International Journal of Hybrid Information Technology
Vol.7, No.5 (2014), pp.401-412
http://dx.doi.org/10.14257/ijhit.2014.7.5.37

Development of a MATLAB Toolbox for 3-PRS Parallel Robot

Guoqiang Chen and Jianli Kang*

Henan Polytechnic University, China
jz97cgq@sina.com, kangjl@hpu.edu.cn

Abstract

Aiming at one kind of 3-PRS parallel robot, the study develops a toolbox in MATLAB. The toolbox includes functions for forward kinematics, inverse kinematics, velocity kinematics, error analysis, schematic representation, and so on. The architecture of the 3-PRS robot is introduced firstly. The instructions of the functions, developing procedure and main algorithms are presented secondly. The toolbox encapsulates complicated mathematical formulas into the single function and provides standard inputs and outputs, which improves the reliability and makes it easy to use. Finally, an example calls the toolbox function, and verifies its correctness, reliability and convenience.

Keywords: *3-PRS parallel robot, toolbox, forward kinematics, inverse kinematics*

1. Introduction

As an important branch of the industrial robot, the parallel robot has many advantages, such as high stiffness, high accuracy, little cumulate error, large load carrying capacity and compact structure [1-7]. It has gained wide applications in all kinds of fields and attracted plenty of study. The 3-PRS parallel mechanism is very typical in the parallel robot family [4]. There are many complicated mathematical formulas in analysis on degrees of freedom, working envelope, kinematics, speed, acceleration, accuracy, and so on [8-18]. Especially, the trigonometric function computation, nonlinear equations and complex matrix make design and analysis tedious and discouraging [1, 3-5, 7-18]. The engineering software MATLAB has powerful performance in numerical computation, symbolic operation and graphical representation, so it is an efficient tool for the robot research [19]. The aim of this paper is to present a computer-aided analysis toolbox developed in MATLAB for one kind of 3-PRS parallel robot, and give key algorithms.

2. Architecture of 3-PRS Parallel Robot

The schematic representation of the 3-PRS parallel robot is shown in Figure 1 [1, 3-5,7-9]. The robot is composed of a moving platform, three limbs, three vertical rails and a fixed base. Three vertical rails vertically link to the base $B_1\ B_2\ B_3$. Moreover, $B_1\ B_2\ B_3$ form an equilateral triangle that lies on a circle with the radius R. The axis of the revolute pair R_i for $i=1,2$ and 3 is perpendicular to the prismatic pair. Each limb L_i for $i=1,2$ and 3 with the length of l_i connects the corresponding rail by a prismatic pair R_i. The moving platform and three limbs are connected by three spherical pairs b_1, b_2 and b_3. Three spherical pairs form an equilateral triangle that lies on a circle with the radius r. The cutter with the length of h is placed at the

*Corresponding Author: Jianli Kang is the corresponding author. Email: kangjl@hpu.edu.cn
This work is supported by Scientific and Technological Project of Henan Province of China with grant No. 132102210430.

ISSN: 1738-9968 IJHIT

center of the moving platform. The feed of the prismatic pair is given as H_i. Angles φ_i for $i = 1, 2$ and 3 are defined from the vertical rail to its corresponding limb L_i. As shown in Figure 1, a fixed Cartesian reference coordinate system OXYZ is located at the center O of the base $B_1B_2B_3$. X-axis and Y-axis are in the base plane $B_1B_2B_3$, X-axis points in the direction of O B_1, and Z-axis is normal to the base plane and points upward. A moving coordinate frame o_T xyz is located at the cutter point o_T. The xy plane is parallel to the moving platform $b_1b_2b_3$, x-axis points in the direction of C_1b_1, and z-axis is normal to the moving platform. The position and orientation of the cutter can be described using the coordinates $(x_{Tool}, y_{Tool}, z_{Tool})$ of the cutter point and Euler angles α, β and γ rotating about Z, Y and X axes of the fixed system. The 3-PRS parallel robot possesses 3- DOF that are rotation α about the Z-axis and β about the Y-axis, and a translational motion z_T along the Z-axis [9]. Three parasitic motions include one rotation γ, one translational motion x_T about the X-axis, and one translational motion y_T about the Y-axis. The three parasitic motions γ, x_T and y_T can be expressed using the three independent motions α, β and z_T.

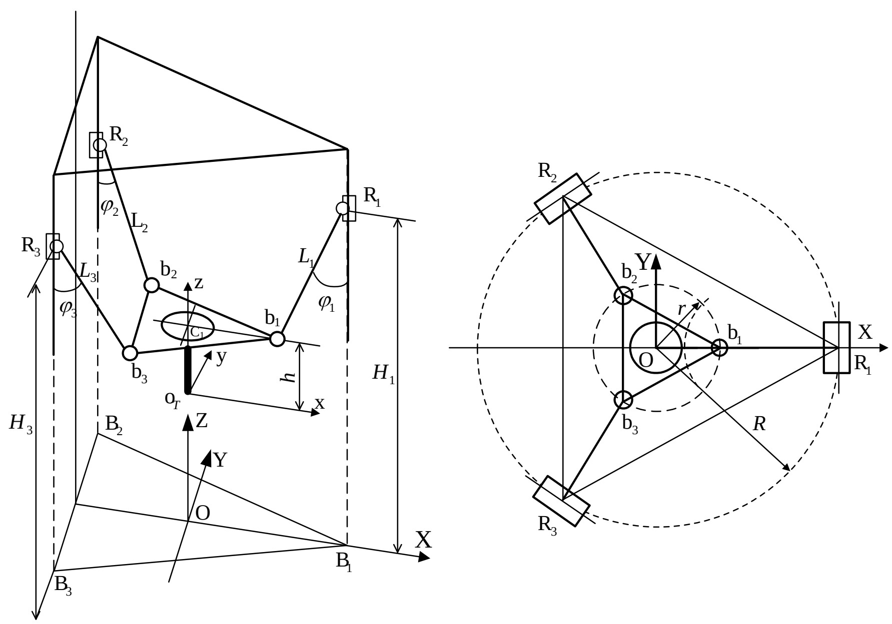


Figure 1. Schematic Representation of the 3-PRS Parallel Robot

3. Toolbox Development in MATLAB

All toolbox function names begin with the prefixion 'TPRS_'. In the toolbox functions, the parameters A, B, G, R, r, h, XT and YT represent α, β, γ, R, r, h, x_T and y_T, respectively. H is a 1-by-3 vector that represents [H_1, H_2, H_3]. TH is a 1-by-3 vector that represents [φ_1, φ_2, φ_3]. B1, B2, B3, R1, R2, R3, b1, b2 and b3 are 1-by-3 vectors of the corresponding points coordinates, respectively.

3.1. Function for Homogeneous Transformation Matrix

T= TPRS_Tran (A, B, G, P) returns a 3-by-4 array that represents the homogeneous transformation matrix computed using Equation (1) from o_T xyz to OXYZ. The input parameter P is the vector $[x_{Tool} \quad y_{Tool} \quad z_{Tool}]^T$.

$$T = \begin{bmatrix} C\beta C\gamma & -C\beta S\gamma & S\beta & x_{Tool} \\ S\alpha S\beta C\gamma + C\alpha S\gamma & -S\alpha S\beta S\gamma + C\alpha C\gamma & -S\alpha C\beta & y_{Tool} \\ -C\alpha S\beta C\gamma + S\alpha S\gamma & C\alpha S\beta S\gamma + S\alpha C\gamma & C\alpha C\beta & z_{Tool} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where $S\alpha$, $S\beta$, $S\gamma$, $C\alpha$, $C\beta$ and $C\gamma$ represent $\sin\alpha$, $\sin\beta$, $\sin\gamma$, $\cos\alpha$, $\cos\beta$ and $\cos\gamma$ respectively.

3.2. Function for Euler Angle γ

G = TPRS_AB2G (A, B) returns the Euler angle γ that can be expressed by the other two angles α and β using Equation (2).

$$\gamma = -\arctan\frac{\sin\alpha \cdot \sin\beta}{\cos\alpha + \cos\beta} \quad (2)$$

3.3. Function for Parasitic Motions x_T and y_T

[XT, YT]= TPRS_XTYT (A, B, G, r, h) returns the parasitic motions x_T and y_T that can be computed β using Equation (3).

$$\begin{cases} x_T = \frac{r}{2}(\cos\beta\cos\gamma + \sin\alpha\sin\beta\sin\gamma - \cos\alpha\cos\gamma) - h\sin\beta \\ y_T = h\sin\alpha\cos\beta - r\sin\alpha\sin\beta\cos\gamma - r\cos\alpha\sin\gamma \end{cases} \quad (3)$$

3.4. Function for Coordinates of B1, B2 and B3

[B1, B2, B3]= TPRS_BiXYZ (R) returns the coordinate vector of B1, B2 and B3 using the following equation system.

$$\begin{cases} \vec{B}_1 = [R \quad 0 \quad 0]^T \\ \vec{B}_2 = \left[-\frac{1}{2}R \quad \frac{\sqrt{3}}{2}R \quad 0\right]^T \\ \vec{B}_3 = \left[-\frac{1}{2}R \quad -\frac{\sqrt{3}}{2}R \quad 0\right]^T \end{cases} \quad (4)$$

3.5. Function for Coordinates of R1, R2 and R3

[R1, R2, R3]= TPRS_RiXYZ (H, R) returns the coordinate vector of R1, R2 and R3 using the following equation.

$$\begin{cases} \vec{R}_1 = \begin{bmatrix} R & 0 & H_1 \end{bmatrix}^T \\ \vec{R}_2 = \begin{bmatrix} -\frac{1}{2}R & \frac{\sqrt{3}}{2}R & H_2 \end{bmatrix}^T \\ \vec{R}_3 = \begin{bmatrix} -\frac{1}{2}R & -\frac{\sqrt{3}}{2}R & H_3 \end{bmatrix}^T \end{cases} \quad (5)$$

3.6. Function for Coordinates of b1, b2 and b3

[b1, b2, b3] = TPRS_ForKinbi(H, TH,PofTPRS) returns the coordinate vector of b1, b2 and b3. PofTPRS is a vector including 4 elements that represent R , l_1 , l_2 and l_3 .

The coordinate vector of b1, b2 and b3 in the system o_T xyz can be computed using Equation (6). Combined with the transformation matrix T, the coordinate vector in the system OXYZ can be gotten.

$$\begin{cases} \vec{b}_1 = \begin{bmatrix} r & 0 & h \end{bmatrix}^T \\ \vec{b}_2 = \begin{bmatrix} -\frac{1}{2}r & \frac{\sqrt{3}}{2}r & h \end{bmatrix}^T \\ \vec{b}_3 = \begin{bmatrix} -\frac{1}{2}r & -\frac{\sqrt{3}}{2}r & h \end{bmatrix}^T \end{cases} \quad (6)$$

3.7. Function for Angles between Vertical Rails and Corresponding Limbs

[TH, Flag]= TPRS_ForKinTH (H, PofTPRS, TH0) computes three feeds and angles using the optimal iteration algorithm based on based on Equation (7). PofTPRS is a 1-by-5 vector that represent r , R , l_1 , l_2 and l_3 . The initial iteration value TH0 is a vector including 3 elements, and has heavy effect on the rationality of the solution.

$$\left|\mathbf{b}_1\mathbf{b}_2\right| = \left|\mathbf{b}_1\mathbf{b}_3\right| = \left|\mathbf{b}_2\mathbf{b}_3\right| = \sqrt{3}r \quad (7)$$

3.8. Function for the Direction Vector of the z-axis

Vz= TPRS_zVectorFromb1b2b3 (b1, b2, b3) returns the direction vector of the z-axis. The direction vector can be computed using the following formula.

$$\vec{e}_3 = \frac{\overrightarrow{b_1b_2} \times \overrightarrow{b_2b_3}}{\left|\overrightarrow{b_1b_2} \times \overrightarrow{b_2b_3}\right|} = \frac{\overrightarrow{b_2b_3} \times \overrightarrow{b_3b_1}}{\left|\overrightarrow{b_2b_3} \times \overrightarrow{b_3b_1}\right|} = \frac{\overrightarrow{b_3b_1} \times \overrightarrow{b_1b_2}}{\left|\overrightarrow{b_3b_1} \times \overrightarrow{b_1b_2}\right|} \quad (8)$$

3.9. Function for the Coordinates of the Cutter Point

[XTool, YTool, ZTool]= TPRS_XTYTZTFromb1b2b3(b1,b2,b3,h,dx,dy) returns the cutter point coordinates $(x_{Tool}, y_{Tool}, z_{Tool})$. dx and dy are the offsets δx and δy along x-axis and y-axis between the center C_1 of the equilateral triangle and the perpendicular projection point C_2 of the cutter point in the equilateral triangle plane, shown in Figure 2. The algorithm can be expressed as follows.

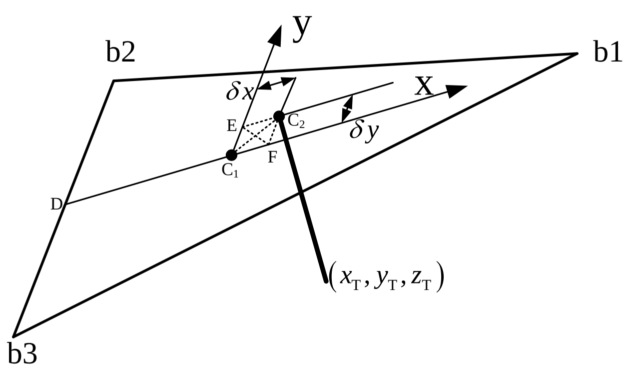


Figure 2. Algorithm Diagram for Function TPRS_XTYTZTFromb1b2b3

The coordinates (x_{C1}, y_{C1}, z_{C1}) of the center C_1 of the equilateral triangle b1b2b3 can be expressed as

$$\begin{cases} x_{C1} = \dfrac{x_{b1} + x_{b2} + x_{b3}}{3} \\ y_{C1} = \dfrac{y_{b1} + y_{b2} + y_{b3}}{3} \\ z_{C1} = \dfrac{z_{b1} + z_{b2} + z_{b3}}{3} \end{cases} \quad (9)$$

Based on the coordinates of b1 and b2, the direction vector (m, n, p) of the y-axis can be computed. The parameter equation for the y-axis line is

$$\begin{cases} x = x_{C1} + mt \\ y = y_{C1} + nt \\ z = z_{C1} + pt \end{cases} \quad (10)$$

The corresponding t of the point E is

$$t = \frac{\delta y}{\sqrt{m^2 + n^2 + p^2}} \quad (11)$$

The coordinates (x_E, y_E, z_E) of E can be computed. Likewise, the coordinates of b2b3 center D can be gotten, the direction vector of the x-axis can be computed, and the coordinates

(x_F, y_F, z_F) of the point F can be gotten. Based on coordinates of points E and F, the center coordinates of the parallelogram C_1FC_2E can be computed, and coordinates (x_{C2}, y_{C2}, z_{C2}) of the point C_2 are

$$\begin{cases} x_{C2} = x_E + x_F - x_{C1} \\ y_{C2} = y_E + y_F - y_{C1} \\ z_{C2} = z_E + z_F - z_{C1} \end{cases} \tag{12}$$

The function TPRS_zVectorFromb1b2b3 is called to compute the direction vector of the line that the cutter resides. Likewise, $(x_{Tool}, y_{Tool}, z_{Tool})$ can be computed.

3.10. Function for Three Euler Angles from the Direction Vector of the Cutter

[A, B, G]= TPRS_VectorOfTool2ABG(VectorOfTool) returns Euler angles. The input parameter VectorOfTool is a vector including three elements that represents the direction of the cutter. The 9 upper-left elements in the transform matrix T form the direction cosine matrix from o_T xyz to OXYZ. The unit vectors of the three coordinate axes of OXYZ are $\vec{E}_1 = [1,0,0]$, $\vec{E}_2 = [0,1,0]$ and $\vec{E}_3 = [0,0,1]$, while the three unit vectors of o_T xyz are $\vec{e}_1$, $\vec{e}_2$ and $\vec{e}_3$. The direction cosine matrix D can be expressed as

$$D = \begin{bmatrix} \vec{E}_1 \cdot \vec{e}_1 & \vec{E}_1 \cdot \vec{e}_2 & \vec{E}_1 \cdot \vec{e}_3 \\ \vec{E}_2 \cdot \vec{e}_1 & \vec{E}_2 \cdot \vec{e}_2 & \vec{E}_2 \cdot \vec{e}_3 \\ \vec{E}_3 \cdot \vec{e}_1 & \vec{E}_3 \cdot \vec{e}_2 & \vec{E}_3 \cdot \vec{e}_3 \end{bmatrix} \tag{13}$$

The unit vector $\vec{e}_3$ is the utilization one of VectorOfTool. The matrix elements in the third row and third row of D can be computed. The Euler angles α and β are computed using Equation (1) and (2), then the Euler angle γ is gotten by calling the function TPRS_AB2G.

3.11. Function for the Inverse Kinematics

[H, TH] = TPRS_BacKin (A, B, ZT, PofTPRS) returns three feeds and three angles. PofTPRS is a vector including 6 elements that represent r, R, l_1, l_2, l_3 and h. The initial iteration value TH0 is a vector including 3 elements, and has heavy effect on the rationality of the solution. The Euler angle γ is computed using TPRS_AB2G, and the transformation matrix T is computed using TPRS_Tran. The coordinates of b1, b2 and b3 in the system o_T xyz can be expressed as $[x_{bi}, y_{bi}, z_{bi}, 1]^T$ $(i = 1, 2, 3)$, and the corresponding coordinates in the system OXYZ can be computed using the transformation matrix T. The feeds of the three prismatic pairs H_i $(i = 1, 2, 3)$ are

$$H_i = Z_{Ri} = Z_{bi} + \sqrt{l_i^2 - (X_{Ri} - X_{bi})^2 - (Y_{Ri} - Y_{bi})^2} \qquad (i=1, 2, 3) \tag{14}$$

where Z_{Ri} is the Z coordinate of R_i, l_i is the length of limb L_i, X_{Ri} and X_{bi} are the X coordinates of R_i and b_i respectively, and Y_{Ri} and Y_{bi} are Y coordinates of R_i and b_i respectively.

3.12. Function for all Solutions of Angles Using Monte Carlo Simulation

[TH,N]= TPRS_ForKinAllTH (H, PofTPRS, THMin, THMax, K, M, Error) returns all solutions of three angles between vertical rails and corresponding limbs using Monte Carlo simulation. The return value N is the number of the solutions. The input parameter PofTPRS has the same meaning as the function TPRS_BacKin, [THMin, THMax] is the interval where the solution resides, K is the number of Monte Carlo sampling, and M is the number of the possible solutions. The solutions are regarded as one if the absolute value of their difference is no more than the limiting error Error. The return number of the solutions N is no more than the set number M.

3.13. Function for the Position and Orientation Error

Error= TPRS_ManError (A, B, ZT, PofTPRS, EPofTPRS) returns the position and orientation error. The return value Error is a vector including 6 elements that represent the errors of the three Euler angles and position coordinates. The input parameter PofTPRS has the same meaning as the function TPRS_BacKin. EPofTPRS is a vector including 11 elements that represent moving platform radius error δr, fixing base radius error δR, three limb-length-errors $\delta l_1, \delta l_2, \delta l_3$, cutter length error δh, three feeds errors $\delta H_1, \delta H_2, \delta H_3$, offsets δx and δy. The computation procedure is as follows.

Step 1: Three feeds H and three angles TH0 are computed using [H, TH0]= TPRS_BacKin (A, B, ZT, PofTPRS).

Step 2: The three angles are computed again using [TH, Flag]= TPRS_ForKinTH (H+EPofTPRS(7:9), PofRobot(1:5)+EPofTPRS(1:5), TH0).

Step 3: The three-dimensional coordinates of b1, b2 and b3 are computed using [b1, b2, b3]= TPRS_ForKinbi(H+EPofTPRS(7:9), TH, PofTPRS(2:5)+ EPofTPRS(2:5)).

Step 4: The direction vector of the z-axis is computed using Vz= TPRS_zVectorFromb1b2b3 (b1, b2, b3).

Step 5: Three Euler angles A, B and G are computed using [A, B, G]= TPRS_VectorOfTool2ABG (-Vz).

Step 6: The coordinates $(x_{\mathrm{Tool}}, y_{\mathrm{Tool}}, z_{\mathrm{Tool}})$ are computed using [XTool, YTool, ZTool] = TPRS_XTYTZTFromb1b2b3 (b1, b2, b3, EPofTPRS(6), EPofTPRS(10), EPofTPRS(11)).

Step 7: The error Error is computed using the actual and nominal values.

3.14. Function for the Jacobian Matrix

Jacobian = TPRS_ Jacobian (A, B, ZT, PofTPRS, Flag) computes the Jacobian matrix according to the position and orientation, and the structure parameter. The input parameter PofTPRS has the same meaning as the function TPRS_BacKin. If Flag is 1, the function returns the Jacobian matrix; else, the inverse Jacobian matrix. The return value Jacobian is a 3-by-3 matrix. The 3-PRS parallel robot possesses 3- DOF α, β and z_{T}. Based on the Equation (1) and (14), three feeds are functions of α, β and z_{T}. The α, β and z_{T} are functions of the time t, and can be expressed as $\alpha(t)$, $\beta(t)$ and $z(t)$. So Equation (14) can be expressed as

$$H_i(t) = F_i\left(\alpha(t), \beta(t), Z(t)\right) \tag{15}$$

So

$$\begin{bmatrix} \frac{dH_1}{dt} & \frac{dH_2}{dt} & \frac{dH_3}{dt} \end{bmatrix}^T = J \begin{bmatrix} \frac{d\alpha}{dt} & \frac{d\beta}{dt} & \frac{dZ}{dt} \end{bmatrix}^T \quad (16)$$

where the Jacobian matrix J is

$$J = \begin{bmatrix} \frac{\partial H_1}{\partial \alpha} & \frac{\partial H_1}{\partial \beta} & \frac{\partial H_1}{\partial Z} \\ \frac{\partial H_2}{\partial \alpha} & \frac{\partial H_2}{\partial \beta} & \frac{\partial H_2}{\partial Z} \\ \frac{\partial H_3}{\partial \alpha} & \frac{\partial H_3}{\partial \beta} & \frac{\partial H_3}{\partial Z} \end{bmatrix} \quad (17)$$

If J has full rank, the inverse Jacobian matrix can be computed. Because of the complexity of Equations (15) to (17), the Jacobian matrix function jacobian (f, v) can be used directly.

3.15. Function for the Acceleration Computation

aH = TPRS_ Acceleration (A, B, ZT,VABX, aABX, PofTPRS) computes the acceleration of three feeds. PofTPRS has the same meaning as the function TPRS_BacKin. VABX and aABX are the velocity and acceleration vectors of the cutter direction and position, and each includes 3 elements. The return value aH is a vector including 3 elements representing the acceleration of three feeds. The transmission relationship can be gotten through differentiation using Equation (16).

3.16. Function for the Structure Diagram Drawing

TPRS_ Plot(B1,B2,B3,R1,R2,R3,b1,b2,b3) plots the scheme of the 3-PRS robot. The function TPRS_XTYTZTFromb1b2b3 is called to compute the coordinates of the cutter point, and the MATLAB function plot3 is used to draw the scheme.

4. Example Result and Discussion

The structure parameters of a 3-PRS parallel robot are as follows. $l_1 = l_2 = l_3 = 1107$, $r = 200$, $R = 350$ and $h = 280$. The trajectory path of the cutter point is the intersection of the spherical surface and the plane $z = z_0$, and the cutter is always perpendicular to the spherical surface. The trajectory path of the cutter point can also be expressed as

$$\begin{cases} x = \sqrt{R_s^2 - (R_s - z_0)^2} \cos t \\ y = \sqrt{R_s^2 - (R_s - z_0)^2} \sin t \qquad (0^0 \le t < 360^0) \\ z = z_0 \end{cases}$$

The direction vector of the cutter is $-(x, y, z_0)$, and the direction vector of the z-axis is (x, y, z_0) . The toolbox developed can be used to analyze the characteristics. If $R_s = 100$, $z_0 = 160$ and $t = 60^0$, the computation procedure and key codes in MATLAB are as follows.

```
>> dRsz0=sqrt(100*100-(100-160)*(100-160));
```



```
>> x=dRsz0*cos(pi/3); y=dRsz0*sin(pi/3);                % Prepare parameters
>> [A,B,G]= TPRS_VectorOfTool2ABG(-[x,y,160]) ;         % Compute three Euler angles
>> [H,TH]= TPRS_BacKin(A,B,160,[200,350,1107,1107,1107,280]);
                                                        % Compute feeds and angles
>> [R1,R2,R3]= TPRS_RiXYZ(H,350);   % Compute coordinates
>> [b1,b2,b3]= TPRS_ForKinbi(H, TH,[350,1107,1107,1107]);
                                                        % Compute coordinates
>> [B1,B2,B3]= TPRS_BiXYZ(350);                         %Compute coordinates
>> TPRS_Plot(B1,B2,B3,R1,R2,R3,b1,b2,b3,280);           % Plots the scheme
```

The feed vector of the three prismatic pairs is H=[1462.50857, 1462.50857,1591.87333], the three angles vector between vertical rails and corresponding limbs is TH =[0.1359, 0.1359, 0.1649], and the scheme of the robot is is shown in Figure 3.

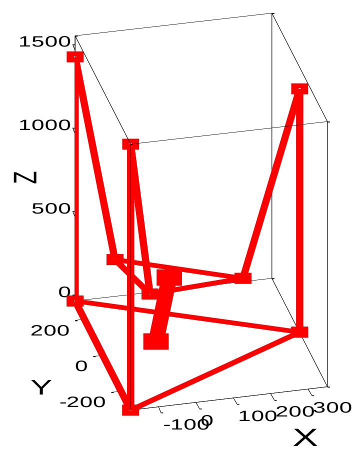
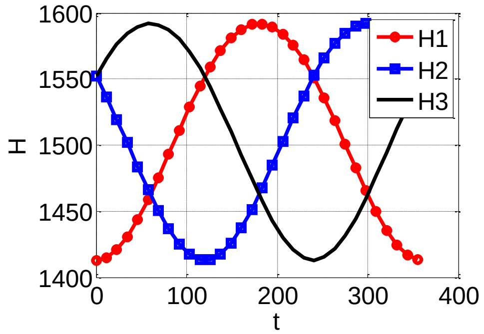
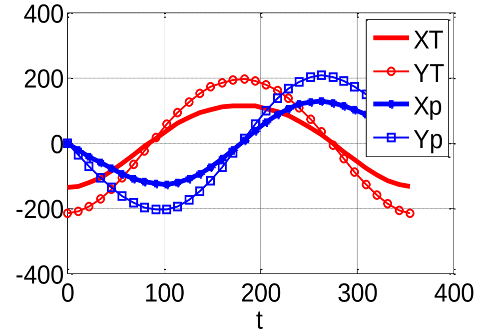


Figure 3. Scheme of the Robot Position and Orientation for Position and Orientation

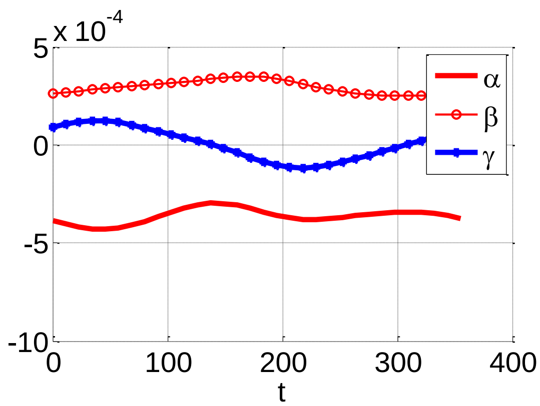
The parameter t is made discrete over the interval $[0^\circ, 360^\circ)$. Each discretization value is computed, and the result is shown in Figure 3. TPRS_BacKin is used for Figure 4 (a), TPRS_XTYT for Figure 4 (b), and TPRS_ManError for Figure 4(c) and (d). The error vector of the 11 parameters is [0.0068, 0.0770, 0.0798, 0.0250, -0.0726, -0.0566, -0.0636, -0.0918, -0.0788, 0.0232, 0.0878] in Figure 4(c) and (d). The three position-coordinates errors and three Euler angles errors differ greatly in value although the error source is same, which demonstrates that the position and orientation has tremendous effect on the error sensitivity. The translational motion of the 3-PRS robot possesses is only one z_T along the Z-axis, and it has two parasitic motions x_T and y_T , as shown in Figure 4(b). An X-Y table is needed to compensate the parasitic motions x_T and y_T , and the compensations Xp and Yp are shown in Figure 4 (b).



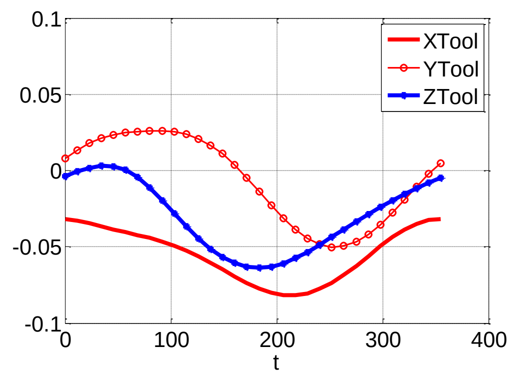
a) Feeds of Three Prismatic Pairs



b) Parasitic Motions and Compensations



c) Three Euler Angles Errors



d) Position Errors

Figure 4. Computing Results of the Example

5. Conclusions

A toolbox for the 3-PRS parallel robot is developed in MATLAB and key algorithms are given. The toolbox includes 16 functions for forward kinematics, inverse kinematics, velocity kinematics, error analysis, schematic representation of the robot, and so on. Finally, an example calls the toolbox function, and verifies its correctness, reliability and convenience. The toolbox is very useful for design and analysis of the 3-PRS robot characteristic. The developed toolbox and its application have several advantages. The toolbox encapsulates complicated mathematical formulas into the single function and provides standard inputs and outputs, which improves the reliability and makes it easy to use. The functions are saved as m-files and all file names end with the extension '.m', so it is helpful and almost necessary for the user to modify the codes for expansion. However, it should be noticed that the 3-PRS robot can be classified into four categories including seven kinds according to limb arrangements as discussed in [9]. Based on the toolbox here, the function for the other kinds can be extended.

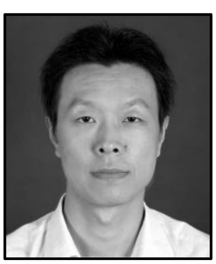
Acknowledgements

This work is supported by Scientific and Technological Project of Henan Province of China with grant No. 132102210430. The authors would like to thank the anonymous reviewers for their valuable work.

References

[1] W. Liu and S. Chang, "Study on structural design of a parallel robot and applications in automobiles", Machinery Design & Manufacture, no. 6, **(2012)**, pp. 141-143.
[2] Z. Huang, "Theory of parallel robot mechanism", 20 Anniversary of Mechanical Science Fundamental Study. Edited by. Y. Lei, Wuhan China: Wuhan University of Technology Press, **(2007)**.
[3] J. Zhao, "Research on the accuracy theory and the measurement device of serial-parallel machine tools", Dotoral dissertation, Wuhan China: Huazhong University of Science and Technology, **(2001)**.
[4] X. Liu, C. Wu , J. Wang and I. BONEV, "Attitude description method of [PP]S type parallel robotic mechanisms", Chinese Journal of Mechanical Engineering, vol. 44, no. 10, **(2008)**, pp. 19-23.
[5] M. S. Tsai and W. H. Yuan, "Dynamic modeling and decentralized control of a 3 PRS parallel mechanism based on constrained robotic analysis", Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 63, no. 3-4, **(2011)**, pp. 525-545.
[6] C. Gong, J. Xiong and C. Huang, "Kinematics and dynamics simulation of Delta parallel robot", Manufacturing Automation, vol. 35, no. 3, **(2013)**, pp. 5-7&14.
[7] G. Abbasnejad, S. Zarkandi and M. Imani, "Forward kinematics analysis of a 3-PRS parallel manipulator", World Academy of Science, Engineering and Technology, vol. 37, **(2010)** January, pp. 329-335.
[8] M. S. Tsai, T. N. Shiau, Y. J. Tsai and T. H. Chang, "Direct kinematic analysis of a 3-PRS parallel mechanism", Mechanism and Machine Theory, vol. 38, no. 1, **(2003)**, pp. 71-83.
[9] Q. Li, Z. Chen, Q. Chen, C. Wu and X. Hu, "Parasitic motion comparison of 3-PRS parallel mechanism with different limb arrangements", Robotics and Computer-Integrated Manufacturing, vol. 27, no. 2, **(2011)**, pp. 389-396.
[10] J. Yu, Y. Hu, S. Bi, G. Zong and W. Zhao, "Kinematics feature analysis of a 3-DOF compliant mechanism for micro manipulation", Chinese Journal of Mechanical Engineering, vol. 17, no. 1, **(2004)**, pp. 127-131.
[11] J. A. Carretero, R. P. Podhorodeski, M. A. Nahon and C. M. Gosselin, "Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator", Journal of Mechanical Design, Transactions of the ASME, vol. 122, no. 1, **(2000)**, pp. 17-24.
[12] P. Huang, J. Wang, L. Wang and J. Qian, "Dimensional synthesis for 3-PRS mechanism based on identifiability performance", Chinese Journal of Mechanical Engineering, vol. 25, no. 2, **(2012)**, pp. 234-240.
[13] K. H. Hunt, "Structural kinematics of in-parallel-actuated robot-arms", Journal of Mechanisms, Transmissions, and Automation in Design, vol. 105, no. 4, **(1983)**, pp. 705-712.
[14] Y. Li and Q. Xu, "Kinematic analysis of a 3-PRS parallel manipulator", Robotics and Computer-Integrated Manufacturing, vol. 23, no. 4, **(2007)**, pp. 395-408.
[15] D. Liu, R. Che, Z. Li and X. Luo, "Research on the theory and the virtual prototype of 3-DOF parallel-link coordinate-measuring machine", IEEE Transactions on Instrumentation and Measurement, vol. 52, no. 1, **(2003)**, pp. 119-125.
[16] N. A. Pouliot, C. M. Gosselin and M. A. Nahon, "Motion simulation capabilities of three-degree-of-freedom flight simulators", Journal of Aircraft, vol. 35, no. 1, **(1998)**, pp. 9-17.
[17] A. A. Selvakumar, R. P. Babu and R. Sivaramakrishnan, "Simulation and singularity analysis of 3-PRS parallel manipulator". Proceedings of the 9th IEEE International Conference on Mechatronics and Automation, Chengdu China, **(2012)** August 5-8, pp. 2203-2207.
[18] J. Wahl, 2000. Articulated tool head, US Patent 6,431,802, **(2002)** August 13.
[19] T. Yu, D. Li and B. Song, "Research on trajectory planning and simulation of industrial robots based on MATLAB-Robotics toolbox", Mechanical Engineer, no. 7, **(2011)**, pp. 81-83.

Author



Guoqiang Chen, is currently an associate professor at School of Mechanical and Power Engineering of Henan Polytechnic University, China. His research interests include robot technology, electric vehicle control, motor control, pulse width modulation technology and software design.