

## Graphical Abstract

**End-to-End Design and Validation of a Low-Cost Stewart Robotic Platform with Nonlinear Control and Estimation**

Benedictus C. G. Cinun<sup>a,b</sup>, Tua A. Tamba<sup>b</sup>, Immanuel R. Santjoko<sup>a,b</sup>, Xiaofeng Wang<sup>c</sup>, Bin Hu<sup>a,d</sup>

## Highlights

### **End-to-End Design and Validation of a Low-Cost Stewart Robotic Platform with Nonlinear Control and Estimation**

Benedictus C. G. Cinun<sup>a,b</sup>, Tua A. Tamba<sup>b</sup>, Immanuel R. Santjoko<sup>a,b</sup>, Xiaofeng Wang<sup>c</sup>, Bin Hu<sup>a,d</sup>

- Developing a low-cost Stewart platform prototype
- Development of a real-time feedback linearization controller, with integration of LQR for optimal control performance
- Deployment of an Extended Kalman Filter (EKF) for robust state estimation using encoder and IMU data
- Implementation of the proposed control system in laboratory-scale prototype.

# End-to-End Design and Validation of a Low-Cost Stewart Robotic Platform with Nonlinear Control and Estimation

Benedictus C. G. Cinun<sup>a,b</sup>, Tua A. Tamba<sup>b</sup>, Immanuel R. Santjoko<sup>a,b</sup>, Xiaofeng Wang<sup>c</sup>, Bin Hu<sup>a,d</sup>

<sup>a</sup>*Department of Electrical and Computer Engineering, University of Houston, , Houston, 77004, TX, USA*

<sup>b</sup>*Department of Electrical Engineering, Parahyangan Catholic University, , Bandung, 40141, West Java, Indonesia*

<sup>c</sup>*Department of Electrical Engineering, University of South Carolina, , Columbia, 29208, SC, USA*

<sup>d</sup>*Department of Engineering Technology, University of Houston, , Houston, 77004, TX, USA*

---

## Abstract

This paper presents the design, construction, and control of a low-cost Stewart platform prototype, addressing the lack of affordable options for research and education compared to expensive commercial systems. The platform achieves full six-DOF motion using six linear actuators connecting a moving platform to a fixed base. A robust trajectory tracking controller based on feedback linearization compensates for the system's nonlinear dynamics, while an Extended Kalman Filter fuses actuator encoder and IMU measurements for accurate real-time state estimation. The proposed framework is validated through simulation and experimental tests on static and dynamic trajectories. Results demonstrate effective trajectory tracking and reliable state estimation under realistic sensor noise and disturbances, confirming the platform's suitability as an accessible research and educational tool.

**Keywords:** low-cost Stewart platform prototype, trajectory tracking controller, Extended Kalman Filter, state estimation

---

## 1. Introduction

The Stewart platform is a six-degree-of-freedom (6-DoF) parallel manipulator renowned for its high stiffness, precision, and ability to support large payloads relative to its size [1, 2, 3]. These properties make it well-suited for applications requiring precise motion and load-bearing capability, including flight simulators [4, 5], wave compensation in maritime operations [6, 7, 8], surgical robotics [9, 10], telescope positioning [11, 12], and more recently, fine adjustment of segmented mirrors on the James-Webb Space Telescope (JWST) [13]. The breadth of these applications highlights the platform's versatility and ongoing relevance in both industrial and research domains [14, 15, 16, 17, 18].

However, access to Stewart platforms for experimental research is often restricted by high costs. Commercial systems typically employ precision actuators, high-resolution sensors, and sophisticated onboard controllers, driving prices beyond the reach of many academic or small-scale laboratories. For example, the Stewart platform by Acrome, which includes the necessary controller and software, is priced at USD 7,499 [19], whereas large-scale platforms for applications such as flight simulation or heavy-duty testing can range from USD 12,000 [20] to EUR 50,000 [21]. In response, various low-cost prototypes have been developed using alternative actuation and sensing strategies to reduce manufacturing costs. Examples include a proof-of-concept educational platform [22] and a surgical motion simulator that use servo motors instead of linear actuators [10], as well as a Stewart platform-based haptic controller driven by hobby servo motors [23]. Another low-cost design [24] used several design changes to evaluate the perfor-

mance of 3D-printed components under simulated ocean wave conditions.

In addition to affordability, accurate state estimation is essential for enabling advanced control strategies and ensuring reliable operation of Stewart platforms. State estimation for Stewart platforms has been approached through several methods. Numerical techniques such as Newton–Raphson and Levenberg–Marquardt are commonly used to solve forward kinematics [25, 26, 27], but these iterative approaches are computationally intensive and less suited for real-time applications [25, 27]. Alternatives include sampling pose–actuator length data for use with a modified Denavit–Hartenberg formulation [28] and data-driven approaches such as machine learning [29], although the latter require large datasets for accurate results. The use of an indoor-GPS was also explored for increasing the pose accuracy of the Stewart platform [30].

Complementing state estimation, various control strategies have been proposed to exploit the platform's capabilities under different operational requirements. Control strategies for Stewart platforms range from classical PID [31, 32, 33] to advanced methods such as model predictive control and adaptive control based on kinematic or dynamic models [34]. Other designs have incorporated sliding-mode control with velocity feedforward compensation [7] or combined linear quadratic regulator (LQR) and active disturbance rejection control [35]. More recently, data-driven methods, including neural networks [36] and reinforcement learning [37], have been explored to enhance performance. While control strategies for Stewart platform are abundant, many low-cost Stewart platforms employ only kinematic controllers without accounting for the system dynamics, which are crucial when handling changes in dynamic model pa-

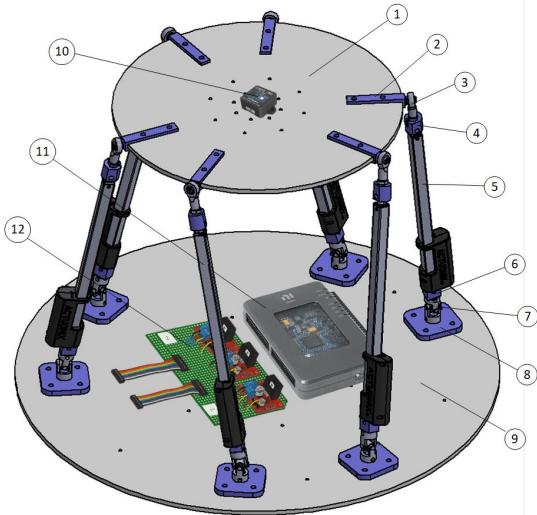


Figure 1: The design of the Stewart platform prototype. Each distinct part of the system is labeled with a corresponding number.

parameters such as component masses.

In this paper, we present a low-cost Stewart platform prototype designed to balance affordability with the capability to support advanced control research. The hardware combines off-the-shelf components with custom-made parts fabricated via computer-aided design (CAD) and low-cost manufacturing. The system software integrates data acquisition, the platform's dynamic model, and real-time control. A feedback linearization control scheme is implemented and integrated with an LQR controller for optimal tracking, and an Extended Kalman Filter (EKF) is developed for real-time state estimation from noisy sensor measurements.

The main contributions of this work are:

1. An end-to-end design methodology for a low-cost Stewart platform prototype, combining off-the-shelf components with custom-fabricated parts using 3D printing and non-commercial electronic boards, while achieving acceptable motion precision for research applications.
2. A real-time feedback linearization control scheme integrated with LQR for improved tracking performance.
3. An EKF-based state estimation method for real-time operation under noisy measurements.

Compared to existing studies, which often focus solely on theoretical modeling or controller design, this work provides a complete framework that integrates hardware development, control system design, and state estimation for a functional Stewart platform prototype.

The remainder of this paper is organized as follows. Section 2 describes the developed prototype of the Stewart platform. Section 3 derives the kinematic and dynamic models of the Stewart platform, and states the objective of the control system design. Section 4 proposes a feedback control framework which consists of an EKF-based estimator and a feedback linearization scheme. Section 5 presents detailed simulation and experimental results of the closed-loop Stewart robotic platform when subjected to varying desired poses. Finally, Section 6 provides concluding remarks and suggestions for future work.

## 2. Prototype Development and Description

The Stewart platform prototype discussed in this paper was developed using a combination of off-the-shelf and custom-made components to balance cost, functionality, and flexibility. Figure 1 illustrates the complete configuration of the developed Stewart platform prototype, highlighting several key components that are fully integrated. The circular shape platform ① is made from 5 mm acrylic material with high precision printing and serves as the end-effector with six degrees of freedom. It is connected via a custom-made 3D-printed platform socket ② to a POS 5 ASB bearing ③, which functions as a passive spherical joint. On the other hand, this bearing is also linked to the upper actuator adapter ④, which is attached to the rod of the linear actuator ⑤, acting as the primary driving mechanism of the system. The Stewart platform is actuated by Actuonix P16-200 linear actuators, allowing each leg to extend by up to 20 cm. Each actuator is equipped with a built-in encoder for direct leg length measurement. Prior to its operation, each actuator was subjected to a detailed manual calibration process to ensure an accurate and almost identical operational characteristic with other actuators. The lower part of the linear actuator is secured by the lower actuator adapter ⑥, which connects to a 10 mm universal joint ⑦, further inserted into the base socket ⑧ mounted firmly on the circular shape base ⑨, forming the fixed structural foundation.

At the center of the platform, an ROS IMU sensor ⑩ is installed to measure the orientation and angular velocity of the platform in real-time. The MyRIO-1900 board ⑪ is used as the data acquisition module to process sensor data and execute control algorithms in coordination with a Personal Computer (PC). The entire electronic system is supported by a custom custom-made Printed Circuit Board (PCB) ⑫, which integrates the motor driver circuits, the power supply, and the communication interface between the MyRIO board and the six linear actuators. The power and control signals for the actuators are managed by the PCB that integrate the L298N motor drivers. The electrical schematic of the custom-made PCB is shown in Figure 2, while its top-layer layout is presented in Figure 3.

The communication architecture of the Stewart platform as illustrated in Figure 4 shows how relevant signals flow between sensors, actuators, and the main control system. Inside the MyRIO board, LabView software is installed with two main components: a low level controller which handles real-time control tasks, and the Input and Output (I/O) Transmitter which manages signal communication to and from the PC. The PC is responsible for high-level control, including trajectory planning

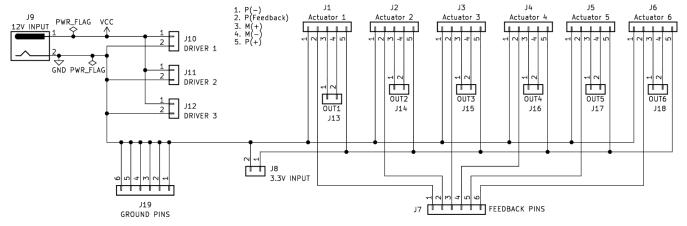


Figure 2: Electrical schematic of the custom-made PCB that interfaces the MyRIO with a motor driver and position feedback of the actuators.

and feedback processing, and features a Human-Machine Interface (HMI) that allows the operator to configure commands and monitor system states. The PC also includes an IMU decoder to process and interpret the orientation and motion sensor data.

The prototype consists of a moving platform with radius  $r_p = 16$  cm and a base with radius  $r_b = 20$  cm. The joints  $b_i$  and  $p_i$  are defined on these elements and symmetrically arranged in pairs, with the joints  $b_i$  defined in the base frame  $\{B\}$  and the joints  $p_i$  in the platform frame  $\{P\}$ . As illustrated in Figure 5, each of the three pairs of joints is centered at  $0^\circ$ ,  $120^\circ$ , and  $240^\circ$  angles on  $\{B\}$ , and at  $60^\circ$ ,  $180^\circ$ , and  $300^\circ$  degrees on  $\{P\}$ , with a  $\pm 20^\circ$  offset within each pair. The circular acrylic platform has an effective radius of 16 cm (including joint connectors), a total mass of  $m_p = 0.528$  kg, and the following moment of inertia:

$$I_p = \begin{bmatrix} 0.03 & 0.01 & 0.01 \\ 0.01 & 0.03 & 0.01 \\ 0.01 & 0.01 & 0.02 \end{bmatrix}. \quad (1)$$

Each leg of the platform comprises a linear actuator with a bottom (static) part of mass  $m_b = 0.1187$  kg and top (moving) part of mass  $m_t = 0.027$  kg. The center of mass of each part is assumed to be at its midpoint, giving distances of  $l_b = 0.13861$  m and  $l_t = 0.1$  m from the respective joints. The corresponding inertia for the top and bottom parts are given as follows:

$$I_t = \begin{bmatrix} \frac{1}{3}m_t l_t^2 & 0 & 0 \\ 0 & \frac{1}{3}m_t l_t^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I_b = \begin{bmatrix} \frac{1}{3}m_b l_b^2 & 0 & 0 \\ 0 & \frac{1}{3}m_b l_b^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (2)$$

These physical parameters and signal configurations form the foundation for the modeling and controller design processes of the developed Stewart platform prototype.

Table 2 lists the Bill of Materials (BoM) for the developed Stewart platform prototype. All components in the table are commercially available, and the listed prices have been converted to US dollars.

### 3. System Modeling and Problem Formulation

This section presents the kinematic and dynamic modeling of the Stewart platform. The kinematic analysis defines the relationship between the platform's pose and the length of the legs, while the dynamic analysis describes the equations of motion of the platform to be used in the controller design.

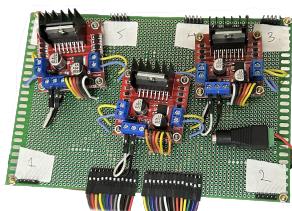


Figure 3: Top-layer layout (front view) of the custom-made PCB, illustrating the connector arrangement for actuators, sensor inputs, and power distribution.

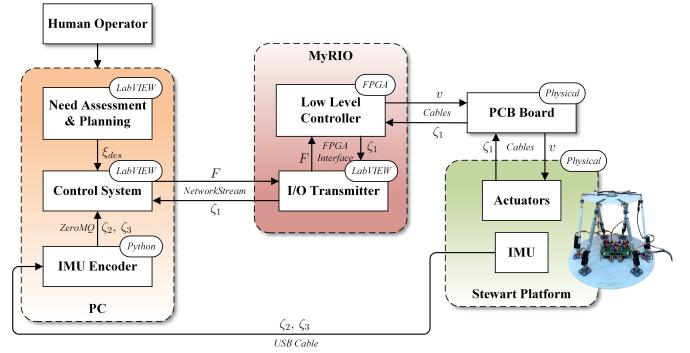


Figure 4: System configuration of the Stewart platform.

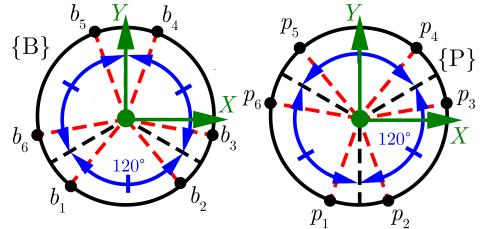


Figure 5: Joint configurations of the Stewart platform: joint  $b_i$  positions on  $\{B\}$  (left) and joints  $p_i$  position on  $\{P\}$  (right).

#### 3.1. Kinematic Model

Kinematic analysis studies how the components of a mechanism move relative to each other, without considering the forces that cause the motion [38]. In robotic systems, kinematic analysis is typically divided into Forward Kinematics (FK) and Inverse Kinematics (IK). FK determines the pose of the end-effector from known joint variables, while IK determines the required joint states to achieve a desired end-effector pose. For parallel manipulators such as the Stewart platform, computing the solution of IK is simpler than that of the FK due to the closed-loop structure of the system [25, 39, 40]. In this work, IK analysis is performed to compute the required lengths of the legs to achieve a specified pose of the platform.

This paper considers the kinematic structure of Stewart platform in Figure 6(a) with three main structural elements: a fixed *base* with an attached base frame  $\{B\}$ , a moving *platform* with a platform frame  $\{P\}$ , and *legs* formed by six linear *actuators* that connect the base to the platform. The prototype developed from these elements is shown in Figure 6(b). The points of attachment of the actuator on the base are denoted as  $b_i$  and those on the platform as  $p_i$ , where  $i = 1, \dots, 6$ . These points serve as passive joints that mechanically connect the legs to the platform and the base, allowing relative motions to occur. The joints in  $b_i$  are universal joints, while those in  $p_i$  are spherical joints.

The structure of the Stewart platform enables six DoF motions of the platform with respect to (w.r.t.) the base, consisting of translational movements along the  $X$ ,  $Y$  and  $Z$  axes, as well as rotational movements about these same axes. The translation of  $\{P\}$  w.r.t.  $\{B\}$  is described by  $t = [X', Y', Z']^T$ , while the orientation is represented by the Euler angle vector  $r = [\phi, \theta, \psi]^T$ . Here,  $\phi$  denotes roll (rotation about the  $X$ -axis),  $\theta$  denotes pitch (rotation about the  $Y$ -axis), and  $\psi$  denotes yaw (rotation about the  $Z$ -axis). The vector of generalized pose of the Stewart plat-

Table 1: The BoM of the Stewart platform prototype.

No	Components	Quantity	Unit Price	Total
1	Actuonix P-16P Linear Actuator	6	\$ 90.00	\$ 540.00
2	L298N Motor Drivers	3	\$ 0.85	\$ 2.56
3	Acrylic (Platform)	1	\$ 5.25	\$ 5.25
4	Acrylic (Base)	1	\$ 14.03	\$ 14.03
5	Universal Joint	6	\$ 7.32	\$ 43.92
6	Rod End Bearing PHS5	6	\$ 0.64	\$ 3.84
7	National Instrument MyRIO 1900	1	\$ 1,156.26	\$ 1,156.26
<b>Total</b>				\$ 1,765.85

form is thus defined as

$$q = [t, r]^\top = [X', Y', Z', \phi, \theta, \psi]^\top \in \mathbb{R}^6. \quad (3)$$

The orientation of the platform is further described by the rotation matrix  $R \in \text{SO}(3)$ , which maps the vectors from  $\{\text{P}\}$  to  $\{\text{B}\}$ . The matrix  $R$  is computed from the Euler angles  $r$  using the ZYX (yaw–pitch–roll) convention as follows [41]:

$$R = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi - c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \quad (4)$$

where  $s_\theta := \sin(\theta)$  and  $c_\phi := \cos(\phi)$ .

The generalized velocity of the platform is defined as  $\dot{q} = [\dot{t}, \dot{r}]^\top$ , where  $\dot{t} = [X', Y', Z']^\top$  is the linear velocity of the platform, and  $\dot{r} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^\top$  is the time derivatives of the Euler angles [31, 37]. The angular velocity expressed w.r.t.  $\{\text{P}\}$  is denoted by  $\omega_p = [\omega_{px}, \omega_{py}, \omega_{pz}]^\top$  and is defined as follows.

$$\omega_p = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (5)$$

The angular velocity in frame  $B$  can be rewritten as follows.

$$\tilde{\omega} = R\tilde{\omega}_p R^\top, \quad (6)$$

where  $\tilde{\omega}$  and  $\tilde{\omega}_p$  are the skew-symmetric matrices that correspond to the angular velocity vectors  $\omega$  and  $\omega_p$ , respectively.

Using the defined generalized variables, the kinematic relationship between the base, the platform, and the connecting legs can now be established. Figure 6(a) illustrates the geometric and vector representations of the Stewart platform used for this

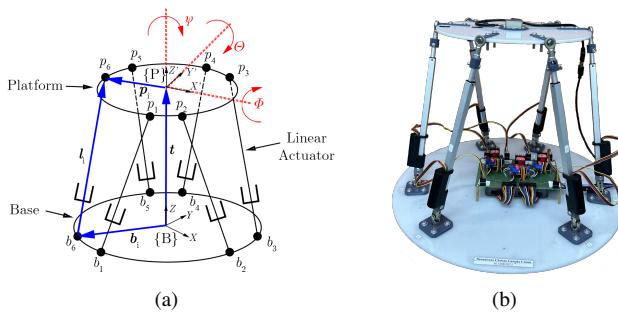


Figure 6: The schematic (a) and developed prototype (b) of Stewart platform.

purpose. In this figure,  $b_i$  denotes the position vector from the origin of  $\{\text{B}\}$  to the base attachment point of the  $i$ -th leg, while  $p_i$  denotes the corresponding vector from the origin of  $\{\text{P}\}$  to the platform attachment point. For given platform's pose  $q$ , the inverse kinematics problem of the Stewart platform can be solved. This involves using the known platform pose and the geometric configuration of the mechanism to determine the required leg lengths. Specifically, the position and orientation of the platform, along with the fixed locations of the joints, are used to calculate the vector form of each leg. The vector of the  $i$ -th leg is given as follows [39, 13]:

$$l_i = t + R p_i - b_i, \quad (7)$$

where  $l_i$  is the leg vector expressed w.r.t.  $\{\text{B}\}$ . The corresponding scalar-valued length of each leg is defined by the Euclidean norm of  $l_i$  as follows:

$$s_i = \|l_i\|. \quad (8)$$

This IK formulation provides the necessary length that each leg must achieve to realize a given pose of the platform.

### 3.2. Dynamics Model

The dynamic model of the Stewart platform is derived by accounting for the dynamics of both the moving platform and the connecting legs. The dynamics of the legs are formulated using Euler–Lagrange method, while the dynamics of the platform are modeled using Newton–Euler formulation [39]. This combined model is essential to accurately capture the coupled motion of the system and serves as the foundation for implementing model-based control strategies. Moreover, the dynamics are expressed in task space, enabling a direct relationship between the actuator forces and the platform motion in terms of position and orientation. This formulation is particularly valuable for the simulation and real-time control of the physical prototype.

#### 3.2.1. Legs Dynamics Analysis

In the prototype, each leg is realized using a linear actuator that functions as a prismatic joint. Each actuator is composed of two primary rigid components: a fixed section, referred to as the bottom part that is mounted to the base, and a movable section, referred to as the top part that is connected to the moving platform. The bottom and top parts have masses of  $m_b$  and

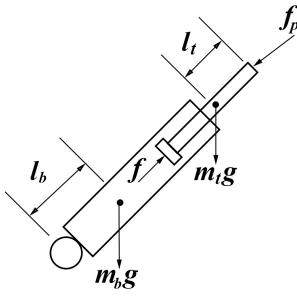


Figure 7: The free-body diagram of the linear actuator used as a leg of the Stewart platform (adapted and modified from [39]).

$m_t$ , respectively, and their centers of mass are located at distances  $l_b$  and  $l_t$  from the base and platform joints, respectively. These components are subjected to external forces including gravity (represented by the gravitational acceleration vector  $g$ ), the actuator-generated force  $f$ , and the constraint force  $f_p$  acting on the platform joint. Figure 7 illustrates the force distribution and structural configuration of the actuator which form the basis for the dynamic modeling of each leg.

To determine the force  $f_p$  on the  $i$ -th leg (defined as  $f_{pi}$ ), a dynamic analysis is used based on Euler–Lagrange equation that takes into account the kinetic and potential energies of the actuator’s components. The resulting expression for the force  $f_{pi}$  is defined as follows:

$$f_{pi} = (\mathbf{M}_1 + \mathbf{M}_2)_i \begin{bmatrix} \mathbf{I}_{3 \times 3} & R\tilde{p}_i^\top R^\top \end{bmatrix} \ddot{q} + \mathbf{C}_{ai} \begin{bmatrix} \mathbf{I}_{3 \times 3} & R\tilde{p}_i^\top R^\top \end{bmatrix} \dot{q}, \\ + (\mathbf{M}_1 + \mathbf{M}_2)_i \tilde{\omega}^2 R p_i - (Q_f + Q_{mg} + Q_{mbg})_i, \quad (9)$$

where  $\tilde{p}_i$  is the skew-symmetric matrix of  $p_i$ , while the detailed expressions for the mass terms  $\mathbf{M}_1$  and  $\mathbf{M}_2$ , the Coriolis term  $\mathbf{C}_a$ , and the generalized force components  $Q_f$ ,  $Q_{mg}$  and  $Q_{mbg}$  are given in the Appendix. The readers may refer to [39] for detailed derivation of (9). In essence, (9) relates the force applied by each actuator to the dynamic motion of the platform.

### 3.2.2. Platform Dynamics Analysis

The first step in the derivation of the platform dynamics is to determine the control point of the platform. This point does not always coincide with the origin of  $\{\mathbf{P}\}$ , but can change depending on the position of the loads on its top. The vector of the platform’s control point w.r.t.  $\{\mathbf{B}\}$  can be written as follows:

$$\mathbf{q}_c = t + R\mathbf{c}_p, \quad (10)$$

where  $\mathbf{c}_p$  is the position vector of the control point w.r.t.  $\{\mathbf{P}\}$ . The acceleration of the control point  $\mathbf{q}_c$  satisfies (11).

$$\ddot{\mathbf{q}}_c = [\mathbf{I} \quad R\tilde{c}_p^\top R^\top] \ddot{q} + \tilde{\omega}^2 R \mathbf{c}_p. \quad (11)$$

Assuming equilibrium forces are acting on the platform, the evaluation of Newton’s law on the platform gives the following:

$$m_p \ddot{\mathbf{q}}_c = - \sum_{i=1}^6 f_{pi} + m_p g, \quad (12)$$

where  $m_p$  is the platform’s mass. By substituting (11) into (12), the platform dynamics w.r.t.  $\{\mathbf{B}\}$  satisfies

$$[m_p \mathbf{I} \quad m_p R \tilde{c}_p^\top R^\top] \ddot{q} + m_p \tilde{\omega}^2 R \mathbf{c}_p = - \sum_{i=1}^6 f_{pi} + m_p g. \quad (13)$$

Combining the dynamic models of the legs and the platform, a compact expression of the dynamic model of the Stewart platform in the task space can be formulated as the following Equations of Motion (EoM) [39, 31]:

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(q)(\mathbf{H}(q)\mathbf{F} - \mathbf{C}(q, \dot{q})\dot{q} - \mathbf{G}(q)), \quad (14)$$

where  $\mathbf{M}(q) \in \mathbb{R}^{6 \times 6}$  is the inertia matrix,  $\mathbf{C}(q, \dot{q}) \in \mathbb{R}^{6 \times 6}$  accounts for Coriolis and centrifugal effects,  $\mathbf{G}(q) \in \mathbb{R}^6$  is the gravity vector, and  $\mathbf{H}(q) \in \mathbb{R}^{6 \times 6}$  is the inverse transpose of the Jacobian matrix that maps the forces on the actuators to generalized forces in the task space. Finally,  $\mathbf{F} \in \mathbb{R}^6$  denotes the force inputs generated by actuators. The detailed derivation of (14) is available in [39] and summarized in the Appendix.

### 3.3. Problem Formulation

Given the EoM for the dynamics of the Stewart platform in (14), our objective is to construct a feedback controller that can move the platform from an initial pose to a desired final pose. To this end, let  $\xi = [q^\top, \dot{q}^\top]^\top$  be the vector of state variables of the platform. One main challenge in designing a feedback controller for (14) is that these state variables cannot be measured directly. In this regard, an estimator should first be developed to provide estimates of the state variables based on measurement data obtained from available sensors. In this paper, the estimator is developed using EKF algorithm, while the controller is developed using the combination of feedback linearization scheme and LQR control method. These estimation and controller design are detailed in the next section.

## 4. Estimation and Control Systems Design

This section presents the design of a controller for the developed Stewart platform using the integration of a feedback linearization scheme and the LQR method. Since the platform state variables cannot be measured directly by the available sensors, an EKF algorithm is designed to calculate the estimate of these state variables and then used in the proposed feedback control strategy. The integration of the feedback controller and the EKF-based estimation scheme helps to achieve more accurate motion execution with low computational demand. This makes the framework suitable for real-time deployment on the Stewart platform prototype.

### 4.1. Feedback Linearization Control Scheme

To achieve a precise motion of the Stewart platform, a practical control strategy based on the feedback linearization scheme is implemented. This scheme essentially transforms the dynamics into a linear system model that is suitable for the linear control design framework [42]. For the EoM of the Stewart platform in (14), a feedback linearizing force control input  $\mathbf{F}$  of the

form (15) is designed to cancel the system nonlinearities:

$$F := H^{-1}(q)(M(q)u + C(q, \dot{q})\dot{q} + G(q)), \quad (15)$$

where  $u$  is a virtual control input to be designed for the resulting linear system. Substituting the force control input (15) into (14) results in a double integrator model  $\ddot{q} = u$  which can be written as the following Linear Time-Invariant (LTI) system:

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0_{3 \times 3} \\ I_{3 \times 3} \end{bmatrix} u. \quad (16)$$

Using  $\xi = [q, \dot{q}]^\top$  as the state variables of the system, (16) can be rewritten as the following compact LTI system model

$$\dot{\xi} = A\xi + Bu. \quad (17)$$

The LTI system model (17) allows for the design of a linear feedback control law of the form

$$u = -K\{\xi - \xi_{des}\}, \quad (18)$$

where  $\xi_{des}$  is the desired state variable and  $K$  is the feedback control gain. In this paper, the feedback control  $u$  is designed using the well-known LQR scheme to ensure the minimization of quadratic performance index  $J(\xi, u)$  as follows [43]:

$$u := \arg \min J(\xi, u), \text{ where } J = \int_0^t (\xi^\top N \xi + u^\top O u) dt, \quad (19)$$

where  $N \in \mathbb{R}^{12 \times 12}$  and  $O \in \mathbb{R}^{6 \times 6}$  are the state and control input weighting matrices, respectively, which directly influence the system's behavior:  $N$  penalizes deviations from the desired state to achieve accurate tracking, while  $O$  regulates the control effort to ensure smooth operation by preventing excessive actuator efforts. The optimal feedback gain  $K$  in (18) is calculated as  $K = O^{-1}B^\top D$ , where  $D$  is a positive semidefinite matrix which satisfies the continuous-time algebraic Riccati equation of the form:  $A^\top D + DA - DBR^{-1}B^\top D + N = 0$ .

#### 4.2. State Estimation Using EKF

The implementation of control methods in Section 4.1 requires feedback information about the state variables from sensors. However, the Stewart platform prototype developed in this research utilizes three types of measurement that do not directly measure the state variables. The first measurement is obtained from a built-in encoder embedded in each linear actuator which measures the length of each leg on the platform. The encoder measurement provides indirect information on the pose of the platform, which can be calculated using the inverse kinematics relationship in (8). The second measurement is given by an inertial measurement unit (IMU) mounted on the moving platform to provide real-time rotational data which capture the orientation state  $r$ . The last measurement is also obtained from the IMU and provides  $\omega_p$  in (5). With this set of measurements, the output function  $\gamma$  of the Stewart platform can be written as the following equation:

$$\gamma = [s_1, s_2, s_3, s_4, s_5, s_6, \phi, \theta, \psi, \omega_{px}, \omega_{py}, \omega_{pz}]^\top + v, \quad (20)$$

where  $v \sim \mathcal{N}(0, V) \in \mathbb{R}^{12}$  is a vector of Gaussian measurement noise with mean zero and covariance matrix of  $V \in \mathbb{R}^{12 \times 12}$ .

Using the output measurement (20), an EKF algorithm is implemented to calculate the estimate of the state variables  $\hat{\xi}$  that will be used to implement the control algorithm in Section 4.1. The EKF is developed using the discrete-time version of the system model in (16) and the output measurement in (20). In particular, by using the forward Euler method for a sampling period of  $\Delta t$ , the discretization of (16) at each discrete time instant  $k \in \mathbb{N}$  of the following form is considered:

$$\begin{aligned} \begin{bmatrix} q_{k|k-1} \\ \dot{q}_{k|k-1} \end{bmatrix} &= \begin{bmatrix} I_{3 \times 3} & \Delta t I_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} q_{k-1|k-1} \\ \dot{q}_{k-1|k-1} \end{bmatrix} + \begin{bmatrix} 0_{3 \times 3} \\ \Delta t I_{3 \times 3} \end{bmatrix} u_{k-1} + w_k, \\ \hat{\xi}_{k|k-1} &= A_d \hat{\xi}_{k-1|k-1} + B_d u_{k-1} + w_k, \end{aligned} \quad (21)$$

where  $w_k \in \mathbb{R}^{12} \sim \mathcal{N}(0, W)$  is an assumed Gaussian process noise with mean zero and covariance matrix of  $W \in \mathbb{R}^{12 \times 12}$ . The prediction model in (21) essentially models the prior state estimate  $\hat{\xi}_{k|k-1}$  as the function of the previous state estimate  $\hat{\xi}_{k-1|k-1}$  and the control input  $u_{k-1}$ . The implementation of the EKF also uses the Jacobian matrix  $\Gamma_k$  of the measurement function (20), evaluated in the forecast state estimate  $\hat{\xi}_{k|k-1}$  as follows

$$\Gamma_k = \frac{\partial \gamma}{\partial \xi} \Big|_{\xi=\hat{\xi}_{k|k-1}}. \quad (22)$$

Algorithm 1 summarizes the EKF that is used to estimate the Probability Density Function (PDF) of the state estimate  $\hat{\xi}$  and the state covariance matrix  $P \in \mathbb{R}^{12 \times 12}$ . Each iteration of the EKF method consists of two stages, namely *Prediction* and *Measurement-Update*. In the *Prediction* stage, the PDF is propagated forward from the prior PDF according to

$$\begin{aligned} \hat{\xi}_{k|k-1} &= A_d \hat{\xi}_{k-1|k-1} + B_d u_{k-1}, \\ P_{k|k-1} &= A_d P_{k-1|k-1} A_d^\top + V. \end{aligned} \quad (23)$$

The *Measurement-update* stage then updates the forecast PDF into a posterior PDF by incorporating the latest observation  $z_k$  provided in the output vector. The innovation vector  $\tilde{z}_k$  in (24) is constructed in this stage and is defined as the difference between the output vector  $z_k$  and the predicted measurement of the output evaluated at the predicted state estimate  $\hat{\xi}_{k|k-1}$

$$\tilde{z}_k = z_k - \gamma \Big|_{\xi=\hat{\xi}_{k|k-1}}. \quad (24)$$

Finally, the posterior state estimate  $\hat{\xi}_{k|k}$  and the posterior covariance  $P_{k|k}$  are calculated by incorporating the innovation vector

$$\begin{aligned} \hat{\xi}_{k|k} &= \hat{\xi}_{k|k-1} + P_{k|k-1} \Gamma_k^\top (\Gamma_k P_{k|k-1} \Gamma_k^\top + W)^{-1} \tilde{z}_k, \\ P_{k|k} &= (I - K_k \Gamma_k) P_{k|k-1}. \end{aligned} \quad (25)$$

The EKF in Algorithm 1 begins with an *Initialization* phase where the initial state  $\hat{\xi}_{0|0}$  is estimated from encoder data using numerical forward kinematics [27], and the initial covariance  $P_{0|0}$  is chosen to be large enough to reflect initial uncertainty.

#### 4.3. Overall Estimation & Control Systems Architecture

The overall architecture of the estimation and control systems

---

**Algorithm 1** Pseudo code of the EKF.

---

```

1: procedure INITIALIZATION
2:   At  $k = 0$ , initialize  $\hat{\xi}_{0|0}$ ,  $P_{0|0}$ 
3:   return  $\hat{\xi}_{0|0}$ ,  $P_{0|0}$ 
4: At each  $k$ , repeat the following procedures:
5:   procedure PREDICTION
6:     Compute forecast PDF data  $\xi_{k|k-1}$  and  $P_{k|k-1}$  using (23)
7:   return  $\hat{\xi}_{k|k-1}$ ,  $P_{k|k-1}$ 
8:   procedure MEASUREMENT-UPDATE
9:     Compute measurement jacobian matrix  $\Gamma_k$  using (22)
10:    Compute information vector  $\tilde{z}$  using (24)
11:    Compute posterior PDF data  $\xi_{k|k}$  and  $P_{k|k}$  using (25)
12:   return  $\hat{\xi}_{k|k}$ ,  $P_{k|k}$ 

```

---

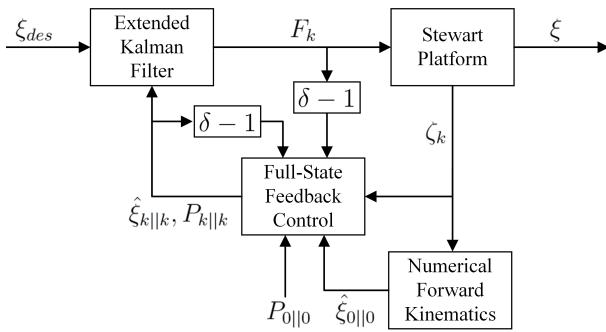


Figure 8: Block diagram of the proposed estimation and control scheme.

for the Stewart platform is shown in Figure 8. In this figure, the desired state variable  $\xi_{des}$  serves as the reference input to the feedback control module, which in turn computes the desired forces  $F$  that drive the current vector of state variables of the Stewart platform to the desired one. The estimate of the state variables used in the controller design is obtained using the EKF module based on measurement data  $\xi_k$  obtained at discrete time step  $k$  from the leg encoders and the IMU sensor. The EKF also incorporates the initial state estimates  $\hat{\xi}_{0|0}$  and their covariance  $P_{0|0}$  calculated using the combination of numerical forward kinematics method and the measured encoder data [27]. The posterior estimates of the state variables  $\hat{\xi}_{k|k}$  and their covariance  $P_{k|k}$  from the EKF are then fed back to the controller module. The proposed estimation and feedback loop allow the controller to adjust its commands based on the real-time estimate of the state of the platform, which eventually enables robust and accurate control action of the platform motion.

## 5. Simulation and Experimental Results

This section presents the simulation and experimental results of the implementation of the proposed feedback control scheme on the Stewart platform's model and prototype. The initial condition of the platform is set to be:  $\xi_0 = [0, 0, 0.32, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ , making the platform horizontally parallel and centered to the base. The performance of the controller is tested under two desired motion scenarios of the platform, namely (a) "step" motion and (b) "dancing" motion.

The "step" scenario was conducted for a duration of 60 seconds, while the "dancing" scenario lasted 20 seconds. Both scenarios used the same sampling interval of 0.01 seconds.

The "step" motion scenario is designed to evaluate the platform's ability to track static desired states as defined in (26).

$$\begin{aligned} q_{des} &= [x_{des}, y_{des}, z_{des}, \phi_{des}, \theta_{des}, \psi_{des}]^T, \\ \dot{q}_{des} &= [0, 0, 0, 0, 0, 0]^T. \end{aligned} \quad (26)$$

In particular, state variables  $q$  are required to follow a piecewise function whose value remained unchanged for a certain period of time. The  $z_{des}$  is set to be a constant value of 0.4 meter, while the other components are defined over time  $t$  as follows.

$$\begin{aligned} x_{des} &= \begin{cases} 0.075 & \text{if } 10 \leq t < 20 \\ 0 & \text{otherwise} \end{cases}, & y_{des} &= \begin{cases} 0.075 & \text{if } 20 \leq t < 30 \\ 0 & \text{otherwise} \end{cases}, \\ z_{des} &= 0.4, & \phi_{des} &= \begin{cases} 0.15 & \text{if } 30 \leq t < 40 \\ 0 & \text{otherwise} \end{cases}, \\ \theta_{des} &= \begin{cases} 0.15 & \text{if } 40 \leq t < 50 \\ 0 & \text{otherwise} \end{cases}, & \psi_{des} &= \begin{cases} 0.15 & \text{if } 50 \leq t \leq 60 \\ 0 & \text{otherwise} \end{cases}, \end{aligned}$$

The "dancing" motion scenario is designed to evaluate the platform's ability to track a dynamic trajectory in (27).

$$\begin{aligned} q_{des} &= [0, 0, 0.4, 0.1 s_t, 0.1 c_t, 0]^T, \\ \dot{q}_{des} &= [0, 0, 0.4, 0.1 c_t, -0.1 s_t, 0]^T. \end{aligned} \quad (27)$$

A video demonstration of the presented results is given in [44].

### 5.1. Simulation Results

For the EKF implementation, the covariance matrix of the process noise is set to  $V = \text{diag}(1, 1, 1, 1, 1, 1, 5, 5, 5, 5, 5)$  while the covariance matrix of the measurement noise is set to  $W = \text{diag}(10, 10, 10, 10, 10, 10, 1, 1, 1, 3, 3, 3)$ . The initial value of the error covariance matrix is set to  $P_0 = 10^{-2} \text{diag}(100, 100, 100, 100, 100, 100, 1, 1, 1, 1, 1, 1)$ . For the LQR controller, the state weighting matrix is chosen as  $N = \text{diag}(30, 30, 5, 30, 30, 200, 3, 3, 1, 3, 3, 20)$ , while the control input weighting matrix is set to  $O = 10I_6$ . These parameters were empirically tuned through several simulation trials to achieve desirable estimation and tracking performance. For the estimation problem, the objective is to ensure the convergence of the state estimation error  $e_l$  according to (28).

$$\lim_{t \rightarrow \infty} e_l \leq r_l, \quad e_l = \|\hat{\xi} - \xi\|. \quad (28)$$

For the control problem, the objective is to ensure the convergence of the tracking error  $e_t$  according to (29).

$$\lim_{t \rightarrow \infty} e_t \leq r_t, \quad e_t = \|\xi - \xi_{des}\|. \quad (29)$$

In (28)-(29),  $r_l$  and  $r_t$  are some specified positive constants that were set in the simulations to be  $r_l = 0.02$  and  $r_t = 0.01$ .

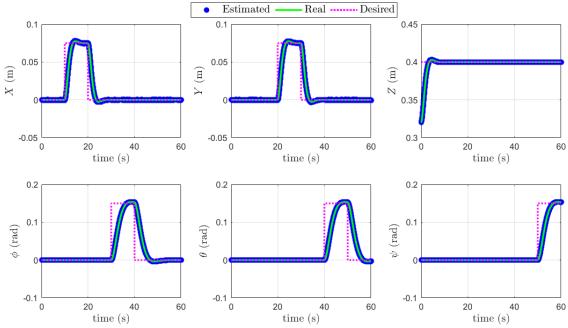


Figure 9: Simulated position variables of the "step" motion scenario.

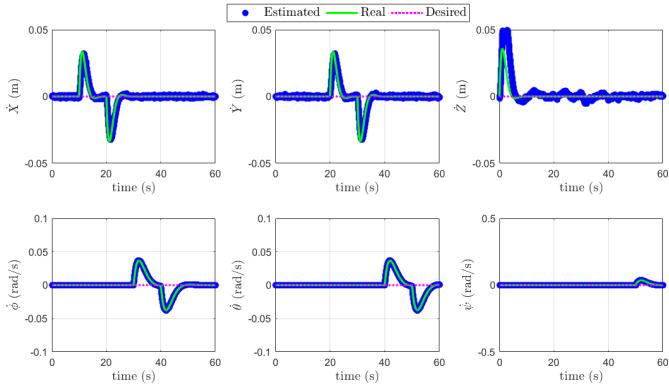


Figure 10: Simulated velocity variables of the "step" motion scenario.

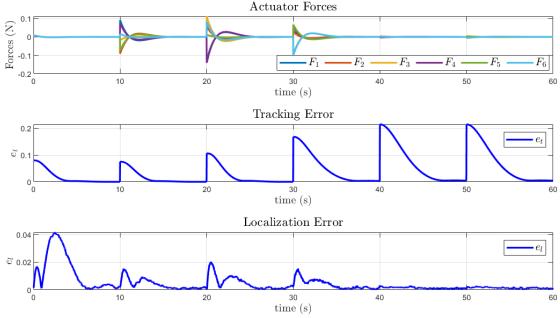


Figure 11: Simulated force control input of the "step" motion scenario.

### 5.1.1. Simulation Result of "Step" Motion

The trajectories of the position and velocity of the platform for this scenario are shown in Figure 9 and Figure 10, respectively, and the corresponding control input and estimation error are shown in Figure 11. In these figures, the actual states are represented by the green lines, whereas the estimated states are plotted in blue dots. The close similarity between the actual and estimated states confirms the effectiveness of the proposed estimation scheme in capturing the system dynamics. As can also be observed in Figure 9, the calculated force control input drives the system to follow the desired pose.

Figure 11 shows that the estimation error  $e_l$  decreases as the state variables reach the desired values with a final value of approximately 0.0012. In addition, the tracking error  $e_t$  shown in this same figure also decreases and eventually converges to

values within the acceptable limit bound. These results demonstrate that the proposed scheme can be applied to the static desired trajectory case.

### 5.1.2. Simulation Result of "Dancing" Motion

The state trajectories of the platform for this scenario are shown in Figure 12, while the corresponding control inputs and the estimation error are shown in Figure 13. The actual states variables are shown as the green lines, while the estimated values are plotted as blue dot line. The estimation scheme effectively captures the dynamics of the system, as evidenced by the close alignment between the estimated and actual states. Furthermore, the plots of variables  $\psi$ ,  $\theta$ , and velocity in Figure 12 demonstrate that the generated force control input successfully drives the platform toward the desired pose. In general, these results demonstrate the effectiveness of the proposed estimation and control method for time-varying reference trajectory.

Compared to the "step" motion scenario, the tracking error in the "dancing" scenario exhibits smoother variations and eventually converges to approximately 0.13. While this value exceeds the tolerance defined in (29), it is attributed to the continuously changing desired states and the accumulation of error across all system states. Moreover, the estimation error also converges to approximately 0.0037, which is well below the tolerance value of  $r_l = 0.01$ . It can be concluded from these results that the platform is able to keep up with varying desired poses using the proposed estimation and control methods.

## 5.2. Experimental Results

Due to the inherent challenges in directly measuring the true state variables in real-time, the experimental results are primarily compared against the outcomes of similarly tuned simulations. This comparative analysis will allow for an assessment of the prototype's performance and the effectiveness of the proposed algorithms in a real-world scenarios, despite the absence of a perfectly known ground truth for the platform's state. In particular, the main control objective in the experiment is to achieve the condition below:

$$\lim_{t \rightarrow \infty} e_{cs} \leq r_{cs}, \quad e_{cs} = \|\xi_{des} - \hat{\xi}\|, \quad (30)$$

where  $r_{cs}$  is a specified positive constant.

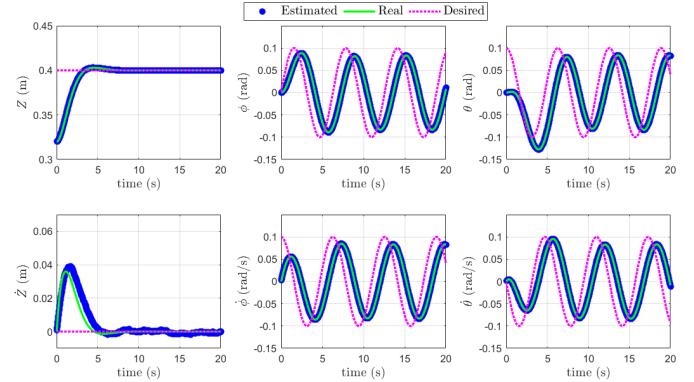


Figure 12: Simulated state variables of the "dancing" motion scenario.

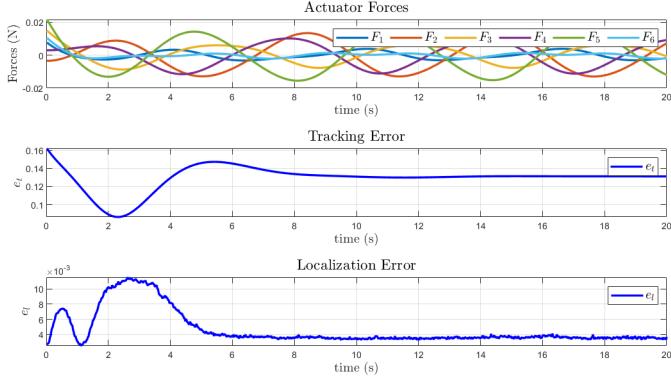


Figure 13: Simulated force control input of the "dancing" motion scenario.

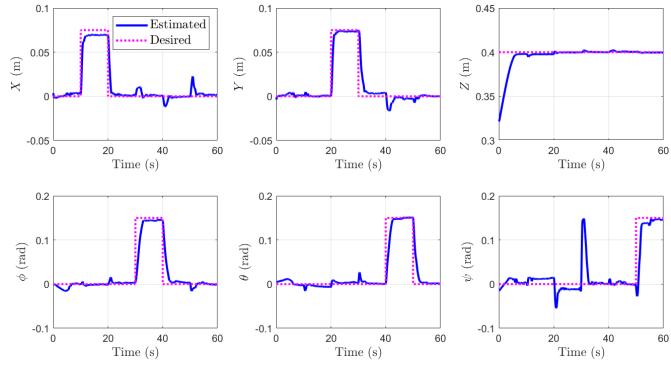


Figure 14: Experimental position variables of the "step" motion scenario.

### 5.2.1. Experiment Result of "Step" Motion

The position and velocity trajectories of the platform in the experiment are shown in Figure 14 and Figure 15, respectively. In both figures, the estimated states are shown by blue lines, while the desired trajectories are shown by purple dotted lines. The generated force control input and the tracking error are shown in Figure 16. It can be seen in these figures that the platform's position and orientation estimates closely follow the desired values, indicating the good performance of the proposed EKF-based state estimation scheme. The results show that the platform can follow changes in the trajectory along the linear axes. While the  $X$  and  $Y$  axes exhibit noticeable fluctuations, the estimated states still capture the general shape and timing of the desired trajectories. Although the  $Z$ -axis shows only minor fluctuations, those in the  $X$  and  $Y$  directions may appear more pronounced due to differences in plot scale. Small disturbances are also observed in the platform's orientation. These fluctuations are expected in experimental settings and do not significantly impact the overall trajectory tracking performance.

The estimated velocity of the platform shows more noticeable fluctuations compared to the position estimates. Although these fluctuations are relatively minor in the linear velocity components ( $\dot{X}$ ,  $\dot{Y}$ ,  $\dot{Z}$ ), they become more pronounced in the angular velocity estimates ( $\dot{\phi}$ ,  $\dot{\theta}$ ,  $\dot{\psi}$ ). At certain time intervals, the velocity profiles exhibit overshoot, which is primarily due to rapid changes in the desired position. These overshoots occur as the platform moves toward the new desired pose before coming to rest and stabilizing. This behavior is expected, as the

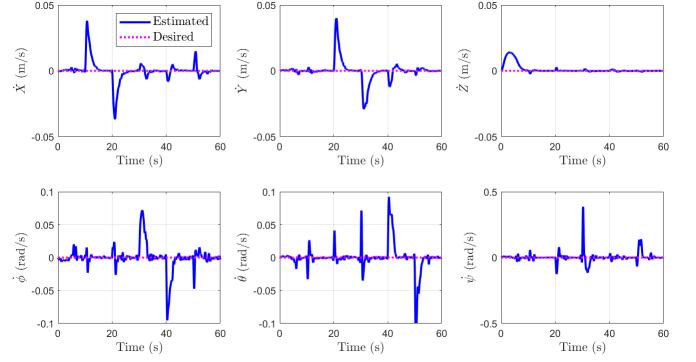


Figure 15: Experimental velocity variables of the "step" motion scenario.

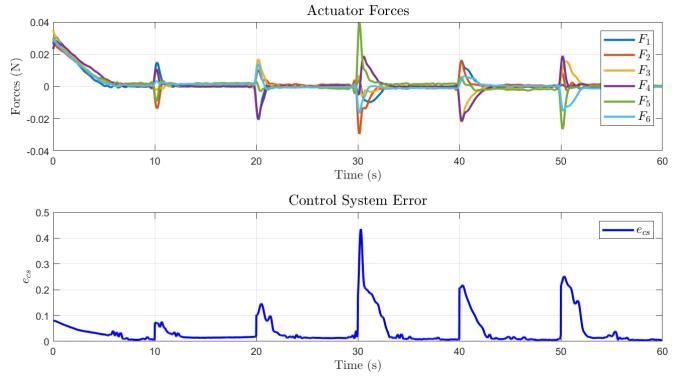


Figure 16: Experimental force control input of the "step" motion scenario.

system must generate sufficient velocity to reach the target pose promptly and accurately.

These results demonstrate that the controller successfully generates the necessary force profiles to reach and stay in the desired states, effectively tracking step-like commands. Despite fluctuations in both position and velocity estimates, the actuator forces remain bounded and exhibit no signs of instability or oscillatory behavior. This indicates that the controller is robust to moderate levels of sensor noise without significant degradation in performance. The consistent and stable force outputs further support the effectiveness of the proposed control scheme, particularly for static reference tracking tasks.

### 5.2.2. Experiment Result of "Dancing" Motion

Figure 17 shows the actual and desired state variables of the closed-loop system where the estimated states are plotted as blue lines along with the desired states that are plotted as purple dotted lines. Only state variables that are subjected to changes in the desired trajectory are shown; the remaining axes are omitted for clarity. The corresponding force control input and error observed during the experiment are shown in Figure 18.

For the angular motion axes  $\phi$  and  $\theta$  where dynamic trajectories were applied, the estimated states closely follow the desired values. The platform's orientation starts from zero and attempts to reach the desired values within specific time steps. As shown in Figure 17, there is a transient period before the platform finally achieves the target orientation. Although some deviations are present, the overall response of the system is consistent with

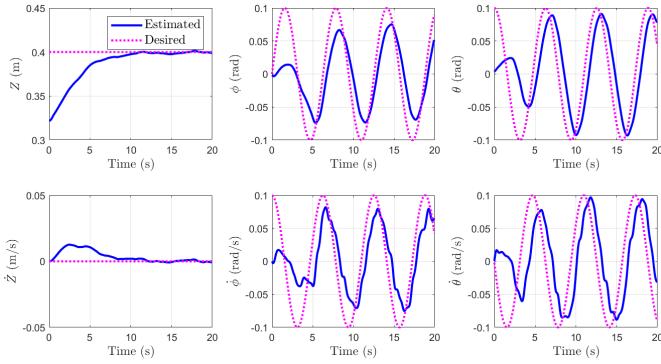


Figure 17: Estimated and desired states of the "dancing" motion scenario.

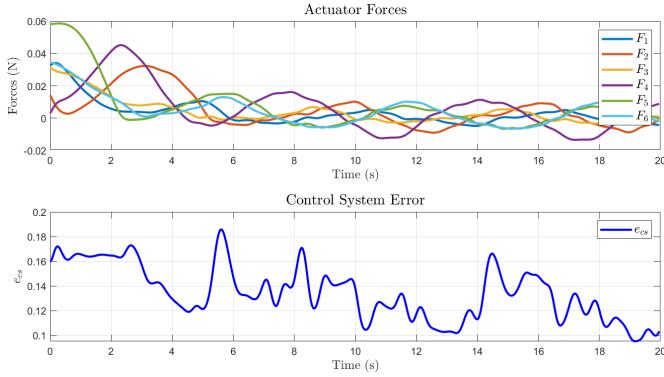


Figure 18: Experimental force control input of the "dancing motion scenario".

the desired profiles, indicating that the controller can effectively handle variations in angular motions. These deviations can be attributed to practical factors, such as actuator speed constraints, sensor noise, and unmodeled dynamics, which affect the system's ability to track rapidly changing references.

The estimated velocity results also show that the platform is generally able to follow the desired velocity profile. For the linear velocity  $\dot{Z}$ , an overshoot occurs as the platform moves along the  $Z$ -axis before settling at the target velocity. This is expected due to the momentum required to reach the desired position. For the angular velocities  $\dot{\phi}$  and  $\dot{\theta}$ , a gradual transition is observed from the initial to desired velocities, reflecting the response time required by the physical system elements. Although these fluctuations influence the generated actuator forces, the platform still demonstrates successful tracking of the desired state variables to achieve the required control objective.

## 6. Conclusion

This paper has presented the development of a prototype and the implementation of a feedback control method on a 6-DoF Stewart platform. The proposed control system combines the feedback linearization scheme with the LQR method, and is implemented using the estimated values of the system states obtained using an EKF method. Simulation and experimental evaluations of the proposed estimation and control design methods are performed w.r.t. static and dynamic reference trajectories. These results showed that the proposed EKF method

produced reliable estimates of the state variables, and that the proposed feedback control scheme generated appropriate forces that help the platform to track the desired motion. These results demonstrate the effectiveness of the proposed estimation and control framework for real-time implementation. Future work will examine more advanced methods to help improve robustness and ensure safety of the platform operation.

## References

- [1] D. Stewart, A platform with six degrees of freedom, Proceedings of the Institution of Mechanical Engineers 180 (1) (1965) 371–386.
- [2] A. Antonov, Parallel–serial robotic manipulators: A review of architectures, applications, and methods of design and analysis, Machines 12 (11) (2024). doi:10.3390/machines12110811.
- [3] H. Cheng, G. Liu, Y. Yiu, Z. Xiong, Z. Li, Advantages and dynamics of parallel manipulators with redundant actuation, in: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No. 01CH37180), Vol. 1, IEEE, 2001, pp. 171–176.
- [4] M.-Y. Wei, S.-A. Fang, J.-W. Liu, Design and implementation of a new training flight simulator system, Sensors 22 (20) (2022) 7933.
- [5] Y. Huang, D. M. Pool, O. Stroosma, Q. P. Chu, M. Mulder, A review of control schemes for hydraulic stewart platform flight simulator motion systems, in: AIAA Modeling and Simulation Technologies Conference, 2016, p. 1436.
- [6] Y. Cai, S. Zheng, W. Liu, Z. Qu, J. Han, Model analysis and modified control method of ship-mounted stewart platforms for wave compensation, IEEE Access 9 (2020) 4505–4517.
- [7] Y. Cai, S. Zheng, W. Liu, Z. Qu, J. Zhu, J. Han, Sliding-mode control of ship-mounted stewart platforms for wave compensation using velocity feedforward, Ocean Engineering 236 (2021) 109477.
- [8] W. Qiu, S. Wang, A. Niu, K. Fan, G. Han, H. Chen, Modeling and analysis of landing collision dynamics for an active helideck based on the stewart platform, Ocean Engineering 297 (2024) 117107.
- [9] S. Kizir, Z. Bingül, Design and development of a stewart platform assisted and navigated transphenoidal surgery, Turkish Journal of Electrical Engineering and Computer Sciences 27 (2) (2019) 961–972.
- [10] V. Patel, S. Krishnan, A. Goncalves, K. Goldberg, Sprk: A low-cost stewart platform for motion study in surgical robotics, in: 2018 International Symposium on Medical Robotics (ISMRR), IEEE, 2018, pp. 1–6.
- [11] X. Li, W. Zhou, D. Jia, J. Qian, J. Luo, P. Jiang, W. Ma, A decoupling synchronous control method of two motors for large optical telescope, IEEE Transactions on Industrial Electronics 69 (12) (2022) 13405–13416.
- [12] I. Jikuya, D. Uchida, M. Kino, M. Kurita, K. Yamada, Structure of distributed control system in seimei telescope, SICE Journal of Control, Measurement, and System Integration 14 (2) (2021) 111–118.
- [13] F. Liang, S. Tan, X. Zhao, J. Fan, Z. Lin, Z. Shi, X. Kang, Kinematics and dynamics simulation of a stewart platform, in: Journal of Physics: Conference Series, Vol. 2333, IOP Publishing, 2022, p. 012026.
- [14] Y. Patel, P. George, Parallel manipulators applications—a survey, Modern Mechanical Engineering 2 (3) (2012) 57–64.
- [15] N. Ghodsian, K. Benfriha, A. Olabi, V. Gopinath, A. Arnou, Mobile manipulators in industry 4.0: A review of developments for industrial applications, Sensors 23 (19) (2023) 8026.
- [16] M. Zarebidoki, J. S. Dhupia, W. Xu, A review of cable-driven parallel robots: Typical configurations, analysis techniques, and control methods, IEEE Robotics & Automation Magazine 29 (3) (2022) 89–106.
- [17] C. Yang, W. Ye, Q. Li, Review of the performance optimization of parallel manipulators, Mechanism and Machine Theory 170 (2022) 104725.
- [18] M. Russo, D. Zhang, X.-J. Liu, Z. Xie, A review of parallel kinematic machine tools: Design, modeling, and applications, International Journal of Machine Tools and Manufacture 196 (2024) 104118.
- [19] Acrome, Stewart platform, <https://acrome.net/product/stewart-platform>, accessed: 2025-08-12 (2025).
- [20] PT-Actuator, 6dof stewart motion platform kits 6d-01, <https://www.pt-actuator.com/product/stewart-motion-platform/>, accessed: 2025-08-12 (2025).

- [21] MotionSystems (Poland), Ps-6tl-800 6-dof motion platform, <https://motionsystems.eu/product/motion-platforms/ps-6tl-800/>, accessed: 2025-08-15 (2025).
- [22] M. Venkat Raaman, S. Aravind, R. Pavel, R. Kuppan Chetty, J. A. Dhana-raj, Design and development of a general-purpose low-cost stewart platform for laboratory teaching: A mechatronics approach, in: Advances in Mechanical Engineering: Select Proceedings of CAMSE 2020, Springer, 2021, pp. 469–479.
- [23] U. Kelesbekov, G. Marini, Z. Bai, W. Johal, E. Velloso, J. Knibbe, Stuet: Dual stewart platforms for pinch grasping objects in vr, in: 2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, 2024, pp. 309–318.
- [24] D. Silva, J. Garrido, E. Riveiro, Stewart platform motion control automation with industrial resources to perform cycloidal and oceanic wave trajectories, *Machines* 10 (8) (2022) 711.
- [25] Z. Bingul, O. Karahan, Dynamic modeling and simulation of Stewart platform, INTECH Open Access Publisher London, UK, 2012.
- [26] Y. Wang, A direct numerical solution to forward kinematics of general stewart–gough platforms, *Robotica* 25 (2007) 121–128. doi:10.1017/S0263574706003080.
- [27] M. Nategh, M. Agheli, A total solution to kinematic calibration of hexapod machine tools with a minimum number of measurement configurations and superior accuracies, *International Journal of Machine Tools and Manufacture* 49 (15) (2009) 1155–1164.
- [28] S. Karmakar, C. J. Turner, Forward kinematics solution for a general stewart platform through iteration based simulation, *The International Journal of Advanced Manufacturing Technology* 126 (1) (2023) 813–825.
- [29] D. K. S. Chauhan, P. R. Vundavilli, Forward kinematics of the stewart parallel manipulator using machine learning, *International Journal of Computational Methods* 19 (08) (2022) 2142009.
- [30] M. de Campos Porath, L. A. F. Bortoni, R. Simoni, J. S. Eger, Offline and online strategies to improve pose accuracy of a stewart platform using indoor-gps, *Precision Engineering* 63 (2020) 83–93.
- [31] A. Şumnu, İ. H. Güzelbey, M. V. Çakir, Simulation and pid control of a stewart platform with linear motor, *Journal of mechanical science and technology* 31 (2017) 345–356.
- [32] A. Koszewnik, K. Troc, M. Slowik, Pid controllers design applied to positioning of ball on the stewart platform, *Acta Mechanica et Automatica* 8 (12 2014). doi:10.2478/ama-2014-0039.
- [33] T. Bernal, M. Gordon, A. Pacheco, F. Vargas, Control of a stewart platform using a mimo pid controller, in: 2024 IEEE ANDESCON, IEEE, 2024, pp. 1–6.
- [34] S. S. Ahmadi, A. Rahmanii, Nonlinear model predictive control of a stewart platform based on improved dynamic model, *International Journal of Theoretical and Applied Mechanics* 5 (2020).
- [35] W. Wang, Y. Ning, Y. Zhang, P. Xu, B. Li, Linear active disturbance rejection control with linear quadratic regulator for stewart platform in active wave compensation system, *Applied Ocean Research* 156 (2025) 104469.
- [36] Y. Shi, W. Sheng, J. Wang, L. Jin, B. Li, X. Sun, Real-time tracking control and efficiency analyses for stewart platform based on discrete-time recurrent neural network, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2024).
- [37] H. Yadavari, V. T. Aghaei, S. I. GLU, Addressing challenges in dynamic modeling of stewart platform using reinforcement learning-based control approach, *Journal of Robotics and Control (JRC)* 5 (1) (2024) 117–131.
- [38] R. N. Jazar, *Theory of Applied Robotics*, Springer, 2010.
- [39] H. Guo, H. Li, Dynamic analysis and simulation of a six degree of freedom stewart platform manipulator, *Proceedings of the IMechE, Part C: J. Mech. Eng. Sci.* 220 (1) (2006) 61–72.
- [40] K. M. Lynch, F. C. Park, *Modern robotics*, Cambridge University Press, 2017.
- [41] S. B. Niku, *Introduction to robotics: analysis, control, applications*, John Wiley & Sons, 2020.
- [42] A. Isidori, *Nonlinear Control Systems*, 3rd Edition, Communications and Control Engineering, Springer, London, 1995. doi:10.1007/978-1-84628-615-5.
- [43] K. J. Åström, R. Murray, *Feedback systems: an introduction for scientists and engineers*, Princeton university press, 2021.
- [44] <https://www.youtube.com/watch?v=IcTK9U4tn8w>, YouTube video (2025).

## Appendix A. Notation

$t$	Platform's position vector w.r.t. {B}
$\dot{t}$	Platform's velocity vector w.r.t. {B}
$r$	Euler angles vector
$\dot{r}$	Euler angles vector time derivatives
$\omega_p$	Angular velocity w.r.t. {P}
$\omega$	Angular velocity w.r.t. {B}
$X', Y', Z'$	Position components
$\phi, \theta, \psi$	Euler angles (roll, pitch, yaw)
$q$	Generalized vector pose
$\dot{q}$	Generalized velocity vector
$\ddot{q}$	Generalized acceleration vector
$R$	3D Rotation matrix
$p_i$	Position vector of the $i$ -th platform joint w.r.t. {P}
$b_i$	Position vector of the $i$ -th base joint w.r.t. {B}
$\beta_i$	Angular position of the $i$ -th on the base
$\rho_i$	Angular position of the $i$ -th on the platform
$I$	Identity matrix
$m_p$	Platform mass
$m_t$	Mass of the actuator's top part
$m_b$	Mass of the actuator's bottom part
$l_t$	Distance from actuator top to platform joint
$l_b$	Distance from actuator bottom to base joint
$I_p$	Inertia matrix of the platform
$I_t$	Inertia matrix of the top actuator
$I_b$	Inertia matrix of the bottom actuator
$M$	Inertia (mass) matrix
$C$	Coriolis and centrifugal matrix
$G$	Gravity vector
$H$	Inverse transpose of the Jacobian matrix
$F$	Actuation force or torque input vector
$g$	Gravity acceleration vector
$e_l$	State estimation error
$e_t$	Tracking error
$e_{cs}$	Control system error
$\xi$	State vector
$\xi_{des}$	Desired state vector
$u$	Control input
$A$	System matrix
$B$	Input matrix
$J$	LQR performance index
$K$	State feedback gain matrix
$N$	State weighting matrix (LQR)
$O$	Control effort weighting matrix (LQR)
$\hat{\xi}$	Estimated state vector
$A_d$	Discrete system matrix
$B_d$	Discrete input matrix
$P$	State covariance matrix
$z$	Output vector
$\hat{z}$	Output model
$\tilde{z}$	Innovation vector
$V$	Process noise covariance
$W$	Measurement noise covariance
$\gamma$	Output function
$\Gamma$	Jacobian of $\gamma$ w.r.t. $\xi$

## Appendix B. Stewart Platform Model Derivation

### Appendix B.1. Additional Kinematic Model Derivations

The unit vector of each Stewart platform leg can be calculated through:

$$n_i = \frac{l_i}{s_i}. \quad (\text{B.1})$$

Additionally, the position vector of the joint platform w.r.t frame {B} is computed by the following:

$$q_p = t + Rp_i \quad (\text{B.2})$$

and the time derivative is given by the following:

$$\dot{q}_{pi} = [I \ R\tilde{p}_i^T R^T] \dot{q}. \quad (\text{B.3})$$

Based on (B.3), the velocity of the legs can be defined by:

$$\dot{s}_i = n_i^T \dot{q}_{pi}. \quad (\text{B.4})$$

On the other hand, the velocity of the upper part and lower part of the Stewart platform legs can be computed through (B.5) and (B.6), respectively

$$v_{ti} = \left( I + \frac{l_t \tilde{n}_i^2}{s_i} \right) \dot{q}_{pi}, \quad (\text{B.5})$$

$$v_{bi} = \left( \frac{l_b \tilde{n}_i^T \tilde{n}_i}{s_i} \right) \dot{q}_{pi}. \quad (\text{B.6})$$

Finally, the angular velocity of the Stewart platform leg is defined by:

$$\omega_{li} = \frac{\tilde{n}_i \dot{q}_{pi}}{s_i}. \quad (\text{B.7})$$

### Appendix B.2. Legs Dynamic Model Derivation

For this subsection, the symbol  $i$  which is used to denote the leg number will be omitted to avoid confusion. The Lagrange equation of the Stewart platform leg is defined as follows:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_p} \right) - \frac{\partial T}{\partial q_p} = Q, \quad (\text{B.8})$$

where  $T$  is the kinetic energy which is defined by:

$$\begin{aligned} T &= \frac{1}{2} v_t^T m_t v_t + \frac{1}{2} \omega_l^T (I_l + I_b) \omega_l \\ &= \frac{1}{2} \dot{q}_p^T \left( \left( I + \frac{l_t \tilde{n}^2}{s} \right)^T m_t \left( I + \frac{l_t \tilde{n}^2}{s} \right) + (I_l + I_b) \tilde{n}^T \frac{\tilde{n}}{s^2} \right) \dot{q}_p \\ &= \frac{1}{2} \dot{q}_p^T (M_1 + M_2) \dot{q}_p. \end{aligned} \quad (\text{B.9})$$

Here,  $M_1$  and  $M_2$  are inertial term defined as follows:

$$M_1 = \left( I + \frac{l_t \tilde{n}^2}{s} \right)^T m_t \left( I + \frac{l_t \tilde{n}^2}{s} \right), \quad (\text{B.10})$$

$$M_2 = (I_l + I_b) \tilde{n}^T \frac{\tilde{n}}{s^2}. \quad (\text{B.11})$$

Next, we define the forces acting upon the leg

$$Q_f = nf, \quad (\text{B.12})$$

$$Q_{m_t g} = \left( I + \frac{l_t \tilde{n}^2}{s} \right) m_t g, \quad (\text{B.13})$$

$$Q_{m_b g} = \left( \frac{l_b \tilde{n}^T n}{s} \right) m_b g. \quad (\text{B.14})$$

The generalized force is defined by:

$$Q = Q_f + Q_{m_t g} + Q_{m_b g} + f_p. \quad (\text{B.15})$$

The constraint force  $f_p$  can be rewritten into (9). with  $C_a$ :

$$\begin{aligned} C_a = & \frac{m_t l_t}{s^2} (n \dot{q}_p \tilde{n}^T n + n^T \dot{q}_p \tilde{n}^T \tilde{n} + \tilde{n}^T \tilde{n} \dot{q}_p n^T) \\ & - \frac{m_t l_t^2}{s^3} (n^T \dot{q}_p \tilde{n}^T \tilde{n} + \tilde{n}^T \tilde{n} \dot{q}_p n^T) \\ & - \frac{2(I_t + I_b)}{s^3} (\tilde{n}^T \tilde{n} \dot{q}_p n^T). \end{aligned} \quad (\text{B.16})$$

### Appendix B.3. Platform Dynamic Model Derivation

Combining equations (12) and (13) will produce the following equation:

$$M_p \ddot{q} + C_p \dot{q} + \begin{bmatrix} m_p I \\ m_p R \tilde{c}_p^T R^T \end{bmatrix} \tilde{\omega}^2 R c_p = H_p F_p + \begin{bmatrix} m_p I \\ m_p R \tilde{c}_p^T R^T g \end{bmatrix}, \quad (\text{B.17})$$

where:

$$M_p = \begin{bmatrix} m_p I & m_p R \tilde{c}_p^T R^T \\ m_p R \tilde{c}_p R^T & m_p R \tilde{c}_p \tilde{c}_p^T R^T + R I_p R^T \end{bmatrix}, \quad (\text{B.18})$$

$$C_p = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{\omega} R I_p R^T \end{bmatrix}, \quad (\text{B.19})$$

$$H_p = \begin{bmatrix} I & \dots & I \\ (R \tilde{p} R^T)_1 & \dots & (R \tilde{p} R^T)_6 \end{bmatrix}, \quad (\text{B.20})$$

$$F_p = [f_{p1} \ f_{p2} \ f_{p3} \ f_{p4} \ f_{p5} \ f_{p6}]^T. \quad (\text{B.21})$$

The complete dynamic equation that combines the models of the platform and the legs are defined in (14), in which the details of each matrices are defined as the following:

$$M(q) = M_p + \sum_{i=1}^6 \begin{bmatrix} I \\ R \tilde{p}_i R^T \end{bmatrix} (M_1 + M_2)_i \begin{bmatrix} I & R \tilde{p}_i R^T \end{bmatrix}, \quad (\text{B.22})$$

$$\begin{aligned} C(q, \dot{q}) \dot{q} = & C_p \dot{q} + \sum_{i=1}^6 \begin{bmatrix} I \\ R \tilde{p}_i R^T \end{bmatrix} (C_a)_i \begin{bmatrix} I & R \tilde{p}_i R^T \end{bmatrix} \dot{q} \\ & + \begin{bmatrix} m_p I \\ m_p R \tilde{c}_p^T R^T \end{bmatrix} \tilde{\omega}^2 R c_p \\ & + \sum_{i=1}^6 \begin{bmatrix} I \\ R \tilde{p}_i R^T \end{bmatrix} (M_1 + M_2)_i \tilde{\omega}^2 (R p_i), \end{aligned} \quad (\text{B.23})$$

$$\begin{aligned} G(q) = & - \begin{bmatrix} m_p I \\ m_p R \tilde{c}_p^T R^T g \end{bmatrix} \\ & - \sum_{i=1}^6 \begin{bmatrix} I \\ R \tilde{p}_i R^T \end{bmatrix} (Q_{m_t g} + Q_{m_b g})_i, \end{aligned} \quad (\text{B.24})$$

$$H(q) = \begin{bmatrix} n_1 & \dots & n_6 \\ (R \tilde{p}_1 R^T) n_1 & \dots & (R \tilde{p}_6 R^T) n_6 \end{bmatrix}, \quad (\text{B.25})$$

$$F = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6]^T. \quad (\text{B.26})$$

**Benedictus C. G. Cinun** is currently pursuing his Ph.D. in Electrical Engineering at University of Houston, USA. He received B.Eng. in Electrical Engineering from Parahyangan Catholic University, Indonesia, in 2024. His research interests include robotics and controls.

**Tua A. Tamba** is an associate professor in the Department of Electrical Engineering at Parahyangan Catholic University, Indonesia. He received both MSEE and Ph.D. in Electrical Engineering from University of Notre Dame (USA) in 2016, M.Sc. in Mechanical Engineering from Pusan National University (Republic of Korea) in 2009, and B.Eng. in Engineering Physics from Institut Teknologi Bandung (Indonesia) in 2006. His main research interests include dynamical systems, control theory, and optimization with applications in mechatronics, robotics, automation systems, and networked systems.

**Immanuel R. Santjoko** is currently pursuing his Ph.D. in Electrical Engineering at the University of Houston, USA. He received his B.Eng. in Electrical Engineering from Parahyangan Catholic University, Indonesia, in 2025. His main research interests include robotics and autonomous systems.

**Xiaofeng Wang** is an associate professor in the Department of Electrical Engineering at the University of South Carolina, Columbia, USA. He earned his B.S. degree in Applied Mathematics and M.S. in Operation Research and Control Theory from East China Normal University in 2000 and 2003, respectively, and obtained his Ph.D. degree in Electrical Engineering

at the University of Notre Dame in 2009. He worked as a post-doctoral research associate in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign, before joining USC in 2012. His research interests include model predictive control, robust adaptive control, networked control systems, multi-agent systems, and machine learning, with their applications in robotics, cyber-physical systems, and autonomous systems. He is currently serving as an associate editor of IEEE Transactions on Aerospace and Electronic Systems, Journal of The Franklin Institute, and IEEE Control Systems Society Conference Editorial Board. He received the best paper award at the Annual Conference of the Prognostics and Health Management Society in 2014 and was the finalist of the best paper award at the International Conference on Cyber-Physical Systems in 2013.

**Bin Hu** is an assistant professor in the Department of Engineering Technology at the University of Houston, USA. Before that, he was an assistant professor in the Department of Engineering Technology at Old Dominion University, Norfolk, Virginia, USA. He received his Ph.D. degree from the Department of Electrical Engineering at the University of Notre Dame in 2016, and the M.S. in Control Science and Engineering from Zhejiang University in 2010. Dr. Bin Hu directs the Networked Autonomous and Intelligent Learning at the University of Houston, which focuses on cutting-edge research in learning-based control, optimization, and machine learning, with applications spanning cybersecurity, autonomous robotics, human-machine automation, IoT systems, and vehicular networks. His research interests include intelligent autonomous systems that can learn, adapt, and operate safely in complex, dynamic environments.