

Kinematics simulation and control system design of the three DOF parallel mechanism

Bin-cheng Li^{1,a}, Tian Yu^{2,b} and Ping Liu^{1,c}

¹Mengxi Road NO.2 Jingkou District Zhenjiang City Jiangsu Province China

^alibincheng2000@126.com, ^b610013170@qq.com, ^c260763614@qq.com

Keywords: 3-DOF parallel mechanism; kinematics simulation; LinuxCNC

Abstract. In this paper, we designed a novel three degree of freedom (DOF) parallel mechanism (PM), which can make the moving platform rotate and incline through the movement of three sliders. We built a model of this mechanism and had a kinematics analysis in MATLAB. We also did the same kinematics simulation in ADAMS, so we compared with the two curves to verify the correction of the mathematical model. Making use of the LinuxCNC, we designed the control system for this mechanism.

Introduction

In recent years, with a extensive research of the parallel mechanisms (PMs), more and more PM configurations have been invented to meet the movement we need. A lower-mobility PM, especially 3-DOF PM has much characteristic, such as better dynamic performance, high stiffness, and doing well in different working environment, which has gradually been a hot research[1].

Because of the complexity of forward kinematics, researchers may choose inverse kinematics to get the kinematics characteristic. It is efficient to utilize dynamics simulation software to intuitively observe the process of simulation[2].

Kinematics analysis of the 3-DOF PM

Model building. The three- dimensional model is shown in Fig. 1. This PM is composed of the base platform connected with a moving platform through four kinematic chains. The central kinematic chain is fixed on the base platform at one end and connected to the moving platform using a spherical joint at the other end. The three outer kinematic chains connect the moving platform using universal joints and the three sliders using spherical joints and connecting rods with fixed-length. Each slider is driven by a stepper motor via a lead screw. The center point of the moving platform is fixed, so the moving platform could be rotate and incline with the sliders making linear movement.

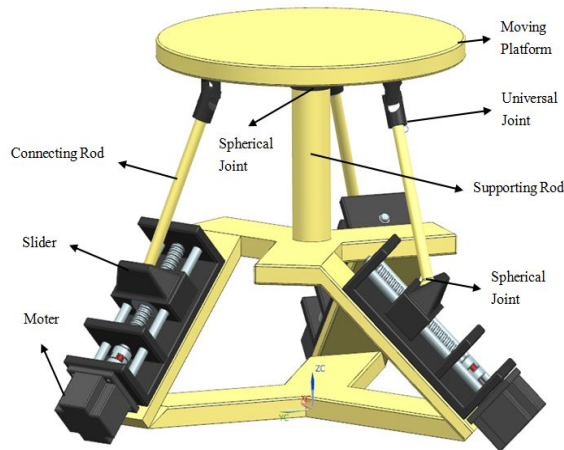


Figure 1. 3-rotational-DOF parallel mechanism

3-DOF PM kinematics position analysis. The PM kinematic position analysis is consist of forward and inverse kinematics. The forward kinematics of the model is that we give the distance the sliders move and then, we can get the position of the moving platform in the Cartesian coordinate system. On the contrary, the inverse kinematics is that we give the moving platform corresponding motion, then we get the distance the sliders move.

The kinematics model of the PM is shown in Fig. 2. With the knowledge of space analytic geometry and Euler coordinate transformation, this inverse kinematics could be solved.

Give a simple motion to the moving platform: Rotate 10 degrees around X, Y, Z axis in the moving coordinate system O_1 -xyz, respectively. Then, the distance B_iC_i ($i=1,2,3$) in the slope will be solved in MATLAB.

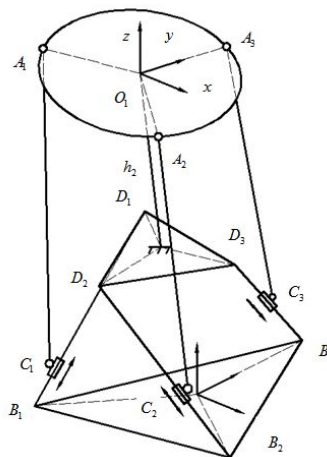


Figure 2. The kinematics model of the PM

Firstly, the following coordinate systems are defined for the model: the reference coordinate system O -xyz is attached to the center of the base platform, and the moving coordinate system O_1 -xyz is located at the center of the moving platform.

Both $\Delta B_1B_2B_3$ and $\Delta D_1D_2D_3$ are equilateral triangles. The height of the base platform and the central strut fixed on the base platform are h_1 , h_2 , respectively. Additionally, the radius of the moving platform is R , and the length of the connecting rods are L .

As shown in Fig .3, the side of $\Delta B_1B_2B_3$ is d . Vertex coordinates in $\Delta B_1B_2B_3$ are following:

$$B_1 = \left(-\frac{d}{2}, -\frac{\sqrt{3}}{6}d, 0\right), B_2 = \left(\frac{d}{2}, -\frac{\sqrt{3}}{6}d, 0\right), B_3 = \left(0, -\frac{\sqrt{3}}{3}d, 0\right)$$

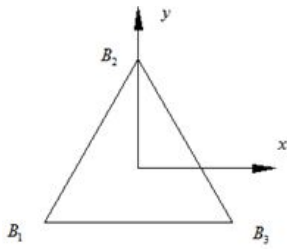


Figure.3 The Vertex coordinates of $\Delta B_1 B_2 B_3$

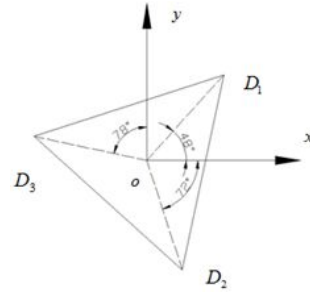


Figure.4 The Vertex coordinates of $\Delta D_1 D_2 D_3$

$$D_1 = (-r \sin 78^\circ, r \cos 78^\circ, h_1) \quad D_2 = (r \cos 72^\circ, -r \sin 72^\circ, h_1) \quad D_3 = (r \cos 48^\circ, r \sin 48^\circ, h_1)$$

$$A_1 = \left(-\frac{\sqrt{3}}{2}R, -\frac{R}{2}, 0\right), \quad A_2 = \left(\frac{\sqrt{3}}{2}R, -\frac{R}{2}, 0\right), \quad A_3 = (0, R, 0)$$

If want to transform the moving coordinate to the reference coordinate system, we need to have a Euler coordinate transformation first. As we all known, the position of the moving platform is represented by a position vector $T = [X_p, Y_p, Z_p]$ and a group of Euler angle α, β, γ . So, we assume the motion law: rotate X axis, then Y axis, last Z axis. The rotation matrix R will be the product of the rotation matrix of X,Y,Z axis. As follows[3]:

$$R = R_\alpha * R_\beta * R_\gamma \tag{1}$$

$$R_\beta = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad R_\gamma = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_\alpha = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The formula of transforming the moving coordinate to the reference coordinate system is follows:

$$\begin{bmatrix} X_{Ai} \\ Y_{Ai} \\ Z_{Ai} \end{bmatrix} = R * \begin{bmatrix} x_{Ai} \\ y_{Ai} \\ z_{Ai} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ H \end{bmatrix} \tag{3}$$

Taking $B_1 C_1$ as example:

Due to the coordinates of point B_1, D_1 , we can get the direct vector $S_1 = (m_1 \ n_1 \ p_1)$ and the equation of line $B_1 D_1$:

$$\frac{x - X_{B1}}{m_1} = \frac{y - Y_{B1}}{n_1} = \frac{z - Z_{B1}}{p_1} \tag{4}$$

The distance of point A_1 to the line $B_1 D_1$:

$$\begin{cases} m_1(x - X_{A1}) + n_1(y - Y_{A1}) + p_1(z - Z_{A1}) = 0 \\ \frac{x - X_{B1}}{m_1} = \frac{y - Y_{B1}}{n_1} = \frac{z - Z_{B1}}{p_1} \end{cases} \tag{5}$$

Then, we can get the coordinate of point E_1 , and length of $A_1 E_1$ [4].

According to the Pythagorean proposition and the length of the connecting rods, we can get the length of C_1E_1 :

$$C_1E_1 = \sqrt{l^2 - A_1E_1^2} . \text{ So, } B_1C_1 = B_1E_1 - C_1E_1 . \quad (6)$$

To sum up, based on the above method, the other two chains can be calculated.

Inverse kinematics analysis in Matlab

Table 1 Parameters about the PM mechanism

Main Parameters	d	r	L	h1	h2	R
values	370 [mm]	112 [mm]	280 [mm]	150 [mm]	200 [mm]	110 [mm]

Then, rotate 10 degrees around X, Y, Z axis in $O_1 - xyz$, respectively. The distance BiCi of the sliders moved will be easily calculated in MATLAB in Fig.5.

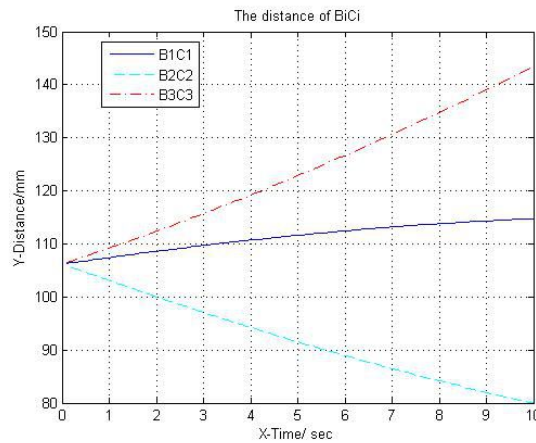


Figure 5. The Length of BiCi in MATLAB

In this picture, the transverse axis is represented time and the longitude one stands for distance.

Inverse kinematics analysis in Adams

We will verify the correction of the inverse kinematics analysis through comparing with the curves in Fig. 5 and Fig. 6. Export the model in UG and import it to ADAMS. Add up kinematic pairs to the model and the drives to the moving platform in order to constrain the degree of the model. The drives in the moving platform are as follows:

MOTION1: X=10d*time MOTION2: Y=10d*time MOTION3: Z=10d*time (d represents degree)

Before having a kinematics analysis, we should verify the degrees of the model. It must be zero after adding up the kinematic pairs and the drivers. So, the results are as follows:

VERIFY MODEL: .MODEL_150825 0 Gruebler Count (approximate DOF)

11 Moving Parts (not including ground) 4 Spherical Joints 3 Universal Joints 4 Fixed Joints

3 Motion 0 Degrees of Freedom for .MODEL_150825

There are no redundant constraint equations. Model verified successfully

After this, modifying the value of time in 10 seconds in the simulation interface, we can have a kinematics analysis in Adams. The result of simulation is as follows:

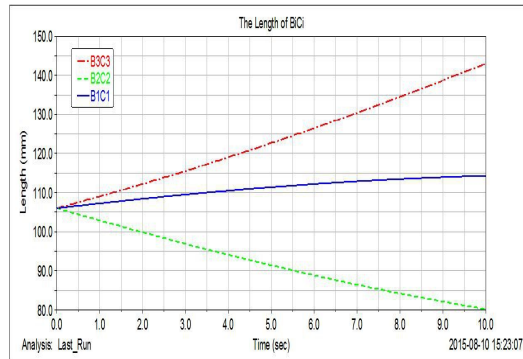


Figure 6. The Length of BiCi in ADAMS

In conclusion, comparing with the curves in Fig. 5 and Fig. 6, the variation of the sliders moved are the same. So, the mathematical model built is verified successfully.

Control system design

The overall plan of the control system. The plan contains the hardware and the software parts. The hardware is consist of the PC, parallel interface board, the stepper controllers and the power supplies, the stepper motors and the lead screws. The way to connect them with wires is shown in Fig. 7.

In the entire control system, the software embedded in the PC is the center of the control system. Debian is the software platform of the system. Software LinuxCNC is applied to substitute for exclusive CNC controller, which can display the real-time position data and the status of each axis[5].

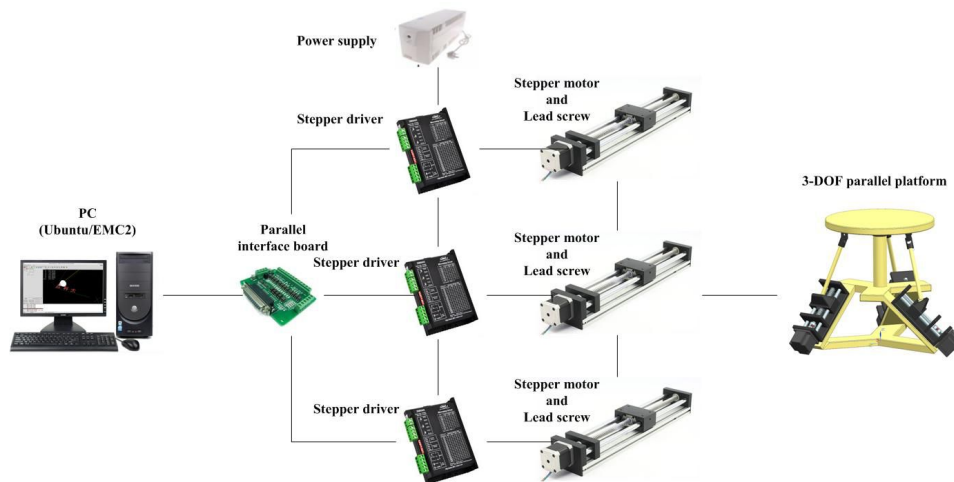


Figure 7. The hardware connection diagram of the 3-DOF PM

The modification of the configuration files in LinuxCNC. LinuxCNC is composed of four parts, which are the EMCMOT, EMCIO, EMCTASK and GUI. When LinuxCNC running, EMCTASK interprets the motion orders that the uses typed to the standard one. Then, the EMCIO calculates the coordinates of the sliders based on the standard orders. Last, LinuxCNC turns the coordinates to control signals in order to write them in the stepper drivers through the parallel interface board. Therefore, We can realize the movement as long as the corresponding configuration files and the EMCMOT module can be successfully modified and compiled.

We utilized the tripod interface to simulate the result of controlling, which I revised the inverse kinematic function and the HAL file. Then, make file[6]. Type the G code in the MDI mode. It showed that the data are the same as which in the matlab. We conclude that the LinuxCNC can be used to combine with the hardware devices.

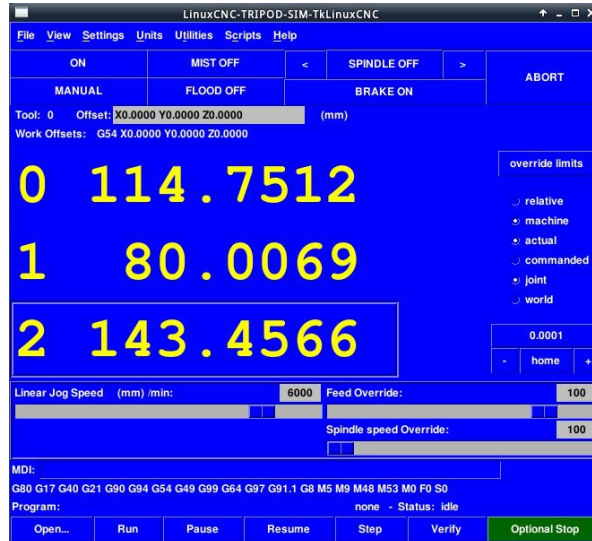


Figure 8. The simulation tripod interface of the 3-DOF PM

Summary

The 3-DOF PM can be applied to many occasions, such as the fixture platform, due to the motion of inclining and rotating. Next, we will make the model into real through buying the materials, machining, assembling and controlling. With the help of LinuxCNC and matlab, the controlling and kinematics analysis of the PM will be more easier.

Reference

- [1] J. Schadlbauer, D.R. Walter, M.L. Husty, The 3-RPS parallel manipulator from an algebraic viewpoint, *Mechanism and Machine Theory* 75 (2014) 161–176.
- [2] Enrique Cuan-Urquizo, Ernesto Rodriguez-Leal, kinematics analysis of the 3-CUP parallel mechanism, *Robotics and Computer-Integrated Manufacturing* 29 (2013) 382–395.
- [3] Wang Hua, Pan Baozhu, Matrix of space coordinate transformation, *Academic research of shijiazhuang institute of technology*, Vol.9 No.3 (2014) 4-6.
- [4] Gao Zunhai, A formula of the distance between point and a line in space, *Studies in college mathematics* Vol.8, No.2 (2005) 4-5.
- [5] EMC overview [EB/OL]. <http://www.tuxcnc.org/pivot/entry/.php?id=5#body>.
- [6] Information on <http://www.linuxcnc.org>.