Innovative Design and Intelligent Manufacturing
L.C. Jain et al. (Eds.)

doi:10.3233/FAIA241130

Modeling and Simulation Analysis of the Stewart Platform System Based on Modelica

Di HUANG [a,1], Lixin HUI[b], Qianqian Lin [a], Bingrui Bao[b], Zhengguo Jin[b]
[a] *Beijing Polytechnic, China*
[b] *Suzhou TongYuan Software&Control Technology Co., Ltd.,China*

Abstract. The Stewart platform is a typical complex mechatronic system, involving multiple disciplines such as mechanics, control and electrical engineering. In order to model and analyze the Stewart platform system in a unified manner, a unified multidisciplinary modeling language is adopted. Based on the object-oriented modeling concept, the system is decomposed and integrated to establish a set of simulation models for the Stewart platform system. This paper, taking advantage of the acausal modeling characteristics of the Modelica language, efficiently and rapidly constructs a set of inverse kinematics control methods that can accurately solve the controlled quantities in the controller based on the inverse model. By conducting multi-scenario simulation analyses based on the Stewart platform system model, The results demonstrate that the inverse kinematics control method can accurately achieve control over the displacement and orientation of the moving platform, proveing the correctness of the Stewart platform system. This provides guidance for subsequent research on the control theory and kinematics dynamics of the Stewart platform.

Keywords. Stewart platform, Modelica, multiple-domain, inverse kinematics control

1. Introduction

Parallel mechanism, in comparison to serial ones, possess several advantages such as greater stiffness, faster dynamic response, higher precision and stronger load-bearing capabilities. Consequently, parallel mechanisms have been garnering increasing attention in industrial applications. With the development of parallel mechanisms, the six-degree-of-freedom platform has also attracted more and more attention, and has been widely used in robotics, manufacturing assembly, medical machinery, aviation, aerospace and other fields[1]. In order to verify the rationality of the design parameters and analyze the kinematic and dynamic characteristics of the Stewart platform, a set of system models of the Stewart platform can be constructed using computer and simulation software[2].

The Stewart platform system consists of components such as the mechanical structure, motors, controllers and sensors, encompassing multiple domains including mechanics, electrical engineering and control systems. In the past, ADMAS software was typically used to construct the interface system, and Simulink was utilized to

[1] Corresponding Author: Di HUANG, huangdi2040@163.com

simplify the construction of the motor and mechanical structure system as well as the control system. An interface-based co-simulation method is adopted to integrate the Stewart platform system.

Traditional multi-domain modeling and simulation generally adopt a co-simulation approach based on interfaces. There are three primary methods currently used in practical engineering applications: (1) Aco-simulation method that integrates multi-domain system models by developing interfaces between various single-domain simulation software. This method largely depends on whether the single-domain simulation software provides or can develop specific interfaces. (2) A co-simulation method based on the High Level Architecture (HLA) standard, typically used for large-scale distributed system co-simulation. This requires the development of interfaces with HLA from the single-domain simulation software. (3) A co-simulation method based on the Functional Mock-up Interface (FMI) standard, which, in contrast to HLA, enables single-domain simulation software to generate dynamic system model C code interfaces in accordance with the standard FMI specifications.

The interface-based co-simulation method can achieve the purpose of multi-domain modeling and simulation to a certain extent. However this method requires software from different domains to provide relevant interfaces and necessitates engineers to decouple each domain and integrate each subsystem, Moreover, using this approach can significantly reduce the accuracy and efficiency of the simulation solution. Therefore, this paper adopts Modelica, a unified multi-domain modeling language, to construct the Stewart platform system.

The advanced features of the Modelica language are reflected in the following four aspects: (1) The Modelica language can construct and analyze system models composed of different disciplines on the same platform according to the generalized Kirchhoff's law, which can effectively improve the solvability and accuracy of the model by avoiding the complex decoupling of different types of models between subsystems[3]. (2) The Modelica language adopts acausal modeling technology, which can avoid engineers deriving the causal relationship of complex simulation systems and improve modeling efficiency[4]. (3) The Modelica language uses object-oriented modeling, which solely focused on the characteristics of the model itself, without considering the impact of external environmental factors, which can greatly improve the reusability and extensibility of the model development[5]. (4) The Modelica language supports continuous-discrete hybrid modeling, which can effectively handle events during the simulation process and simulate the system under variable working conditions with different control sequences. Hence, the Modelica language is well-suited for the modeling tasks of complex engineering systems and fully leverages its features of unified multi-domain modeling and simulation.

The objective of this paper is, in accordance with the structural components of the Stewart platform(6-UPS), to employ the multi-domain modeling and simulation language Modelica along with the multi-domain modeling platform MWorks. Sysplorer, with the application goal of simulating and analyzing the Stewart platform. The paper aims to construct a comprehensive multi-domain system model that integrates the control, mechanical and electrical systems of the Stewart platform. Furthermore, it will perform kinematic and dynamic analysis based on the Stewart system simulation model, laying the foundation for the implementation of a digital twin in the later stages.

2. Systems Analysis of the Stewart Platform System

The Stewart platform system is an integrated mechatronic and control system. Based on the object-oriented modeling features of Modelica and the actual physical topology of the Stewart platform system, a top-down system decomposition and a bottom-up system integration approach are adopted for modular modeling of the Stewart platform.

2.1. Top-down System Decomposition

According to its actual physical topological relationship, the Stewart platform system can be decomposed into the trajectory planning subsystem, the inverse kinematics programming subsystem and the Stewart platform ontology (6-UPS) subsystem.

The trajectory planning subsystem simulates external command signals providing the Stewart platform with target motion instructions, including displacement and attitude angles.

The inverse kinematics planning subsystem calculates the target values for the Stewart platform's servo drives based on the target motion signals from the trajectory planning subsystem. This involves first determining the extension lengths of the six linear actuators and then calculating the target angular values for the driving motors within the linear actuators, ensuring that the Stewart platform can move according to the trajectory planning.

The Stewart platform main body subsystem consists of two parts, the linear servo actuators and the Stewart platform mechanism body (6-UPS). The servo actuators based on the target values calculated by the inverse kinematics planning subsystem, provide driving displacement to the Stewart platform mechanism through the transmission devices. According to the function of the linear drive, it can be decomposed into servo motor and ball screw, with the servo motor being further broken down into components such as the motor body, controller and drive circuit. The Stewart platform mechanism (6-UPS) is composed of 6 spherical joints, 6 Hook joints and 6 linear bearings between the upper and lower platforms. Displacement can be provided to the linear bearings by the linear actuators, enabling the Stewart motion platform to move according to the trajectory planning. Additionally, it is possible to view the motion and force conditions of each rod and bearing in real-time.

According to the functional decomposition of the Stewart platform as described, the model library architecture for the Stewart platform is constructed within the MWorks.Sysplorer software.

2.2. Bottom-up System Integration

Based on the decomposition results of the Stewart platform system, using the Modelica standard library and the developed Stewart platform model library, the corresponding components, subsystems and system models are constructed by graphical modeling approach, and the system framework of the Stewart platform is shown in Figure 1.

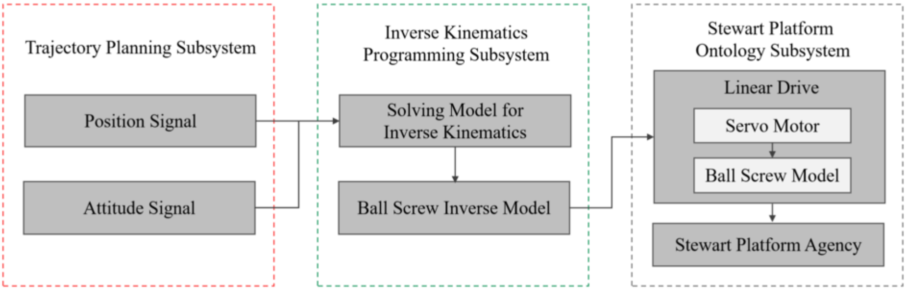


Figure1. Stewart Platform System Framework

After the modeling is completed, necessary validations are conducted on models at various levels. Ultimately, kinematic and dynamic analyses are performed based on the validated Stewart platform system model.

3. Modelling of the Stewart Platform Based on Modelica

Following the aforementioned system analysis approach, the construction of the Stewart platform model is achieved through the modeling and integration of the trajectory planning subsystem, the inverse kinematics planning subsystem, and the Stewart platform ontology (6-UPS) subsystem.

3.1. Model Interface Classification and Construction

The Modelica language, grounded in the generalized Kirchhoff's law, achieves unified modeling and simulation across multiple domains, meaning that energy is transferred between interfaces of physical components from different disciplines within a system. The energy at different discipline interfaces can be divided into potential variables (where the node potentials are equal at interfaces) and flow variables (where the node flows sum to zero across interfaces). Through the interfaces of different disciplines, information transfer between models is accomplished.

The Stewart platform system encompasses control, electrical and mechanical disciplines, it is necessary to establish different interface models, including signal interfaces, electrical interfaces, 1D translational interfaces, 1D rotational interfaces and 3D multi-body interfaces.

The signal interface needs to determine the data flow direction, hence the causal interface is employed, which is defined by specifying real numbers and using the input/output prefix to determine the flow of control signals. All other interfaces are constructed based on the generalized Kirchhoff's laws as physical interfaces, allowing for bidirectional flow of information within them. The potential and flow variables defined in these interface are shown in Table 1.

Table 1. Variables of physical interfaces for each discipline

Profession	Potential variable	Flow variable
Electric	Electric potential (v)	Current (i)
One-dimensional translation	Displacement(s)	Force(f)
One-dimensional rotation	Angle (ϕ)	Torque (τ)
Three-dimensional multi-body	Displacement(S) Attitude (R)	Force(F) Torque (T)

3.2. Trajectory Planning Subsystem Modeling

The Stewart motion platform can achieve movement in six degrees of freedom, i.e., the upper platform can perform translational motion in the X-Y-Z direction, or it can perform rotational motion around the X-Y-Z axis. The control commands are divided into translational and rotational motion control signals based on the motion type of the Stewart upper platform. The control commands define the translational and rotational control signals for the motion platform and transmit these control signals to the inverse kinematics planning subsystem for the resolution of inverse kinematics.

3.3. Inverse Kinematic Planning Subsystem Modeling

The inverse kinematics planning subsystem initially applies the spatial vector method to carry out a position inverse solution for the Stewart mechanism, thereby determining the extension lengths of the linear actuators. Subsequently, by utilizing the non-causal modeling characteristic of the Modelica language, it ascertains the extension lengths for the ball screw mechanism and calculates the reverse angular velocity needed for the motor drive, supplying the target angle signal to the servo motor controller.

3.3.1. Inverse Kinematics Solution Model Modeling

The inverse kinematics of Stewart motion platform is to use the spatial vector method to calculate the elongation of each linear actuator based on the desired position and orientation of the motion platform, so as to achieve the intended position and orientation. As shown in Figure 2, the upper and lower platforms of the Stewart platform have their own independent coordinate systems, with the upper platform coordinate system is p-frame, and the lower platform coordinate system is b-frame. Pi($i \in (1,6)$) represents the six hinges points in the p-frame coordinate system of the upper platform, and bi($i \in (1,6)$) represents the six hinges points in the b-frame coordinate system of the lower platform. The goal of the inverse kinematics of the Stewart platform is to solve for the modulus length of the vector li which determines the elongation of the individual linear actuators.

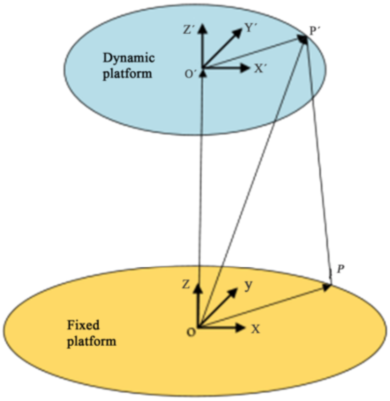


Figure 2. Stewart Platform Simplified Schematic

Subsequently the spatial vector method is used to calculate li(i∈(1,6)) (Note: all spatial vectors involved in the calculation must be represented in the b-frame coordinate system).

Equation (1) is used to calculate the coordinates of the six hinges of the upper platform in the *b-frame* coordinate system, i.e., the $\boldsymbol{q}_i(i \in (1,6))$ vector. Where $\boldsymbol{T}$ represents the position of the origin of the *p-frame* coordinates in the *b-frame*, and $\boldsymbol{R}_p^b$ is the rotation matrix from the *p-frame* to the *b-frame* based on the desired orientation angles, which is used to convert the $\boldsymbol{p}_i$ vector from the *p-frame* to the *b-frame*.

$$\boldsymbol{q}_i = \boldsymbol{T} + \boldsymbol{R}_p^b * \boldsymbol{p}_i \tag{1}$$

Since $\boldsymbol{b}_i$ and $\boldsymbol{q}_i$ are known, the vector $\boldsymbol{l}_i$ is calculated by equation (2), and then the modulus length of the vector $\boldsymbol{l}_i$ is calculated by equation (3), and the elongation e_i of the linear drive is obtained by subtracting the original length L_i of the linear drive.

$$\boldsymbol{l}_i = \boldsymbol{q}_i - \boldsymbol{b}_i \tag{2}$$

$$e_i = norm(\boldsymbol{l}_i) - L_0 \tag{3}$$

3.3.2. Ball Screw Direct/Inverse Model Modeling

Before modeling the inverse model of a ball screw, it is essential to first establish the model of the ball screw itself. The ball screw consists of a screw shaft, a nut and balls. Its function is to convert rotational motion into linear motion. The threaded grooves between the screw shaft and the nut are filled with balls. When the screw shaft or the nut rotates, the balls roll along these threaded grooves, resulting in rolling friction as the screw shaft and the nut move relative to each other. The primary principle formula for the ball screw is as follows:

$$w = \frac{d\theta}{dt} \tag{4}$$

$$v = \frac{ds}{dt} \tag{5}$$

$$w = v \times k \tag{6}$$

$$T \times w \times \eta = F \times v \tag{7}$$

Wherein: θ is the rotation angle of the nut, w is the angular velocity of the nut, s is the displacement of the nut, v is the velocity of the nut, k is the ball screw transmission ratio, T is the torque applied to the nut, F is the force applied to the nut and η is the transmission efficiency of the ball screw.

According to the non-causal modeling characteristics of the Modelica language, where the "=" in the code only constrains the equality of both sides without dictating the assignment relationship, it is possible to solve for the movement displacement through the rotation angle, and also to solve for the rotation angle through the movement displacement.

3.4. Stewart Platform Ontology (6-UPS) Subsystem Modeling

The Stewart platform main body consists of 6 linear actuators and the Stewart platform mechanism, as shown in Figure3.

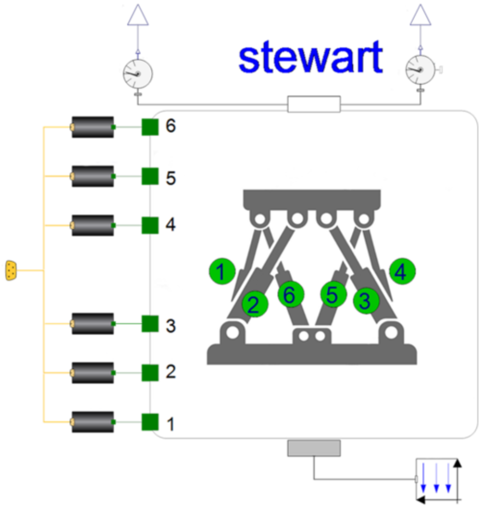


Figure 3. Stewart Platform Ontology (6-UPS) Subsystem Model

3.4.1. Linear Actuator Modeling

Given that the six linear actuators share the same structure, leveraging the object-oriented modeling features of the Modelica language, it is only necessary to construct a single set of linear actuator models.

The linear actuator is composed of a servo motor and a ball screw. The model of the servo motor itself directly adopts the permanent magnet synchronous motor model (PMSM) from the Motor_TY motor library in MWorks.Sysploer.

A cascaded PID control method is employed to achieve control over the position, angular velocity and current of the permanent magnet synchronous motor. Based on the control block diagram, a permanent magnet synchronous motor controller model is established, as shown in Figure 4, the permanent magnet synchronous motor controller can be set to either position control mode or angular velocity control mode according to the control requirements.

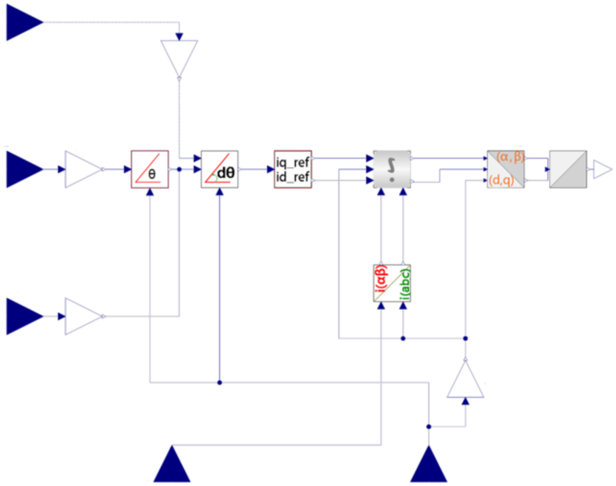


Figure 4. Control Loop Model of PMSM

In the position control mode, the difference between the expected angle control signal and the angle signal feedback from the sensor yields an angle deviation signal. This signal, after being processed by the position controller, is output as an angular velocity control signal. The position controller uses a PI controller with anti-integration saturation, and the output signal of the controller is limited according to the maximum angular velocity of the actual motor to prevent the integrator from falling into saturation, resulting in system overshoot and dullness.

The difference between the desired angular velocity signal and the actual angular velocity signal provided by the sensor feedback results in an angular velocity error signal, which acts as the input to the angular controller. The angular velocity controller employs a PI controller, and its output signal is the desired q-axis current of the motor. The output signal is limited according to the maximum current that the motor can handle.

The current control circuit expects a current of 0 on the d-axis. The discrepancy between this expected value and the actual d-axis current feedback produces a d-axis current error signal, which is then managed by a d-axis current PI controller to generate a d-axis voltage control signal. The q-axis current error signal, derived from the difference between the desired q-axis current output from the angular velocity controller and the actual q-axis current feedback, is the input to the q-axis current PI controller. To account for the voltage control error due to the motor's counter EMF, a feed forward compensation for current control has been incorporated into the current control loop, which serves to mitigate the disturbances caused by the motor's counter EMF.

The motor controller, motor body model and ball screw model are integrated to form a linear actuator based on their physical topological relationships, as shown in Figure 5.

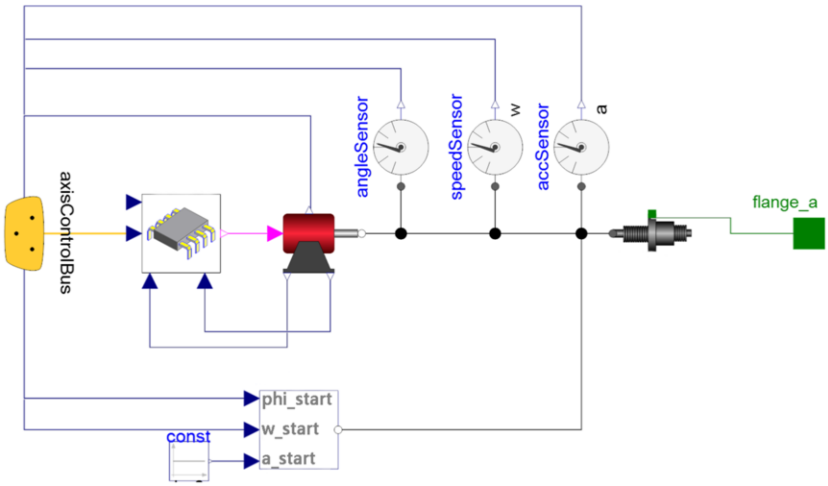


Figure 5. Linear Drive Model

Taking advantage to the object-oriented characteristics of Modelica, the models for the motor, controller, ball screw and other components in the system can be interchanged according to the system model.

3.4.2. Stewart Platform Structural Model Modeling

The three-dimensional assembly model of the Stewart platform from CAD software is transformed into a Modelica kinematic simulation model using the Kinetrans plug-in within MWorks.Sysplorer, as depicted in Figure 6. The Stewart platform comprises 6 parallel linear mechanisms, the upper platform and the linear mechanism are connected by the spherical joints, and the lower platform is attached to them through Hooke joints. The linear mechanism in the middle can extend or retract based on commands, thereby facilitating the movement of the upper platform.

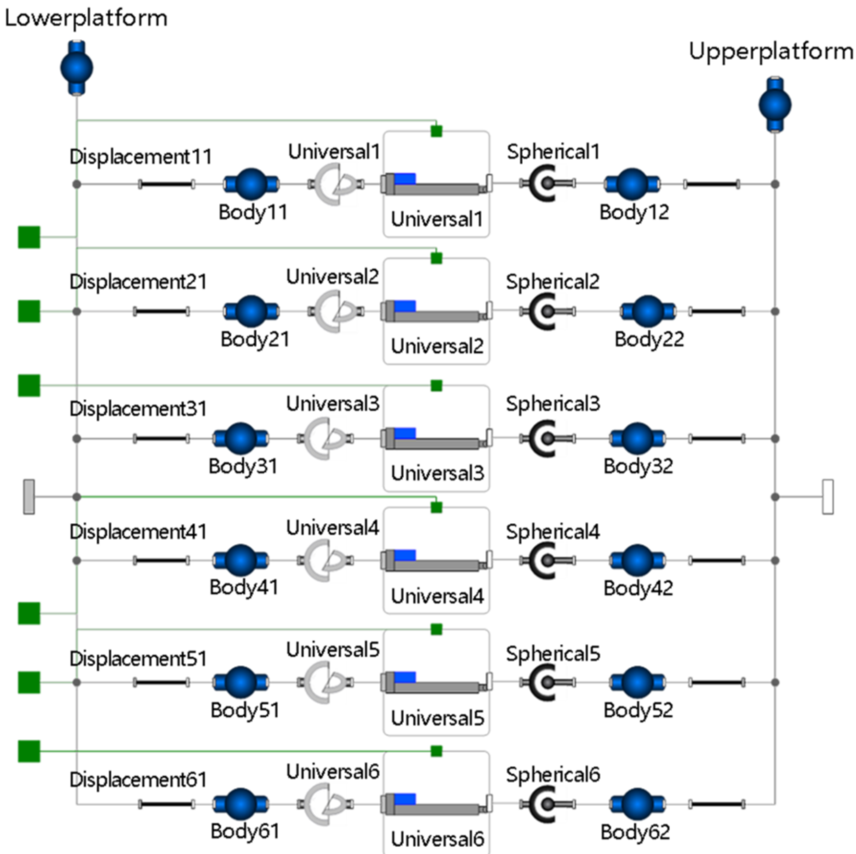


Figure 6. Stewart Platform Architecture Model

3.5. Stewart Platform System Integration

By utilizing the completed models of the trajectory planning subsystem, inverse kinematics planning subsystem and the Stewart platform body (6-UPS) subsystem, the Stewart platform is integrated from the bottom up in accordance with the system architecture illustrated in Figure 1, as shown in Figure7.

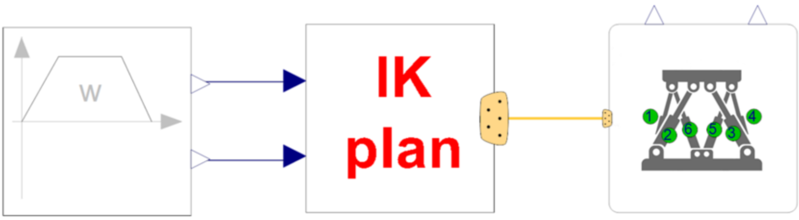


Figure 7. Stewart Platform System Model

4. Simulation Analysis of the Stewart Platform System

Considering the multidisciplinary nature of the Stewart platform, employing a single-domain simulation software cannot concurrently create the control system and the object being controlled. Consequently, it is impossible to affirm the control efficiency while simultaneously monitoring the intricate features of the motors, linear actuators and moving platforms.

In order to delve deeper into the analysis of the inverse motion control efficiency of the Stewart platform system and the status of the controlled object, the Stewart platform system model established in Figure7 is simulated and analyzed in MWorks.Sysplorer, with different trajectory planning path conditions given. Through the simulation results, the motion state of the Stewart moving platform and theextension length and force magnitude required by the liner actuators can be monitored. This paperconducts an analysis of three different trajectory path conditions, all starting to move at 1s, with a movement duration of 5s, The target position (x, y, z) and attitudes (roll, pitch, yaw) are shown in Table 2.

Table 2. Trajectory Planning Path Parameters

Case	x/m	y/m	z/m	roll/deg	pitch/deg	yaw/deg
1	0	0	0.2	0	0	0
2	0	0	0	10	0	0
3	0	0.2	0.2	10	0	0

The main parameters within the system are adjusted, utilizing the International System of Units (SI units) for all values, and identical parameters are applied to the six linear actuators. The parameter settings are detailed in Table 3.

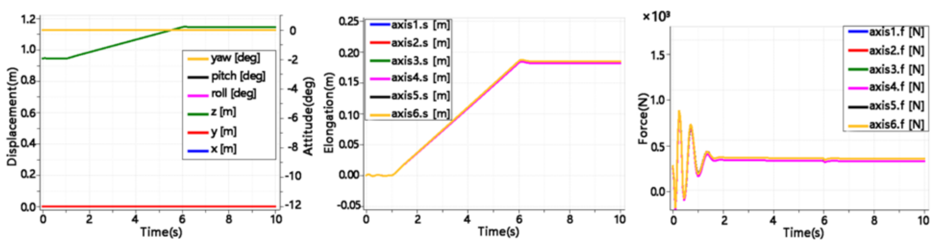
Table 3. Main Parameters of the System

Name of the component	Name of the parameter	Parameter description	Value
Ball screw	ratio	Gear ratio	50
Permanent magnet synchronous motor	V_DC	DC supply voltage	48
	p	Number of electrode pole pairs	2
	fsNominal	Rated frequency	50
	Rs	Phase resistance of the stator winding	1.275
	Lssigma	Leakage inductance of the stator phase winding	0.00047
	Lmd	d-axis inductance	0.00975
	Lmq	q-axis inductance	0.00975
Permanent magnet synchronous motor controller	N	Controller selection	position control
	kp_i	Current loop proportionality factor	10
	Ti_i	Time constant for the integration of the current loop	4

kp_w	Speed loop scale factor	10
Ti_w	Time constant for the integration of the velocity loop	4
kp_phi	Angular loop scale factor	20
Ti_ phi	Time constant integrated by the angular loop	2.8

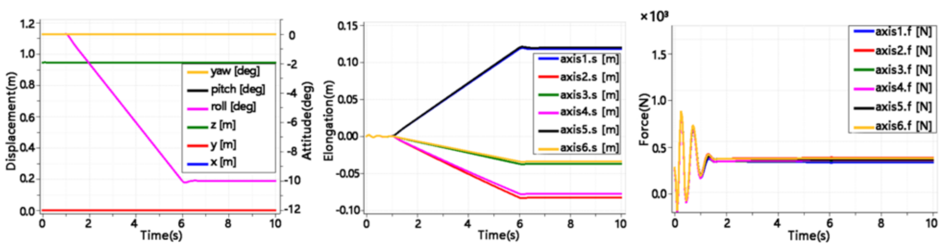
By observing the simulation results curves, the motion state of the Stewart moving platform and the extension length and force magnitude required by the linear actuators can be monitored. The simulation results are shown in Figures 8 to 10.

Analyzing the simulation results under the three working conditions, under Condition 1, the moving platform moves along the z direction according to the reference motion trajectory as shown in Figure 8-a, Since the moving platform has an initial height, it starts moving based on this initial height; Fig. 8-b and Fig. 8-c are the elongation curves and stress curves of the six linear drives under the condition of case 1 respectively. As the movement is only in one direction, the six curves in both figures basically coincide. Wherein Fig. 8-c, the fluctuation in the force curve from 0-1s is due to the six linear actuators balancing the gravity of the moving platform, and it can be seen that the force curves in the three working conditions are basically the same at 0-1s. As shown in Fig. 9, under the condition of working condition 2, the roll angle of the moving platform (rotation around the x-axis) rotates 10 degrees clockwise, and the motion trajectory curve of the moving platform in Fig. 9-a is basically consistent with the reference trajectory signal. In Fig. 9-b and Fig. 9-c, the motion and force curves of axis 1 and axis 5 are basically consistent, the motion and force curves of axis 3 and axis 6 are basically the same, and the motion and force curves of axis 2 and axis 4 are basically consistent. As shown in Figure 10, the motion and force curves under condition 3 are also basically in line with the movement trend of the Stewart platform under this condition.



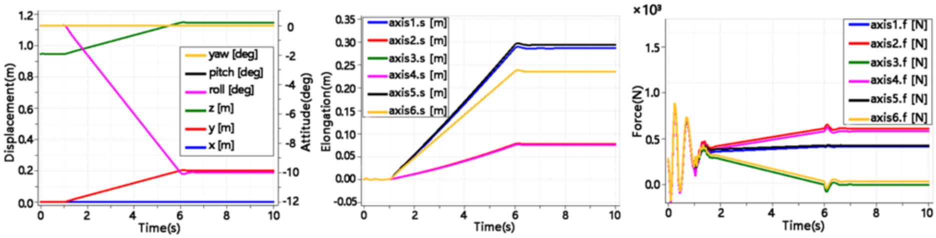
a) Movement Track Curve **b)** Linear Actuator Elongation Curve **c)** Linear Actuator Force Curve

Figure 8. Case 1 Simulation Result Curve



a) Movement Track Curve **b)** Linear Actuator Elongation Curve **c)** Linear Actuator Force Curve

Figure 9. Case2 Simulation Result Curve



a) Movement Track Curve **b)** Linear Actuator Elongation Curve **c)** Linear Actuator Force Curve

Figure 10. Case 3 Simulation Result Curve

5. Conclusion

This paper presents a Stewart platform system model established based on the Modelica language, integrating multiple disciplines such as mechanics, electrical engineering and control systems. It overcomes the limitation of single-domain robot simulation software, which can only simulate a single aspect of the system, allowing for a more comprehensive and complete representation of the system's overall characteristics. Furthermore, by Combining the MWorks.Sysplorer software with the rich model library of Modelica, different types of Stewart motion platforms can be rapidly analyzed and optimized, which provides an important research method for the study of complex multi-domain robotic systems.

References

[1] C. Tripodo, S. Lorenzi, A. Cammi, Object-oriented modeling, simulation and control of a 6-DoF parallel kinematic manipulator for remote handling in DONES facility, *Fusion Engineering and Design* **184** (2022).
[2] J.Y. Zhao, R.B. Zhang, L. Sun, Position Inverse Solution of Stewart Platform, *Chinese Hydraulics & Pneumatics***12** (2017):40-47.
[3] G. Bardaro, L. Bascetta, F. Casella, M. Matteucci, Using Modelica for advanced multi-body modelling, in: *Proceedings of the 12th International Modelica Conference* (2017) , 887–894.
[4] C.X. Wang, W.C. Luo, R.L. Liu, Multi-domain Modeling and Virtual Debugging of CNC Machine Tool Based on Modelica, *Modular Machine Tool & Automatic Manufacturing Technique***10** (2018), 102-105.
[5] J.J. Zhao, J.W. Ding, F.L. Zhou, L.P. Chen, Modelica and Its Mechanism of Multi-domain Unified Modeling and Simulation, Journal of System Simulation S2 (2006), 570-573.

Acknowledgment

The authors would like to acknowledge Beijing Polytechnic and Suzhou TongYuan Software&Control Technology Co., Ltd., and this study is supported by the Top Young Talents Program of Beijing Polytechnic (CJGX2024-028-14) and the Project of Cultivation for young top-notch Talents of Beijing Municipal Institutions.