzarra, B. Şandru, Ch. Pinto, y V. Petuya, “A symmetric parallel Schönflies-motion manipulator for pick-and-place operations”, *Robotica*, vol. 29, núm. 06, pp. 853–862, oct. 2011.

[10] S. Amine, S. Caro, P. Wenger, y D. Kanaan, “Singularity analysis of the H4 robot using Grassmann–Cayley algebra”, *Robotica*, vol. 30, núm. 07, pp. 1109–1118, dic. 2012.

[11] O. Company, S. Krut, y F. Pierrot, “Internal singularity analysis of a class of lower mobility parallel manipulators with articulated traveling plate”, *IEEE Transactions on Robotics*, vol. 22, núm. 1, pp. 1–11, feb. 2006.

[12] R. J. Cui, Z. H. Guo, Z. X. Yin, y S. S. Zhu, “Analysis and Design of Lower-Mobility Parallel Mechanism of Non-Symmetrical Based on Variable Topology Theory”, *Advanced Materials Research*, vol. 201–203, pp. 1907–1912, feb. 2011.

[13] S. Guo, Y. Fang, y H. Qu, “Type synthesis of 4-DOF nonoverconstrained parallel mechanisms based on screw theory”, *Robotica*, vol. 30, núm. 01, pp. 31–37, ene. 2012.

[14] A. Hernández, O. Altuzarra, C. Pinto, y E. Amezua, “Transitions in the velocity pattern of lower mobility parallel manipulators”, *Mechanism and Machine Theory*, vol. 43, núm. 6, pp. 738–753, jun. 2008.

[15] B. Hu y Z. Huang, “Kinetostatic model of overconstrained lower mobility parallel manipulators”, *Nonlinear Dynamics*, vol. 86, núm. 1, pp. 309–322, oct. 2016.

[16] Z. Huang y Q. C. Li, “General Methodology for Type Synthesis of Symmetrical Lower-Mobility Parallel Manipulators and Several Novel Manipulators”, *The International Journal of Robotics Research*, vol. 21, núm. 2, pp. 131–145, feb. 2002.

[17] Z. Huang y Q. C. Li, “Type Synthesis of Symmetrical Lower-Mobility Parallel Mechanisms Using the Constraint-Synthesis Method”, *The International Journal of Robotics Research*, vol. 22, núm. 1, pp. 59–79, ene. 2003.

[18] D. Kanaan, P. Wenger, S. Caro, y D. Chablat, “Singularity Analysis of Lower Mobility Parallel Manipulators Using Grassmann #x2013;Cayley Algebra”, *IEEE Transactions on Robotics*, vol. 25, núm. 5, pp. 995–1004, oct. 2009.

[19] Q. C. Li y Z. Huang, “A Family of Symmetrical Lower-Mobility Parallel Mechanisms with Spherical and Parallel Subchains”, *Journal of Robotic Systems*, vol. 20, núm. 6, pp. 297–305, jun. 2003.

[20] J. Meng, G. Liu, y Z. Li, “A Geometric Theory for Analysis and Synthesis of Sub-6 DoF Parallel Manipulators”, *IEEE Transactions on Robotics*, vol. 23, núm. 4, pp. 625–649, ago. 2007.

[21] C. Pinto, J. Corral, O. Altuzarra, y A. Hernández, “A methodology for static stiffness mapping in lower mobility parallel manipulators with decoupled motions”, *Robotica*, vol. 28, núm. 05, pp. 719–735, sep. 2010.

[22] Qinchuan Li y J. M. Herve, “Parallel Mechanisms With Bifurcation of Schoenflies Motion”, *IEEE Transactions on Robotics*, vol. 25, núm. 1, pp. 158–164, feb. 2009.

[23] Qinchuan Li y Zhen Huang, “Mobility analysis of lower-mobility parallel manipulators based on screw theory”, 2003, vol. 1, pp. 1179–1184.

[24] Y. Xu, W. Liu, J. Yao, y Y. Zhao, “A method for force analysis of the overconstrained lower mobility parallel mechanism”, *Mechanism and Machine Theory*, vol. 88, pp. 31–48, jun. 2015.

[25] J. Yu, J. S. Dai, S. Bi, y G. Zong, “Type synthesis of a class of spatial lower-mobility parallel mechanisms with orthogonal arrangement based on Lie group enumeration”, *Science China Technological Sciences*, vol. 53, núm. 2, pp. 388–404, feb. 2010.

[26] Y. Zhao, J. F. Liu, y Z. Huang, “A force analysis of a 3-RPS parallel mechanism by using screw theory”, *Robotica*, vol. 29, núm. 07, pp. 959–965, dic. 2011.

[27] S. Zhu y Z. Huang, “A Novel Method for Jacobian Analysis of Lower-Mobility Parallel Mechanism”, 2006, pp. 2139–2143.

[28] S. Briot y W. Khalil, *Dynamics of Parallel Robots*, vol. 35. Cham: Springer International Publishing, 2015.

[29] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 8.5)*. 2019.