actuators

MDPI

Article

Dynamic Parameter Estimation and Trajectory Control of Two-Wheeled Mobile Manipulator on an Inclined Surface

Sertaç Emre Kara [1,*] and Oğuz Yakut [2]

[1] Department of Mechanical Engineering, Faculty of Engineering, Firat University, 23119 Elazig, Türkiye
[2] Mechatronic Engineering Department, Faculty of Engineering, Fırat University, 23119 Elazig, Türkiye; oyakut@firat.edu.tr
* Correspondence: sekara@firat.edu.tr

Abstract

In this study, the hardware design, manufacturing, and control of a Two-Wheeled Inverted Pendulum Manipulator (TWIPM) are successfully achieved. The system comprises a two-degree-of-freedom independently driven chassis integrated with a three-degree-of-freedom robotic manipulator. To enable the robot to navigate and reach designated target coordinates on inclined terrain, an autonomous trajectory planning scheme is implemented using the A* algorithm, and the system's mathematical model is comprehensively updated. To ensure balance stability and enhance robustness against external perturbations, a PI-based supplementary control architecture is proposed to support the core PID controllers governing wheel angular position and chassis tilt. The efficacy of the proposed control strategy is initially validated via numerical simulations under various reference trajectories and disturbance inputs on flat surfaces. Subsequently, experimental validations conducted on a physical testbed featuring an inclined ramp demonstrate the robot's autonomous trajectory tracking and precise trajectory tracking positioning capabilities, confirming its real-world viability. Finally, future research directions are outlined, focusing on end-effector position optimization, the integration of a dynamic payload estimator under varying weights, and the fully onboard execution of navigation algorithms on the central microcontroller.

Keywords: two-wheeled inverted pendulum manipulator (TWIPM); autonomous trajectory planning; supplementary control architecture

1. Introduction

In today's social life and industrial sectors, robots minimize human error and eliminate manual labor by undertaking tasks that are necessary yet hazardous or monotonous for humans. Considering mobile robots operating on land, wheeled systems stand out due to their higher agility and speed compared to legged robots. Two-wheeled self-balancing robots are frequently utilized for transport tasks in high-tech factories or warehouses. These Two-Wheeled Inverted Pendulum Robot (TWIPR) systems can perceive complex environments autonomously using sensors such as obstacle detection, LiDAR, and orientation, and can reach designated points within the scope of their task definition (Ha and Yuta [1], Zhai and Zhang [2]).

Integrating manipulation capabilities into these mobile systems is one of the most crucial features, bringing the robot closer to humans, offering advantages over humans in hazardous environments or in fields such as healthcare support, domestic chores, and industrial tasks. Although Two-Wheeled Inverted Pendulum Manipulators (TWIPM) stand

Check for updates

Academic Editors: Guanghong Yang, Yuanxin Li and Chao Deng

Received: 20 June 2026
Revised: 30 July 2026
Accepted: 8 August 2026
Published: 11 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

out in this domain due to their point-turning capabilities and small footprints (Klokowski et al. [3], Lentini et al. [4], Grasser et al. [5], Tirmant et al. [6], Stilman et al. [7]), the manipulator structure added to the naturally unstable chassis renders the balance control highly complex. In two-wheeled robots, manipulator movements are considered disturbance inputs; simultaneously, the motion of the mobile chassis complicates the position control of the end-effector.

In the literature, the dynamic models of TWIPR and TWIPM systems are generally derived using Lagrange's Equations (Önkol and Kasnakoğlu [8]), Newton's Laws (Zhang et al. [9]), or the Kane Method (Muhammad et al. [10]), which offers computationally faster solutions. For the control of these systems, linear and nonlinear methods such as Linear Quadratic Regulator (LQR) (Oryschuk et al. [11], Petcu and Andreescu [12]), PID (Albani et al. [13], Velagić et al. [14]) and Sliding Mode Control (SMC) (Pham et al. [15], Hou et al. [16]) are widely preferred. However, an analysis of existing studies reveals that the dynamic models of these complex structures are generally derived by linearization around the equilibrium point on flat ground. Research regarding inclined surfaces is mostly limited to systems where only the inverted pendulum chassis is considered (Kausar et al. [17], Zhou et al. [18], Dai et al. [19], Zhang et al. [20], Sun et al. [21]). Although the ability to climb inclined surfaces has been investigated in systems with manipulation capabilities, structures where the robot's heading direction changes on inclined surfaces and the robot configuration shapes accordingly have not been encountered. Furthermore, most of these studies consist of simulation work, and the physical manipulation interaction is not sufficiently observed.

Kinematic formulations and path planning receive comparatively less attention in the TWIPM literature, despite being just as critical for real-world deployment. Unlike fixed-base manipulators, mobile manipulators require forward and inverse kinematics that jointly account for chassis pose and arm configuration, which substantially complicates the solution space (Feng et al. [22]). Workspace characterization for wheeled manipulators has been studied (Zahid et al. [23]), though the effect of chassis inclination on reachable volume is rarely treated. For navigation, graph-search methods such as A* and sampling-based planners like RRT are well established for flat-surface mobile robots (Candra et al. [24], Lin et al. [25]), and adaptations to uneven terrain have been proposed (Chatzisavvas et al. [26], Jiang et al. [27], Hong et al. [28], Pu et al. [29]). Most of these frameworks, however, model the robot as a point mass and do not couple the arm kinematics into the planning problem. As a result, generating joint trajectories for 3D end-effector targets on inclined surfaces has not been systematically handled.

Motivated by these gaps in the literature, this study aims to design and control a two-wheeled self-balancing robot capable of autonomously orienting itself on an inclined surface and performing manipulation tasks at given coordinates. Within the scope of this work, a low-cost, modular TWIPM platform independent of proprietary software was developed, and its physical design was achieved. To ensure the autonomous capability of the system, the forward/inverse kinematics of the robot and the kinematics of the mobile chassis were derived, and the A* algorithm was utilized for path planning. Furthermore, an original, supplementary-controlled total control law was designed to stably maintain body balance against disturbance effects arising from both the inclined surface and the manipulator. Finally, the success of this developed design and control architecture was validated through experimental studies where physical parameters were estimated, thereby introducing a comprehensive and hardware-based solution to the literature.

2. Material and Method

2.1. Robot Design

The design, manufacturing, and control processes of the two-wheeled self-balancing mobile manipulator developed within the scope of this study were carried out at the Machine Theory and Dynamics Laboratory of the Department of Mechanical Engineering at Fırat University. All experimental applications of the study were conducted on this platform. The distinguishing aspect of the robot is its specific manipulator configuration, which enables the manipulator's end-effector to remain stationary at a designated position on inclined surfaces. Throughout the development process, the dynamic model of the robot was formulated, unknown system parameters were estimated using experimental data, and a supplementary-controlled control law was proposed and tested to ensure the balance of the system (Kara and Yakut [30], Kara [31]).

2.1.1. Mechanical Design and Manufacturing

The developed robot fundamentally consists of two drive wheels, a rigid main chassis, and a 3-degree-of-freedom (3-DoF) manipulator positioned on the chassis. The structural components of the robot (the chassis and manipulator links) were fabricated from PLA (Biopolymer Polylactic Acid) material using a 3D printer.

The wheels of the system are driven by two DC motors (Maxon A-Max, 6 V, 1216 CPR) operating at a voltage of 9 V. The system is powered by a 12 V supply. The total mass of the robot is 2.8 kg.

The workspace reachability of the manipulator arm is a critical factor that determines the robot's interaction with its environment. The workspace (reach limits) that the end-effector can attain is illustrated in Figure 1. The envelope shown in Figure 1, approximately 480 mm of vertical and 270 mm of horizontal reach, is the mechanically reachable workspace of the end-effector, determined by the link lengths and the joint angle limits of the servo actuators;. In the experimental validation, the manipulation sequence exercised a vertical stroke of 0.158 m along the tool-frame z-axis within this envelope (Section 4), the full envelope not being swept because the task at the ramp apex required only vertical extension. The static reachable envelope is thus bounded by mechanical constraints alone; stability considerations do not further restrict the reachable set at the inclinations tested, but they dictate the quasi-static deployment speed of the arm, as discussed in Section 2.3.

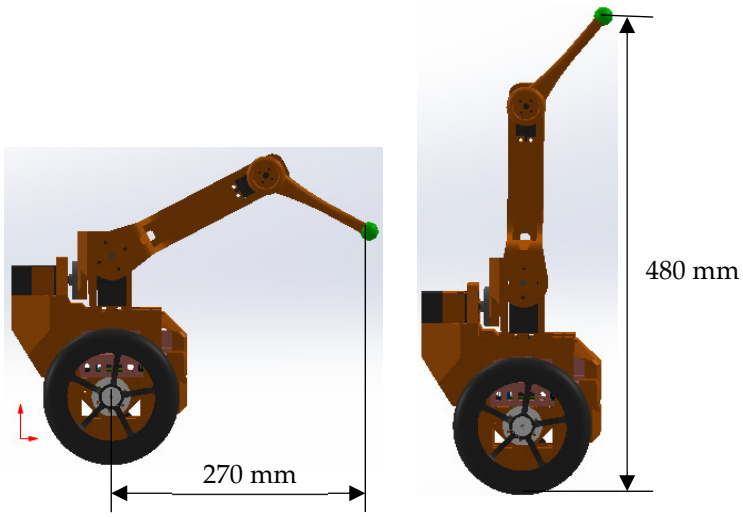


Figure 1. End-effector reachable workspace dimensions.

2.1.2. Electronics and Control Hardware

The electronic control architecture, which constitutes the brain of the system, is structured around a 600 MHz ARM Cortex-M7-based microcontroller (Teensy 4.0) featuring high-speed processing capabilities. The schematic diagram of the electronic hardware is provided in Figure 2.

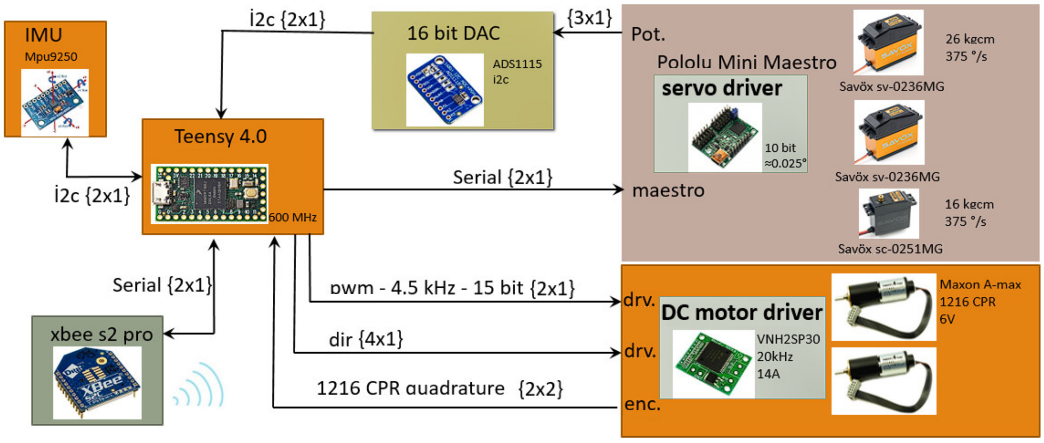
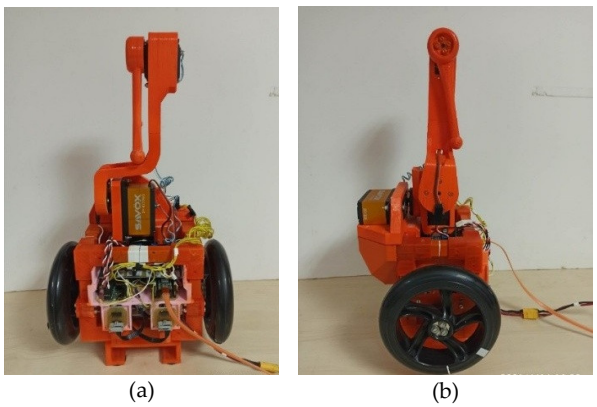


Figure 2. Electronic system architecture.

The chassis tilt angles required for balance control are obtained using an MPU9250 inertial measurement unit (IMU). Data acquired from this sensor are transferred to the main controller via the I2C communication protocol. The instantaneous angular positions of the servo motors constituting the manipulator are determined through analog signals received from the motors' built-in potentiometers. To ensure high precision, these analog signals are transmitted to the microcontroller via a 16-bit resolution ADS1115 Analog-to-Digital Converter (ADC), again using the I2C protocol. The microcontroller converts these signals into angular data and performs the inverse kinematics calculations necessary for the manipulator to reach the desired target.

The Maxon drive motors in the propulsion system feature encoders with a resolution of 1216 CPR (Counts Per Revolution) and are controlled via a VHN2SP30 motor driver. The motor driver can generate PWM (Pulse Width Modulation) signals at a frequency of 20 kHz to ensure precise speed and torque control. The electronic components and power circuitry of the robot are integrated into a custom 3D-printed enclosure to form a compact structure. This design consists of two primary layers: the lower layer houses the electronic control circuits, such as the microcontroller and sensors, while the upper layer accommodates the power components, including the voltage regulators. The final front and side views of the fully manufactured robot are shown in Figure 3.



(a) (b)

Figure 3. Fabricated TWIPM prototype: (**a**) front view, (**b**) side view.

2.2. Forward and Inverse Kinematic Equations

Robot kinematics is a discipline that investigates the motions and positions of robots, encompassing the mathematical modeling and analysis of their links, joints, and ranges of motion. This discipline is generally examined under two fundamental headings: inverse kinematics and forward kinematics. Inverse kinematics aims to calculate the required joint angles for the robot's end-effector to reach a target coordinate. In forward kinematics, the position and motion of the end-effector are analyzed based on the current joint angles. These mathematical models form the foundation for programming control systems and optimizing the robot's position, velocity, acceleration, and workspace constraints.

In this study, robot kinematics is addressed under two independent headings: the differential-drive kinematics of the mobile chassis and the kinematics of the robot arm. While planning the path for the mobile chassis to reach a target location from a specific starting point, the position analysis of the robot arm's end-effector was carried out relative to the center of the robot chassis.

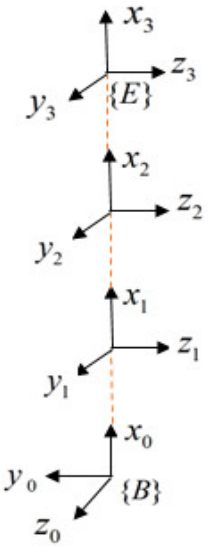
2.2.1. Forward Kinematic Modeling of the Robot Arm

In this study, the Denavit–Hartenberg (D-H) method was employed to derive the forward kinematic equations of the robot manipulator. This systematic approach, which forms the foundation for the position, velocity, and acceleration analyses as well as the inverse kinematic solutions of the robot, consists of four consecutive steps. In the first step, local coordinate axes representing the position and orientation of each joint are defined. In the second step, the axes of rotation and common normals between consecutive joints are determined. In the third step, the fundamental D-H parameters (d, θ, a, α) that establish the relationship between joints are assigned. In the final step, using these parameters, homogeneous transformation matrices are constructed to transform the local coordinate system of one joint into the coordinate system of the subsequent joint.

Each coordinate frame enabling the displacement from point {B}, accepted as the chassis center of the robot used in the study, to point {E}, the end-effector of the robot, is shown from the same perspective in Figure 4. The axes z_0, z_1, and z_2 represent the rotation axes of the robot arm (Kara [31]).



(a)



(b)

Link	θ_i	α_i	a_i	d_i
Link1	θ_1	$\pi/2$	a_1	0
Link 2	θ_2	0	a_2	0
Link 3	θ_3	0	a_3	0

(c)

Figure 4. (**a**) 3D view of the robot arm coordinate frame alignment, (**b**) Coordinate frames belonging to the robot configuration, (**c**) Link Table of the Robot Arm.

The forward kinematic transformation of the manipulator is expressed as in Equation (1).

$$P_{3x1} = \begin{bmatrix} a_1 \cos\theta_1 + a_2 \cos\theta_1 \cos\theta_2 + a_3\theta_1 \cos\theta_2 \cos\theta_3 - a_3 \cos\theta_1 \sin\theta_2 \sin\theta_3 \\ a_1 \sin\theta_1 + a_2 \cos\theta_2 \sin\theta_1 + a_3 \cos\theta_2 \cos\theta_3 \sin\theta_1 - a_3 \sin\theta_1 \sin\theta_2 \sin\theta_3 \\ a_2 \sin\theta_2 + a_3 \cos\theta_2 \sin\theta_3 + a_3 \cos\theta_3 \sin\theta_2 \end{bmatrix} \quad (1)$$

2.2.2. Numerical Inverse Kinematics for Robot Manipulators

The numerical inverse kinematics of the robot arm is a method that enables the calculation of the required joint angles through iterative algorithms so that the end-effector can reach the desired target position. Due to the increasing complexity of analytical inverse kinematic solutions as degrees of freedom expand, which imposes a high computational load on microcontroller-based hardware, a numerical analysis approach was adopted in this study. The process fundamentally consists of defining the target spatial position, assigning initial joint values, and executing iterative optimization steps. During the iteration phase, the amount of error between the target position and the estimated end-effector position, calculated based on current joint angles, is determined. If the resulting error exceeds the specified tolerance limits, the joint angles are updated using optimization algorithms such as Newton-Raphson or gradient descent. As a result of this loop, which is repeated until the system error falls below the targeted tolerance value, the final joint angles that will drive the robot to the respective target are obtained.

Equations (2) and (3) represent the workspace. Thus, the infinitesimal displacement dy depends on the joint space (dq), where J denotes the Jacobian matrix (Kara [31]).

$$y_{nx1} = f(q_{nx1}) \Rightarrow dy = Jdq \quad (2)$$

$$\dot{y}_{nx1} = \frac{d}{dt} f(q_{nx1}) = \frac{df(q)}{dq}\frac{dq}{dt} = \frac{df(q)}{dq}\dot{q} \quad (3)$$

Following some algebraic operations, Equation (4) is derived.

$$\dot{y}_{nx1} = J_{nxn}\dot{q}_{nx1} \quad (4)$$

The Jacobian matrix is calculated in Equation (5):

$$J = \frac{df(q)}{dq} \quad (5)$$

By linearizing the function around small angles, Equation (6) is obtained.

$$\vec{f}(\vec{q}) \approx \vec{f}(\vec{q}_0) + J(\vec{q}_0)(\vec{q} - \vec{q}_0) \quad (6)$$

Here, representing the initial angle and the desired angle, rearranging yields Equation (7).

$$\vec{f}(\vec{q}_d) \approx \vec{f}(\vec{q}) + J(\vec{q})(\vec{q}_d - \vec{q}) \quad (7)$$

Here, $\Delta\vec{q} = \vec{q}_d - \vec{q}$ is expressed as the angular error, and $\vec{t} = f(\vec{q}_d)$ denotes the desired workspace coordinate. In general, the workspace error presented in Figure 5 is approximately given by Equation (8) as $\vec{t} - \vec{r} \approx J\Delta\vec{q}$

$$\vec{e} \approx J\Delta\vec{q} \quad (8)$$

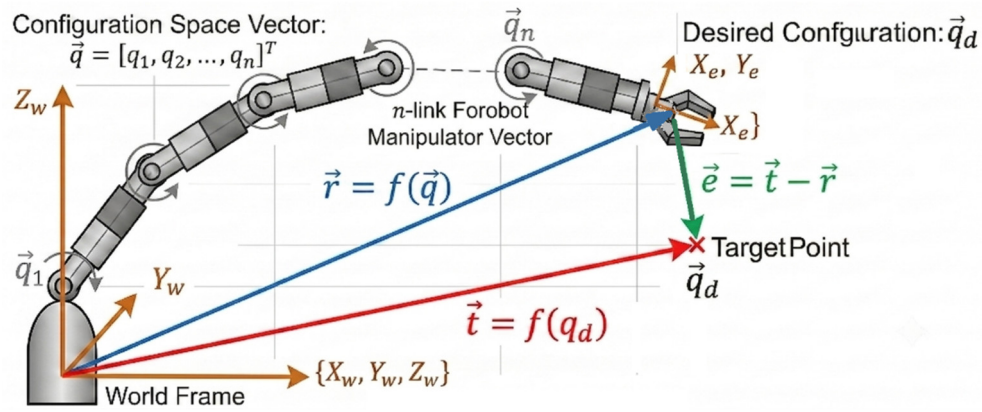


Figure 5. Geometric representation of inverse kinematics: configuration space vector, end-effector position, and error vector in world frame.

2.3. Robot Dynamic Equations

The dynamics of a mobile manipulator comprise the combined dynamics of the self-balancing mobile chassis and the manipulator arms. Addressing the system through a total dynamics approach is often avoided due to mathematical complexity. The dynamic model of the mobile inverted pendulum chassis is derived independently, with the manipulator treated as a lumped mass acting on the chassis (Kara [31]).

Representing the manipulator as a lumped mass rigidly coupled to the chassis is a deliberate modelling decision rather than a simplification of convenience, and its scope of validity is delimited as follows. First, the balance loop of an inverted-pendulum platform must be executed at a sampling rate sufficient to stabilise an inherently unstable plant; deriving and evaluating the fully coupled chassis–manipulator dynamics at every control interval would impose a computational burden incompatible with the onboard microcontroller-based implementation targeted in this work, and would conflict with the objective of a low-cost platform independent of proprietary software. Second, the assumption is admissible when the manipulator is actuated slowly relative to the bandwidth of the balance loop, and when the manipulator mass constitutes a limited fraction of the total system mass. Both conditions are satisfied in the present configuration: the three-degree-of-freedom arm is lightweight relative to the 2.8 kg platform, and in the experimental scenario it is deployed and retracted quasi-statically at the ramp apex rather than during transit. Under these conditions the manipulator acts predominantly as a slowly varying inertial load and a shift in the centre of mass, both of which are captured by the lumped-mass representation. This treatment is furthermore consistent with the prevailing approach in the two-wheeled mobile manipulator literature, in which manipulator motion is regarded as a disturbance input acting on the balancing chassis rather than as a fully coupled dynamic subsystem.

The limits of this assumption were nevertheless not left unexamined. The residual coupling that the lumped-mass representation does not reproduce is precisely what becomes observable in the experimental results, where the extension and retraction of the manipulator at the ramp apex induce a transient oscillation of the chassis. The proposed control architecture attenuates this transient, and the platform reaches its target configuration without loss of balance, which indicates that the unmodelled coupling remains within the disturbance-rejection envelope of the controller for the operating regime considered here. It is expected, however, that this margin would be eroded under rapid manipulator motion or substantially increased payload, in which case the inertial and Coriolis coupling terms would no longer be negligible, and a fully coupled dynamic formulation would be required. The derivation of such a coupled model, together with a payload-dependent

dynamic estimator, is identified as a direction for subsequent work. The kinematic diagram of the mobile manipulator chassis is shown in Figure 6.

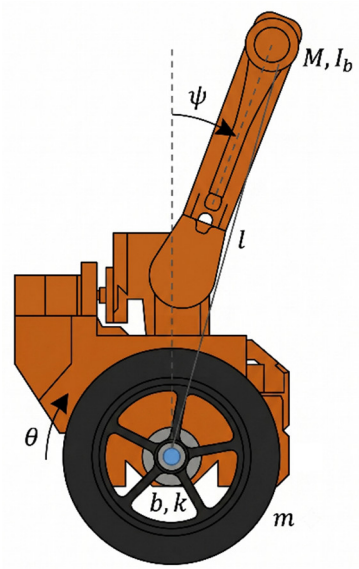


Figure 6. Kinematic diagram of the mobile inverted pendulum chassis.

The equations of motion of the inverted pendulum chassis were derived using Lagrange's Equations method. Denoting the total kinetic energy of the system as T, it can be expressed as in Equation (9).

$$T = \frac{1}{2}M(\dot{x_b}^2 + \dot{y_b}^2) + \frac{1}{2}I_b\dot{\psi}^2 + \frac{1}{2}m\dot{x}^2 + \frac{1}{2}I_w\dot{\theta}^2 \tag{9}$$

Here, M, m, I_b, and I_w denote the mass of the robot chassis, the mass of the robot wheel, the mass moment of inertia of the robot chassis, and the mass moment of inertia of the robot wheel, respectively. In addition, θ and ψ represent the generalized coordinates of the system. Furthermore, x_b and y_b express the position components of the robot chassis's center of mass in the Cartesian coordinate system. These position components and their corresponding velocity components are given by Equations (10) and (11).

$$x_b = x + l\sin\psi;\ \dot{x}_b = \dot{x} + \dot{\psi}l\cos\psi \tag{10}$$

$$y_b = l\cos\psi;\ \dot{y}_b = -\dot{\psi}l\sin\psi \tag{11}$$

Here, l is the distance between the wheel shaft center and the center of mass of the robot chassis. Given that θ represents the rotation of the chassis and ψ represents the rotation of the wheel, the relative rotation and relative rotational velocity of the motor shaft can be calculated as in Equation (12).

$$\theta_m = \theta - \psi;\ \dot{\theta}_m = \dot{\theta} - \dot{\psi} \tag{12}$$

By rearranging equations and applying the non-slip rolling condition, where r denotes the wheel radius, the following expression can be written in Equation (13) by performing the transformation $x = r$:

$$T = \frac{1}{2}M\dot{\theta}^2r^2 + Mlr\dot{\theta}\dot{\psi}\cos\psi + \frac{1}{2}M\dot{\psi}^2l^2 + \frac{1}{2}I_b\dot{\psi}^2 + \frac{1}{2}m\dot{\theta}^2r^2 + \frac{1}{2}I_w\dot{\theta}^2 \tag{13}$$

Denoting the total potential energy of the robot system as V, it can be expressed by Equation (14).

$$V = Mgl\cos\psi \tag{14}$$

The Lagrange function, denoted as L, is defined as $L = T\text{-}V$ in Equation (15).

$$L = \frac{1}{2}M\dot{\theta}^2r^2 + Mlr\dot{\theta}\dot{\psi}\cos\psi + \frac{1}{2}M\dot{\psi}^2l^2 + \frac{1}{2}I_b\dot{\psi}^2 + \frac{1}{2}m\dot{\theta}^2r^2 + \frac{1}{2}I_w\dot{\theta}^2 - Mgl\cos\psi \tag{15}$$

In its most general form, Lagrange's equations can be written as in Equation (16).

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i - F_{di} - F_{ci} \quad (i = 1, 2) \tag{16}$$

Here, q represents the generalized coordinates, where $q_1 = \theta$ and $q_2 = \psi$. Since there are two generalized coordinates, two dynamic equations belonging to the system will be obtained. F, F_c, and F_d denote the DC motor torque, Coulomb friction, and viscous friction forces, respectively, and can be calculated in Equation (17).

$$F = \frac{K_t}{R}V_a - \frac{K_tK_e}{R}(\dot{\theta} - \dot{\psi});\ F_c = k\frac{\dot{\theta}}{\left|\dot{\theta}\right|};\ F_d = b(\dot{\theta} - \dot{\psi}) \tag{17}$$

Here, k and b represent the Coulomb and viscous friction coefficients belonging to the wheel shaft center. K_t, K_e, and R denote the torque constant, electromotive force (EMF) constant, and motor resistance, respectively. Following some arrangements, Equations (18) and (19) are obtained.

$$\left\{(M+m)r^2 + I_w\right\}\ddot{\theta} + Mlr\ddot{\psi}\cos\psi - Mlr\dot{\psi}^2\sin\psi + \frac{K_tK_e}{R}(\dot{\theta} - \dot{\psi}) + b(\dot{\theta} - \dot{\psi}) + k\frac{\dot{\theta}}{\left|\dot{\theta}\right|} = \frac{K_t}{R}V_a \tag{18}$$

$$Mlr\ddot{\theta}\cos\psi + Ml^2\ddot{\psi} + I_b\ddot{\psi} - Mgl\sin\psi + \frac{2K_tK_e}{R}(\dot{\theta} - \dot{\psi}) + 2b(\dot{\theta} - \dot{\psi}) + 2k\frac{\dot{\theta}}{\left|\dot{\theta}\right|} = \frac{2K_t}{R}V_a \tag{19}$$

In its most general form, the nonlinear state-space model can be written as in Equation (20). Here, $x(t)$ represents the state variables, $u(t)$ denotes the system inputs, and $y(t)$ expresses the output matrix.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(t, x(t), u(t))\\ \mathbf{y}(t) &= \mathbf{h}(t, x(t), u(t))\end{aligned} \tag{20}$$

Given that the wheel rotation angle is $\theta = x_1$, the chassis angle is $\psi = x_2$, and x_3 and x_4 represent their respective derivatives, it can be explicitly written in the nonlinear state-space form in Equations (21)–(23):

$$\dot{x}_1 = x_3;\ \dot{x}_2 = x_4 \tag{21}$$

$$\begin{aligned}\dot{x}_3 = {} & (I_bK_tu + K_tl^2Mu - I_bRksign(x_3) - I_bK_eK_tx_3 + I_bK_eK_tx_4 - I_bRbx_3 + I_bRbx_4\\ & -l^2MRksign(x_3) - K_eK_tl^2Mx_3 + K_eK_tl^2Mx_4 - l^2MRbx_3 + l^2MRbx_4 - 2K_tlMrucos(x_2)\\ & +l^3M^2Rrx_4{}^2sin(x_2) + 2K_eK_tlMrx_3cos(x_2) - 2K_eK_tlMrx_4cos(x_2) + 2lMRbrx_3cos(x_2)\\ & -2lMRbrx_4cos(x_2) - l^2M^2Rgrcos(x_2)sin(x_2) + I_blMRrx_4{}^2sin(x_2)\\ & +2lMRkrcos(x_2)sign(x_3))/(R(I_b * I_w + l^2M^2r^2 + I_wl^2M + I_bMr^2\\ & +I_bmr^2 + l^2Mmr^2 - l^2M^2r^2cos(x_2)^2))\end{aligned} \tag{22}$$

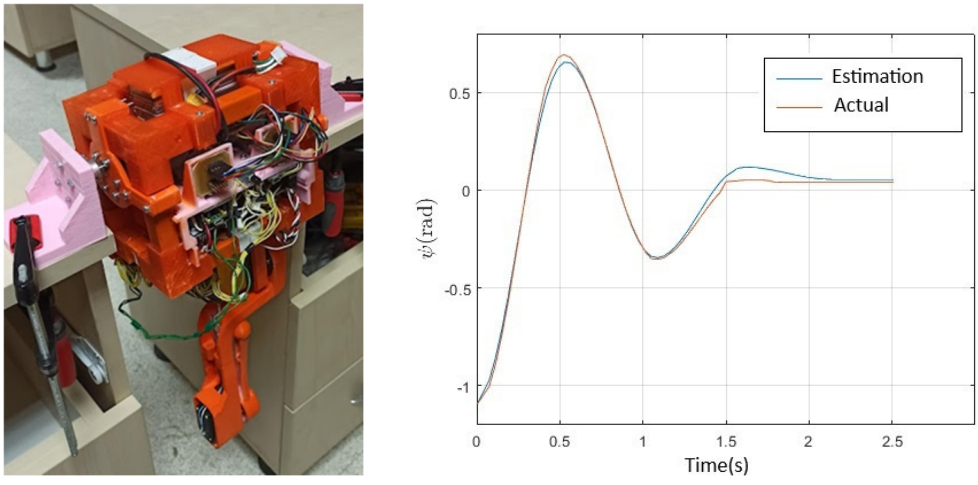
$$\begin{aligned}\dot{x}_4 = {} & (2I_wK_tu + 2K_tMr^2u - 2I_wRksign(x_3) + 2K_tmr^2u - 2I_wK_eK_tx_3 + 2I_wK_eK_tx_4 - 2I_wRbx_3 + 2I_wRbx_4 \\ & -2MRkr^2sign(x_3) - 2Rkmr^2sign(x_3) - 2K_eK_tMr^2x_3 + 2K_eK_tMr^2x_4 - 2MRbr^2x_3 + 2MRbr^2x_4 \\ & -2K_eK_tmr^2x_3 + 2K_eK_tmr^2x_4 - 2Rbmr^2x_3 + 2Rbmr^2x_4 - K_tlMrucos(x_2) + lM^2Rgr^2sin(x_2) \\ & +I_wlMRgsin(x_2) + K_eK_tlMrx_3cos(x_2) - K_eK_tlMrx_4cos(x_2) + lMRbrx_3cos(x_2) \\ & -lMRbrx_4cos(x_2) - l^2M^2Rr^2{x_4}^2cos(x_2)sin(x_2) + lMRgmr^2sin(x_2) + lMRkrcos(x_2)sign(x_3)) \\ & /(R(I_bI_w + l^2M^2r^2 + I_wl^2M + I_bMr^2 + I_bmr^2 + l^2Mmr^2 - l^2M^2r^2cos(x_2)^2))\end{aligned} \tag{23}$$

Parameter Estimation of the Mobile Chassis Dynamics

Due to the complex structure of the mobile manipulator dynamics, the mobile chassis dynamics were considered separately; the physical parameters in the proposed mathematical model were compared with the physical system operated experimentally, and the mathematical model parameters were updated accordingly (Kara [31]).

For this purpose, the mobile manipulator chassis was first modeled as a simple pendulum. From this simple pendulum model, the mass moment of inertia of the chassis I_b, the pendulum length l, and the viscous damping coefficient at the DC motor shaft b were obtained. The chassis mass M is known. The experimental oscillatory motion of the robot was compared with the pendulum model given by Equation (24). To achieve this, the wheels on the robot were first removed, and both motor shafts were connected to a fixed bearing from their ends. In this manner, the robot chassis was reduced to a simple pendulum model. Subsequently, the robot chassis was released from the vertical axis with an initial condition of 60° to allow it to oscillate. The parameter estimation was obtained using the least-squares method, and the graphical result of the simulation is given in Figure 7.

$$\ddot{\theta} = -\frac{2b\dot{\theta} + Mgl\sin\theta + sign(\dot{\theta})k}{Ml^2} \tag{24}$$



(a) (b)

Figure 7. (**a**) Derivation of Simple Pendulum Parameters, (**b**) Model update graph for obtaining simple pendulum parameters.

To find the drive motor parameters, the wheels were reattached to the robot chassis, and the robot chassis was fixed. In this case, an increasing input signal (chirp) between 0 and 1 Hz was applied to the DC motor. This ensures excitation within the bandwidth range where the motor can operate and enables the determination of motor parameters. The unknown parameters of the drive motor were obtained on the condition of minimizing the error value ($e_{\ddot{\theta}}$) resulting from the difference between the actual angular acceleration

values obtained from experimental data ($\ddot{\theta}_{real}$) and the estimated acceleration values ($\hat{\ddot{\theta}}$) calculated using the equation given in Equation (25). The stated error value was calculated as in Equation (26).

$$\ddot{\theta} = \frac{1}{I_w}\left(K_t i - b\dot{\theta} - ksign(\dot{\theta})\right) \tag{25}$$

$$e_{\ddot{\theta}} = \hat{\ddot{\theta}} - \ddot{\theta}_{real} \tag{26}$$

The block diagram of the proposed method is presented in Figure 8.

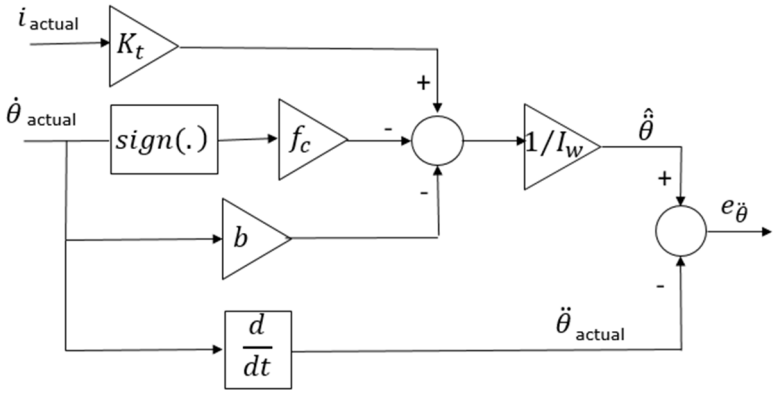


Figure 8. Block diagram of the acceleration error based on the electro-mechanical model.

A comparison between the actual and estimated acceleration values of the system within the 0–100 s time interval is provided in Figure 9. From this plot, it can be clearly seen that the updated (estimated) model closely converges to the actual system.

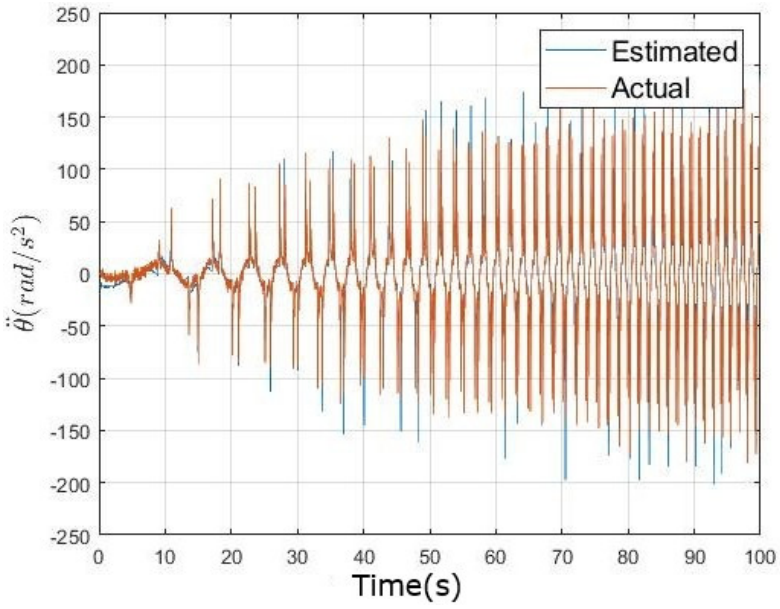


Figure 9. Actual and estimated acceleration values.

Since the updated K_t value belonging to the electric motor is equal to the K_e value, the resistance value R of the electric motor can be calculated using Equations (27) and (28) in the next stage.

$$T_m = K_t i \tag{27}$$

$$i = \frac{(V_a - K_e\dot{\theta}_m)}{R} \tag{28}$$

Here, denoting i as the armature current and e_i as the error of the current value, the update was performed by taking the difference between the experimentally obtained current value and the current value obtained from the formula, and the unknown R value was calculated. The flowchart of the utilized algorithm is given in Figure 10.

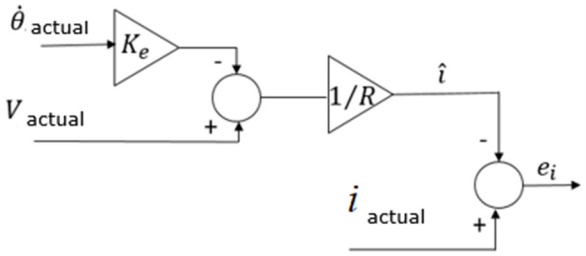


Figure 10. Current error block diagram from the electrical model.

A comparison between the actual and estimated current values of the system within the 0–100 s time interval is provided in Figure 11. From this plot as well, it can be clearly understood that the updated (estimated) model closely converges to the actual system.

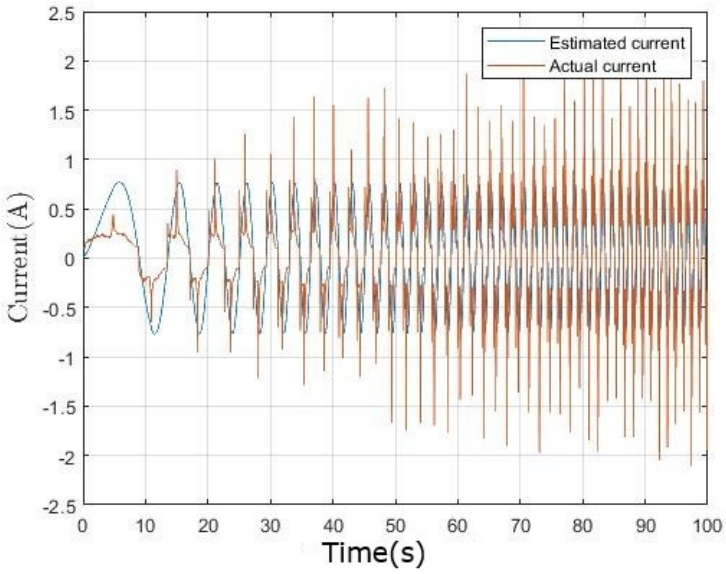


Figure 11. Actual and estimated current values.

2.4. Control System Design

In this study, a PI-based supplementary control architecture and effective control method with auxiliary control that can be utilized for two-wheeled inverted pendulum-type robots has been developed (Kara [31]). This developed method consists of chassis control, wheel control, and chassis tilt control structures.

The controller provided in Figure 12 consists of two separate PID controllers that perform the angular control of the wheels and the control of the chassis tilt angle, in addition to an auxiliary control unit (chassis tilt control) that enhances the balancing capability of the chassis and provides robustness against disturbing inputs. This auxiliary control unit operates according to the PI control law. According to the control law, γ and α are the weighting coefficients of the chassis and wheel controllers, respectively, and were determined via the trial-and-error method. These coefficients provide the user with the flexibility to select which controller will be utilized more effectively. Introduced as a ramp function, ρ ensures a smooth startup for the inverted pendulum chassis, allowing the initial overshoot caused by the integral action of the controller to be gradually softened.

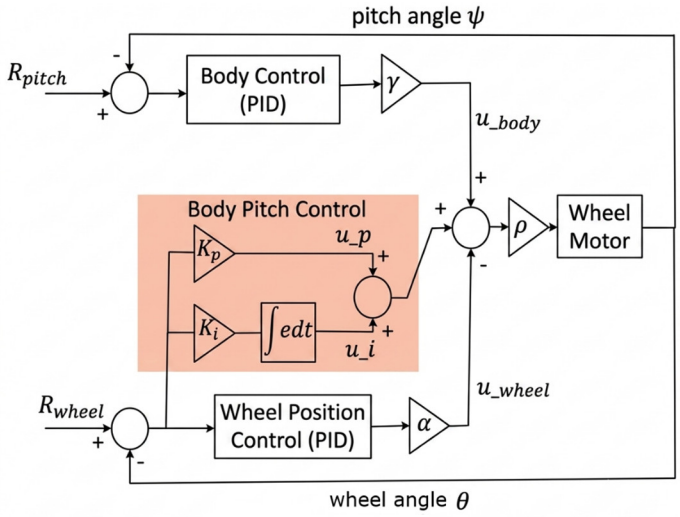


Figure 12. Block diagram of the control system.

The control system was implemented separately for both wheels. The control laws to be applied to the drive motors of the right and left wheels are given in Equations (29) and (30), respectively. Here, u_i and u_p denote the integral and proportional control inputs of the auxiliary control unit.

$$u_{right} = (\gamma u_{body} - \alpha u_{rightwheel} + u_i + u_p)\rho \tag{29}$$

$$u_{left} = (\gamma u_{body} - \alpha u_{leftwheel} + u_i + u_p)\rho \tag{30}$$

Equations (18) and (19) were linearised around the upright equilibrium point, yielding the transfer functions of Equations (31) and (32). Small-angle approximations ($\cos\psi \approx 1$, $\sin\psi \approx \psi$, $\psi^2 \approx 0$) were applied to the body tilt angle, and incremental changes in wheel angle θ were treated as small over each discrete time step ($\cos\theta \approx 1$, $\sin\theta \approx \theta$), yielding the following transfer functions.

$$\frac{\theta}{V_a} = \frac{A_6(B_1 s + 2A_4 s + B_2) - 2A_6 s(A_2 s - A_4)}{(A_1 s^2 + (A_4 + A_5)s)(B_1 s^2 - 2A_4 s + B_2) - (A_2 s^2 - A_4 s)(A_2 s^2 + 2s(A_4 + A_5))} \tag{31}$$

$$\frac{\psi}{V_a} = \frac{s(A_1 s + A_4 + A_5)2A_6 - A_6(s(A_2 s + 2(A_4 + A_5)))}{s(A_1 s + A_4 + A_5)(B_1 s^2 - 2A_4 s + B_2) - s^2(A_2 s - A_4)(A_2 s + 2(A_4 + A_5))} \tag{32}$$

Here A_1, A_2, A_3, A_4, A_5, A_6, B_1 and B_2 are given in Table 1.

Table 1. Lumped coefficients of the linearized transfer functions are presented at the upright equilibrium (ψ = 0).

$A_1 = (M+m)r^2 + I_w$	$A_2 = Ml\cos\psi$	$A_3 = -Mlr\sin\psi$	$A_4 = \frac{K_t K_e}{R} + b$
$A_5 = ksign(\dot{\theta})$	$A_6 = \frac{K_t}{R}$	$B_1 = Ml^2 + I_b$	$B_2 = -Mgl$

3. Numerical Simulations

In this section, the results of the numerical simulation studies are presented to validate the performance of the designed controller. Within the scope of the analyses, the dynamic responses of the robot chassis against different reference inputs (sine and step) and disturbing effects (unit step and half-sine) were investigated.

3.1. Sinusoidal Reference Input—Disturbance-Free Case

In this scenario (Figure 13), a sinusoidal reference input with an amplitude of 0.1 m and a frequency of 0.25 Hz was applied to the robot chassis for 20 s without any disturbing input. When the obtained linear displacement and velocity plots are examined, an inverse response (phase difference) caused by the chassis inertia that must be overcome for the robot to initiate its motion is observed. In the following seconds, the robot chassis advances by oscillating approximately 2.5° around its equilibrium position, and this tilt angle exhibits perfect agreement with the linear displacement of the chassis. Furthermore, the wheel drive motors successfully completed the reference tracking without exceeding the 9 V operating limit (without undergoing saturation).

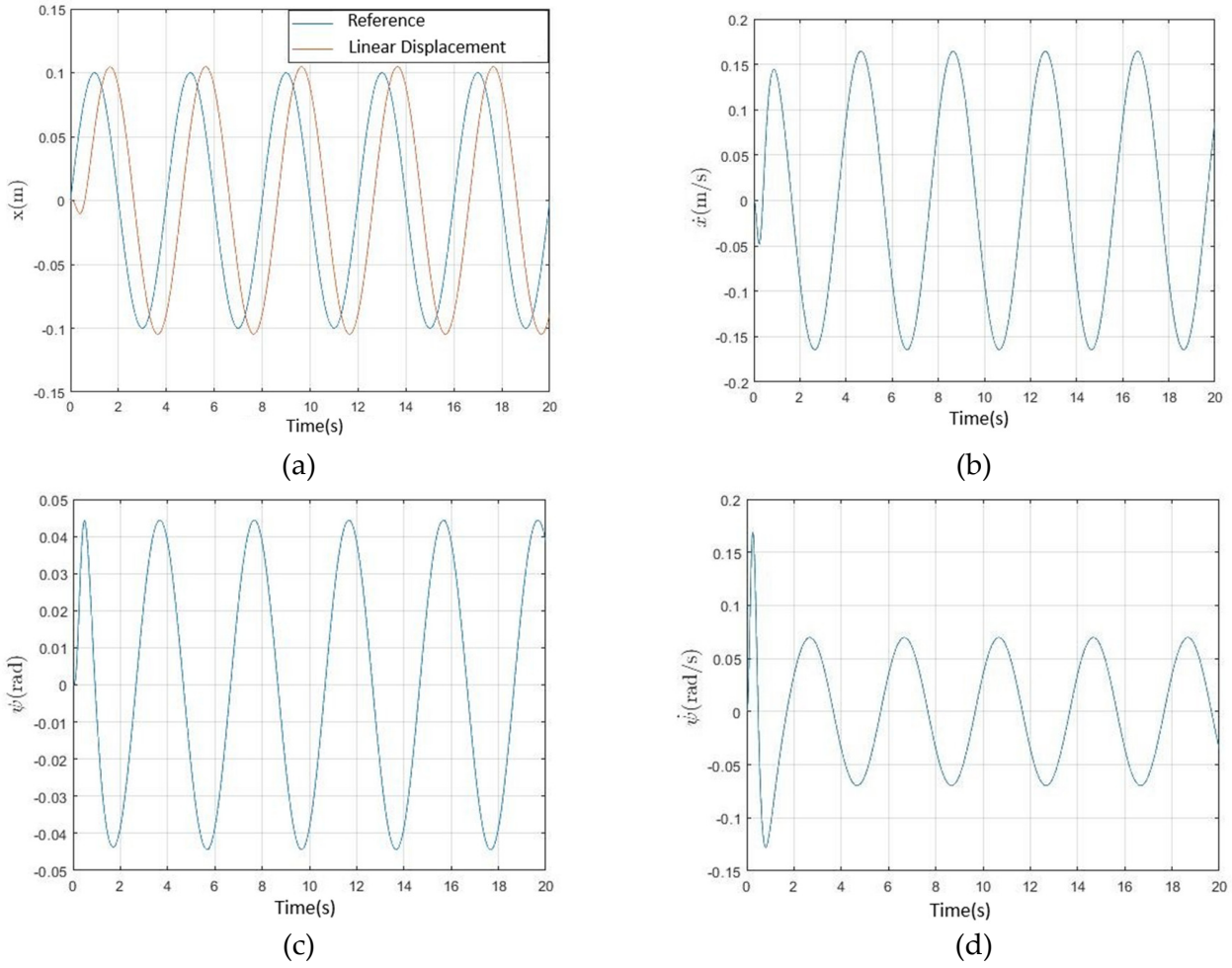


Figure 13. (**a**) Linear Displacement (**b**) Linear Velocity (**c**) Pitch Angle (**d**) Pitch Velocity.

3.2. Sinusoidal Reference Input—Unit Step Disturbance Input Effect

Under the same reference input values (0.1 m amplitude, 0.25 Hz frequency), a unit step disturbance input was applied to the system at t = 2 s. In the t = 2–3 s interval where the disturbance effect was introduced, the displacement phase difference increased; however, the system damped this effect in a short time and returned to a steady state with a constant phase difference. When the tilt angle and the chassis angular velocity are examined, it is determined that the system returned to its stable oscillation (approximately 2.5°) as of approximately t = 7 s. Despite the externally applied disturbance, the motor voltages achieved successful control without undergoing saturation (Figure 14).

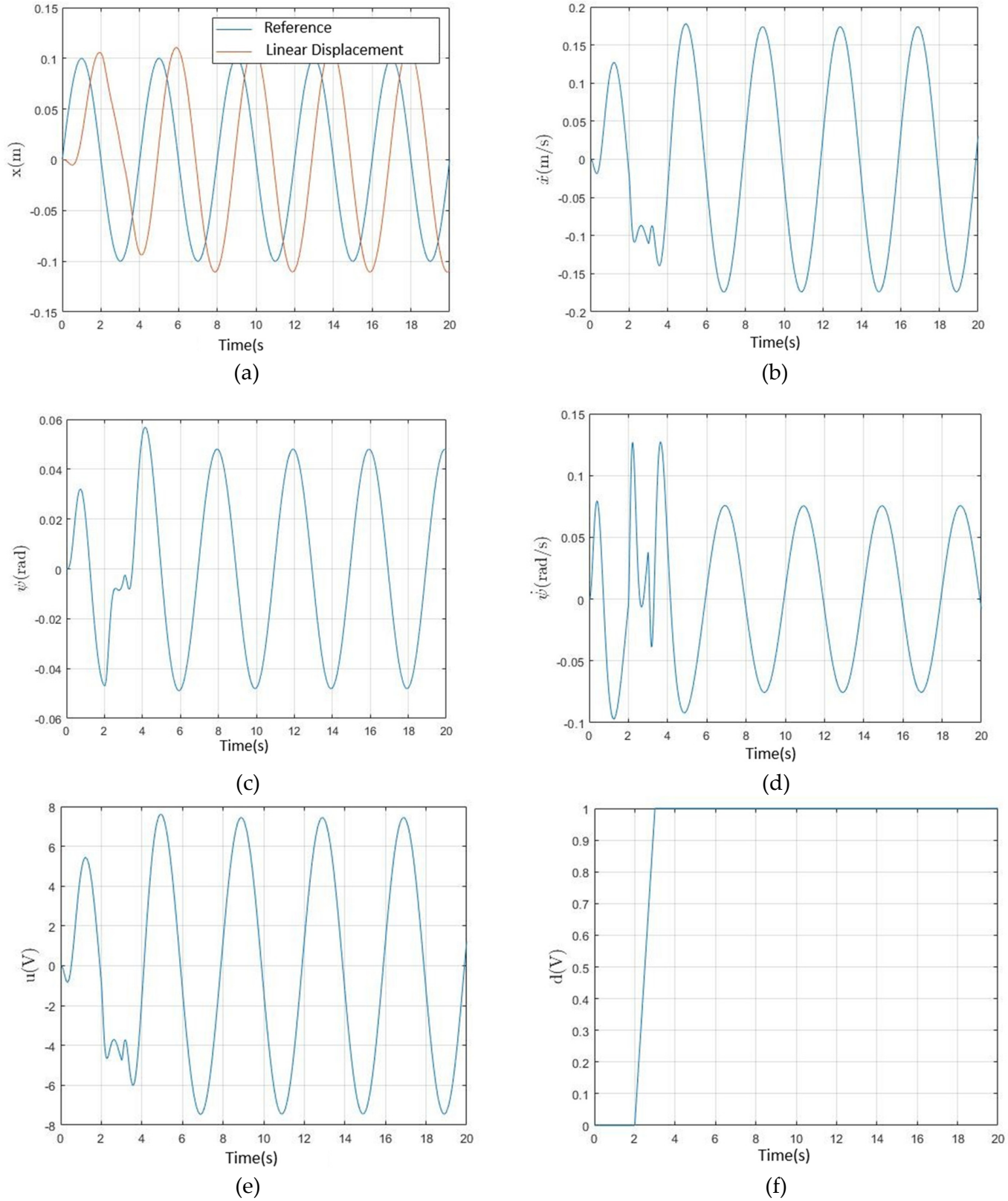


Figure 14. Under disturbance effect: (**a**) linear displacement versus reference, (**b**) linear velocity, (**c**) chassis tilt angle, (**d**) chassis tilt angular velocity, (**e**) drive motor input, (**f**) disturbance input.

3.3. Step Reference Input—Disturbance-Free Case

In this application, a step reference input with an amplitude of 0.1 m was applied to the robot chassis. Looking at the linear displacement response, it is observed that following the initial inverse response due to inertia, the system exhibits a maximum overshoot of only

0.01 m and settles precisely at the targeted reference position at t = 3 s. Similarly, the chassis tilt angle reached a fixed and stable position along the vertical axis at t = 2 s (Figure 15).

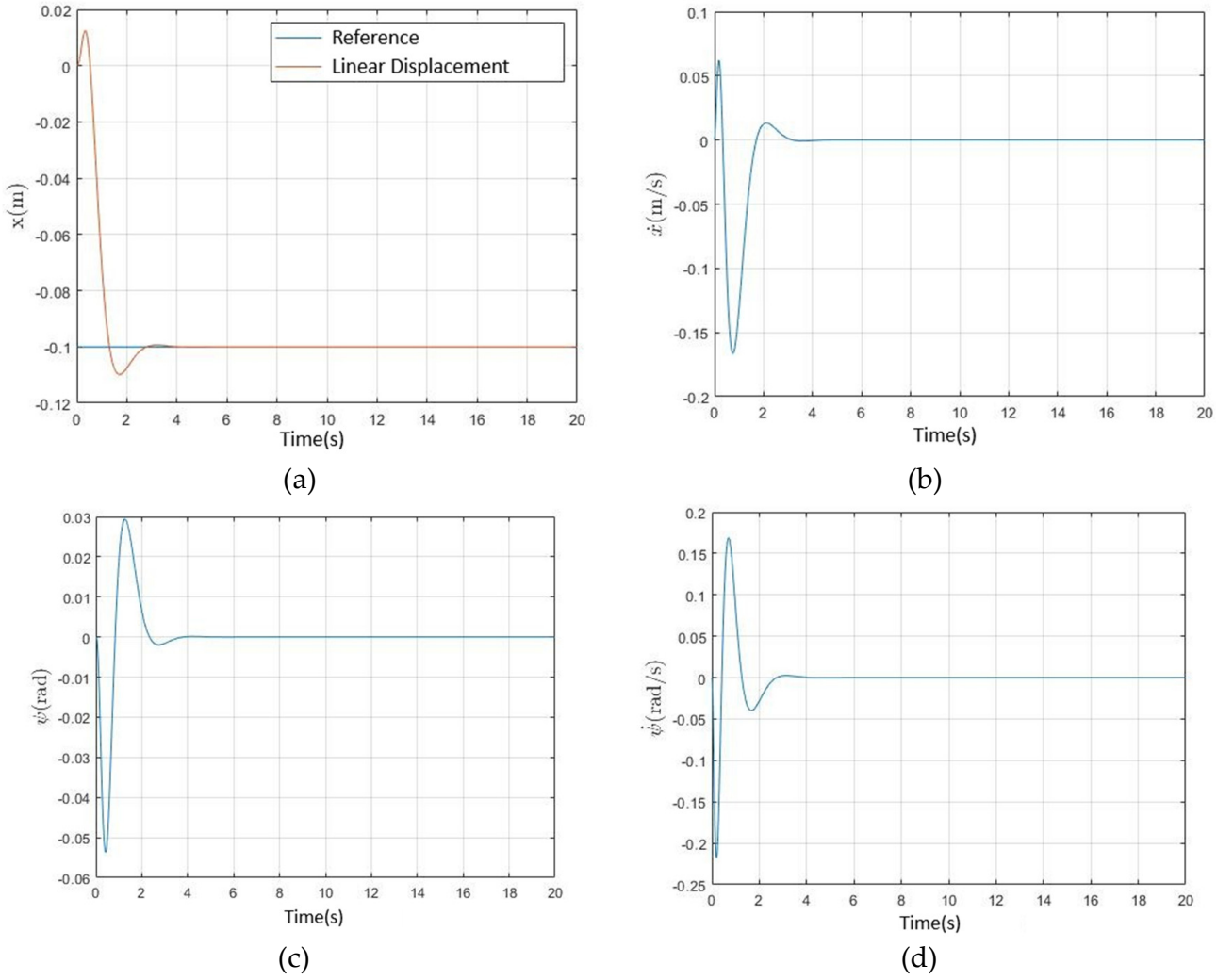


Figure 15. (**a**) Linear displacement versus reference, (**b**) linear velocity, (**c**) chassis tilt angle, (**d**) chassis tilt angular velocity.

3.4. Step Reference Input—Unit Step Disturbance Input Effect

During the step reference (0.1 m) tracking, the system was subjected to a unit step disturbance input at t = 2 s, while it had made its maximum overshoot but not yet fully settled into the steady state. These initial high-amplitude oscillations were suppressed thanks to the effective performance of the controller, and transition to the steady state was achieved in both linear displacement and chassis orientation at t = 7 s (Figure 16).

3.5. Sinusoidal Reference Input—Half-Sine Disturbance Input Effect

In the final scenario, during the sinusoidal reference tracking with a 0.1 m amplitude and a 0.25 Hz frequency, a periodic half-sine disturbance input with a 1 V amplitude and a 0.25 Hz frequency starting at t = 2 s was applied to the system. Since the disturbance input continues periodically, it is clearly observed that the reference tracking overshoots are higher within the intervals where the system is exposed to this effect. Similarly, distinct deviations occurred in the chassis tilt amplitudes at the moments when the externally applied disturbance input took effect (Figure 17).

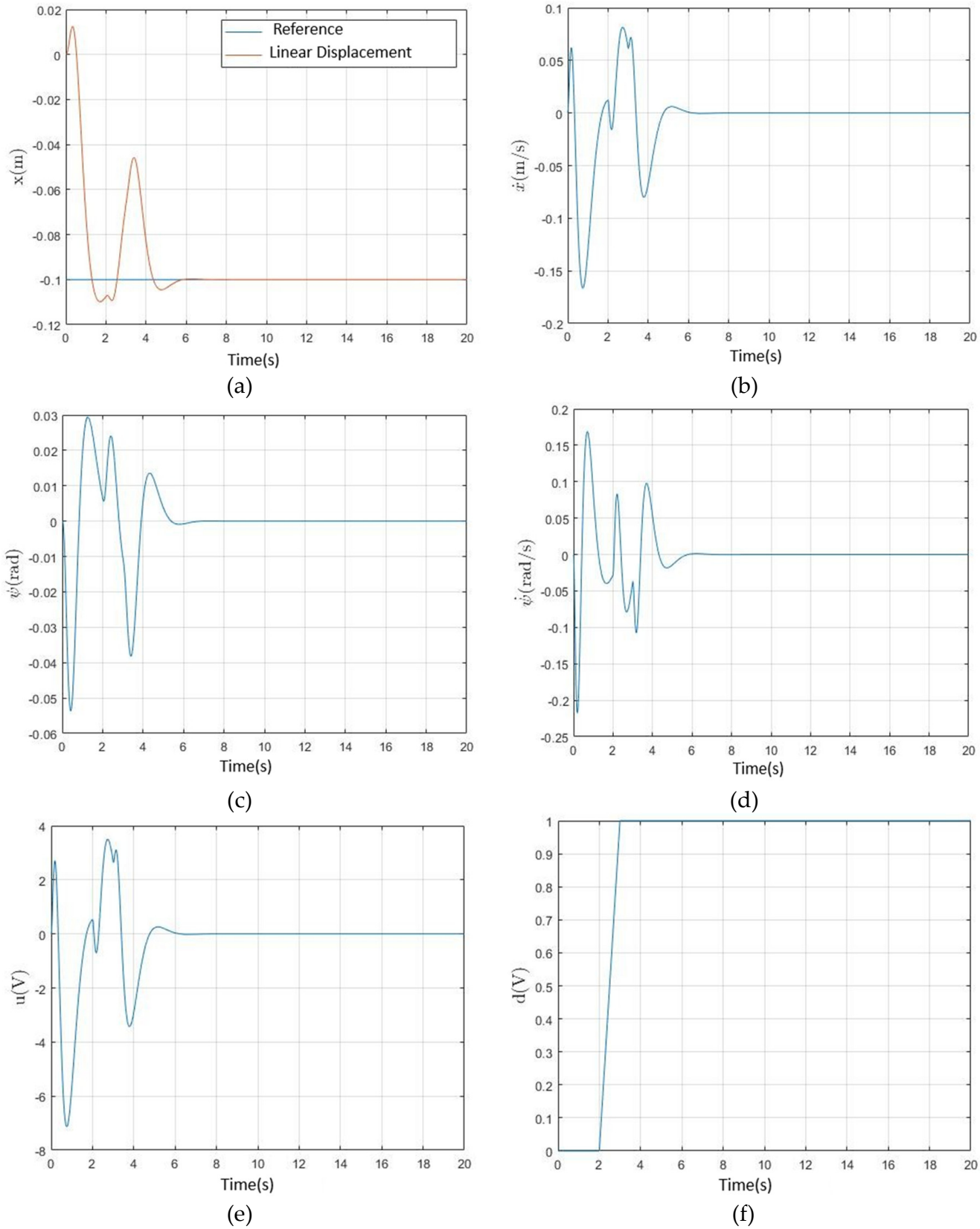


Figure 16. Under the effect of a disturbance input: (**a**) linear displacement versus reference, (**b**) linear velocity, (**c**) chassis tilt angle, (**d**) chassis tilt angular velocity, (**e**) drive motor input, (**f**) disturbance input.

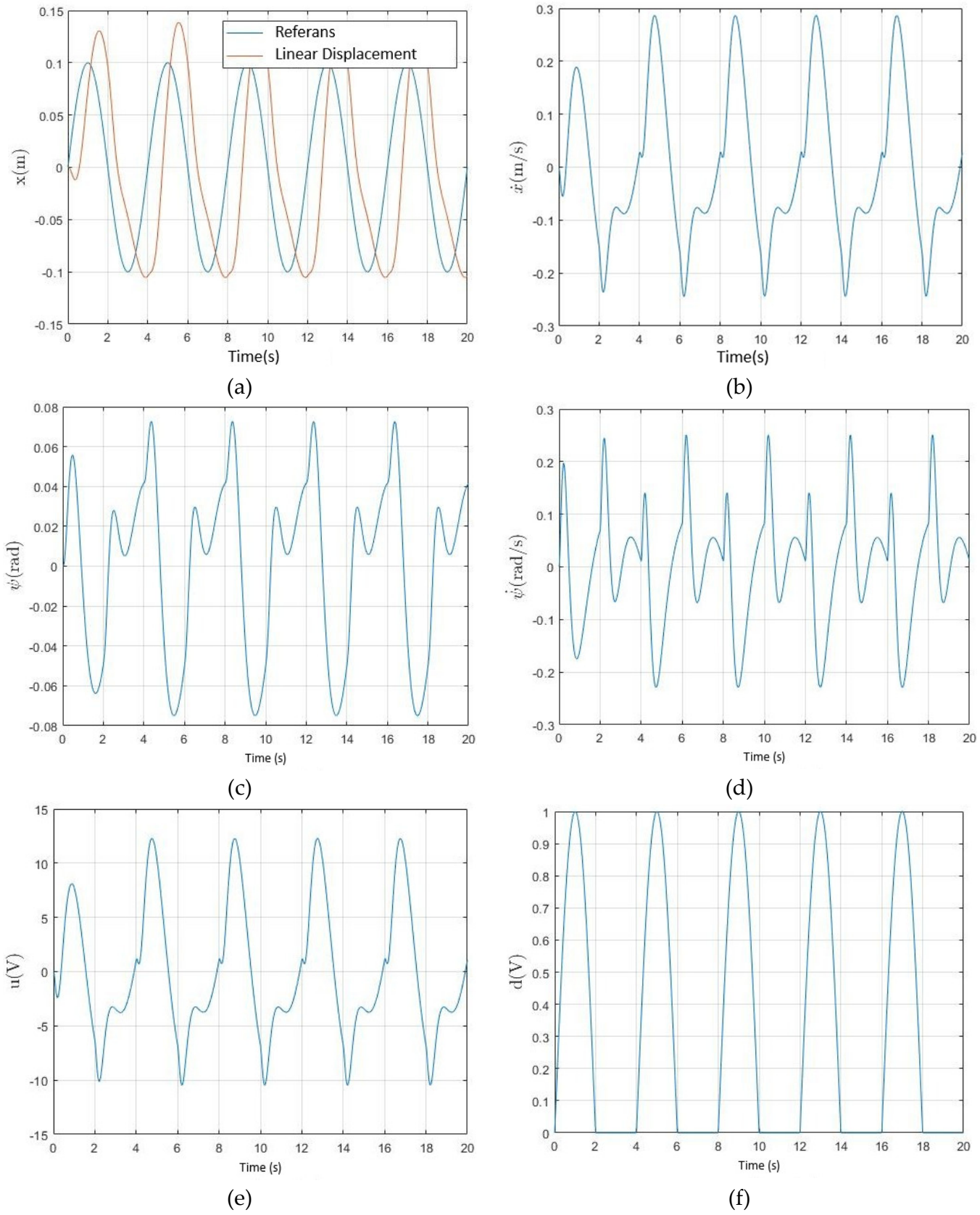


Figure 17. Under disturbance input effect: (**a**) linear displacement versus reference, (**b**) linear velocity; under disturbance effect: (**c**) chassis tilt angle, (**d**) chassis tilt angular velocity; (**e**) drive motor input, (**f**) disturbance input.

3.6. Comparative Analysis with Classical and Optimal Control Strategies

To quantify the contribution of the proposed supplementary control architecture, the proposed dual-PID with auxiliary PI (hereafter 'Proposed') is compared against three reference controllers implemented on the same reconstructed analytical model: a baseline dual-PID without the auxiliary PI block (hereafter 'Baseline'), a Linear Quadratic Regulator augmented with integral action (LQI), and a Hierarchical Sliding Mode Controller (HSMC). All four controllers were tuned on the nominal model using the same differential-evolution cost function and the same robustness constraint; each controller was required to maintain balance under a 10% simultaneous perturbation of chassis mass M and moment of inertia I_b before its nominal-model gains were accepted. The 20% perturbation scenario (Section 3.6.4) was therefore a genuine extrapolation test; no controller had been exposed to it during tuning.

Two methodological decisions merit explicit statement. First, the LQI formulation was adopted in place of the conventional proportional-only LQR because the physical system exhibits significant Coulomb friction at the wheel shaft. A pure LQR without integral action produces a non-zero steady-state position error under Coulomb friction, which would constitute an unfair handicap unrelated to the controller architecture under evaluation; integral augmentation restores a common baseline. Second, the HSMC was implemented in a cascaded form, in which an outer loop regulating the longitudinal position supplies a tilt-angle reference to an inner balance loop, following the subsystem-decomposition principle established for underactuated two-wheeled systems in the hierarchical sliding mode control literature [15,16]. In our implementation, attempts to drive longitudinal tracking and balance simultaneously from a single aggregated second-level sliding surface produced a persistent conflict between the two objectives, attributable to the non-minimum-phase relationship between motor voltage and chassis position; the cascaded arrangement, by assigning the balance objective priority through the inner loop, resolved this conflict and yielded lower tracking error under the tuning protocol adopted here (step-reference IAE reduced by more than 75% under the tuning protocol adopted here). This behaviour is specific to the present tuning and cost-function settings and is not a general statement about the single-surface formulation, which has been applied successfully in the cited works.

It should be noted that the controller gains used in this comparison were re-identified for the reconstructed simulation model and do not correspond directly to the hardware gains used in the physical experiments of Section 4, since the analytical model and the physical platform differ in unmodelled dynamics, sensor noise characteristics, and communication delays. The comparison is therefore valid as a relative performance assessment on a common analytical platform, not as a prediction of absolute hardware performance.

3.6.1. Step Reference with Nominal Parameters

A step reference of 1 rad in wheel angle was applied to all four controllers under nominal model parameters. The time-domain responses are shown in Figure 18a, and the performance metrics are summarised in Table 2.

Table 2. Controller performance comparison with step reference, nominal parameters.

Controller	OS (%)	Ts, 5% (s)	IAE (rad·s)	ISE (rad^2·s)	RMSE (rad)	$\int u^2$ (V^2·s)	max ψ (°)
Baseline PID	9.04	>12	1.0090	0.5080	0.2058	42.0	6.39
Proposed	1.47	2.08	0.6923	0.4333	0.1900	39.5	6.86
LQI	36.36	11.56	1.0148	0.5041	0.2050	49.1	9.00
HSMC	0.00	2.04	0.5636	0.4146	0.1859	44.0	7.99

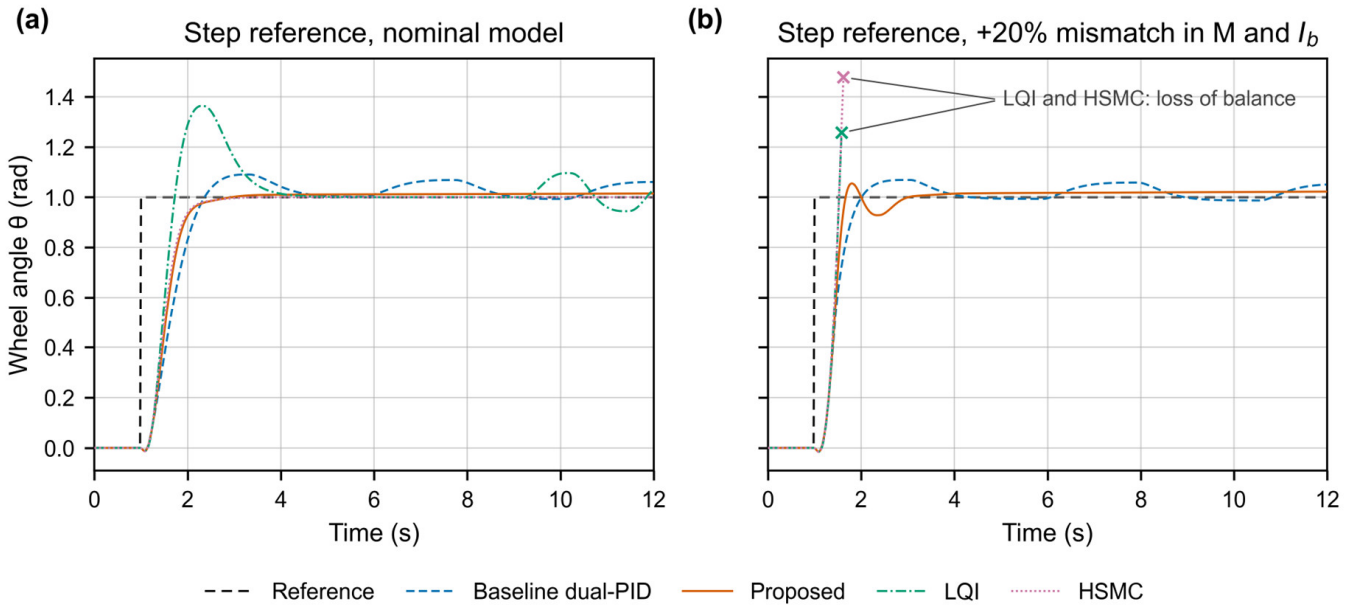


Figure 18. Position responses of the four controllers: (**a**) nominal parameters; (**b**) 20% mismatch in M and I_b. Loss of balance is marked where applicable.

Under nominal conditions, the HSMC achieves the lowest tracking error (IAE = 0.56 rad·s, RMSE = 0.186) together with the lowest overshoot (0.00%), while the Proposed controller attains a comparable settling time (2.08 s) with an overshoot of 1.47%, an order of magnitude below the Baseline (9.04%). The Baseline controller settles beyond the 12 s observation window, confirming that the auxiliary PI block provides a meaningful contribution to transient response. The LQI exhibits elevated overshoot (36.36%), which is a known consequence of the integral weight required to overcome Coulomb friction in an LQR framework; the integrator accumulates error during the initial inverse response of the non-minimum-phase plant, producing a large corrective action. This behaviour is inherent to the plant-controller interaction rather than to a suboptimal gain selection, and it is consistent with observations reported in similar two-wheeled platforms [15]. This nominal ranking is, however, reversed under parameter mismatch, as shown in Section 3.6.4.

3.6.2. Step Reference with External Disturbance

Under the same step reference, a disturbance equivalent to a 5° instantaneous inclination impulse was injected at t = 2 s. All four controllers maintained balance. The Proposed controller recovered to within 5% of the reference in 2.96 s following the disturbance event, while the Baseline required more than 12 s due to the absence of the integrating action in the auxiliary channel. The LQI disturbance recovery improved relative to its nominal-case settling (4.04 s vs. 11.56 s), reflecting the integral state reset following the impulse.

3.6.3. Sinusoidal Reference Tracking

A sinusoidal reference of 0.25 Hz and ±0.5 rad amplitude was tracked over 20 s. Table 3 reports the tracking metrics.

The Proposed controller achieves the lowest tracking error across all metrics (RMSE = 0.113) while the HSMC maintains the smallest chassis tilt excursion (1.39°) at the cost of reduced position tracking fidelity. The LQI and HSMC show higher IAE in the sinusoidal scenario relative to the step scenario, a consequence of the phase lag introduced by the integral states in a tracking context. The Proposed controller's auxiliary PI acts on the balance error rather than the position error directly, which limits its adverse effect on tracking bandwidth.

Table 3. Controller performance comparison with sinusoidal reference.

Controller	IAE (rad·s)	ISE (rad^2·s)	RMSE (rad)	$\int u^2$ (V^2·s)	max ψ (°)
Baseline PID	1.8314	0.3490	0.1706	74.7	3.50
Proposed	1.2065	0.1527	0.1128	67.4	4.00
LQI	3.5127	1.2986	0.3293	63.3	2.80
HSMC	2.8392	0.8394	0.2647	37.2	1.39

3.6.4. Robustness to Parameter Mismatch

To evaluate robustness, the chassis mass M and moment of inertia I_b were simultaneously increased by 20% above the values used during controller design, while all gain sets remained unchanged. This scenario emulates payload addition or model identification uncertainty. Table 4 presents the results.

Table 4. Controller performance under +20% simultaneous M and I_b mismatch. (—): simulation terminated at $|\psi| = 60°$.

Controller	OS (%)	Ts, 5% (s)	IAE (rad·s)	RMSE (rad)	$\int u^2$ (V^2·s)	max ψ (°)	Stable
Baseline PID	6.89	>12	0.7588	0.1806	81.6	8.03	Yes
Proposed	5.49	2.60	0.6445	0.1750	84.9	8.62	Yes
LQI	—	—	13.4686	1.1822	74.1	60.00	No
HSMC	—	—	15.7170	1.3833	85.3	60.00	No

Under +20% parameter mismatch, the LQI and HSMC lose stability, whereas both PID-based controllers maintain balance. The time-domain responses under nominal and mismatched conditions are contrasted in Figure 18, in which the loss of balance of the model-based controllers under mismatch is directly visible. To characterise the stability boundary more precisely, a gradual mismatch sweep was conducted at 10%, 15%, and 20% perturbation levels; the resulting settling times and stability outcomes are summarised in Figure 19.

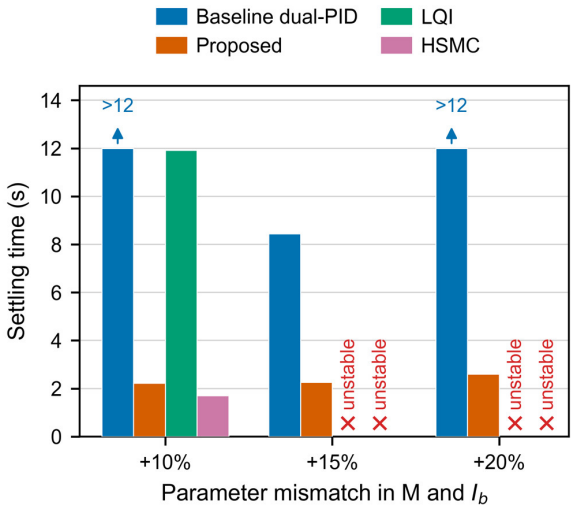


Figure 19. Settling time under progressive parameter mismatch. Red crosses denote loss of stability; arrows denote settling times exceeding 12 s.

The LQI and HSMC controllers lose stability between 10% and 15% mismatch. This behaviour is consistent with the sensitivity of model-based controllers to parameter uncertainty; the LQI gain matrix was computed from the nominal linearised system, and the HSMC sliding surface coefficients were derived from nominal dynamics; both rely on

accurate model knowledge for their stability guarantees. The proposed architecture, by contrast, does not require a dynamic model for its auxiliary PI channel; the integral action in that channel compensates for the force imbalance introduced by the parameter shift without explicit knowledge of its source. The Baseline controller also maintains stability under mismatch, but its settling time degrades severely (>12 s at 20%), confirming that the auxiliary PI block is responsible for the preserved transient performance of the Proposed controller rather than the dual-PID structure alone.

It is noted that the instability of the LQI and HSMC under mismatch reflects a deliberate test condition rather than a fundamental inadequacy of these methods; a model-based controller explicitly designed for robustness, for example, through H-infinity synthesis or mu-synthesis, would be expected to exhibit superior robustness margins. The present comparison evaluates the nominal design variants of each method under a common tuning protocol, as would be encountered in practical deployment where exact model knowledge is unavailable.

3.6.5. Weighting Coefficient Sensitivity

The proposed controller employs two scalar weighting coefficients, γ and α, that determine the relative contributions of the chassis tilt and wheel position channels to the total motor command (Equations (29) and (30)). These were initially selected by trial and error. To assess the sensitivity of the controller to variations in these parameters, a 5×5 grid sweep spanning $\pm 20\%$ perturbations around the nominal values was conducted, yielding 25 operating points (Figure 20). Of these, 19 maintained stable closed-loop operation. The unstable region forms a systematic pattern rather than a scattered set; instability occurs exclusively in the region where γ is reduced while α is increased, that is, where the weighting of the chassis balance channel is diminished relative to the wheel position channel. This pattern admits a direct physical interpretation, as the balance channel loses authority in the composite control law, wheel position corrections increasingly excite the chassis tilt without adequate compensation, and the inverted pendulum escapes its region of attraction. The nominal operating point selected by trial and error lies within the stable region with a margin of at least 10% in every direction of the (γ, α) plane, confirming that the selection constitutes a deliberate and robust operating point rather than a fragile one. Within the stable region, performance varies smoothly with no abrupt cliffs. The resulting stability boundary map provides the systematic characterisation of the weighting coefficient selection that a purely empirical tuning report would lack, and delineates the safe adjustment range available to a practitioner adapting the controller to a modified platform.

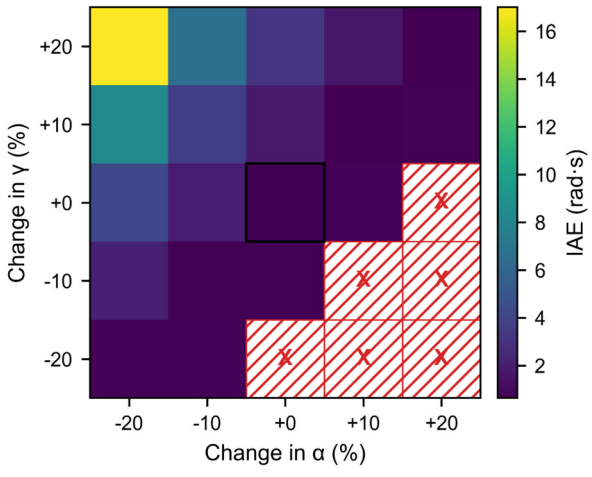


Figure 20. Sensitivity map of the weighting coefficients γ and α ($\pm 20\%$ grid). Colour indicates IAE; hatched cells denote loss of stability. The black frame marks the nominal operating point.

4. Experimental Studies

Within the scope of this study, a control application of the robot was performed on a trajectory that terminates with a ramp having a 0.2 slope. The trajectory of the robot was computed on a computer using the A* path planning algorithm, which determined a collision-free route around the obstacle by navigating the robot through a series of waypoints, moving 500 mm forward, turning left and advancing 500 mm, then turning right to reach the ramp, prior to uploading the planned path to the microcontroller. The balance of the robot at the moment it extended its arm on the inclined surface was observed. The interaction of the robot end-effector with any target point was not investigated. The trajectory planning that the robot is required to follow is provided in Figure 21.

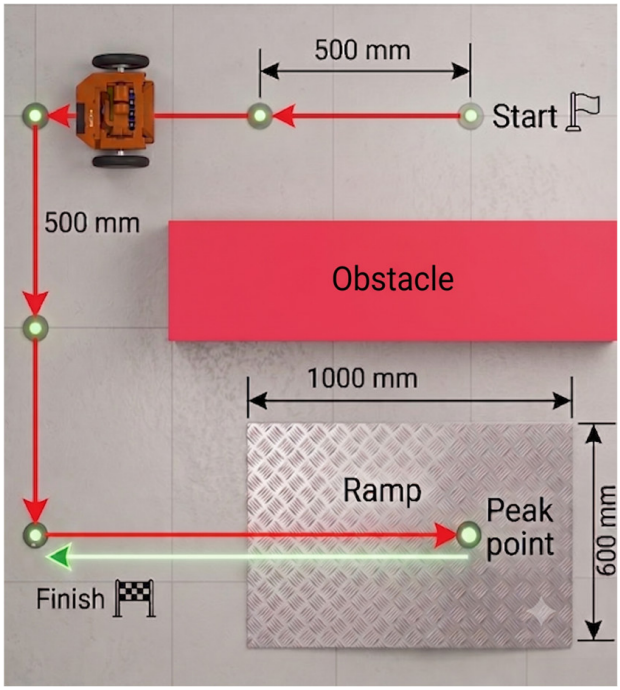


Figure 21. Experimental trajectory plan with ramp obstacle and waypoints.

The A* algorithm is employed here as an established and computationally inexpensive waypoint generator; no novelty is claimed with respect to path planning itself, and the contribution of this study resides in the control architecture that enables balanced navigation and manipulation on inclined terrain. The planning and control layers are deliberately separated in this architecture; the planner operates on the projected horizontal workspace and supplies a sequence of geometric waypoints, whereas terrain inclination is handled at the control layer through the inclination-dependent chassis tilt reference. As reported in this section the equilibrium tilt reference is adjusted according to the surface gradient encountered, so that the ramp is negotiated without requiring the slope to be embedded in the planning cost function. This separation keeps the planning stage light enough to be executed with the modest computational resources of the onboard platform, while confining the terrain-dependent behaviour to the controller, where it is directly observable and can be tuned independently of the route geometry. The manipulator is likewise not represented within the planning problem, since in the adopted operating sequence it is deployed only after the platform has reached and stabilised at the target waypoint, and therefore does not influence the geometric feasibility of the planned route.

In the trajectory plan, red arrows indicate the forward path while the green arrow denotes the return direction. The green dots represent the A* computed waypoints uploaded to the microcontroller. Upon reaching the peak point of the ramp, the robot performs a 90°

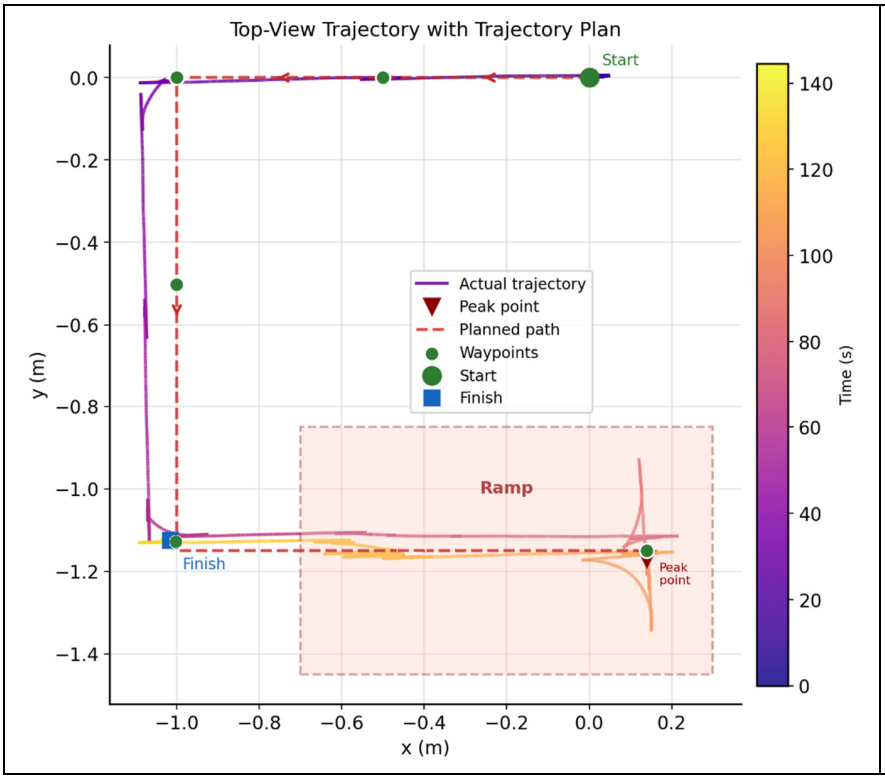
rotation, executes the manipulator arm extension-retraction sequence, and then completes a second 90° rotation before descending to the finish point. Throughout this maneuver, stable balance on the inclined surface was successfully maintained. Figure 22 shows the experimental testbed and representative snapshots taken during the physical implementation of the proposed trajectory.



(a) (b) (c)

Figure 22. Robot movement stages: (**a**) initialization, (**b**) cruising, (**c**) ramp climbing.

Figure 23 illustrates the experimental trajectory executed under the proposed control architecture. Beginning at the origin (point 1, Cartesian coordinates {0, 0, 0}), the robot autonomously navigated through six waypoints pre-loaded into the microcontroller, while the balance controller continuously maintained postural stability throughout the motion. Successive waypoints 1 through 5 are separated by 0.5 m intervals, whereas the final segment between points 5 and 6 spans 1 m and traverses a ramp with a 0.2 slope. The motion commands assigned to each waypoint are depicted below:



- Move 0.5 m from 1 to 2 and wait,
- Move 0.5 m from 2 to 3 and wait,
- Rotate 90° at 3,
- Move 0.5 m from 3 to 4 and wait,
- Move 0.5 m from 4 to 5 and wait,
- Rotate 90° at 5,
- Move 1 m from 5 to 6 and wait,
- Rotate 90° at 6 and wait,
- Deploy robot arm to {x, y, z}={0, 0, 0.175} and wait,
- Retract robot arm to {x, y, z}={0, 0, 0} and wait,
- Rotate 90° at 6 and wait,
- Move 1 m from 6 to 5 and wait.

Figure 23. Real-time odometry-based trajectory of the robot in the x-y plane. Motion commands: move 0.5 m (waypoints 1–5), rotate 90° at waypoints 3 and 5, move 1 m to ramp apex, deploy arm to {0, 0, 0.175} and retract, return.

The real-time trajectory plot provided in Figure 23 was generated based on the position data acquired from the encoders located in the wheel drive motors. Between points 1 and 3, it is observed that the robot successfully passes through the designated coordinate points. At the turning points, the robot first stops, executes the rotation, and then proceeds to advance. During the 90° turn after point 3, a deviation of approximately 10 cm was observed. Subsequently, despite this deviation, the robot's advancement between points 3 and 5 appears to be stable. A similar deviation was also observed when the robot performed a 90° turn at point 5. Between points 5 and 6, a stable trajectory tracking toward the ramp is again visible. Due to the inertial effects resulting from the extension and retraction of the robot manipulator at the peak of the ramp, it can be stated that the robot's balance was affected, causing a slight oscillation.

Figure 24 displays the voltage plots belonging to the right and left drive motors. Between 67 and 81 s, the robot climbs the ramp. In this interval, it is observed that the motors reach saturation. From the 114th second onwards, the robot descends the ramp.

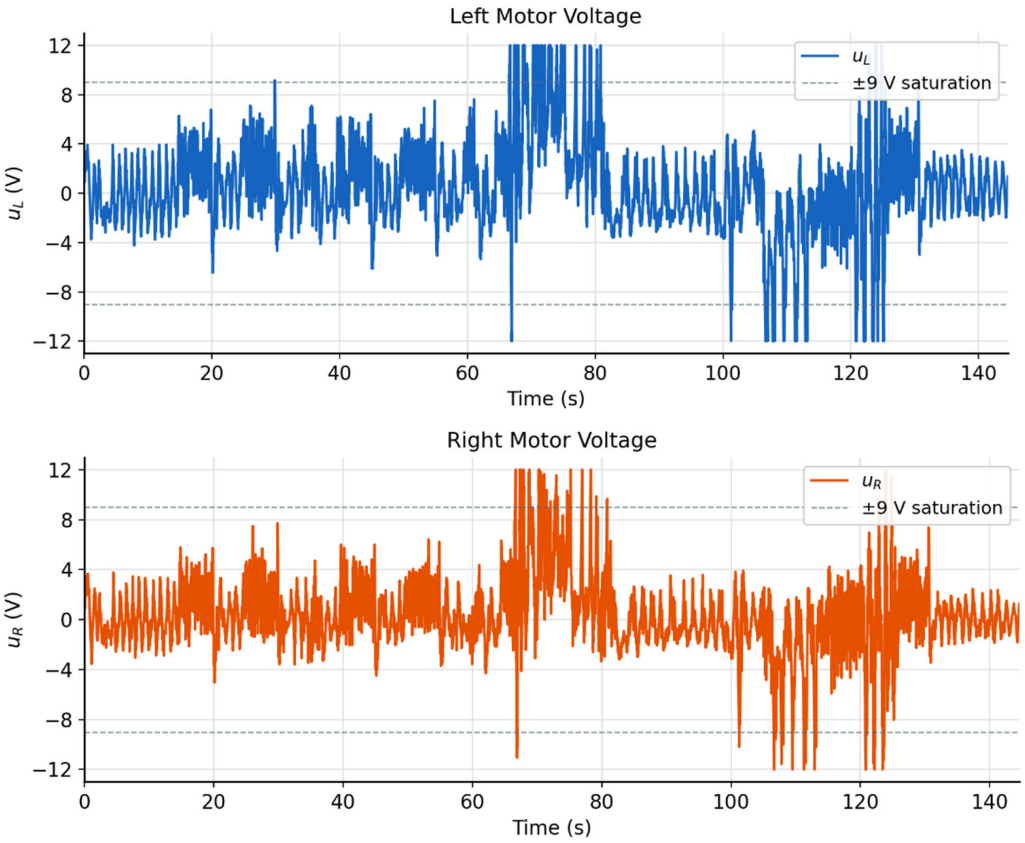


Figure 24. Input values of the robot drive motors.

Figure 25 illustrates the robot tilt angle, ψ. Since the projection of the robot's center of gravity falls outside the wheel wheelbase area, the reference input on flat ground is $-8°$. The reference input required for the robot to climb the 0.2-gradient ramp is $-32°$. On the other hand, the reference value required for the robot to descend the ramp is 10°.

In the inverse kinematics solution of the robot, the robot arm is targeted to move along the z-axis of its own coordinate frame, from an initial position of 0.02 m to a reference position of 0.178 m in XYZ coordinates. The initial coordinate of 0.02 m stems from the mechanical limitations of the robot arm (Figure 26).

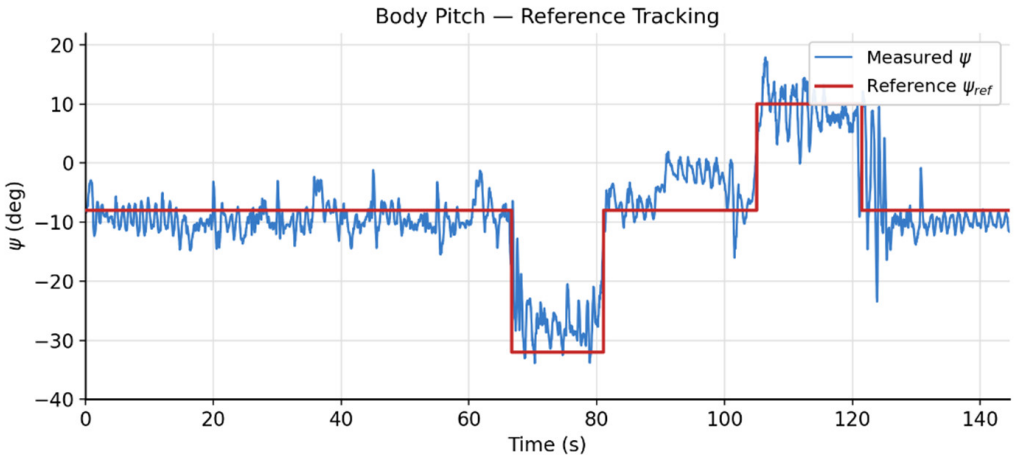


Figure 25. Body tilt angle response under step reference inputs.

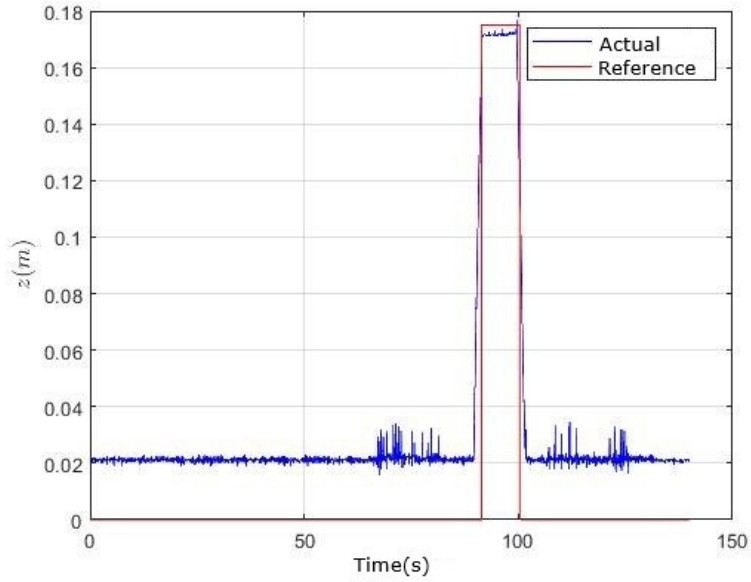


Figure 26. End-effector displacement along the z-axis in the tool frame.

4.1. Quantitative Validation Metrics

The experimental results presented above are here accompanied by quantitative metrics derived directly from the recorded data to complement the graphical results.

4.1.1. Dynamic Parameter Estimation Accuracy

The parameter identification procedure described in Section Parameter Estimation of the Mobile Chassis Dynamics produced estimated time-series for angular acceleration (Figure 9) and motor current (Figure 11). The accuracy of these estimates was evaluated by computing error metrics between the estimated and measured series over the full 0–100 s identification interval. Results are reported in Table 5.

Table 5. Parameter estimation accuracy metrics. Fit (%) = 100 × (1 − RMSE/std(measured)).

Signal	RMSE	MAE	Fit (%)	Correlation (r)
Angular accel. ($\ddot{\theta}$)	18.44 rad/s^2	10.30 rad/s^2	61.0	0.921
Motor current (i)	0.354 A	0.294 A	23.4	0.752

The angular acceleration estimate achieves a correlation of r = 0.921 with the measured signal, indicating that the identified mechanical parameters capture the dominant dynamics of the system. The fit percentage of 61.0% reflects the high-amplitude, high-frequency content of the chirp-excited signal, which includes components outside the bandwidth of the identification model of Equation (25); the correlation metric is more indicative of model fidelity in this context. The current estimate yields a lower fit percentage (23.4%) while maintaining r = 0.752; the residuals are dominated by unmodelled brush-commutation transients and switching noise from the motor driver, which are not represented in the electromechanical model of Equation (25). These unmodelled components do not affect the identified parameters used in the control design, since the parameter extraction minimises the error on the acceleration signal rather than the current signal directly.

4.1.2. Trajectory Tracking Performance

The body pitch angle was controlled to three successive reference levels during the experimental trajectory: −8° on level ground, −32° during ramp ascent, and 10° during ramp descent (Figure 25). Steady-state metrics, computed after excluding the 2 s transient following each reference transition, are reported in Table 6.

Table 6. Body pitch tracking metrics—experimental results.

Phase	Reference	Mean (°)	RMSE (°)	MAE (°)
Level ground	−8°	−8.36	3.52	2.73
Ramp ascent	−32°	−27.14	5.90	4.94
Ramp descent	+10°	+8.10	4.25	3.08
Overall	—	—	3.89	2.97

The controller maintains the target pitch angle on level ground with an RMSE of 3.52°. During ramp ascent, the RMSE increases to 5.90° and a mean bias of +4.86° is observed (measured −27.14° vs. reference −32°), attributable to the higher motor torque demand during climbing, which reduces the margin available for attitude correction. The largest instantaneous deviations coincide with the intervals of actuator saturation identified in Section 4.1.3. During ramp descent, the RMSE is 4.25° with a mean bias of −1.90°.

4.1.3. Actuator Utilisation

The voltage commands to the drive motors were recorded throughout the experimental trial (Figure 24). Table 7 summarises the actuator utilisation statistics by phase, where saturation is defined as $|u| \geq 9$ V (operating voltage limit) and full saturation as $|u| = 12$ V (supply rail).

Table 7. Actuator utilisation statistics by phase.

Phase	Duration (s)	Left $\geq$ 9 V (%)	Left 12 V (%)	Right $\geq$ 9 V (%)	Right 12 V (%)
Level ground	~104	1.3	0.5	0.6	0.2
Ramp ascent	16.5	26.9	9.7	9.9	2.8
Ramp descent	18.9	11.3	3.9	5.5	1.2
Overall	**~140**	5.0	1.8	2.1	0.5

On level ground the controller operates well within the actuator limits, with saturation occurring in less than 1.3% of samples. During ramp ascent, the left motor command exceeds 9 V in 26.9% of samples and reaches the 12 V supply rail in 9.7% of samples, confirming the elevated torque demand during climbing noted qualitatively above. The

asymmetry between left and right motor saturation rates (26.9% vs. 9.9% at $\geq$9 V during ascent) indicates a lateral mass distribution offset, consistent with the non-symmetric placement of the electro-mechanical parts. Despite these saturation intervals, the balance loop was maintained throughout, demonstrating the robustness of the proposed controller to temporary actuator limitations.

The control loop was executed at 50 Hz on the Teensy 4.0 microcontroller (ARM Cortex-M7, 600 MHz), with hardware encoder sampling at 200 kHz via the dedicated GPT1 timer peripheral and IMU data acquisition at 10 kHz via I^2C. Under full telemetry load, the effective control update rate was measured at approximately 44 Hz due to communication jitter. Motor PWM was generated at 18.31 kHz (13-bit resolution). This execution profile demonstrates that the proposed control architecture is compatible with resource-constrained embedded hardware without requiring dedicated real-time processors.

4.1.4. End-Effector Positioning Accuracy

During the manipulation sequence at the ramp apex, the robot arm was commanded to extend along its tool-frame z-axis from an initial position to a reference displacement of $\Delta z = 0.158$ m (corresponding to the 0.020 m to 0.175 m stroke shown in Figure 26, where the 0.020 m initial offset reflects the mechanical zero reference of the manipulator arm as defined by its joint angle convention). The recorded displacement reached $\Delta z = 0.1525$ m, yielding an amplitude error of -3.5%. During the plateau phase, the end-effector position was maintained with an RMSE of 2.2 mm and a MAE of 1.4 mm, demonstrating millimetre-level positioning accuracy despite the balance disturbances induced by the arm motion.

4.1.5. Planar Trajectory Accuracy and Error Analysis

The trajectory was defined by six waypoints including a ramp segment with a 0.2 gradient (Figure 21). The robot successfully traversed all waypoints and returned to within 2.26° of its initial heading at the conclusion of the trial.

Trajectory deviations of approximately 10 cm were observed during the 90° turn manoeuvres at waypoints 3 and 5. Three mechanisms contribute to these deviations. First, differential-drive turning produces instantaneous velocity differentials between the wheels that induce slip at the tyre–ground interface; the odometry system integrates encoder pulses and therefore registers slip as forward motion, introducing an immediate position error. Second, any heading error accumulated during a turn propagates as a lateral offset over the subsequent straight segment: a 5° heading error over a 0.5 m segment produces approximately 4.4 cm of lateral displacement. Third, the balance controller continues to issue pitch-correction commands during the turn, altering the effective normal load distribution between the wheels and amplifying the asymmetric slip. The saturation statistics of Table 7 support this interpretation: the elevated motor saturation during ramp climbing, where lateral asymmetry was most pronounced, is consistent with unequal wheel loading.

Mitigation strategies for future work include: (i) IMU yaw-odometry fusion via a complementary or extended Kalman filter to bound heading drift; (ii) trapezoidal angular velocity profiling during turns to reduce peak slip torque; (iii) post-turn localisation correction using distance or landmark measurements; and (iv) coordinated reduction in translational speed during turning to reduce centrifugal load transfer.

5. Results and Discussions

In this study, the hardware design, manufacturing, and control of a two-wheeled inverted pendulum manipulator (TWIPM), which consists of a two-degree-of-freedom independently driven chassis and a three-degree-of-freedom manipulator, were success-

fully achieved. To enable the system to reach target coordinates on inclined surfaces, an inclination-aware control algorithm was developed to detect and compensate for the ramp gradient, and the required trajectory was planned using the A* algorithm. To maintain the system's balance stability and enhance its robustness against external disturbances, a PI-based 'supplementary control architecture' was developed to support the PID controllers regulating the wheel angular position and chassis tilt.

The proposed control method was first validated through numerical simulations under various reference and disturbance inputs on a flat surface. Subsequently, experimental studies conducted on physical hardware featuring an inclined ramp demonstrated the system's capacity for autonomous trajectory tracking and its successful attainment of targeted coordinates, thereby confirming the controller's validity under real-world conditions.

In light of the successful findings obtained, various studies are envisioned to advance the system's autonomy and operational capabilities further. Within this scope, testing alternative control strategies on the developed configuration and integrating new hardware into the manipulator's end-effector to expand the robot's manipulation capacity are planned. Additionally, optimizing current position control algorithms for the end-effector to reach targeted spatial coordinates with much higher precision and integrating a payload estimator capable of providing dynamic response under varying weights are aimed. Finally, running navigation and trajectory planning algorithms directly on the robot's onboard central microcontroller without requiring an external computer, alongside a performance comparison of different autonomous algorithms, will be among the focal points of future research.

6. Conclusions

In this study, the comprehensive hardware development, manufacturing, and robust control implementation of a Two-Wheeled Inverted Pendulum Manipulator (TWIPM) have been successfully accomplished. By combining a two-degree-of-freedom independently driven mobile platform with a three-degree-of-freedom robotic manipulator, a highly agile yet dynamic system configuration was achieved.

The core contributions and crucial findings of this research can be summarized as follows:

- Autonomous Trajectory Generation: The integration of the A* algorithm enabled the robot to autonomously drive collision-free paths around obstacles toward target coordinates, which were subsequently executed via the onboard microcontroller.
- Robust Control Performance: To tackle the inherent instability of the dual-wheeled structure, a PI-based supplementary control architecture was introduced to assist the main PID loops. Numerical simulations and real-world experimental validations confirmed that this control scheme significantly enhances balance stability and provides superior robustness against external disturbances.
- Experimental Validation on Inclined Terrains: The experimental trials performed on a physical 0.2-gradient ramp demonstrated that the robot preserves its stabilization during flat-to-inclined terrain transitions. Despite minor tracking errors (approximately 10 cm) during sharp 90° turns and transient inertial oscillations induced by the rapid extension/retraction of the manipulator at the peak of the ramp, the proposed controller successfully drove the system to its final destination without tipping over.
- Actuator Saturation Analysis: Motor voltage profiles indicated that while the drive actuators underwent saturation during the steep ramp-climbing phase (between 67 and 81 s), the system maintained its structural integrity and balance loop without failing.

https://doi.org/10.3390/act15080433

Several limitations of the present study delimit the scope of its conclusions. The physical validation was conducted on a single prototype and a single 0.2-gradient ramp; this gradient being fixed by the available testbed. Although the experimental trial spans level-ground, ascent, and descent conditions with per-phase performance quantified in Table 6, the full range of inclinations over which the model and controller maintain their performance has not been established experimentally. The actuator utilisation statistics of Table 7 indicate that this range will be bounded primarily by motor saturation during climbing, where the commanded voltage exceeded 9 V for 26.9% of the ascent phase. With respect to payload, the robustness analysis of Section 3.6.4 shows that the controller tolerates a +20% deviation in chassis mass and inertia without loss of stability; heavier or asymmetric payloads, as well as rapid manipulator motion, fall outside the validated envelope of the lumped-mass model as discussed in Section 2.3. A systematic experimental campaign across multiple inclinations, payload configurations, and surface conditions, with repeated trials and statistical characterisation, constitutes the planned next stage of this work. Within that stage, the payload estimator identified above is intended to integrate directly with the existing architecture: operating on the discrepancy between commanded torque and observed chassis response, its output would supply an adaptive feedforward correction to the auxiliary PI channel, the component shown in Section 3.6.4 to absorb mass-parameter deviations, while onboard execution of the planning layer would replace the host-computer A* run with an embedded implementation on the same waypoint interface, requiring no modification of the control architecture.

In conclusion, the proposed configuration and control architecture have proven their viability under real-world physical constraints and environmental disturbances. Future work will focus on optimizing the end-effector positioning loops for higher spatial precision, embedding a dynamic payload estimation algorithm to handle varying cargo weights, and migrating the trajectory planning algorithms to run completely onboard on the robot's central microcontroller to achieve full operational autonomy.

Since the present study adopts A* as an offline planner executed prior to task initiation, sampling-based and reactive planning strategies such as RRT*, dynamic window approaches, and optimisation-based planners capable of online replanning were not evaluated. Incorporating such planners, together with terrain gradient and manipulator configuration as explicit terms in the planning cost, would extend the system towards operation in previously unmapped and dynamically changing environments, and constitutes a natural continuation of this work.

Author Contributions: Conceptualization, methodology, software, data curation, writing—original draft preparation, visualization, S.E.K.; writing—review and editing, supervision, O.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by the Firat University Scientific Research Projects Management Unit (FUBAP) with Project Number MF.25.138.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ha, Y.; Yuta, S. Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot. *Robot. Auton. Syst.* **1996**, *17*, 65–80. [CrossRef]
2. Zhai, Y.; Zhang, S. A Novel LiDAR–IMU–Odometer Coupling Framework for Two-Wheeled Inverted Pendulum (TWIP) Robot Localization and Mapping with Nonholonomic Constraint Factors. *Sensors* **2022**, *22*, 4778. [CrossRef] [PubMed]
3. Klokowski, P.; Eßer, J.; Gramse, N.; Pschera, B.; Plitt, M.; Feldmeier, F.; Bajpai, S.; Jestel, C.; Bach, N.; Urbann, O.; et al. evoBOT—Design and Learning-Based Control of a Two-Wheeled Compound Inverted Pendulum Robot. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2023; pp. 10425–10432. [CrossRef]
4. Lentini, G.; Settimi, A.; Caporale, D.; Garabini, M.; Grioli, G.; Pallottino, L.; Catalano, M.G.; Bicchi, A. Alter-Ego: A Mobile Robot With a Functionally Anthropomorphic Upper Body Designed for Physical Interaction. *IEEE Robot. Autom. Mag.* **2019**, *26*, 94–107. [CrossRef]
5. Grasser, F.; D'Arrigo, A.; Colombi, S.; Rufer, A.C. JOE: A mobile, inverted pendulum. *IEEE Trans. Ind. Electron.* **2002**, *49*, 107–114. [CrossRef]
6. Tirmant, M.; Baloh, L.; Vermeiren, T. B2, an alternative two wheeled vehicle for an automated urban transportation system. In *Intelligent Vehicle Symposium*; IEEE: Piscataway, NJ, USA, 2002; Volume 2. [CrossRef]
7. Stilman, M.; Wang, J.; Teeyapan, K.; Marceau, R. Optimized control strategies for wheeled humanoids and mobile manipulators. In *9th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS09*; IEEE: Piscataway, NJ, USA, 2009; pp. 568–573. [CrossRef]
8. Önkol, M.; Kasnakoğlu, C. Adaptive Model Predictive Control of a Two-wheeled Robot Manipulator with Varying Mass. *Meas. Control* **2018**, *51*, 38–56. [CrossRef]
9. Zhang, L.; Wang, Z.; Li, L. Design and implementation of two-wheeled mobile robot by variable structure Sliding Mode Control. In *Proceedings of the 2016 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 5869–5873. [CrossRef]
10. Muhammad, M.; Buyamin, S.; Ahmad, M.N.; Nawawi, S.W. Dynamic Modeling and Analysis of a Two-Wheeled Inverted Pendulum Robot. In *Proceedings of the 2011 Third International Conference on Computational Intelligence, Modelling & Simulation, Langkawi, Malaysia, 20–22 September 2011*; IEEE: Piscataway, NJ, USA, 2011; pp. 159–164. [CrossRef]
11. Oryschuk, P.; Salerno, A.; Al-Husseini, A.M.; Angeles, J. Experimental Validation of an Underactuated Two-Wheeled Mobile Robot. *IEEE/ASME Trans. Mechatron.* **2009**, *14*, 252–257. [CrossRef]
12. Petcu, R.; Andreescu, G.D. Two-Wheeled Inverted-Pendulum Self-Balancing Robot with LQR or PID Control. In *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*; IEEE: Piscataway, NJ, USA, 2022; pp. 000345–000350. [CrossRef]
13. Albani, M.; Mustofa, E.; Binugroho, E.H.; Ningrum, E.S. Robust Balancing on One-wheeled Human Transporter Based on Rider's Weight Variation Using PID-AFC Method. In *Proceedings of the 2019 6th International Conference on Instrumentation, Control, and Automation (ICA), Bandung, Indonesia, 31 July–2 August 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 131–136. [CrossRef]
14. Velagić, J.; Kovač, I.; Panjević, A.; Osmanović, A. Design and Control of Two-Wheeled and Self-Balancing Mobile Robot. In *Proceedings of the 2021 International Symposium ELMAR, Zadar, Croatia, 13–15 September 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 77–82. [CrossRef]
15. Pham, V.C.; Tran, N.; Ngo, H.H.; Mac, P.; Tran, H. Robust Optimal Control for Two-Wheeled Inverted Pendulum via Hierarchical Sliding Mode Control and High-Order Disturbance Observer with Off-Policy Adaptive Dynamic Programming. *J. Control Autom. Electr. Syst.* **2026**, *37*, 746–763. [CrossRef]
16. Hou, M.; Zhang, X.; Chen, D.; Xu, Z. Hierarchical Sliding Mode Control Combined with Nonlinear Disturbance Observer for Wheeled Inverted Pendulum Robot Trajectory Tracking. *Appl. Sci.* **2023**, *13*, 4350. [CrossRef]
17. Kausar, Z.; Stol, K.; Patel, N. The Effect of Terrain Inclination on Performance and the Stability Region of Two-Wheeled Mobile Robots. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 218. [CrossRef] [PubMed]
18. Zhou, H.; Li, X.; Feng, H.; Li, E.; Ding, P.; Zhai, Y.; Zhang, S.; Fu, Y. Control of the Two-wheeled Inverted Pendulum (TWIP) Robot Moving on the Continuous Uneven Ground. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*; IEEE: Piscataway, NJ, USA, 2019; pp. 1588–1594. [CrossRef]
19. Dai, F.; Gao, X.; Jiang, S.; Liu, Y.; Li, J. A multi-DOF two wheeled inverted pendulum robot climbing on a slope. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*; IEEE: Piscataway, NJ, USA, 2014; pp. 1958–1963. [CrossRef]
20. Zhang, J.; Zhao, T.; Guo, B.; Dian, S. Fuzzy fractional-order PID control for two-wheeled self-balancing robots on inclined road surface. *Syst. Sci. Control Eng.* **2022**, *10*, 289–299. [CrossRef]
21. Sun, Z.; Dong, R.; Zhao, N. Standard dynamic model and control of a two-wheeled inverted pendulum robot on inclined planes. *Int. J. Dyn. Control* **2026**, *14*, 56. [CrossRef]

22. Feng, T.; Liu, T.; Wang, X.; Xu, Z.; Zhang, M.; Han, S.C. Modeling and implementation of two-wheel self-balancing robot equipped with supporting arms. In *2011 6th IEEE Conference on Industrial Electronics and Applications*; IEEE: Piscataway, NJ, USA, 2011; pp. 713–718. [CrossRef]
23. Zahid, T.; Kausar, Z.; Shah, M.F.; Saeed, M.T.; Pan, J. Workspace Design and Trajectory Planning of a Five Degree of Freedom Mobile Welding Manipulator for Spherical Objects. *IEEE Access* **2021**, *9*, 73843–73856. [CrossRef]
24. Candra, A.; Budiman, M.A.; Hartanto, K. Dijkstra's and A-Star in Finding the Shortest Path: A Tutorial. In *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*; IEEE: Piscataway, NJ, USA, 2020; pp. 28–32. [CrossRef]
25. Lin, Z.; Wu, K.; Shen, R.; Yu, X.; Huang, S. An Efficient and Accurate A-Star Algorithm for Autonomous Vehicle Path Planning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 9003–9008. [CrossRef]
26. Chatzisavvas, A.; Dossis, M.; Dasygenis, M. Optimizing Mobile Robot Navigation Based on A-Star Algorithm for Obstacle Avoidance in Smart Agriculture. *Electronics* **2024**, *13*, 2057. [CrossRef]
27. Jiang, J.; Han, Z.; Li, J.; Wang, Y.; Wang, J.; Xu, S. Global Path Planning of UGVs in Large-Scale Off-Road Environment Based on Improved A-star Algorithm and Quadratic Programming. In *2023 IEEE Intelligent Vehicles Symposium (IV)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–7. [CrossRef]
28. Hong, Z.; Sun, P.; Tong, X.; Pan, H.; Zhou, R.; Zhang, Y.; Han, Y.; Wang, J.; Yang, S.; Xu, L. Improved A-Star Algorithm for Long-Distance Off-Road Path Planning Using Terrain Data Map. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 785. [CrossRef]
29. Pu, H.; Wan, X.; Song, T.; Schonfeld, P.; Peng, L. A 3D-RRT-star algorithm for optimizing constrained mountain railway alignments. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107770. [CrossRef]
30. Kara, S.E.; Yakut, O. Design and Control of a Self-Balancing Two-Wheeled Mobile Manipulator on Inclined Surface. In Proceedings of the International Scientific Research Congress Dedicated to the 30th Anniversary of Baku Eurasia University, Baku, Azerbaijan, 27–28 April 2022; pp. 341–348.
31. Kara, S.E. Design of Self-Balancing Two-Wheel Mobile Manipulator on Inclined Plane. Ph.D. Thesis, Fırat University, Elazığ, Türkiye, 2023.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.