International Journal of Multidisciplinary Research and Growth Evaluation.

Solving the kinematic problem for parallel 6 degrees of freedom hexapod system applied in azimuth and elevation observation systems by using virtual coordinate system

Vu Duc Tuan
Control, Automation in Production and Improvement of Technology Institute (CAPITI), Academy of Military Science and Technology (AMST), Hanoi, Vietnam

* Corresponding Author: **Vu Duc Tuan**

Article Info

ISSN (online): 2582-7138
Impact Factor: 5.307 (SJIF)
Volume: 05
Issue: 02
March-April 2024
Received: 05-02-2024;
Accepted: 09-03-2024
Page No: 346-353

Abstract
For parallel robot systems, due to limitations in direction angle, to calculate kinematic problem using the usual method, we have to convert from 2 angles - azimuth and elevation to 3 angles - roll, pitch, yaw. The solution to this conversion problem is the arcsin and arccos functions, making the kinematic problem more complicated, especially research problems on dynamics. This article proposes a simple solution for solving kinematic problem of 6-degree-of-freedom parallel systems applied in azimuth and elevation observation systems (AEOS) by using virtual coordinate systems, helping to minimize system calculation parameters and bring convenience for future control problems. Research results are calculated, simulated, and tested on Matlab.

DOI: https://doi.org/10.54660/.IJMRGE.2024.5.2.346-353

Keywords: parallel robot, hexapod, kinematic problem, azimuth and elevation observation system

Introduction
Currently, the six-degree-of-freedom parallel motion system is applied in many fields such as semi-realistic simulation, stabilization and balancing system. A small field of 6-degree-of-freedom parallel motion systems is applied to AEOS to capture satellites. This mechanism is implemented by synchronous motion control of six linear actuators, so it has many advantages in terms of accuracy, rigidity, and load-bearing capacity [2]. However, publications on kinematic problems in this application are few. According to the conventional method to solve the kinematic problem, we have to find the rotation transformation matrix on real coordinate systems. Previous publications often use it directly without providing an explicit method to determine the rotation transformation matrix [1, 3, 5]. The research content of the article presents in detail how to set up the rotation transformation matrix according to the conventional method on the real coordinate system. On that basis, a new approach is proposed to the inverse kinematic problem for AEOS through the use of virtual coordinate systems.

Methodology
A. Parallel motion systems with 6 degrees of freedom and inverse kinematic problems for AEOS using the conventional method
1. Parallel motion systems with 6 degrees of freedom
The 6 degrees of freedom parallel motion system, also known as the SGP 6-dof parallel robot, is normally composed of a base plate (base platform - B), an upper plate (moving platform - P) and 6 driving legs [2, 6].

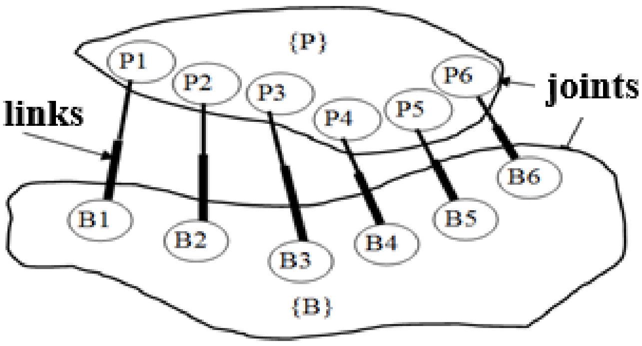


Fig 1: The structure of parallel robot

These driving legs are linked together by a translational joint - prismatic, so their length can be changed and linked to two planes through ball joints at the ends of each leg. The joints between the leg with the base plane and the moving plane are arranged as shown in Figure 1.

The symmetrical SGP 6DOF parallel robot has joints located on the base plane or moving plane arranged in symmetrical pairs and belonged to two circles [1]. Thanks to that, the geometric representation of the base plane and the moving plane can be simply expressed by the following 4 variables:

- r_B, r_P: Radius of circle created by joints on the base plane, and on the moving plane.
- α_B, α_P: Angle created by the symmetrical joint pair on the base plane, and on the moving plane.

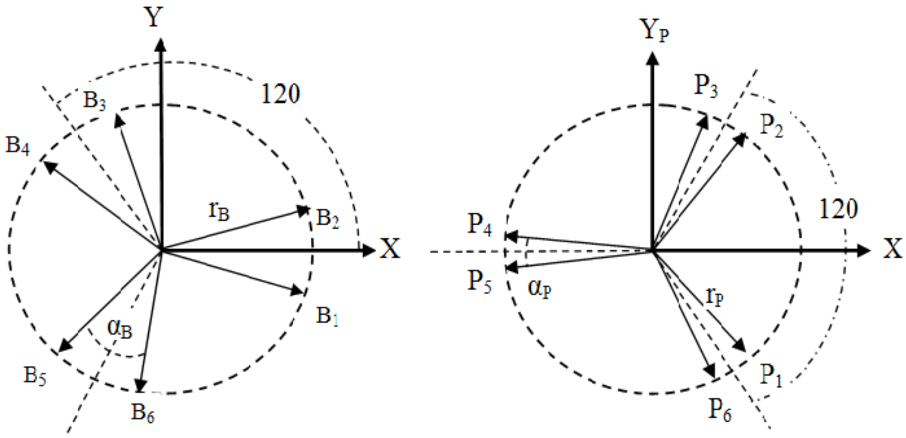


Fig 2: Arrangement of the $B_i$and P_i joints on the base plane and moving plane

2. Inverse kinematic problems for azimuth and elevation observation systems using the conventional method

To calculate inverse kinematic for the AEOS according to the conventional method, we need to find the transformation matrix from the new coordinate system after rotating the direction angle by an angle β and the elevation angle by an angle τ to the original coordinate system, then establish the equation of motion of the system. So first, we need to set up the coordinate system for them. In the 6-degree-of-freedom parallel motion system, the coordinate system includes a fixed coordinate system attached to the base platform, and a moving coordinate system attached to the moving platform [5, 6]. The layout of the X, Y coordinate system is shown in Figure 2 and the Z axis is the vertical axis, parallel to the gravity vector. The fixed coordinate system is denoted as $\{B\}$ with the center as O_B, and the coordinate system of the moving platform is denoted as $\{P\}$ with the center as O_P. The coordinate system of the driving legs of a parallel robot is a set of length variables. For SGP 6DOF - parallel robot, it is composed of 6 sliding joints. Written as a vector in joint space, we have:

$$L=[l_1\ l_2\ l_3\ l_4\ l_5\ l_6]^T \tag{1}$$

The inverse kinematic problem is to find the length vector L_i from the given end-effector coordinate q. The coordinate vector of the joints on the base plane {B} and the coordinate vector of the joints on the moving plane {P}:

$$B_i = \begin{bmatrix} r_B\cos(\varphi_i) \\ r_B\sin(\varphi_i) \\ 0 \end{bmatrix} = \begin{bmatrix} B_{ix} \\ B_{iy} \\ B_{iz} \end{bmatrix} \qquad \begin{aligned} \varphi_i &= \frac{(i-1)\pi}{3} - \frac{\alpha_B}{2} \text{ with i=1,3,5;} \\ \varphi_i &= \varphi_{i-1} + \alpha_B \text{ with i=2,4,6;} \end{aligned} \tag{2}$$

$$P_i = \begin{bmatrix} r_P\cos(\theta_i) \\ r_P\sin(\theta_i) \\ 0 \end{bmatrix} = \begin{bmatrix} P_{ix} \\ P_{iy} \\ P_{iz} \end{bmatrix} \qquad \begin{aligned} \theta_i &= \frac{(i-2)\pi}{3} + \frac{\alpha_P}{2} \text{ with i=1,3,5;} \\ \theta_i &= \theta_{i-1} + \frac{\pi}{3} - \alpha_P \text{ with i=2,4,6;} \end{aligned} \tag{3}$$

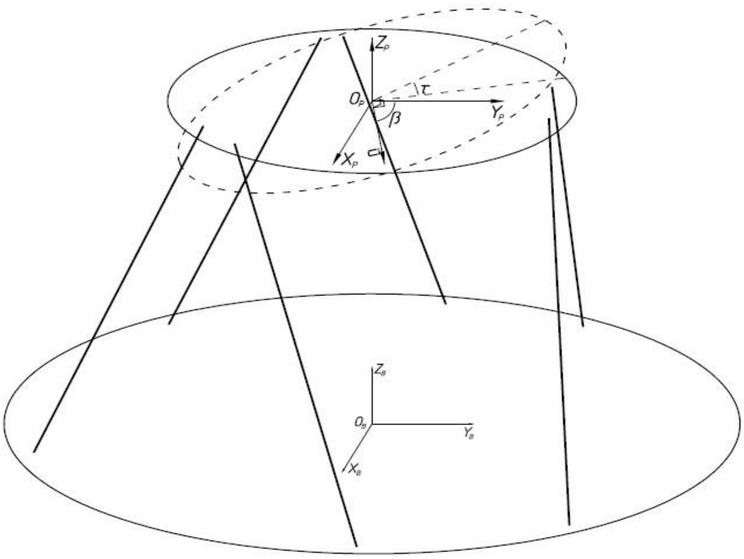


Fig 3: Position of the moving platform after rotating an azimuth angle β and an elevation angle τ.

Normally, to solve the problem of inverse kinematic for a hexapod system, a transformation matrix from 3 rotation angles, roll, pitch, yaw [2] is calculated. However, due to the limitation of the yaw angle of the moving platform (rotation angle around the Z axis), it is not possible to rotate a full 360 degree like the azimuth angle of the target, so to rotate the hexapod system with an azimuth angle β and an elevation angle τ, we need to perform a direct rotation around vector $\vec{u}$ (on the plane $O_PX_PY_P$ and offset by an angle β from the O_PZ_P axis) by an angle τ as shown in Figure 3. To find the transformation matrix of this rotation, we perform the following basic rotations:
+ First, we rotate around the O_PZ_P axis by an angle β to bring the O_PY_P axis of the $O_PX_PY_P$ coordinate system coincident with the vector $\vec{u}$.
+ Rotate around the O_PY_P axis at an angle τ so that the moving platform of the motion system tilts relative to the original coordinate system at an angle of about τ.
+ Rotate around the Z axis at an angle -β to change the OY axis to position OY'.
Therefore, the rotation transformation matrix is calculated as:

$$R_T = R_Z(\beta).R_Y(\tau).R_Z(-\beta) = \begin{bmatrix} s^2\beta + c^2\beta c\tau & c\beta s\beta c\tau - c\beta s\beta & c\beta s\tau \\ c\beta s\beta c\tau - c\beta s\beta & c^2\beta + s^2\beta c\tau & s\beta s\tau \\ -c\beta s\tau & -s\beta s\tau & c\tau \end{bmatrix} \tag{4}$$

Formula (4) is the rotation transformation matrix of the 6-degree-of-freedom parallel motion system for the AEOS. This formula is completely consistent with previous studies [1, 3, 5]. However, this rotation transformation matrix is a trigonometric matrix with complex components. The article offers a different approach to the components of a simpler rotation transformation matrix through the use of virtual coordinate systems.

B. Solving inverse kinematic problem using virtual coordinate systems

1. Positional inverse kinematic problem

To solve the inverse kinematic problem of a 6-degree-of-freedom motion system applied to the AEOS, we use the virtual coordinate system $O_PX_{PA}Y_{PA}Z_{PA}$ on the plane of the moving plate with the O_PZ_{PA} axis coinciding with the O_PZ_P axis, and the O_PX_{PA}, O_PY_{PA} axes are respectively offset by an angle β compared to the O_PX_P, O_PY_P axes. The virtual coordinate system $O_BX_{BA}Y_{BA}Z_{BA}$ on the base plate has corresponding axes parallel to the axes of the $O_PX_{PA}Y_{PA}Z_{PA}$ coordinate system as shown in Figure 4.
Then, the transformations are calculated on the virtual coordinate system. To do this, we need to follow these 2 steps:

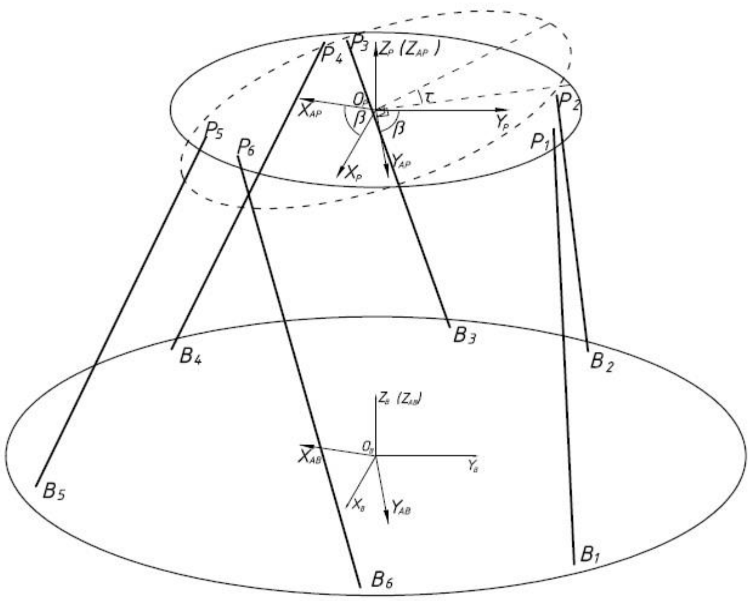


Fig 4: Position of the moving platform after rotating an azimuth angle β and an elevation angle τ.

Step 1: Find the coordinates of the joints on virtual coordinate system.
Because the O_BX_{BA}, O_BY_{BA} axes of the virtual coordinate system $O_BX_{BA}Y_{BA}Z_{BA}$ are respectively offset by an angle β compared to the O_BX_B, O_BY_B axes, the position of the joints B_{Ai} relative to the virtual coordinate system is calculated as:

$$B_{Ai} = \begin{bmatrix} r_B cos(\varphi_i - \beta) \\ r_B sin(\varphi_i - \beta) \\ 0 \end{bmatrix} = \begin{bmatrix} B_{Aix} \\ B_{Aiy} \\ B_{Aiz} \end{bmatrix} \tag{5}$$

Where $\varphi_i = \frac{(i-1)\pi}{3} - \frac{\alpha_B}{2}$ when i=1, 3, 5;
$\varphi_i = \varphi_{i-1} + \alpha_B$ when i=2,4,6;

Similarly, position of the joints P_{Ai} relative to the virtual coordinate system is calculated as:

$$P_{Ai} = \begin{bmatrix} r_P cos(\theta_i - \beta) \\ r_P sin(\theta_i - \beta) \\ 0 \end{bmatrix} = \begin{bmatrix} P_{Aix} \\ P_{Aiy} \\ P_{Aiz} \end{bmatrix} \tag{6}$$

Where $\theta_i = \frac{(i-2)\pi}{3} - \frac{\alpha_P}{2}$ when i=1,3,5;
$\theta_i = \theta_{i-1} + \frac{\pi}{3} - \alpha_P$ when i=2,4,6;

Step 2: Perform rotation around the O_PY_{PA} axis of the virtual coordinate system by an angle τ. Then:

$$R_{AT} = R_Y(\tau) = \begin{bmatrix} cos(\tau) & 0 & sin(\tau) \\ 0 & 1 & 0 \\ -sin(\tau) & 0 & cos(\tau) \end{bmatrix} \tag{7}$$

Normally, with problems for the AEOS, we are only interested in the elevation angle τ and the azimuth angle β when considering the center position of the base plate to be non-moving. To be more general, assume that the expected position of the end-effector on the moving plate relative to the origin mounted on the base plate is represented by a vector:

$${}^{B}P = [x_d \ \ y_d \ \ z_d]^T \tag{8}$$

Then:
Desired position of the moving plate relative to the virtual coordinate system is:

$${}^{B}P_A = R_Z(-\beta).[x_d \ \ y_d \ \ z_d]^T \tag{9}$$

The vector that represent length of legs is determined by:

$$L_i = P_i + {}^{B}P - B_i \tag{10}$$

Considering the virtual coordinate system mounted on the base plate:

$${}^{B}L_{Ai} = R_{AT} \cdot P_{Ai} + {}^{B}P_A - B_{Ai} \tag{11}$$

The length of the driving leg of the SGP is determined as the Euclidean norm of L_i as follows:

$$l_i = |L_i| = \sqrt{L_{Aix} + L_{Aiy} + L_{Aiz}} \tag{12}$$

Thus, using formulas (11), (12) we can calculate the length vector of the legs L_i, the Euclidean norm l_i if we know the position and direction angles of the parallel motion system. With the use of virtual coordinate systems, we have a simpler approach to the rotation transformation matrix through just a single rotation around the OY axis.

2. Inverse velocity problem and Jacobian matrix

The inverse velocity problem is the problem of finding a matrix (called the Jacobian) that transforms the velocity of the actuating joints (actuator) to the velocity of the end-effector. For the parallel motion system applied to the AEOS, the used expression of the Jacobian matrix is given as follows:

$$\dot{l} = J^{-1}.\dot{t} \tag{13}$$

Where:
- $\dot{l} = \left(\dot{l}_1, \dot{l}_2, \dot{l}_3, \dot{l}_4, \dot{l}_5, \dot{l}_6\right)^T$ is velocity vector of the legs,
- $\dot{t} = \left(\dot{x}_d, \dot{y}_d, \dot{z}_d, \dot{\beta}, \dot{\tau}\right)^T$ is velocity vector of end-effector,
- Matrix J^{-1} is the reverse transformation matrix from the end-effector to the driving joints.

To solve the velocity problem, we introduce the concept of unit vector of the driving cylinder legs calculated in the virtual coordinate system. This vector is denoted u_{Ai}, where i is the index of the corresponding leg. The vector u_{Ai} is calculated as:

$$u_{Ai} = \frac{L_{Ai}}{l_i} = \left(u_{Axi}, u_{Ayi}, u_{Azi}\right);\ i = 1,..,6 \tag{14}$$

Thus, (11) can be written as:

$$u_{Ai}.l_i = R_{AT} \cdot P_{Ai} + {}^{B}P_A - B_{Ai} \tag{15}$$

Differentiating both sides of equation (15) with respect to time we get:

$$\dot{l}_i.u_{Ai} + l_i.\dot{u_{Ai}} = \omega \times R_{AT} \cdot P_{Ai} + R_{AT}.\dot{P}_{Ai} + {}^{B}\dot{P}_A - \dot{B}_{Ai} \tag{16}$$

Because u_i is a unit vector, then $u_i.u_i = 1, u_i.\dot{u}_i = 0$. Thus, (16) become:

$$\dot{l}_i = {}^{B}\dot{P}_A.u_{Ai} + \omega \times (R_{AT} \cdot P_{Ai}).u_{Ai} + R_{AT}.\dot{P}_{Ai}.u_{Ai} - \dot{B}_{Ai}.u_{Ai} \tag{17}$$

Where:

$$\omega = \omega_Y(\tau) = [0\ 1\ 0]^T.\dot{\tau} \tag{18}$$

$${}^{B}\dot{P}_A = -\omega_Z(\beta) \times [R_Z(-\beta).[x_d\ \ y_d\ \ z_d]^T] + R_Z(-\beta).[\dot{x}_d\ \ \dot{y}_d\ \ \dot{z}_d]^T \tag{19}$$

$$\dot{P}_{Ai} = P'_{Ai}.\dot{\beta};\quad \dot{B}_{Ai} = B'_{Ai}.\dot{\beta}; \tag{20}$$

where $P'_{Ai} = \begin{bmatrix} r_P cos(\theta_i - \beta) \\ -r_P sin(\theta_i - \beta) \\ 0 \end{bmatrix}$; $B'_{Ai} = \begin{bmatrix} r_B cos(\varphi_i - \beta) \\ -r_B sin(\varphi_i - \beta) \\ 0 \end{bmatrix}$

Thus, (16) can be explicitly written as:

$$\dot{l}_i = u_{Ai}.R_Z(-\beta)\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{z}_d \end{bmatrix} + \left(\left[R_Z(-\beta).\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} + R_{AT}.P'_{Ai} - B'_{Ai}\right] \times u_{Ai}\right).e_z.\dot{\beta} + ([R_{AT}.P_{Ai}] \times u_{Ai}).e_y.\dot{\tau} \tag{21}$$

Or:

$$\dot{l}_i = \begin{bmatrix} u_{A1}.Rz(-\beta) & \left([R_z(-\beta).\,^BP + R_{AT}.P'_{A1} - B'_{A1}] \times u_{A1}\right).e_z & ([R_{AT}.P_{A1}] \times u_{A1}).e_y \\ u_{A2}.Rz(-\beta) & \left([R_z(-\beta).\,^BP + R_{AT}.P'_{A2} - B'_{A2}] \times u_{A2}\right).e_z & ([R_{AT}.P_{A2}] \times u_{A2}).e_y \\ u_{A3}.Rz(-\beta) & \left([R_z(-\beta).\,^BP + R_{AT}.P'_{A3} - B'_{A3}] \times u_{A3}\right).e_z & ([R_{AT}.P_{A3}] \times u_{A3}).e_y \\ u_{A4}.Rz(-\beta) & \left([R_z(-\beta).\,^BP + R_{AT}.P'_{A4} - B'_{A4}] \times u_{A4}\right).e_z & ([R_{AT}.P_{A4}] \times u_{A4}).e_y \\ u_{A5}.Rz(-\beta) & \left([R_z(-\beta).\,^BP + R_{AT}.P'_{A5} - B'_{A5}] \times u_{A5}\right).e_z & ([R_{AT}.P_{A5}] \times u_{A5}).e_y \\ u_{A6}.Rz(-\beta) & \left([R_z(-\beta).\,^BP + R_{AT}.P'_{A6} - B'_{A6}] \times u_{A6}\right).e_z & ([R_{AT}.P_{A6}] \times u_{A6}).e_y \end{bmatrix}.\dot{t} = J^{-1}.\dot{t} \quad (22)$$

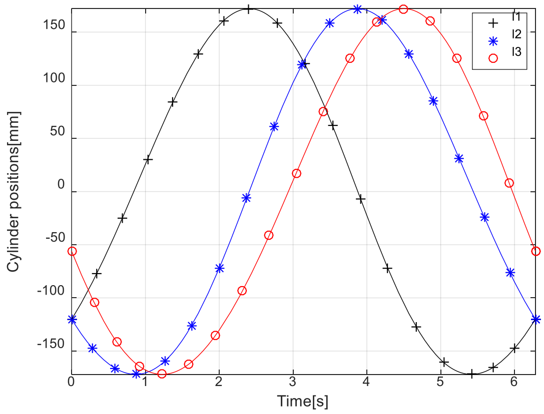
The Jacobian matrix J^{-1} is a 6x5 matrix with values $u_{A1}.Rz(-\beta)$ consisting of 3 components along the x, y, z axes. The matrix J^{-1} has been explicitly determined by the authors as a single matrix, with the usual method the Jacobian matrix is determined as the product of 2 matrices [7].

C. Simulations
The research results were verified and simulated on Matlab with a 6-degree-of-freedom parallel robot system with the following parameters:
- r_B: Radius of circle created by joints on the base plate (=1.0m).
- r_P: Radius of the circle created by the joints on the moving plate (=0.55m).
- α and θ: Angles created by the symmetrical joint pair on the base plate and moving plate (=30^0).
- The moving plane height in the intermediate position is 1.378m, corresponding to the average length of the cylinder leg being 1.5m

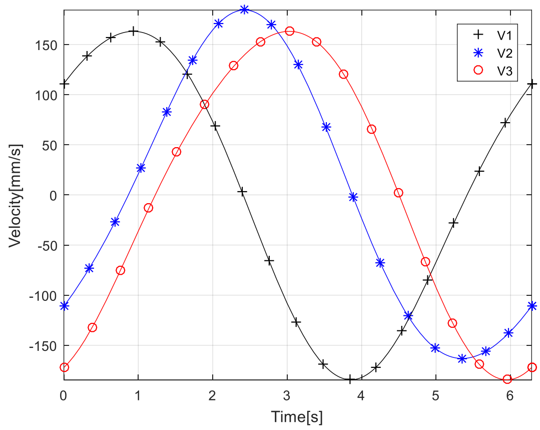
The simulation process is performed with 2 cases:
+ Case 1: keep the elevation angle the same and change the azimuth angle
+ Case 2: keep the azimuth angle the same and change the elevation angle
Each case performs simulation with the conventional method (through lines with +, *, o) and with the method using virtual coordinate systems (through solid lines). Simulation results using both methods are shown on the same graph.

Results and Discussion
Figure 5, figure 6 are the position and velocity of the cylinder legs in case 1 when keeping the elevation angle about 20 degrees constant, and changing the azimuth angle around the OZ axis with the equation $\beta = t$ (t=0÷2.π). In figure 5, cylinder legs 1,2,3 are shown; figure 6 shows cylinder legs 4,5,6.

Figure 7 describes the position and velocity of the cylinder legs in case 2 with a fixed azimuth angle at the initial position, and changing the elevation angle around the OY axis with the equation τ=20.sin(t) ((t=0÷2.π). When keeping the azimuth angle constant and changing the elevation angle, the position and velocity of cylinder leg pairs 1, 2; 3, 6; and 4, 5 have the same variation.

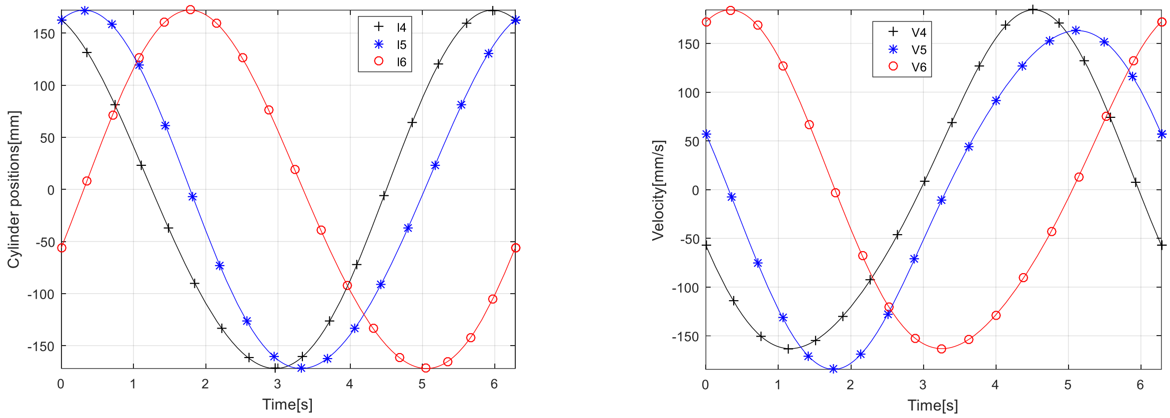


a) Positions of the cylinder legs 1, 2, 3



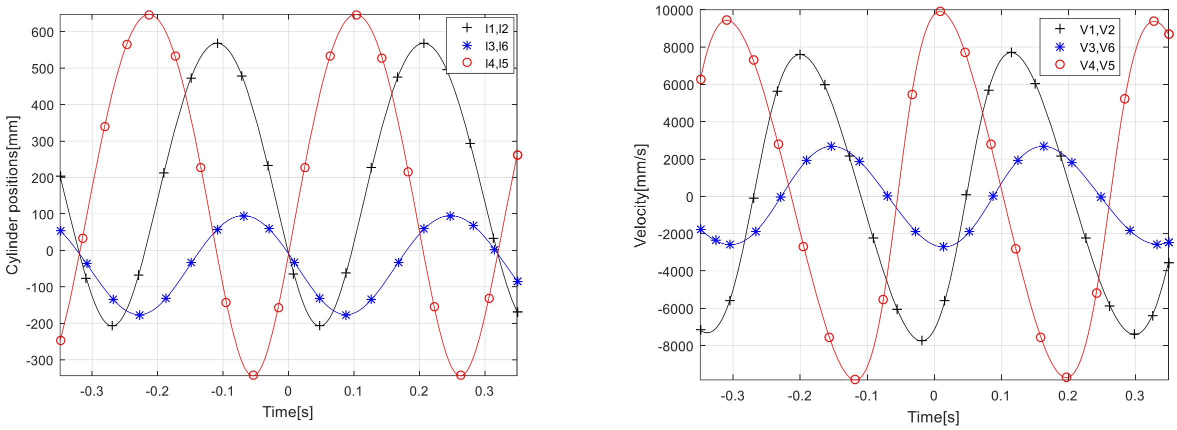
b) Velocities of the cylinder legs 1, 2, 3

Position and velocity of the cylinder legs 1, 2, 3 when rotating azimuth angle β



a) Positions of the cylinder legs 4, 5, 6 b) Velocities of the cylinder legs 4, 5, 6

Fig 6: Position and velocity of the cylinder legs 4, 5, 6 when rotating azmuth angle β



a) Positions of the cylinder legs b) Velocities of the cylinder legs

Fig 7: Position and velocity of the cylinder legs when rotating elevation angle $-20^0 \div 20^0$

From the graph of the position and speed of the driving cylinder legs, we can see that using the virtual coordinate system method gives us results that completely coincide with the conventional method when using less by one rotation transformation matrix. Thereby demonstrating the correctness and the enhancement of the chosen method.

Conclusions

This article has presented a new approach to the problem of solving inverse kinematic for parallel motion systems with 6 degrees of freedom applied to AEOS by using a virtual coordinate system. The simulation, comparison and evaluation results have shown the correctness of the method. Using a virtual coordinate system gives us a simpler and more effective approach:

- In the position problem, with the use of virtual coordinate systems, the rotation transformation matrix only uses two rotations: around the Y axis and around the Z axis. Meanwhile, with the conventional method, the final rotation transformation is the product of 3 rotation matrices.
- In the velocity problem, with the use of virtual coordinate systems, the author has built an explicit formula for the Jacobian matrix in the form of a single matrix. While with the conventional method, the Jacobian matrix is built as the product of two matrices.

Therefore, this approach helps solve problems quickly, conveniently and consumes less computational resources than conventional methods.

References

1. Ngo Duy Tan, Truong Tuan Anh và nnk, Research and Development of a Hexapod Tracking Controller for VNREDSat-1 Satellite. 2021; 4(3). ISSN 2581-7175.
2. Stewart D. A Platform with Six Degrees of Freedom, Proceedings of the Institution of Mechanical Engineers, 1965.
3. Ngô Duy Tân, Đặng Quốc Đại, Vương Huy Hoàng, "Nghiên cứu và thiết kế bệ đỡ kiểu hexapod cho hệ thống anten bám theo vệ tinh nhỏ quan sát Trái đất", Tạp chí Tự động hóa ngày nay, Số 218-4/2019.Tr.28 – 30, ISSN 1859-0551.
4. Dasgupta, TS Mruthyunjaya. A Newton-Euler formulation for the inverse dynamics of the Stewart–Gough Platform manipulator, Mech. Mach.Theory. 1998; 33(8):1135-1152.
5. Likhit W, Rapirat R, Jarunee K. First Development S-Band Antenna System for THAICHOTE Satellite Control Ground Station, IEEE, 2016.
6. Sébastien Briot, Wisama Khalil. Dynamics of Parallel Robots, Mechanisms and Machine Science, 35, 139-198.

7. A Sumnu. Simulation and PID control of a Stewart platform with linear motor, Journal of Mechanical Science and Technology. 2017; 31(1):345-356.