electronics

MDPI

Article

Smoothly Weighted Hybrid NMPC–LQR Control for Slope-Dependent Uphill Motion of a Two-Wheeled Self-Balancing Wheelchair

Yaozhi Gu, Haomin Sun, Jiangdi Xu, Xinying Zhang, Hongyan Tang *, Qiaoling Meng and Hongliu Yu

Institute of Rehabilitative Engineering & Technology, University of Shanghai for Science and Technology, Shanghai 200093, China; 2335054710@st.usst.edu.cn (Y.G.); 2335054919@st.usst.edu.cn (H.S.); 2335054626@st.usst.edu.cn (J.X.); 2335054906@st.usst.edu.cn (X.Z.); qiaoling_meng@126.com (Q.M.); hlyu_usst@163.com (H.Y.)

* Correspondence: hytang@usst.edu.cn

Abstract

Sustained uphill motion of two-wheeled self-balancing wheelchairs is challenging. Slope-induced gravity changes the equilibrium condition, driving-torque demand, and velocity response. This paper proposes a smoothly weighted hybrid control method. The method combines nonlinear model predictive control and a linear quadratic regulator for pitch stabilization and uphill velocity tracking. The control design accounts for the slope-dependent equilibrium condition and steady-state torque demand. NMPC handles large-deviation recovery, velocity regulation, and actuator constraints. LQR improves local stabilization near the equilibrium point. The two controller outputs are coordinated by a continuously varying weight, which provides a smooth transfer of control authority across the transition region. MATLAB/Simulink simulations compare the proposed method with standalone NMPC, LQR, and SMC under several slope angles. Disturbance-recovery tests are also conducted under external torque disturbances. The results show stable uphill motion under the tested slope conditions. After finite-duration disturbances, the controller recovers both pitch posture and uphill velocity. Under a sustained torque disturbance, pitch stability is retained, but velocity regulation degrades. The proposed method improves pitch stabilization and velocity maintenance under the tested conditions. The recorded wheel-end torque remains bounded without sustained saturation in the three hybrid-controller cases. These results demonstrate numerical feasibility under the specified nominal simulation conditions, while uncertainty-robust and real-time performance remain to be validated.

Keywords: two-wheeled self-balancing wheelchair; uphill motion on slopes; wheeled inverted pendulum; nonlinear model predictive control; linear quadratic regulator; smooth hybrid control

Check for updates

Academic Editor: Bai Li

Received: 16 July 2026
Revised: 30 August 2026
Accepted: 2 September 2026
Published: 10 September 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.

1. Introduction

Wheelchairs provide essential mobility for people with lower-limb impairment. Two-wheeled self-balancing wheelchairs offer a compact footprint and high maneuverability in confined spaces [1]. Their longitudinal dynamics resemble those of an underactuated wheeled inverted pendulum [1,2]. Wheelchair users may also need to negotiate ramps in public and healthcare environments. On an incline, gravity changes the equilibrium posture and required drive torque. These effects complicate pitch stabilization and uphill velocity tracking. Reliable uphill control is therefore essential for practical operation.

Early studies established balance and longitudinal motion control for two-wheeled wheelchairs [3,4]. Later work addressed integrated design [5], nonlinear dynamic modeling [6], and nonholonomic formulations [7]. Fuzzy and optimization-based methods improved stability during lifting and under movable payloads [8,9]. These studies provided effective solutions for specific operating conditions. Most of this work, however, focused on level-ground motion, structural transformation, or predefined payload cases. Sustained uphill motion creates a different control problem. The controller must regulate pitch and velocity. It must also compensate for the gravity component along the slope.

Model-based methods provide a systematic framework. Ahmad and Tokhi applied a linear quadratic regulator (LQR) to wheelchair lifting and stabilization [10]. LQR offers efficient local feedback near a prescribed operating point. Its performance depends on the linearized model. Large pitch errors and slope-dependent equilibria can therefore limit its effective range. Pathak et al. coordinated velocity and position control through partial feedback linearization [11]. Nonlinear model predictive control (NMPC) can use nonlinear dynamics directly and impose actuator constraints explicitly [12,13]. Constrained optimal planning has also been combined with local LQ feedback for a wheeled inverted pendulum [14]. CasADi supports the numerical implementation of nonlinear optimal control [15]. Liu et al. applied NMPC to a self-balancing wheelchair. Their simulations demonstrated recovery from large pitch deviations on level ground [16]. That study did not consider sustained uphill motion. It also omitted slope-dependent equilibrium and steady-state torque compensation.

Related studies have examined wheeled inverted pendulums on slopes. Nasrallah et al. analyzed controllability and posture regulation on an inclined plane [17]. Peng et al. and Liang et al. modeled slope dynamics and designed balance controllers [18,19]. Kim and Choi combined MPC, disturbance estimation, and a sliding mechanism for slope traversal [20]. Other slope-control studies used fuzzy fractional-order PID control for a self-balancing robot on an inclined road [21] and dedicated motion control for a two-wheeled wheelchair in a slope environment [22]. Related robust-control studies have employed pitch-angle disturbance observation for a two-wheeled wheelchair [23], sliding-mode velocity control for a mobile wheeled inverted pendulum [24], and adaptive robust control for a two-wheeled inverted-pendulum system subject to underactuation and uncertainty [25]. Together, these studies confirm the importance of slope-dependent equilibrium, gravity compensation, and disturbance rejection. Their platforms differ from human-carrying wheelchairs. Some methods also require an additional sliding mechanism or dedicated slope-sensing hardware. A wheelchair driven only by its wheel motors still needs a unified solution for nonlinear recovery, local stabilization, and uphill velocity tracking. Direct switching between controllers can introduce torque discontinuities and pitch oscillations.

This paper proposes a smoothly weighted hybrid NMPC–LQR controller for sustained uphill motion of a two-wheeled self-balancing wheelchair. Unlike gain scheduling of a single controller or hard switching between control modes, the proposed method continuously blends independently generated NMPC and LQR wheel-end torque commands according to the slope-referenced pitch error. NMPC handles large-deviation recovery, velocity regulation, and constraints, whereas LQR strengthens local pitch stabilization. A nonzero NMPC contribution is retained near equilibrium to preserve uphill-velocity regulation.

The main contributions of this study are threefold. First, a reduced nonlinear longitudinal model is formulated in the slope coordinate frame, and a unified definition of the slope-dependent equilibrium pitch angle and steady-state wheel-end compensation torque is established. Second, a continuously weighted NMPC–LQR architecture is developed to combine nonlinear large-deviation recovery with local linear stabilization. The

smoothstep weighting law provides zero weight-gradient conditions at both transition boundaries, while the retained NMPC contribution preserves the uphill-velocity objective near equilibrium. Third, the proposed method is evaluated against standalone NMPC, a slope-aware LQR controller with velocity feedback, and a nonlinear SMC benchmark under matched model and actuator constraints. The evaluation further includes wheel-end torque histories, deterministic and random torque disturbances, hard-switching comparison, weighting-parameter sensitivity, and computational-time measurements.

The remainder of this paper is organized as follows. Section 2 presents the slope-dependent dynamic model. Section 3 describes the proposed controller. Section 4 reports the comparative simulations, disturbance tests, parameter-sensitivity analysis, and computational results. Section 5 concludes the paper.

2. System Model

2.1. System Simplification and Modeling Assumptions

The system considered in this study is a two-wheeled self-balancing wheelchair–occupant system. To describe its longitudinal balance dynamics during uphill motion, the system is simplified as a wheeled inverted-pendulum model in the slope coordinate frame. The wheelchair body and the occupant are treated as an equivalent rigid upper body. The left and right driving wheels are treated as two identical supporting and actuating units. The wheel axle center is selected as the reference point for model simplification. Only longitudinal motion along the slope is considered. Yaw motion, roll motion, lateral motion, and left–right asymmetry are neglected. The two driving wheels are assumed to have identical structural and inertial parameters. The wheels roll on the ground without slipping. Tire deformation and local ground unevenness are also neglected. The wheelchair body and the occupant have no relative motion.

Here, M_p denotes the combined equivalent mass of the wheelchair upper-body assembly and the occupant, which are treated as a single rigid body; M_c denotes the mass of one driving wheel; r is the wheel radius; l is the distance from the wheel-axle center to the center of mass of the equivalent upper body; γ is the slope angle.

The generalized coordinate vector is defined as $\boldsymbol{\eta} = [\ s \quad \theta \quad \alpha_r \quad \alpha_l\ ]^T$, where s is the displacement of the wheel-axle center along the slope, with the positive direction defined uphill, where θ is measured from the slope normal. Its positive direction corresponds to forward pitching toward the uphill direction. Accordingly, $\theta = 0^\circ$ indicates that the equivalent upper body is aligned with the slope normal, whereas $\theta = \gamma$ indicates that it is aligned with the world vertical. Therefore, $\theta - \gamma$ represents the upper-body pitch angle relative to the world vertical. The variables α_r and α_l denote the right- and left-wheel rotation angles, respectively.

According to these assumptions, the wheelchair–occupant system is reduced to a wheeled inverted-pendulum model on a slope, as shown in Figure 1.

2.2. Dynamic Model on the Slope

NMPC is a model-based optimal control method. Therefore, a nonlinear dynamic model is required to describe the uphill motion on the slope. Based on the simplified wheeled inverted-pendulum model, the system dynamics are derived using energy analysis and the Lagrange formulation. In the world frame, the positions of the wheel axle center and the equivalent center of mass can be expressed in terms of s, θ, γ, and l. Based on the geometric relationship in Figure 1, the position of the equivalent center of mass in the world frame is expressed in Equation (1),

$$\begin{aligned} W_{xp} &= 0 \\ W_{yp} &= (s + l\sin\theta)\cos\gamma - (r + l\cos\theta)\sin\gamma \\ W_{zp} &= (s + l\sin\theta)\sin\gamma + (r + l\cos\theta)\cos\gamma \end{aligned} \tag{1}$$

Differentiating Equation (1) with respect to time gives the corresponding linear velocity of the equivalent center of mass, as shown in Equation (2),

$$W_{vp} = [0, (\dot{s} + l\dot{\theta}cos\theta)cos\gamma + l\dot{\theta}sin\theta sin\gamma, (\dot{s} + l\dot{\theta}cos\theta)sin\gamma - l\dot{\theta}sin\theta cos\gamma] \tag{2}$$

The total kinetic energy is obtained by summing the translational and rotational kinetic energies of the equivalent upper body and the two driving wheels, and is written as $T = T_p + T_c$. Similarly, the total potential energy is written as $V = V_p + V_c$, and the Lagrangian is defined as $L = T - V$. These energy terms are written as follows:

$$\begin{aligned} T_p &= \frac{1}{2}M_p(\dot{s}^2 + 2l\dot{s}\dot{\theta}\cos\theta + l^2\dot{\theta}^2) + \frac{1}{2}I_{xx}\dot{\theta}^2, \\ T_c &= \frac{1}{2}(M_c r^2 + I_c)(\dot{\alpha}_r^2 + \dot{\alpha}_l^2), \\ V_p &= M_p g[ssin\gamma + lcos(\theta - \gamma)], \\ V_c &= 2M_c g ssin\gamma. \end{aligned}$$

Applying Lagrange's equations to the generalized coordinates yields the constrained dynamic equation in Equation (3),

$$M(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + h(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = B\boldsymbol{\tau} - \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{\lambda}, \tag{3}$$

where $M(\boldsymbol{\eta})$ is the inertia matrix, $h(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ includes the Coriolis, centrifugal, and gravity terms, B is the input matrix, and $\boldsymbol{\tau}$ is the wheel driving torque vector. A is the constraint matrix, and λ is the vector of constraint forces associated with the no-slip rolling condition.

The main matrix terms in Equation (3) are given as follows:

$$M(\eta) = \begin{bmatrix} M_p & M_p l cos\theta & 0 & 0 \\ M_p l cos\theta & M_p l^2 + I_{xx} & 0 & 0 \\ 0 & 0 & M_c r^2 + I_c & 0 \\ 0 & 0 & 0 & M_c r^2 + I_c \end{bmatrix}$$

$$h(\eta, \dot{\eta}) = \begin{bmatrix} -M_p l sin\theta\dot{\theta}^2 + M_p g sin\gamma + 2M_c g sin\gamma \\ -M_p g l sin(\theta - \gamma) \\ 0 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

According to the no-slip rolling assumption, the nonholonomic constraint is written in Equation (4),

$$A(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} = 0, \tag{4}$$

where $A(\boldsymbol{\eta})$ is the nonholonomic constraint matrix and $A(\boldsymbol{\eta})$, $S(\boldsymbol{\eta})$ and their relationship is given below:

$$A(\boldsymbol{\eta}) = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 2 & 0 & -r & -r \end{bmatrix}, S(\boldsymbol{\eta}) = \begin{bmatrix} r & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, A(\boldsymbol{\eta})S(\boldsymbol{\eta}) = 0.$$

Under the adopted longitudinal no-slip constraint, both A and S are constant matrices. Therefore, $\dot{S} = 0$. To avoid solving the constraint force explicitly, a null-space projection method is used. Let $S(\boldsymbol{\eta})$ denote the null-space basis of $A(\boldsymbol{\eta})$, such that $A(\eta)S(\eta) = 0$. The generalized velocity can then be expressed as $\dot{\eta} = S(\eta)\nu$, where $\boldsymbol{\nu}$ is the independent velocity vector under the rolling constraint. Substituting $\dot{\eta} = S(\eta)\nu$ into Equation (3) and premultiplying the result by $S^T(\eta)$ gives the reduced dynamic equation in Equation (5),

$$S^T M S\dot{\nu} + S^T M\dot{S}\nu + S^T h(\eta, \dot{\eta}) = S^T B\tau \tag{5}$$

For the longitudinal balance and uphill velocity control problem, the state vector is defined as $X = \begin{bmatrix} s & \theta & \dot{s} & \dot{\theta} \end{bmatrix}^T$. Based on Equation (5), the reduced nonlinear dynamics are written in the state-space form in Equation (6), where $f(X,u)$ is given below:

$$\dot{X} = f(X, u) \tag{6}$$

$$f(X,u) = \begin{bmatrix} \dot{s} \\ \dot{\theta} \\ \dfrac{(I_{xx} + M_p l^2)[2\tau_e + M_p l r \dot{\theta}^2 \sin\theta - (M_p + 2M_c) g r \sin\gamma] - l M_p r cos\theta [2\tau_e - g M_p l sin(\theta - \gamma)].}{2I_c I_{xx} + l^2 {M_p}^2 r^2 \sin\theta^2 + 2I_c l^2 M_p + 2I_{xx} M_c r^2 + I_{xx} M_p r^2 + 2l^2 M_p M_c r^2} \\ \dfrac{-(2I_c + 2M_c r^2 + M_p r^2)[2\tau_e - M_p l g sin(\theta - \gamma)] - l M_p r cos\theta [2\tau_e + M_p l r \dot{\theta}^2 sin\theta - (M_p + 2M_c) g r \sin\gamma]}{2I_c I_{xx} + l^2 {M_p}^2 r^2 \sin\theta^2 + 2I_c l^2 M_p + 2I_{xx} M_c r^2 + I_{xx} M_p r^2 + 2l^2 M_p M_c r^2} \end{bmatrix}$$

In the explicit dynamic expression, τ_e denotes the reduced equivalent input obtained after symmetric wheel-actuation reduction. The controller input u denotes the wheel-end torque command used by the NMPC, LQR, and blended controllers. These two inputs satisfy $\tau_e = -u/2$. Therefore, the state function $f(X,u)$ in Equation (6) is evaluated by substituting $\tau_e = -u/2$ into the displayed nonlinear dynamics. This input convention is also used in the equilibrium calculation and linearized controller design.

Under the coordinate definition introduced in Section 2.1, $\theta - \gamma$ is the pitch angle of the equivalent upper body relative to the world vertical. Therefore, the terms containing $sin(\theta - \gamma)$ describe the gravitational moment of the equivalent upper body about the wheel axle. In contrast, the terms containing $\sin\gamma$ describe the component of the total gravitational force acting along the slope. The signs of these terms are consistent with positive displacement being uphill and positive pitch being directed forward toward the uphill direction.

This model captures the nonlinear coupling among the slope-induced gravitational component, pitch motion, and rolling constraint. It is used as the prediction model of the NMPC controller.

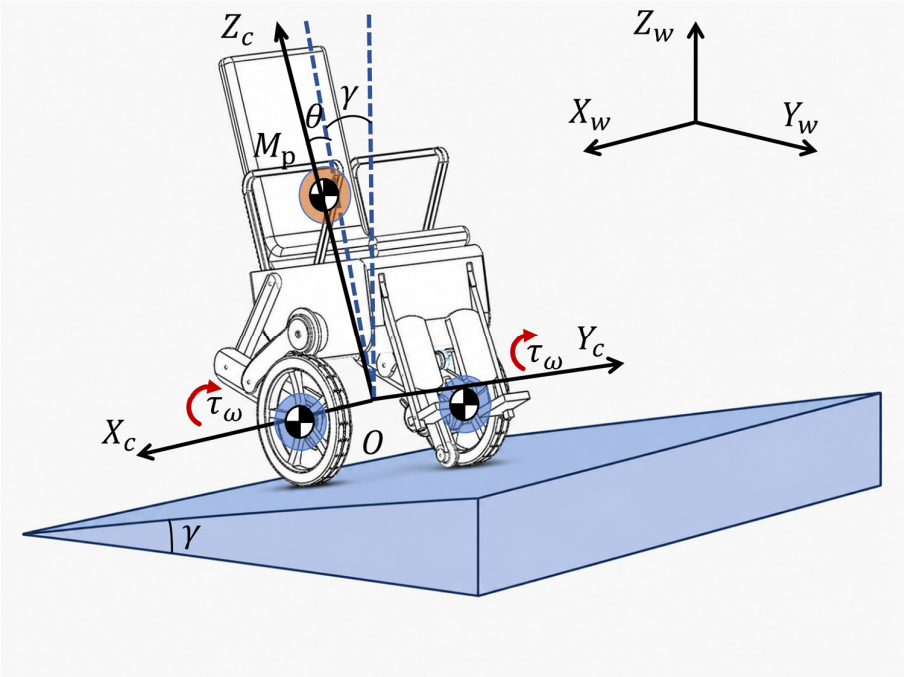


Figure 1. Dynamic model of the two-wheeled self-balancing wheelchair on an inclined plane.

2.3. Equilibrium Point and Linearized Model Around the Equilibrium Point

The equilibrium condition on a slope is different from that on flat ground. Previous work showed that terrain inclination shifts the equilibrium point and reduces the pitch-stability region [26]. Robust balancing on uncertain terrain has also used a terrain-dependent pitch reference and nonlinear-model-based disturbance compensation [27].

On a slope, the gravity component along the slope changes the static balance posture and the steady-state driving torque. Therefore, the slope equilibrium angle θ_{eq} and the steady-state compensation torque input u_{eq} are first calculated from the slope angle. At the steady uphill operating condition, the wheelchair moves at a constant along-slope velocity, while its pitch posture remains constant. Therefore, the along-slope acceleration, pitch angular velocity, and pitch angular acceleration are zero. Substituting these conditions into the nonlinear model in Equation (6) gives the following two static balance relations.

$$2\tau_{e,eq} = (M_p + 2M_c)gr\ sin\ \gamma, 2\tau_{e,eq} = M_p l\ g\ sin(\theta_{eq} - \gamma).$$

The first relation balances the component of the total gravitational force acting along the slope. The second relation balances the gravitational moment of the equivalent upper body about the wheel axle. Eliminating the equilibrium input from these two relations gives the following equilibrium condition:

$$sin(\theta_{eq} - \gamma) = \frac{(M_p + 2M_c)r\ sin\ \gamma}{M_p l}$$

Selecting the near-upright solution that is continuous with the flat-ground equilibrium gives the slope-dependent equilibrium pitch angle in Equation (7),

$$\theta_{eq} = \gamma + \arcsin\left(\frac{(M_p + 2M_c)r\sin\gamma}{M_p l}\right) \quad (7)$$

This angle defines the pitch posture required to compensate for the gravity component on the slope. The reduced equivalent input τ_e and the wheel-end torque command u satisfy

$\tau_e = -u/2$. Therefore, the reduced equilibrium input and the corresponding wheel-end compensation torque are given in Equation (8).

$$\begin{aligned} \tau_{e,eq} &= \frac{1}{2}(M_p + 2M_c)gr\sin\gamma \\ u_{eq} &= -2\tau_{e,eq} = -\left(M_p + 2M_c\right)gr\sin\gamma \end{aligned} \tag{8}$$

In the present simulation framework, the slope angle γ is prescribed from the orientation of the simulated slope plane and is supplied directly to the controller. The controller does not estimate the slope angle online. The prescribed value is used to calculate the equilibrium pitch angle, steady-state compensation torque, and slope-dependent LQR gain. In all reported simulations, the slope value supplied to the controller is equal to the actual slope angle used in the plant model. Therefore, the reported results do not evaluate an estimator-in-the-loop configuration.

Although closed-loop simulations with slope-estimation errors were not conducted, the static sensitivity of the slope-dependent reference can be evaluated from Equations (7) and (8). Let

$$C = \frac{\left(M_p + 2M_c\right)r}{M_p l}$$

When γ is expressed in radians, differentiation of the equilibrium angle and compensation torque gives

$$\frac{\partial\theta_{eq}}{\partial\gamma} = 1 + \frac{Ccos\gamma}{\sqrt{1 - C^2\sin\gamma^2}}, \frac{\partial u_{eq}}{\partial\gamma} = -(M_p + 2M_c)grcos\gamma.$$

Using the nominal physical parameters, a 1° slope-angle error over the tested 4°–8° range produces an equilibrium-reference error of approximately 1.541°–1.544° and a wheel-end feedforward-torque error of approximately 1.650–1.662 N·m. These values quantify only the static propagation of the slope-angle error into the controller reference and feedforward term. They do not provide a closed-loop robustness or stability bound.

This feedforward compensation offsets the steady-state input bias induced by the gravitational component along the slope. It allows the feedback controller to regulate the state deviations around the equilibrium point. The linearization point is selected as $X_{eq} = \begin{bmatrix} s_{eq} & \theta_{eq} & v_{eq} & 0 \end{bmatrix}^T$, where v_{eq} is the desired along-slope velocity. In this paper, $v_{eq} = v_{ref} = 0.5\ \mathrm{m{\cdot}s^{-1}}$. The position s_{eq} is used only to define the local operating point instead of being used as a fixed position target. Linearizing the nonlinear model in Equation (6) about $\left(X_{eq}, u_{eq}\right)$ gives the local linear model in Equation (9),

$$\delta\dot{X} = A\delta X + B\delta u, \tag{9}$$

where $\delta X = X - X_{eq}$ and $\delta u = u - u_{eq}$.

The corresponding state matrix A and input matrix B are given as follows,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{M_p^2 l^2 gr\cos\theta_{eq}\cos(\theta_{eq}-\gamma)}{D_i} & 0 & 0 \\ 0 & \frac{(2I_c+2M_c r^2+M_p r^2)M_p lg\cos(\theta_{eq}-\gamma)}{D_i} & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \frac{-(I_{xx}+M_p l^2)+lM_p r\cos\theta_{eq}}{D_i} \\ \frac{2I_c+2M_c r^2+M_p r^2+lM_p r\cos\theta_{eq}}{D_i} \end{bmatrix},$$

where

$$D_i = 2I_c I_{xx} + l^2 M_p^2 r^2 \sin^2\theta_{eq} + 2I_c l^2 M_p + 2I_{xx} M_c r^2 + I_{xx} M_p r^2 + 2l^2 M_p M_c r^2$$

Thus, a local linear model around the slope equilibrium point is obtained. This model provides the basis for the LQR controller design.

3. Control Method

3.1. Overall Structure of the Control Method

This paper proposes an NMPC–LQR smooth hybrid control method based on the pitch-angle error. The controller consists of an NMPC controller, an LQR controller, and a weight assignment module. NMPC is used for posture recovery and velocity regulation in the large-deviation region. LQR is used for local pitch stabilization near the equilibrium point. A continuous smoothstep weight is used to blend the two controller outputs.

When the system is far from the slope equilibrium point, the control input is mainly generated by NMPC. This enables the pitch angle to approach the slope-dependent equilibrium angle rapidly. It also reduces the deviation between the along-slope velocity and the target velocity.

When the system enters the smooth transition region, the NMPC weight decreases gradually and the LQR weight increases gradually. Thus, the controller weights change continuously, reducing abrupt changes in control authority.

When the system enters the equilibrium neighborhood, a fixed NMPC–LQR hybrid weight is used. In this region, LQR mainly suppresses the pitch-angle error and pitch angular velocity. NMPC retains a certain weight to maintain the along-slope velocity and the uphill motion.

This method combines large-deviation recovery, local stabilization near the equilibrium point, and a smooth transfer of control authority.

3.2. NMPC Controller Design

NMPC uses a receding-horizon formulation. It solves a finite-horizon constrained optimization problem at each sampling instant [28]. The nonlinear state equation derived in Equation (6) is used as the prediction model of the NMPC controller. At each sampling instant, the current state is used as the initial condition. The controller solves the optimal control sequence over the prediction horizon and applies the first control input to the system. This receding-horizon process enables NMPC to recover the pitch angle, regulate the along-slope velocity, and handle the input constraint in the large-deviation region. The total NMPC cost is decomposed as $J_N = J_1 + J_2$, where J_1 is the tracking-error term and J_2 is the predicted-pitch penalty. The tracking-error term is defined in Equation (10),

$$J_1 = \sum_{k=0}^{N-1} \left[e_{X,k}^T Q_N e_{X,k} + R_N \left(u_k - u_{eq} \right)^2 \right] \tag{10}$$

In Equation (10), the state-error vector is defined as $e_{X,k} = \begin{bmatrix} 0 \\ \theta_k - \theta_{eq} \\ \dot{s}_k - v_{eq} \\ \dot{\theta}_k \end{bmatrix}$, where $\theta_k - \theta_{eq}$ is the pitch-angle error, $\dot{s}_k - v_{eq}$ is the along-slope velocity error, and $\dot{\theta}_k$ is the pitch angular velocity.

The displacement-error term is set to zero so that NMPC does not impose a fixed along-slope position target. Instead, the controller mainly regulates the pitch posture and along-slope velocity. The input error is defined relative to the equilibrium input, $u_k - u_{eq}$, rather than relative to zero. This is necessary because uphill motion on an inclined surface requires a nonzero steady-state compensation input. This term limits unnecessary deviation of the control input from the steady-state compensation input.

The NMPC weights are selected as $Q_N = \mathrm{diag}(0, 18000, 5000, 5000)$, $R_N = 0.01$. The four entries of Q_N correspond to the along-slope displacement, pitch angle, along-slope velocity, and pitch angular velocity, respectively. A large pitch-angle weight promotes rapid posture recovery, while the velocity weight maintains the desired uphill speed. The pitch-angular-velocity weight suppresses excessive angular motion during recovery. The small input weight limits excessive control effort relative to u_{eq}. To reduce overshoot during large-deviation recovery, a predicted pitch error is defined in Equation (11),

$$e_{f,k} = \theta_k + T_b\dot{\theta}_k - \theta_{eq}, \tag{11}$$

where T_b is the prediction time coefficient. This term estimates the short-term pitch deviation using the current pitch angle and pitch angular velocity. When $\dot{\theta}_k$ is large, the system may still overshoot the target posture even if the current pitch angle is close to the equilibrium angle. The predicted pitch penalty reduces this overshoot tendency before it appears in the measured pitch error. The predicted pitch penalty is then written as

$$J_2 = \sum_{k=0}^{N-1} W_b e_{f,k}^2, \tag{12}$$

where W_b is the weight of the predicted pitch error. The prediction-time coefficient and predicted-pitch weight were set to $T_b = 0.18$ s and $W_b = 6000$, respectively. These values were kept unchanged in the standalone NMPC and hybrid-controller simulations.

Combining Equations (10)–(12), the total NMPC cost function is obtained as Equation (13):

$$J_N = \sum_{k=0}^{N-1} \left[e_{X,k}^T Q_N e_{X,k} + R_N \left(u_k - u_{eq}\right)^2 + W_b \left(\theta_k + T_b\dot{\theta}_k - \theta_{eq}\right)^2 \right] \tag{13}$$

In the numerical implementation, the NMPC transcription uses 20 state and control nodes. Adjacent nodes are separated by 0.01 s, resulting in 19 integration intervals and an effective prediction duration of 0.19 s. A separate shorter control horizon is not used; the control horizon spans the same set of intervals as the prediction horizon. The state transition and stage cost are discretized using the Hermite-Simpson method.

The nonlinear programming problem was formulated using the CasADi Opti interface and solved using IPOPT. Symbolic expression expansion was enabled before the optimization problem was passed to the solver. At each controller evaluation, the initial guess for all 20 control nodes was set to the slope-dependent equilibrium input u_{eq}. The initial guess for all 20 state nodes was set to $\left[s_{\mathrm{current}}, \theta_{\mathrm{eq}}, v_{\mathrm{eq}}, 0\right]^T$, where s_{current} denotes the current along-slope displacement. The first state node was additionally constrained to the current system state. No custom IPOPT convergence tolerances were specified in the simulations. Therefore, the default IPOPT 3.14.11 values were used: *tol* $= 1 \times 10^{-8}$, *dual_inf_tol* $= 1$, *constr_viol_tol* $= 1 \times 10^{-4}$, *compl_inf_tol* $= 1 \times 10^{-4}$, *acceptable_tol* $= 1 \times 10^{-6}$, *acceptable_iter* $= 15$, *acceptable_dual_inf_tol* $= 1 \times 10^{10}$, *acceptable_dual_inf_toll* $= 1 \times 10^{-2}$, *acceptable_compl_inf_tol* $= 1 \times 10^{-2}$, and *acceptable_obj_change_tol* $= 1 \times 10^{20}$. Solver screen output and per-solve timing output were disabled during the reported simulations. The optimization problem was reconstructed at each controller evaluation. Solver reuse and warm starting were not applied. If IPOPT failed to return a solution, the controller implementation terminated the simulation rather than applying a fallback control input.

3.3. LQR Controller Design Based on the Slope Equilibrium Point

The LQR component within the proposed hybrid architecture is designed specifically for local pitch stabilization. The two-wheeled wheelchair is underactuated, and pitch

stabilization and along-slope velocity regulation share the same wheel-torque input. During controller tuning, assigning additional velocity-regulation authority to the local LQR component weakened its pitch-stabilization effect near the equilibrium point.

For this reason, the LQR component used in the proposed hybrid controller feeds back only the pitch-angle error and pitch angular velocity. Along-slope velocity regulation is retained in the NMPC component. This division of control roles allows LQR to focus on local pitch stabilization, while NMPC maintains the uphill velocity objective and handles the actuator constraint.

The LQR feedback state is selected as $X_L = \begin{bmatrix} \theta - \theta_{eq} \\ \dot{\theta} \end{bmatrix}$. The LQR gain is obtained by minimizing the quadratic performance index in Equation (14),

$$J_{\mathrm{LQR}} = \int_0^{\infty} (X_L^T Q_L X_L + \delta u^T R_L \delta u) dt, \tag{14}$$

where Q_L is the local state weighting matrix and R_L is the input weighting matrix.

The weighting matrices are selected as $Q_L = \mathrm{diag}(1600, 450)$, $R_L = 0.8$. This weighting choice makes LQR focus on the pitch-angle error and the pitch angular velocity near the equilibrium point, while limiting the input magnitude. By solving the continuous-time algebraic Riccati equation, the LQR feedback gain is obtained. The feedback gain is extended to the full state vector $K_{\mathrm{LQR}} = [\ 0 \quad k_\theta \quad 0 \quad k_{\dot{\theta}}\ ]$. With the feedforward equilibrium input included, the equivalent input generated by the LQR controller is given in Equation (15),

$$u_{\mathrm{LQR}} = u_{eq} - K_{\mathrm{LQR}}(X - X_{eq}) = u_{eq} - k_\theta(\theta - \theta_{eq}) - k_{\dot{\theta}}\dot{\theta}, \tag{15}$$

where u_{eq} is the feedforward compensation input at the slope equilibrium point.

The numerical LQR gains were recalculated at each slope-dependent equilibrium point. For the 4°, 6°, and 8° slopes, the full-state gain vectors were [0, 140.7827, 0, 37.7767], [0, 141.2156, 0, 37.9650], and [0, 141.8264, 0, 38.2294], respectively. The zero entries indicate that along-slope displacement and velocity were not included in the local LQR feedback.

The closed-loop poles of the local pitch subsystem were −2.6015 and −9.7012 for the 4° slope, −2.6007 and −9.6000 for the 6° slope, and −2.5996 and −9.4627 for the 8° slope. All poles have negative real parts. Therefore, the LQR component provides local asymptotic stabilization of the linearized pitch subsystem around each slope-dependent equilibrium point.

3.4. NMPC–LQR Smooth Weighting Strategy

Hard switching in an NMPC–LQR controller can cause torque discontinuities and repeated switching near the threshold. Scheduled MPC studies have used bumpless transfer to mitigate similar transients [29]. Unlike gain scheduling of a single controller or discrete handover between controllers, the proposed method continuously blends independently generated NMPC and LQR torque commands according to the pitch-angle error. This preserves NMPC-based velocity regulation and constraint handling while increasing the LQR contribution near the slope-dependent equilibrium.

This strategy blends the outputs of NMPC and LQR continuously in the transition region. The pitch-angle error is defined as $e_\theta = \theta - \theta_{eq}$, and its magnitude $E_\theta = | e_\theta |$ is used for weight assignment, where θ is the current pitch angle and θ_{eq} is the slope-dependent equilibrium angle. Let ε_θ be the threshold of the equilibrium neighborhood and Δ_θ be the width of the smooth transition region. Based on the magnitude of E_θ, the system response is divided into the large-deviation recovery region, the smooth transition region, and the equilibrium neighborhood.

The nominal equilibrium-neighborhood threshold and transition width were set to 10° and 2°, respectively. These values determine when the controller begins to reduce the NMPC contribution and how rapidly the control authority is transferred toward LQR. Their effects on closed-loop stability, convergence speed, and transition smoothness are evaluated through the parameter-sensitivity analysis in Section 4.6.

When $E_\theta \geq \varepsilon_\theta + \Delta_\theta$, the system is in the large-deviation recovery region, and NMPC is assigned the full weight, i.e., $w_N = 1$. When $E_\theta \leq \varepsilon_\theta$, the system enters the equilibrium neighborhood. Since the control objective is continuous uphill motion rather than fixed-point balancing, pure LQR control is not adopted in this region. Instead, a minimum NMPC weight w_{min} is retained. In the equilibrium neighborhood, LQR mainly suppresses the pitch-angle error and pitch angular velocity, whereas the retained NMPC component contributes to along-slope velocity regulation and constraint handling. The minimum NMPC weight is treated as an empirical tuning parameter rather than an analytically derived stability bound. A one-factor-at-a-time sensitivity test was conducted under the nominal 8° slope condition using minimum NMPC weights of 0.3, 0.4, and 0.5. All other model parameters, controller settings, switching parameters, initial physical conditions, input constraints, and simulation durations were kept unchanged. The results in Section 4.6 show that the minimum weight of 0.4 provides the most suitable overall compromise between pitch convergence, steady-state pitch regulation, and uphill-velocity tracking. It was therefore selected as the nominal value.

Therefore, in the equilibrium neighborhood, the controller weights are $w_N = 0.4$ and $w_L = 0.6$.

In the transition region, $\varepsilon_\theta < E_\theta < \varepsilon_\theta + \Delta_\theta$, the normalized variable and the smoothstep-based weights are defined in Equation (16):

$$\begin{gathered} \xi = \frac{E_\theta - \varepsilon_\theta}{\Delta_\theta}, \sigma(\xi) = 3\xi^2 - 2\xi^3, \\ w_N = w_{\min} + (1 - w_{\min})\sigma(\xi), w_L = 1 - w_N \end{gathered} \tag{16}$$

Since the first derivative of $\sigma(\xi)$ is zero at both endpoints, the NMPC weight changes smoothly at the boundaries of the transition region. According to the assigned weights, the blended equivalent input is given by Equation (17),

$$u = w_N u_{\text{NMPC}} + w_L u_{\text{LQR}} \tag{17}$$

To prevent the actuator input from exceeding the actual limit, the actual wheel-end torque is constrained by Equation (18):

$$| u | \leq 60 \text{ N} \cdot \text{m} \tag{18}$$

Here, u is the wheel-end torque command applied by the longitudinal controller.

The smoothstep weighting law provides a continuous transfer of control authority and has zero slope at both boundaries of the transition region. It therefore avoids the instantaneous change in controller weights caused by hard switching. However, continuous weights do not by themselves guarantee continuous actuator torque, which also depends on the individual controller outputs, the NMPC solution, and input saturation. The resulting transition behavior is evaluated numerically through the hard-switching comparison in Section 4.6.

The reported LQR poles establish local asymptotic stability only for the linearized pitch subsystem around each slope-dependent equilibrium. Smooth blending does not automatically guarantee stability of the complete nonlinear controller. In addition, the present NMPC formulation does not include an explicit terminal cost, terminal constraint,

or invariant terminal set, and no common Lyapunov function has been established for the blended closed loop. Therefore, global asymptotic stability and recursive feasibility are not formally claimed. The present conclusions are limited to the convergence and bounded responses observed under the tested simulation conditions.

Because a nonzero NMPC weight is retained in the equilibrium neighborhood, the current implementation continues to evaluate the NMPC optimization problem throughout the simulation. The LQR component changes the allocation of control authority and improves local pitch stabilization, but it does not eliminate the nonlinear optimization step. Therefore, the present hybrid structure should not be interpreted as a method for reducing the computational cost of NMPC. Its additional computational overhead and the overall execution time are evaluated separately in Section 4.7.

4. Simulation Results and Analysis

4.1. Simulation Platform and Parameter Setting

The simulation platform was built in MATLAB/Simulink. The model includes the wheelchair body, driving wheels, slope surface, equivalent mass and inertia parameters, external disturbance input, and state-output modules. The wheelchair body is modeled as a rigid body that can rotate about the wheel axle. The driving torque is applied through the wheel actuator modules. The slope is modeled as a ground plane rotated in the world coordinate frame and an appropriate friction coefficient is set to the wheel–ground contact. The model outputs include the pitch angle θ, pitch angular velocity $\dot{\theta}$, displacement along the slope s, and velocity along the slope $\dot{s}$. The disturbance channel is used in the robustness tests. The disturbance torque is introduced additively at the longitudinal wheel-axle drive-torque input and therefore acts about the wheel axle. It is not applied directly to the pitch coordinate of the equivalent upper body. The simulation model and physical parameters are shown in Figure 2.

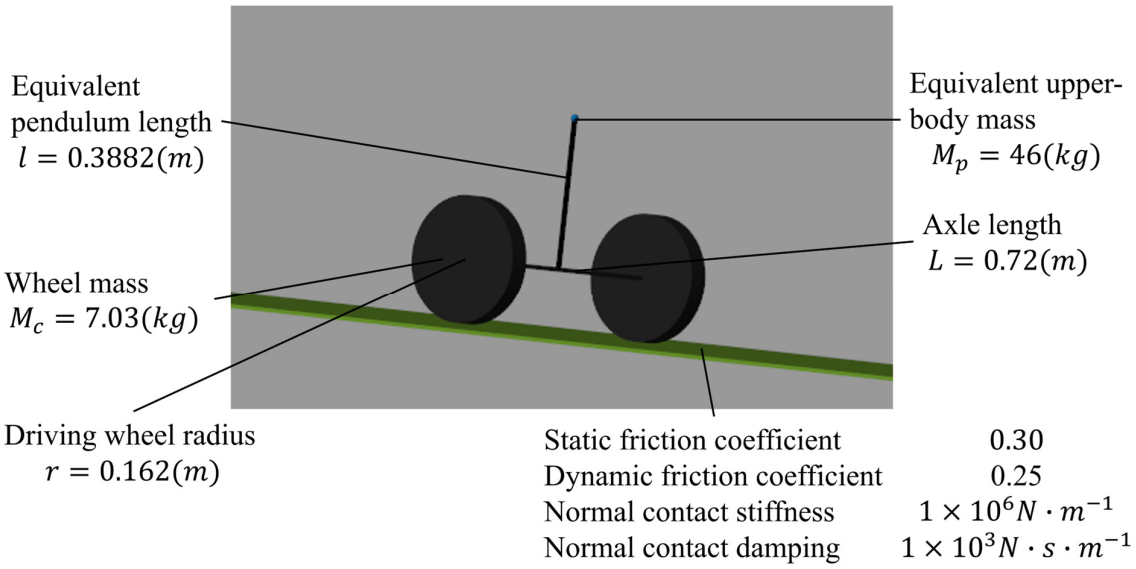


Figure 2. Simulation model and main physical parameters of the wheelchair system.

The nominal parameters of the reduced longitudinal model were $M_p = 46$ kg, $Mc = 7.03$ kg for each wheel, $l = 0.3882$ m, $r = 0.162$ m, $g = 9.81$ m·s^{-2}, $I_{xx} = 6.64$ kg·m^2, and $I_c = 0.0262$ kg·m^2. Here, M_p represents the combined equivalent mass of the wheelchair upper body and occupant, l is the distance from the wheel-axle center to the equivalent center of mass, I_{xx} is the equivalent pitch moment of inertia of the upper body about its center of mass, and I_c is the rotational inertia of one wheel used in the

reduced model. The axle length was $L = 0.72$ m, and the wheel-axis damping coefficient was 1×10^{-4} N·m·s·rad^{-1}. The static and dynamic friction coefficients were 0.30 and 0.25, respectively. The normal contact stiffness and damping coefficients were 1×10^{6} N·m^{-1} and 1×10^{3} N·s·m^{-1}, respectively.

The equivalent upper-body mass $M_p = 46$ kg was calculated by combining the mass of the prototype upper-body assembly with the occupant mass represented in the model. The values of l, I_{xx} and I_c were obtained from the mass-property calculation of a SolidWorks 2024 CAD model constructed using the geometry and component mass properties of the physical prototype. The same wheelchair–occupant equivalent-body definition used for M_p was used to determine l and I_{xx}. The resulting combined center of mass gave $l = 0.3882$ m. The value $I_{xx} = 6.64$ kg·m^2 is the pitch moment of inertia of the equivalent upper body about the lateral axis through its center of mass. The value $I_c = 0.0262$ kg·m^2 is the rotational inertia of one driving wheel about its spin axis. These parameters were derived from the prototype-based CAD model rather than selected through controller tuning.

During the offline Simulink simulations, the controller block was evaluated at a simulation update period of 0.001 s. This simulation block-evaluation period should be distinguished from the 0.01 s prediction-grid interval used inside the NMPC formulation. The reported simulations were conducted offline and do not demonstrate that the optimization can be completed within either update period on an embedded processor.

All reported closed-loop simulations used the nominal physical parameters listed above. The plant states were supplied directly to the controller without measurement noise, quantization, or state-estimation error. No sensing delay, communication delay, or actuator delay was included. The prescribed slope value supplied to the controller was identical to the slope angle used in the plant model. Consequently, the simulations represent nominal full-state-feedback conditions rather than a sensor- and estimator-in-the-loop implementation.

The signal flow of the control system is shown in Figure 3.

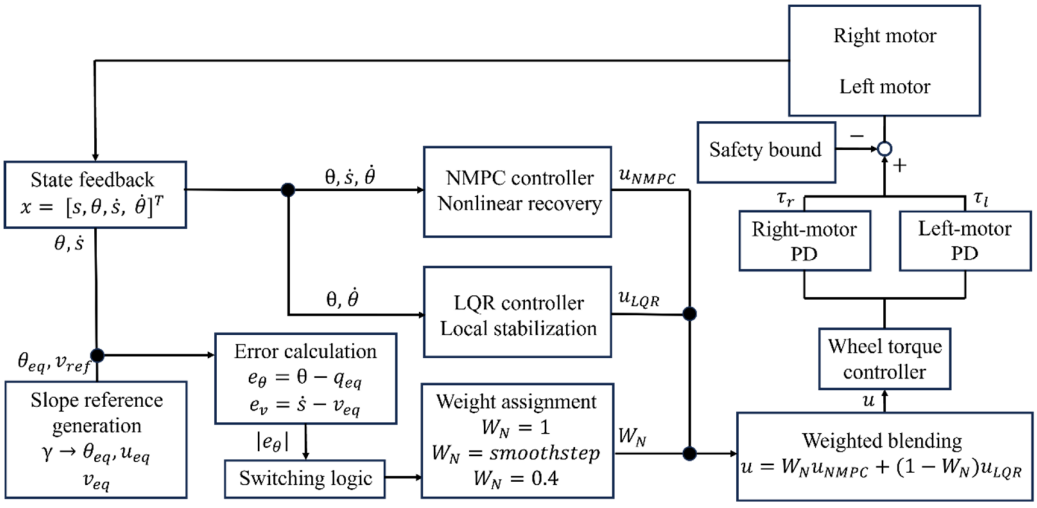


Figure 3. Signal flow of the proposed NMPC–LQR smooth hybrid control system.

4.2. Comparison with Standalone NMPC and LQR Controllers

To evaluate the proposed NMPC–LQR smooth hybrid control method, simulations were conducted on 4°, 6°, and 8° slopes. Standalone LQR, standalone NMPC, and the proposed hybrid controller were tested using the same simulation model, physical parameters, and input constraint. For the standalone comparison, the LQR controller was configured separately from the local LQR component used in the proposed hybrid controller. The stan-

dalone LQR included the pitch-angle error, along-slope velocity error, and pitch angular velocity as feedback variables. It was therefore designed as a full-task slope-aware baseline rather than as a pitch-only controller.

The standalone LQR weighting matrices were $Q_L = \mathrm{diag}(1600,\ 120,\ 450)$, $R_L = 0.8$. The three state penalties correspond to pitch-angle error, along-slope velocity error, and pitch angular velocity, respectively. These settings explicitly include the velocity-tracking objective. For the 4°, 6°, and 8° slope conditions, the resulting full-state gain vectors were [0, 520.0092, 12.2474, 96.9165], [0, 519.6269, 12.2474, 97.4890], and [0, 519.1650, 12.2474, 98.2938] , respectively. These gains correspond to the state order $[s,\ \theta,\ \dot{s},\ \dot{\theta}]$ and to the wheel-end torque command used in the simulations.

For all three slope conditions, the initial pitch angle was set to $\theta(0) = 0°$, which means that the equivalent upper body was initially aligned with the slope normal. According to Equation (7), the corresponding equilibrium pitch angles were 6.1782°, 9.2650°, and 12.3489°, respectively. Therefore, the initial pitch-angle errors were −6.1782°, −9.2650°, and −12.3489°. The desired along-slope velocity was 0.5 m·s^{-1}, and the slope-dependent equilibrium pitch angle was used as the pitch reference. The pitch responses are shown in Figure 4.

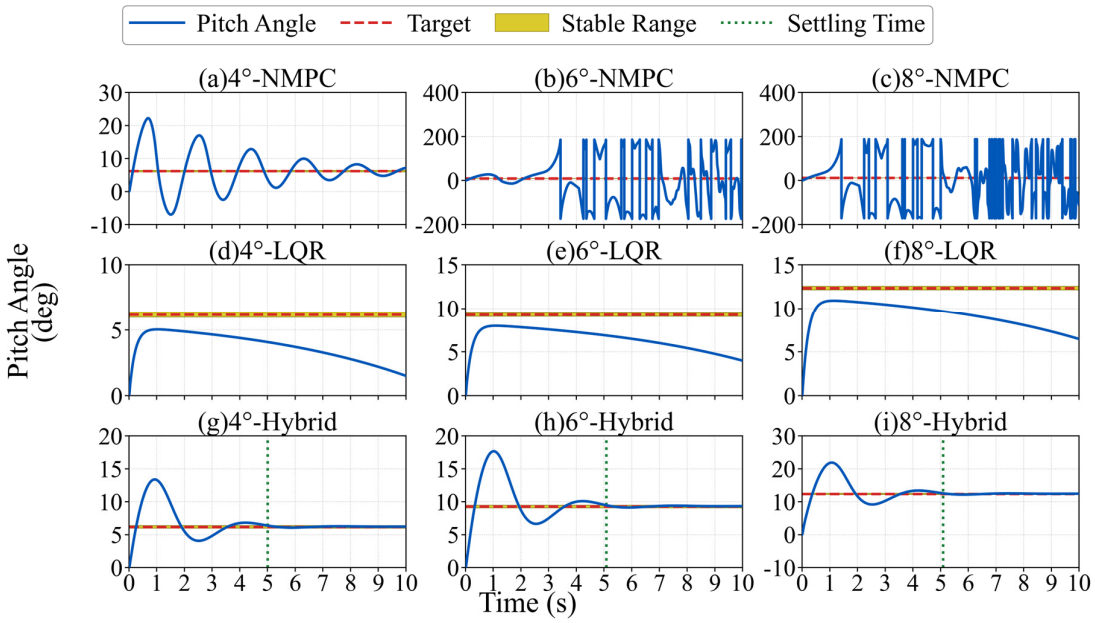


Figure 4. Pitch-angle responses of the three controllers under different slope conditions. Standalone NMPC reached the operational fall boundary of $|\theta| = 90°$ at 3.232 s on the 6° slope and at 1.223 s on the 8° slope. The trajectories after these times are retained only to illustrate numerical divergence and do not represent physically meaningful post-fall motion.

The velocity responses are shown in Figure 5.

The actual wheel-end torque histories are shown in Figure 6. Since the longitudinal left- and right-wheel torque commands are identical, the command of one wheel is presented. Under the proposed hybrid controller, the torque remained bounded and gradually approached the slope-dependent steady-state compensation values. The approximate steady-state values were −6.66, −9.98, and −13.28 N·m on the 4°, 6°, and 8° slopes, respectively. After the initialization transient, no sustained saturation or high-frequency torque switching was observed in the three hybrid-controller cases.

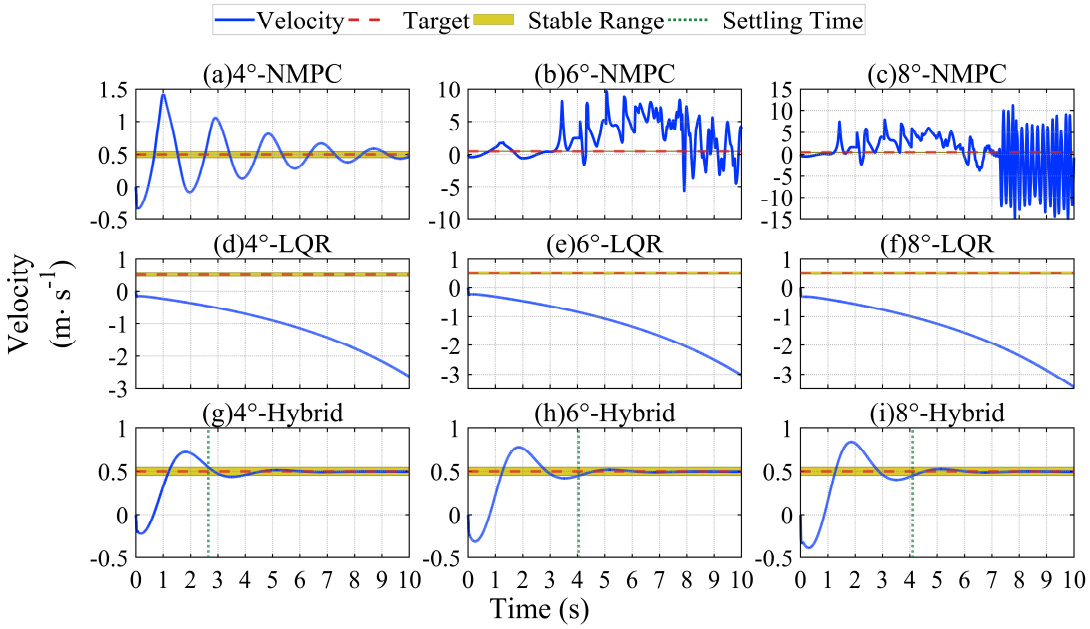


Figure 5. Along-slope velocity responses of the three controllers under different slope conditions. For standalone NMPC, the responses after 3.232 s on the 6° slope and 1.223 s on the 8° slope correspond to post-fall numerical continuation of the contact-free model.

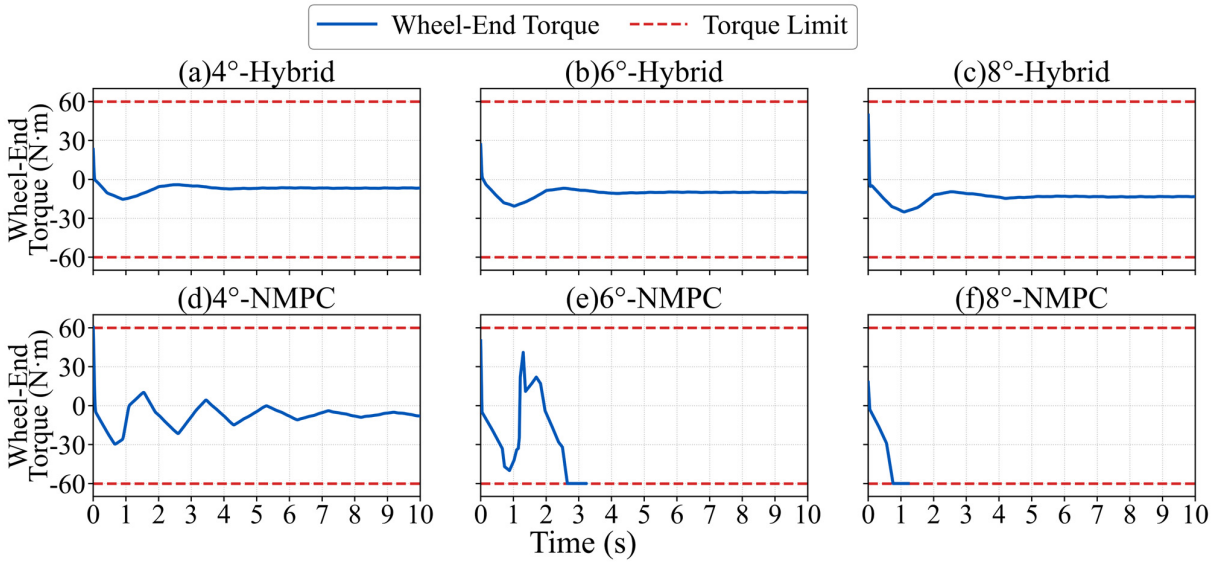


Figure 6. Wheel-end torque histories of the proposed hybrid controller and standalone NMPC under different slope conditions. The dashed horizontal lines indicate the ±60 N·m torque limits. The standalone NMPC trajectories on the 6° and 8° slopes are truncated at the corresponding operational fall times.

The standalone NMPC torque remained oscillatory on the 4° slope. In the 6° and 8° cases, the torque reached the −60 N·m limit before the operational fall boundary was crossed. This indicates that the available actuator authority was insufficient to arrest the rapidly increasing pitch motion under the implemented NMPC settings. Torque values after the fall boundary were excluded because the reduced model does not describe body–ground contact or post-fall motion.

These torque histories provide numerical evidence of bounded actuator behavior under the tested hybrid-controller conditions. However, they do not constitute a general

proof of actuator-command continuity. The final torque also depends on the individual NMPC and LQR outputs, the optimization solution, and actuator saturation.

The main performance indices are listed in Table 1.

Table 1. Performance comparison of the LQR, NMPC, and hybrid controllers under different slope angles.

Slope Angle (°)	Controller	Final Pitch Error (°)	Pitch Std (°)	Final Displacement (m)	Mean Velocity (m · s^{-1})
4	Hybrid	0.042	0.0027	4.60	0.498
4	LQR	−4.672	0.198	−10.55	−2.385
4	NMPC	0.998	0.837	4.82	0.460
6	Hybrid	0.064	0.0044	4.51	0.497
6	LQR	−5.269	0.224	−12.49	−2.758
6	NMPC	-	-	-	-
8	Hybrid	0.089	0.0078	4.46	0.497
8	LQR	−5.885	0.251	−14.52	−3.147
8	NMPC	-	-	-	-

The 6° and 8° standalone NMPC cases reached the operational fall boundary at 3.232 s and 1.223 s, respectively. Steady-state indices were therefore not calculated.

On the 4° slope, the hybrid controller made the pitch angle converge to the equilibrium angle. The final pitch error and the steady-state fluctuation were small. The displacement increased in the uphill direction. The mean velocity after stabilization was close to 0.5 m·s^{-1}. The standalone NMPC controller also achieved uphill motion, but its steady-state pitch oscillation and final pitch error were larger. Under standalone LQR control, the displacement changed in the negative direction, and the target uphill motion was not achieved.

On the 6° slope, the hybrid controller remained stable throughout the simulation. The standalone LQR controller maintained a bounded pitch response but produced downhill motion. Standalone NMPC showed increasing pitch and velocity oscillations and reached the operational fall boundary at 3.232 s. It was therefore classified as unrecoverable from this time onward.

On the 8° slope, the hybrid controller remained stable throughout the simulation and converged near the slope-dependent equilibrium angle. The standalone LQR controller showed a stronger downhill trend. Standalone NMPC exhibited more rapid divergence and reached the operational fall boundary at 1.223 s. It was therefore classified as unrecoverable from this time onward.

To distinguish physically meaningful operation from numerical divergence, the standalone NMPC response was classified as unrecoverable when the absolute pitch angle first reached the operational fall boundary of $|\theta| = 90°$, where it is measured from the slope normal. At this boundary, the equivalent upper-body axis is parallel to the slope and the upright posture has been completely lost. The 4° standalone NMPC case remained within this boundary throughout the 10 s simulation. In contrast, the 6° and 8° cases first reached the boundary at 3.232 s and 1.223 s, respectively. The added torque histories show that the wheel-end torque had already reached the −60 N·m limit before these boundary crossings. The numerical simulations were allowed to continue to 10 s only to illustrate the subsequent divergence of the contact-free reduced model. Responses after the boundary-crossing times do not represent physically meaningful wheelchair motion and are not interpreted as recoverable control behavior.

Although the standalone LQR explicitly included the along-slope velocity error, it did not achieve satisfactory uphill velocity tracking. The controller continued to prioritize pitch stabilization because the pitch-angle and pitch-rate gains were substantially larger than the velocity gain. In addition, pitch and velocity regulation shared the same wheel-torque

input. The available control authority was therefore primarily used to maintain the local pitch response and could not fully compensate for the slope-induced velocity deviation.

Increasing the velocity penalty during preliminary tuning strengthened the velocity response but substantially degraded pitch regulation. The reported weighting therefore represents a compromise that preserves closed-loop pitch stability while retaining an explicit velocity-tracking objective. Nevertheless, the standalone LQR produced negative displacement and poor velocity tracking under the tested slope conditions.

The standalone NMPC baseline used the same nonlinear prediction model, state and input weights, prediction horizon, discretization method, equilibrium reference, initial guess, solver settings, and torque constraint as the NMPC component of the proposed hybrid controller. The only difference was the final control allocation. In the standalone case, the NMPC output was applied directly, corresponding to an NMPC weight of one throughout the simulation. In the hybrid case, the same NMPC output was blended with the local LQR output according to the pitch-error-dependent weight.

Across all three standalone NMPC simulations, IPOPT returned a successful status at every controller evaluation; no failed solve, infeasible status, or maximum-iteration termination occurred before the 6° and 8° cases reached the fall boundary. The divergence therefore reflects the closed-loop behavior of the implemented standalone NMPC under the common model, weights, prediction horizon, solver settings, and torque constraint, rather than optimization failure. Under these settings, the hybrid controller achieved a more favorable allocation between NMPC-based recovery and velocity regulation and LQR-based local pitch stabilization. This result is specific to the tested standalone configuration and should not be interpreted as a general limitation of NMPC.

4.3. Benchmark Comparison with Sliding-Mode Control

Sliding-mode control (SMC) was selected as a representative nonlinear robust-control benchmark because it has been widely used for balancing and velocity regulation of wheeled inverted-pendulum systems [24]. The SMC benchmark and the proposed hybrid controller were tested using the same nonlinear model, physical parameters, slope-dependent references, initial conditions, target velocity, and wheel-end torque constraint. The standalone LQR provides a full-task linear baseline, whereas SMC provides a nonlinear robust-control baseline. Both comparison controllers include pitch stabilization and along-slope velocity regulation. This enables the proposed hybrid method to be evaluated against both linear optimal control and nonlinear robust control under the same nominal model and torque constraint.

The implemented benchmark adopted a cascaded SMC structure consisting of an outer velocity loop and an inner pitch sliding-mode loop. The along-slope velocity error was defined as $e_v = \dot{s} - v_{\mathrm{eq}}$. The outer velocity loop converted this error into a bounded temporary pitch reference, as given in Equation (19),

$$\theta_r = \theta_{\mathrm{eq}} + \Delta\theta_{\max}\,\mathrm{sat}\left(\frac{e_v}{\phi_v}\right), \tag{19}$$

where θ_r is the temporary pitch reference, $\Delta\theta_{\max}$ is the maximum pitch-reference adjustment, and ϕ_v is the velocity-error boundary-layer width. The saturation function used in both the velocity outer loop and the sliding-mode reaching term is defined in Equation (20),

$$\mathrm{sat}(z) = \begin{cases} -1, & z < -1, \\ z, & |z| \le 1, \\ 1, & z > 1. \end{cases} \tag{20}$$

The inner loop used the wrapped pitch-angle error to avoid angular discontinuity. The pitch error and the corresponding sliding variable are defined in Equation (21),

$$e_\theta = \text{atan2}[\sin(\theta - \theta_r), \cos(\theta - \theta_r)], s_\theta = \dot{\theta} + \lambda e_\theta. \tag{21}$$

For the SMC implementation, the pitch dynamics in the nonlinear model were expressed in input-affine form. The drift and input coefficients were evaluated directly from the fourth component $f_\theta(X, u)$ of the nonlinear state function in Equation (6),

$$\begin{gathered} \ddot{\theta} = a_\theta(X) + b_\theta(X)u, \\ a_\theta(X) = f_\theta(X, 0), \\ b_\theta(X) = f_\theta(X, 1) - f_\theta(X, 0), \end{gathered}$$

Based on these quantities, the SMC wheel-end torque command was calculated using nonlinear dynamics compensation, a linear reaching term, and a boundary-layer switching term, as given in Equation (22),

$$u_{\text{SMC}} = \frac{-a_\theta(X) - \lambda\dot{\theta} - k s_\theta - \eta\,\text{sat}(s_\theta/\phi)}{b_\theta(X)}. \tag{22}$$

Here, λ is the pitch-error coefficient, k is the linear reaching gain, η is the switching gain, and ϕ is the boundary-layer width of the sliding variable. To avoid numerical division by a near-zero input coefficient, the control command was set to zero when $| b_\theta(X) | < 10^{-6}$. The resulting SMC command was subject to the same wheel-end torque constraint defined in Equation (18). The parameters were fixed at $\lambda = 8\ \text{s}^{-1}$, $k = 6\ \text{s}^{-1}$, $\eta = 3\ \text{rad} \cdot \text{s}^{-2}$, $\phi = 0.05\ \text{rad} \cdot \text{s}^{-1}$, $\phi_v = 0.5\ \text{m} \cdot \text{s}^{-1}$, and $\Delta\theta_{\max} = 3^\circ$. The same parameter set was used for the 4°, 6°, and 8° slope conditions.

For all three cases, the initial pitch angle was $\theta(0) = 0^\circ$, corresponding to the equivalent upper body being initially aligned with the slope normal. The equilibrium pitch references were 6.1782°, 9.2650°, and 12.3489°, respectively, and the target uphill velocity was $0.5\ \text{m} \cdot \text{s}^{-1}$.

Each simulation lasted 10 s. The pitch settling time was defined as the time after which the pitch-angle error entered and remained within ±0.2°. The steady-state pitch error, pitch standard deviation, and mean uphill velocity were calculated over the final 0.5 s of each simulation.

Figure 7 shows that the SMC benchmark stabilizes the pitch angle and maintains uphill motion under all three slope conditions. For the SMC benchmark, the pitch-angle error entered and remained within ±0.2° at approximately 4.98 s, 5.43 s, and 5.94 s on the 4°, 6°, and 8° slopes, respectively. The corresponding settling times of the proposed hybrid controller were 5.00 s, 5.08 s, and 5.09 s. Therefore, the two controllers produced nearly identical pitch settling times on the 4° slope, whereas the proposed controller settled 0.35 s and 0.85 s earlier on the 6° and 8° slopes, respectively. The quantitative results are summarized in Table 2.

The results show that both controllers eventually achieve stable uphill motion under all three slope conditions. In the coordinate system defined earlier, the final along-slope displacements of the SMC-controlled system are 3.883 m, 3.694 m, and 3.485 m on the 4°, 6°, and 8° slopes, respectively. Its mean steady-state uphill velocities are $0.492\ \text{m}\cdot\text{s}^{-1}$, $0.490\ \text{m}\cdot\text{s}^{-1}$, and $0.488\ \text{m}\cdot\text{s}^{-1}$. Thus, the implemented SMC benchmark successfully tracks the uphill direction, although its steady-state velocity error increases slightly as the slope angle increases.

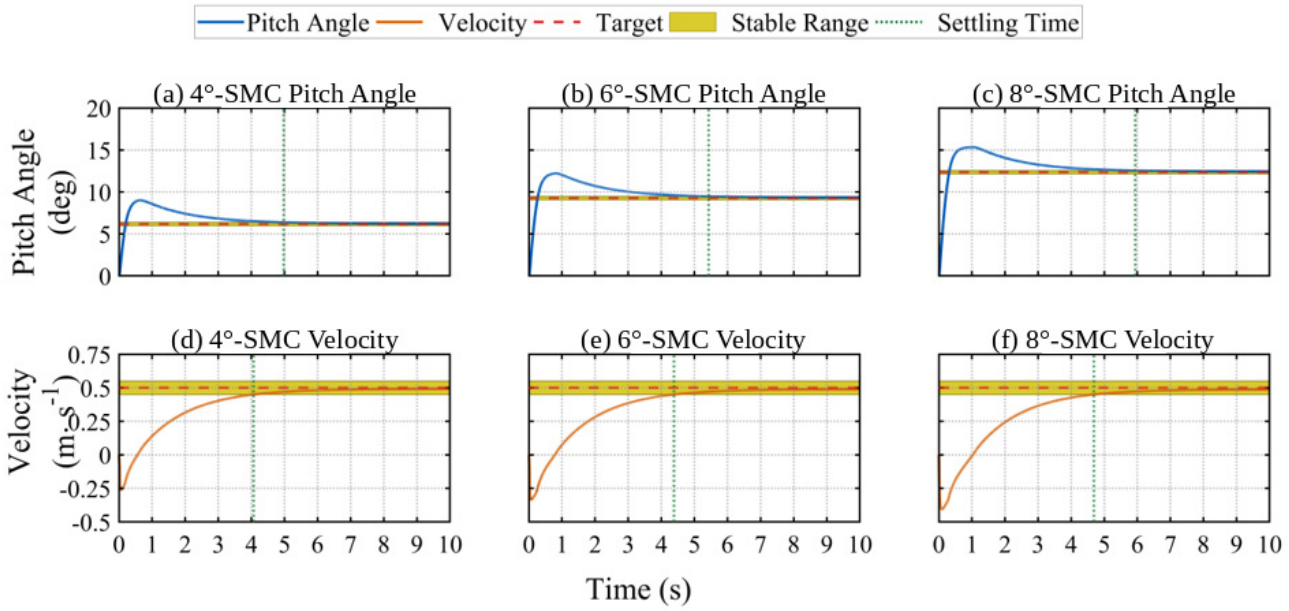


Figure 7. Pitch-angle and along-slope velocity responses of the SMC benchmark under different slope angles.

Table 2. Performance comparison between the proposed hybrid controller and the SMC benchmark.

Slope Angle (°)	Controller	Steady-State Pitch Error (°)	Pitch Std (°)	Mean Velocity ($m \cdot s^{-1}$)	Pitch Settling Time (s)
4	Hybrid	0.042	0.0027	0.498	5.00
4	SMC	0.051	0.0006	0.492	4.98
6	Hybrid	0.064	0.0044	0.497	5.08
6	SMC	0.075	0.0007	0.490	5.43
8	Hybrid	0.089	0.0078	0.497	5.09
8	SMC	0.099	0.0008	0.488	5.94

The proposed hybrid controller achieves final along-slope displacements of 4.600 m, 4.510 m, and 4.460 m, with corresponding mean steady-state uphill velocities of 0.498 $m \cdot s^{-1}$, 0.497 $m \cdot s^{-1}$, and 0.497 $m \cdot s^{-1}$. These values are closer to the target velocity of 0.5 $m \cdot s^{-1}$ than those obtained using SMC. The proposed controller also produces smaller mean steady-state pitch errors under all three slope conditions. In contrast, the SMC benchmark produces smaller pitch-angle standard deviations, indicating smoother steady-state pitch regulation.

Overall, both controllers achieved pitch stabilization and uphill motion under the tested slope conditions. The SMC benchmark produced smaller steady-state pitch fluctuations and a slightly shorter pitch settling time on the 4° slope. However, the proposed hybrid controller achieved faster pitch stabilization on the 6° and 8° slopes, smaller steady-state pitch errors, more accurate velocity tracking, and larger uphill displacements. These results indicate better integrated pitch and velocity regulation for the considered uphill task. This conclusion is limited to the implemented SMC benchmark, selected parameters, and common torque constraint.

4.4. Performance Under Different Prescribed Slope Angles

The hybrid-controller results from Section 4.2 were further analyzed to evaluate its performance under different prescribed slope conditions. Since the equilibrium pitch angle changes with the slope angle, the absolute pitch-angle error $E_\theta = |\theta - \theta_{eq}|$ and the absolute along-slope velocity error $E_v = |\dot{s} - v_{eq}|$ were used as the main performance indices. The

system was considered settled when $E_\theta \leq 0.2^\circ$ and $E_v \leq 0.05$ m·s^{-1} were both satisfied and remained satisfied thereafter. The first time after which both conditions remained continuously satisfied was defined as the settling time.

Figure 8 shows the pitch-angle-error responses.

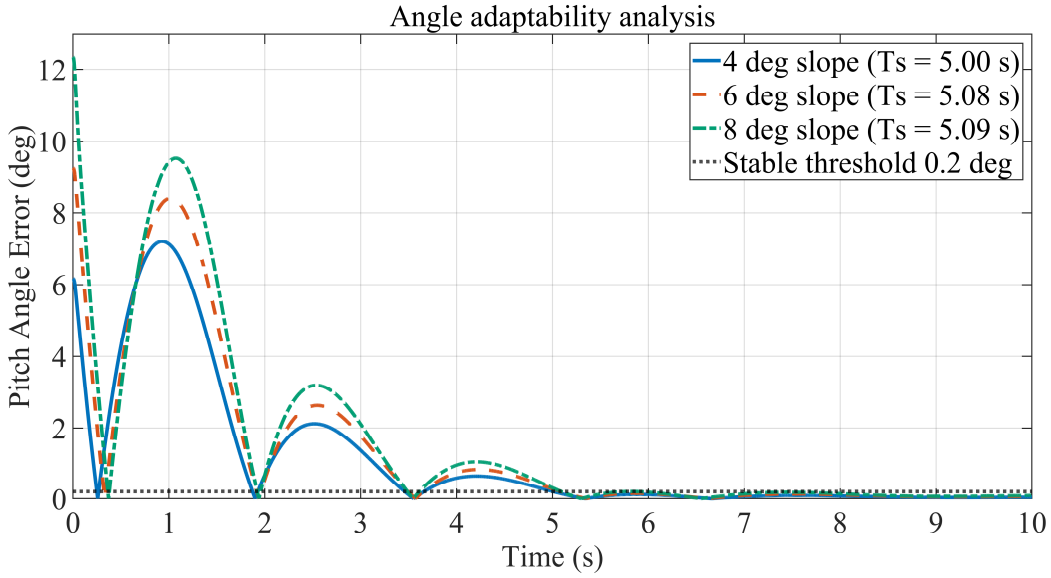


Figure 8. Absolute pitch-angle-error responses of the hybrid controller under different slope angles.

The pitch-angle error decreases gradually in all cases and finally enters the stable range. No sustained oscillation or divergence is observed.

Figure 9 shows the velocity-error responses.

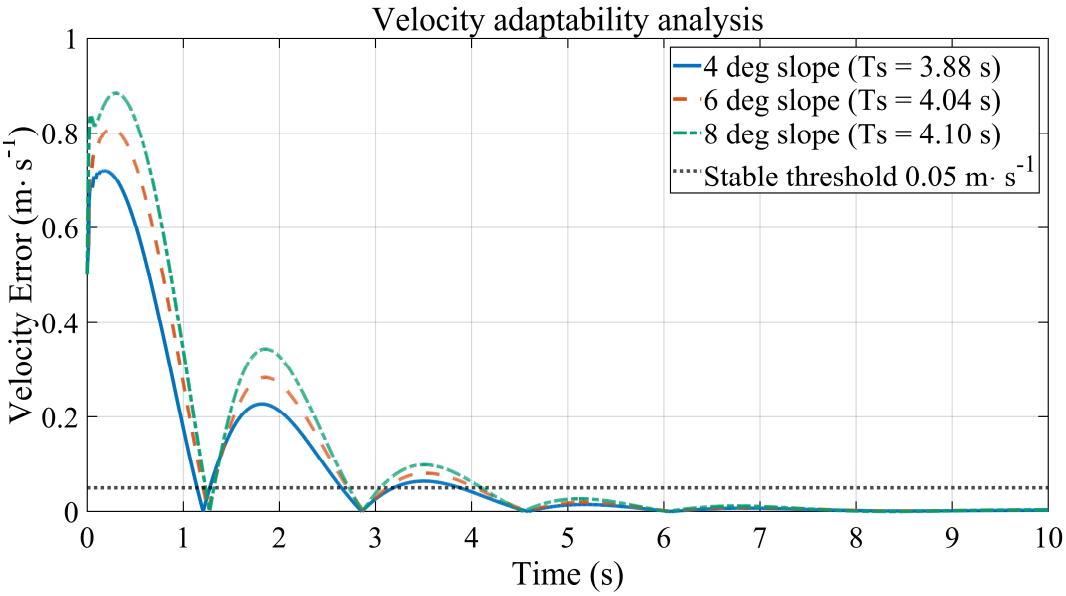


Figure 9. Absolute along-slope velocity-error responses of the hybrid controller under different slope angles.

The velocity error decreases after the transient stage and remains close to zero in the final stage. This indicates that the controller maintains the target uphill velocity after posture recovery.

Table 3 lists the performance indices under different slope angles.

Table 3. Adaptability indices of the proposed hybrid controller under different slope angles.

Performance Index	4°	6°	8°
Equilibrium angle (°)	6.1782	9.2650	12.3489
Settling time (s)	5.0000	5.0800	5.0900
Steady-state pitch error (°)	0.0420	0.0640	0.0890
Steady-state std (°)	0.0027	0.0044	0.0078
Mean steady-state velocity ($m \cdot s^{-1}$)	0.4980	0.4970	0.4970
Velocity reference ($m \cdot s^{-1}$)	0.5000	0.5000	0.5000

The settling times under 4°, 6°, and 8° slopes are about 5.00 s, 5.08 s, and 5.09 s, respectively. The final pitch errors are small, and the mean steady-state velocities are close to the target value. The similar settling times show that the slope-dependent equilibrium reference and compensation input provide consistent nominal performance when the prescribed slope parameter is changed from 4° to 8°. This comparison does not evaluate online slope estimation or robustness against a mismatch between the actual and controller-supplied slope angles.

4.5. Response to External Torque Disturbances

To evaluate the disturbance-recovery ability of the proposed hybrid controller, an additive disturbance torque was introduced at the longitudinal wheel-axle drive-torque input. The disturbance acted about the wheel axle rather than directly on the equivalent upper body. In the primary test, a constant +25 N·m rectangular torque pulse was applied from 5.2 to 5.4 s, after the wheelchair had entered the stable uphill-motion stage. The disturbance magnitude was selected by considering the expected practical wheel-axis loading and the ±60 N·m actuator capability of the prototype. The selected magnitude corresponds to approximately 41.7% of the wheel-end torque limit and represents a substantial but recoverable input disturbance. All other simulation conditions were identical to those used in Section 4.4. Figures 10 and 11 show the corresponding pitch-angle and along-slope velocity responses, and the main recovery indices are summarized in Table 4.

After the disturbance was applied at 5.2 s, the pitch angle showed a transient deviation under all three slope conditions. The maximum slope-referenced pitch angles on the 4°, 6°, and 8° slopes were 12.017°, 15.001°, and 17.965°, respectively. The corresponding maximum pitch-angle errors remained 5.837°, 5.741°, and 5.615°, respectively. The pitch oscillations subsequently decreased, and the pitch angle converged again toward the corresponding slope-dependent equilibrium value. During the final 0.5 s, the mean pitch-angle errors were −0.149°, −0.142°, and −0.155°, respectively.

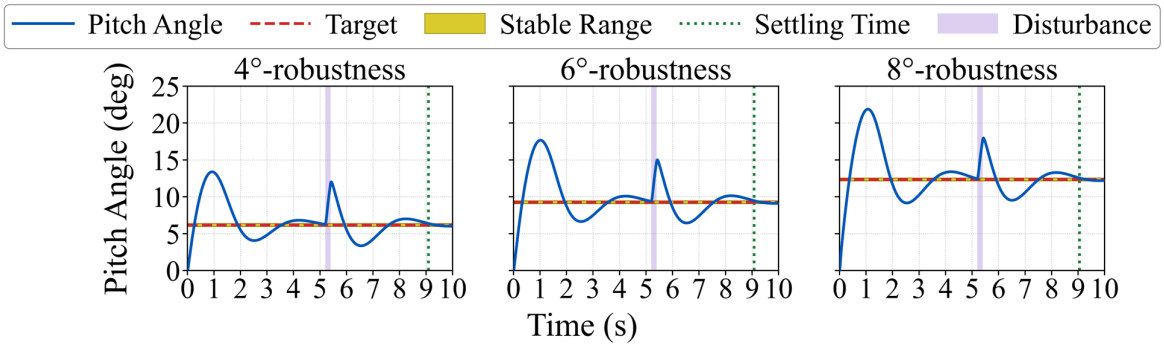


Figure 10. Pitch-angle responses of the hybrid controller under a short +25 N·m external torque disturbance.

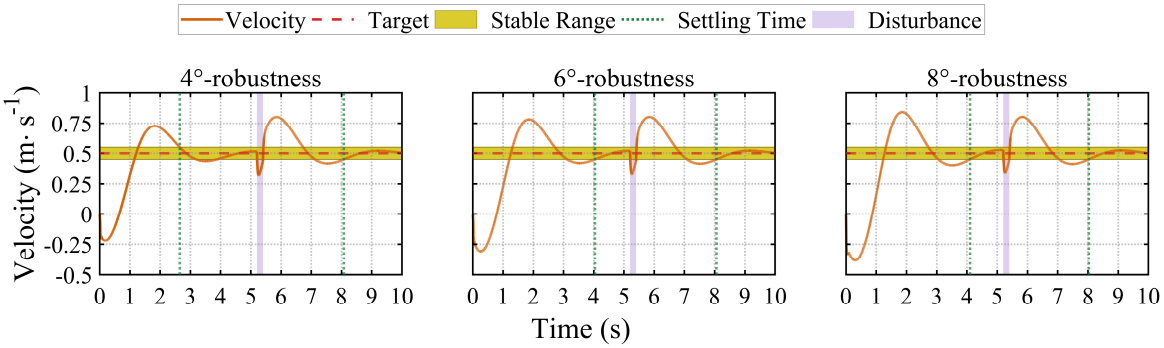


Figure 11. Along-slope velocity responses of the hybrid controller under a short +25 N·m external torque disturbance.

Table 4. Robustness-test results under a short +25 N·m external torque disturbance.

Performance Index	4°	6°	8°
Maximum pitch angle (°)	12.017	15.001	17.965
Maximum pitch error (°)	5.837	5.741	5.615
Minimum velocity $(\mathrm{m{\cdot}s^{-1}})$	0.363	0.370	0.378
Maximum velocity $(\mathrm{m{\cdot}s^{-1}})$	0.803	0.803	0.804
Mean final pitch error (°)	−0.149	−0.142	−0.155
Mean final velocity $(\mathrm{m{\cdot}s^{-1}})$	0.511	0.509	0.508
Velocity recovery duration (s)	2.876	2.859	2.828
Pitch recovery duration (s)	3.890	3.884	3.847

The disturbance also caused fluctuations in the along-slope velocity. The minimum along-slope velocities under 4°, 6°, and 8° slopes were 0.363 $\mathrm{m{\cdot}s^{-1}}$, 0.370 $\mathrm{m{\cdot}s^{-1}}$, and 0.378 $\mathrm{m{\cdot}s^{-1}}$, respectively. The maximum along-slope velocities were 0.803 $\mathrm{m{\cdot}s^{-1}}$, 0.803 $\mathrm{m{\cdot}s^{-1}}$, and 0.804 $\mathrm{m{\cdot}s^{-1}}$, respectively. The velocity then returned close to the target value. In the last 0.5 s, the mean along-slope velocities were 0.511 $\mathrm{m{\cdot}s^{-1}}$, 0.509 $\mathrm{m{\cdot}s^{-1}}$, and 0.508 $\mathrm{m{\cdot}s^{-1}}$, respectively.

Table 4 lists the recovery indices after the disturbance.

The velocity-error re-settling times under the 4°, 6°, and 8° slopes were 2.876 s, 2.859 s, and 2.828 s, respectively. The corresponding pitch-error re-settling times were 3.890 s, 3.884 s, and 3.847 s. The recovery times show only small differences among the three slope cases. Overall, the proposed controller recovered both pitch posture and uphill velocity after the short +25 N·m torque disturbance. This result indicates that the controller can suppress transient pitch deviation and restore the along-slope velocity under finite-duration torque disturbance.

The 8° slope case produced the largest pitch excursion and steady-state pitch fluctuation in the preceding tests. It was therefore selected for the supplementary disturbance tests.

Figure 12 compares three deterministic disturbance modes on the 8° slope: a short positive torque of +25 N·m from 5.2 to 5.4 s, a short negative torque of −25 N·m over the same interval, and a sustained positive torque of +25 N·m from 5.2 s to the end of the simulation. Under the two short disturbances, both pitch and velocity showed transient deviations but recovered after the torque was removed. Under the sustained positive disturbance, the pitch angle remained bounded and converged to a stable posture. However, the along-slope velocity did not return to the 0.5 $\mathrm{m{\cdot}s^{-1}}$ reference. This indicates weakened velocity-maintenance capability under a persistent torque bias.

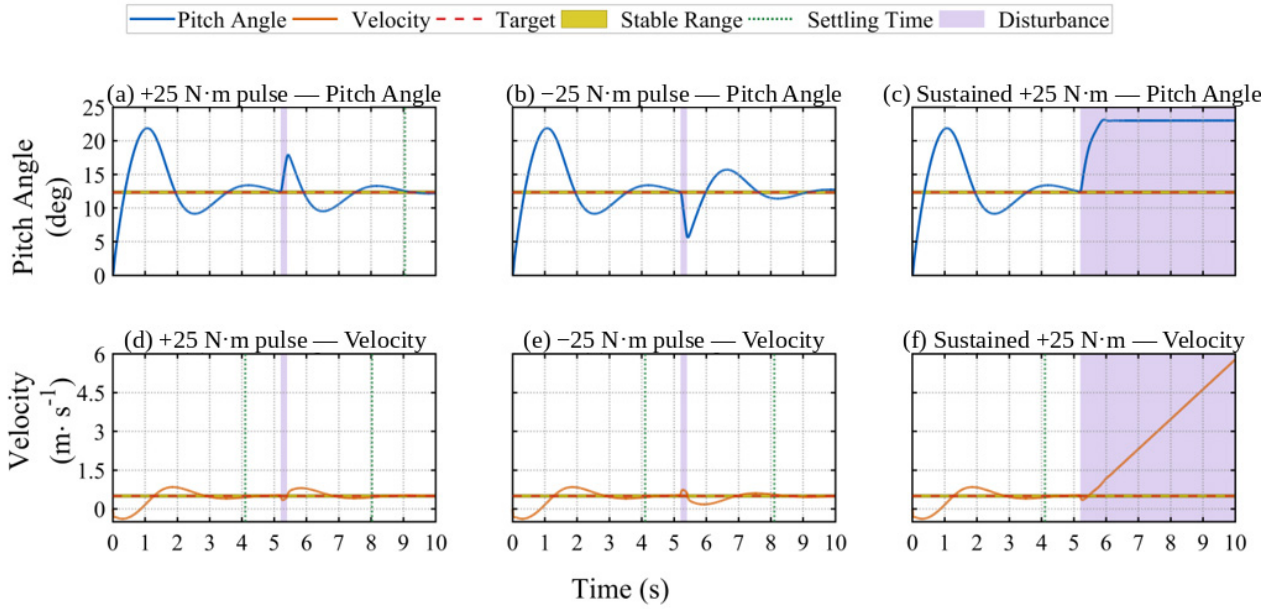


Figure 12. Responses under positive and negative rectangular torque pulses and a sustained positive torque step on the 8° slope.

The response to time-varying torque disturbances was further evaluated using five random-torque tests. In each test, the disturbance torque was updated every 0.2 s from 5.2 s to 7.2 s and remained within ±25 N·m. The wheelchair remained upright in all five runs. The largest peak absolute pitch error was 8.428°. After the random disturbance sequence ended, both pitch posture and along-slope velocity recovered. The final-1-s mean velocity was 0.494 ± 0.022 m·s^{-1}. These results show that the controller can tolerate rapidly varying bidirectional torque disturbances without losing stability.

Overall, the controller recovered both pitch posture and uphill velocity after the tested short positive, short negative, and time-varying random torque disturbances. Under the sustained positive torque disturbance, pitch stability was retained, but velocity regulation degraded. These results demonstrate recovery from the applied wheel-axis torque disturbances under the nominal model parameters. They should not be interpreted as evidence of robustness against parameter uncertainty, slope-estimation error, measurement noise, or sensing and actuation delays, because these effects were not included in the present simulations.

Hard switching and smooth switching were also compared under the 8° slope condition. The two cases used the same initial state, switching threshold, minimum NMPC weight, controller parameters, and input constraints. A one-sided quadratic fit was applied to the 10 ms intervals before and after the switching boundary. With hard switching, the estimated changes in pitch acceleration and along-slope acceleration were 18.97 rad·s^{-1} and 6.68 m·s^{-1}, respectively. With smooth switching, these values decreased to 0.79 rad·s^{-1} and 0.13 m·s^{-1}. The corresponding reductions were 95.8% and 98.0%. These results show that smooth weighting reduces the estimated changes in pitch acceleration and along-slope acceleration at the switching boundary while maintaining comparable stabilization performance.

4.6. Sensitivity Analysis of the Weighting and Switching Parameters

The sensitivity of the minimum NMPC weight was first evaluated under the 8-degree slope condition. Minimum weights of 0.3, 0.4, and 0.5 were tested. The slope angle, initial physical posture, target uphill velocity, switching threshold, transition width, controller parameters, input constraints, and simulation duration were kept unchanged. Pitch settling

time was defined as the time after which the absolute pitch error entered and remained within 0.2 degrees. The mean absolute pitch error and pitch-angle standard deviation were calculated over the final 1 s. The mean velocity error was calculated as the absolute difference between the final-1-s mean uphill velocity and the reference velocity of 0.5 m/s.

As shown in Table 5, the three minimum weights produced different compromises between local pitch regulation and velocity maintenance. The minimum weight of 0.3 produced the smallest pitch-angle standard deviation of 0.0033 degrees. However, its pitch settling time was 5.907 s, and its mean velocity error was the largest among the three cases. Increasing the minimum weight to 0.4 shortened the pitch settling time to 5.090 s and reduced the final-1-s mean absolute pitch error to 0.0741 degrees. These were the best values among the tested cases. Its mean velocity error was 0.00289 m/s, corresponding to approximately 0.58% of the velocity reference.

Table 5. Sensitivity of closed-loop performance to the minimum NMPC weight under the 8-degree slope condition.

Minimum NMPC Weight	Pitch Settling Time (s)	Mean Absolute Pitch Error Over Final 1 s (°)	Pitch-Angle SD Over Final 1 s (°)	Mean Velocity Error Over Final 1 s ($m \cdot s^{-1}$)
0.3	5.907	0.0998	0.0033	0.00586
0.4	5.090	0.0741	0.0078	0.00289
0.5	6.775	0.1107	0.0196	0.00031

The minimum weight of 0.5 produced the smallest mean velocity error. However, its pitch settling time increased to 6.775 s. Its mean absolute pitch error and pitch-angle standard deviation also increased to 0.1107 degrees and 0.0196 degrees, respectively. These results indicate that increasing the retained NMPC contribution to 0.5 improves velocity maintenance but weakens local pitch regulation. Therefore, the minimum weight of 0.4 provides the most suitable overall compromise between pitch convergence, steady-state pitch accuracy, and uphill-velocity tracking under the tested condition. This value is used as the nominal setting, but it is not considered a universal optimum or an analytical stability bound.

Additional simulations were conducted to evaluate the effects of the equilibrium-neighborhood threshold and transition width. The 8° slope case was used because it produced the largest initial pitch deviation among the stable slope cases. The initial pitch angle was 0°, the target uphill velocity was 0.5 $m \cdot s^{-1}$, the minimum NMPC weight was 0.4, and the wheel-end torque was limited to ± 60 N·m.

A one-factor-at-a-time procedure was used. In the threshold-sensitivity test, the transition width was fixed at 2°, while the equilibrium-neighborhood threshold was set to 8°, 9°, 10°, and 11°. In the transition-width test, the threshold was fixed at 10°, while the width was set to 0.5°, 1°, and 2°. The initial absolute pitch error was approximately 12.35°. A transition width of 3° would extend the transition region from 10° to 13° and would therefore place the initial state directly inside the transition region. In that case, the controller would begin with a reduced NMPC weight rather than the full NMPC weight used for large-deviation recovery. The 3° width was therefore excluded to ensure that all tested width cases started in the same full-NMPC region. All other model parameters, controller settings, initial conditions, and input constraints were kept unchanged.

The system was considered settled when the absolute pitch-angle error was no greater than 0.2° and the absolute along-slope velocity error was no greater than 0.05 $m \cdot s^{-1}$, with both conditions remaining satisfied thereafter. The first time at which these two conditions were continuously satisfied was defined as the settling time. For each stable threshold case,

the final pitch error was evaluated at the end of the 10 s simulation, whereas the pitch-angle standard deviation and mean steady-state velocity were calculated over the final 1 s.

Table 6 shows that the threshold mainly affects transient recovery and closed-loop stability. The 8° threshold failed to maintain stable operation during the 10 s simulation. The 9° threshold restored stability, but increased the settling time. The nominal 10° threshold gave the shortest settling time. When the threshold was increased to 11°, the initial pitch-error magnitude lay inside the transition interval, so the controller started with a reduced NMPC weight. This explains the longer settling time.

Table 6. Sensitivity of closed-loop performance to the equilibrium-neighborhood threshold under the 8° slope condition with a fixed transition width of 2°.

Threshold (°)	Stability Outcome	Pitch Settling Time (s)	Final Pitch Error (°)	Steady-State Pitch SD (°)	Mean Steady-State Velocity ($m \cdot s^{-1}$)
8	Unstable	-	-	-	-
9	Stable	7.558	0.090	0.0107	0.4970
10	Stable	5.090	0.089	0.0078	0.4971
11	Stable	5.926	0.089	0.0079	0.4971

For the stable threshold cases from 9° to 11°, the final pitch errors remain between 0.089° and 0.090°, while the mean steady-state velocities remain between 0.4970 and 0.4971 $m \cdot s^{-1}$. Therefore, the steady-state performance is relatively insensitive within this range. However, the threshold still affects the recovery time and whether stable operation is maintained in the tested simulation. The 10° threshold was consequently selected because it provides the fastest stable response while preserving both pitch regulation and uphill velocity tracking.

The transition-width effect was concentrated near the initial transfer of control authority from NMPC to LQR. Outside this interval, the responses for 0.5°, 1°, and 2° almost overlapped. Therefore, the effect was evaluated using the maximum absolute pitch angular velocity during the initial transition. Increasing the width reduced this peak value. The 2° width gave the smallest peak value among the admissible tested cases. A 3° width was excluded because it would place the initial state inside the transition region and change the initial control regime.

For transition widths of 0.5°, 1°, and 2°, the maximum absolute pitch angular velocities during the initial transition were 0.984, 0.953, and 0.880 $rad \cdot s^{-1}$, respectively. The monotonic decrease indicates that increasing the transition width makes the transfer of control authority less aggressive. Compared with the 0.5° width, the 2° width reduced the peak value by approximately 10.6%. A 3° width was not considered because the initial absolute pitch error of approximately 12.35° would lie inside the corresponding 10–13° transition region, causing the controller to begin with a reduced NMPC weight. The 2° width is therefore the largest evaluated width that preserves the initial full-NMPC recovery region and provides the smoothest transition among the admissible tested widths. It was consequently selected as the nominal transition width.

4.7. Computational Cost and Real-Time Limitations

The computational cost of the current controller implementation was evaluated under the nominal 8° slope condition. The initial pitch angle was 0°, the target uphill velocity was 0.5 $m \cdot s^{-1}$, the minimum NMPC weight was 0.4, and the wheel-end torque was constrained to ±60 N·m. The execution times of the LQR calculation, NMPC optimization, and complete hybrid controller were recorded separately.

The timing test was conducted using MATLAB/Simulink and CasADi on a computer equipped with an Intel Core Ultra 5 125H processor with a base clock frequency of 1.20 GHz, 32 GB RAM, and Windows 11 Home, version 25H2. A total of 10,001 controller evaluations were recorded during the 10 s simulation. The reported statistics include the initialization call. The mean value, standard deviation, 95th percentile, and maximum value were calculated from all recorded calls. The simulations were performed using MATLAB/Simulink R2023b, CasADi version 3.7.2, and the IPOPT version distributed with the installed CasADi package IPOPT 3.14.11.

The execution-time statistics are summarized in Table 7.

Table 7. Execution-time statistics for 10,001 controller evaluations under the nominal 8° slope condition.

Controller Component	Mean (ms)	Standard Deviation (ms)	95th Percentile (ms)	Maximum (ms)
LQR calculation	0.0127	0.0081	0.0220	0.4511
NMPC optimization	433.6704	23.6380	458.8047	1517.4224
Complete hybrid controller	433.6993	23.6509	458.8348	1519.6724

The timing results show that the computational cost is dominated by NMPC optimization. The LQR calculation required only 0.0127 ms per call on average, whereas the NMPC optimization required 433.6704 ms. The complete hybrid controller required 433.6993 ms per call. The difference between the complete hybrid-controller time and the NMPC time was only 0.0289 ms, corresponding to 0.0067% of the total average execution time. Therefore, the LQR calculation and smooth-weighting operation introduce negligible overhead. Since the NMPC problem is still solved at every controller call, the present hybrid structure should not be interpreted as a method for reducing the computational cost of NMPC.

The measured computation time is substantially greater than both the 1 ms Simulink block-invocation interval and the 10 ms NMPC prediction-grid interval used in the present offline implementation. The 10,001 recorded evaluations represent the Simulink scheduling sequence during the 10 s offline simulation and should not be interpreted as a verified physical controller sampling frequency. IPOPT returned a successful solution status in all 10,001 recorded evaluations, corresponding to a solver convergence rate of 100% in the nominal timing run.

The measured computation time therefore confirms that the current implementation should be interpreted only as an offline simulation prototype and not as a real-time embedded controller. Further implementation-level optimization and real-time validation will be required before practical deployment can be considered. Future work will include persistent solver construction, warm starting, compiled code generation, and hardware-in-the-loop evaluation. Recent NMPC toolchains and embedded optimal-control frameworks provide possible support for these steps [30,31].

4.8. Summary of Simulation Results

The simulation results show that the proposed NMPC–LQR smooth hybrid control method achieves stable uphill motion on 4°, 6°, and 8° slopes. On the 4° slope, both the hybrid controller and standalone NMPC remained within the operational fall boundary, although the hybrid controller produced a smaller final pitch error and lower steady-state fluctuation. On the 6° and 8° slopes, the proposed hybrid controller remained stable throughout the 10 s simulations, whereas standalone NMPC reached the operational fall boundary at 3.232 s and 1.223 s, respectively. The standalone NMPC torque reached the actuator limit before both fall events.

The prescribed-slope comparison shows similar settling times of 5.00 s, 5.08 s, and 5.09 s for the 4°, 6°, and 8° slopes when the plant and controller use matching slope parameters. This indicates consistent nominal performance across the tested prescribed slopes but does not establish robustness against slope-estimation errors. However, the settling times remain close, with values of 5.00 s, 5.08 s, and 5.09 s, respectively. This result shows consistent nominal performance across the tested prescribed slopes. It does not establish robustness against slope-estimation errors.

The disturbance tests characterize the controller response to the applied wheel-axis torque disturbances under nominal model parameters. The controller recovered both pitch posture and uphill velocity after the short positive, short negative, and time-varying random torque disturbances. Under the sustained torque disturbance, the pitch posture remained stable, but velocity regulation degraded. The hard-switching comparison also shows that smooth weighting reduces the estimated pitch- and along-slope-acceleration change at the switching boundary.

The parameter-sensitivity results support the selected weighting and switching parameters under the tested 8° slope condition. The minimum-weight comparison showed that a value of 0.4 produced the shortest pitch settling time and the smallest final-1-s mean absolute pitch error, while maintaining a small velocity-tracking error. Thresholds from 9° to 11° produced similar steady-state pitch and velocity performance, while the 10° threshold gave the shortest settling time. The 8° threshold did not maintain stable operation. Increasing the transition width from 0.5° to 2° reduced the peak pitch angular velocity during the initial transition. Therefore, $w_{min} = 0.4$, $\varepsilon_\theta = 10°$, and $\Delta_\theta = 2°$ were selected as the nominal settings.

The computational results show that the LQR and smooth-weighting calculations add negligible overhead to the NMPC computation. However, the complete hybrid controller remains dominated by NMPC optimization and does not satisfy the computation-time requirement for real-time implementation under the present simulation settings. Therefore, the present results demonstrate offline simulation performance, but not real-time embedded feasibility.

The validation scope of the present study should be interpreted carefully. All closed-loop simulations were conducted using nominal occupant mass, center-of-mass location, equivalent pendulum length, inertial parameters, friction coefficients, and actuator characteristics. Exact state feedback and the exact prescribed slope angle were used. Parameter uncertainty, payload variation, slope-estimation error, measurement noise, quantization, sensing delay, communication delay, and actuator delay were not evaluated. Therefore, the torque-disturbance tests establish disturbance-recovery behavior only for the applied additive wheel-axis torque inputs under the nominal simulation model.

These untested effects cannot be inferred directly from the existing torque-disturbance results. Changes in occupant mass and center-of-mass location modify M_p, l, and I_{xx}, thereby producing persistent prediction-model mismatch. Friction and actuator variations alter the available longitudinal control authority. Slope-estimation error biases both the equilibrium reference and the feedforward torque. Measurement noise affects the state-feedback terms and the pitch-error-dependent blending weight, while sensing and actuation delays may degrade the predictive and local stabilization responses. No quantitative tolerance, operational-safety claim, or closed-loop robustness bound is therefore established for these effects.

Accordingly, the present results demonstrate nominal offline-simulation performance under the specified model, prescribed slope inputs, and tested torque disturbances. They do not establish practical robustness or operational safety for a physical wheelchair.

5. Conclusions

This paper proposed an NMPC–LQR smooth hybrid control method for the pitch stabilization and velocity maintenance of a two-wheeled self-balancing wheelchair during uphill motion on slopes.

First, the wheelchair–occupant system was modeled as a wheeled inverted-pendulum system in the slope coordinate frame. A reduced nonlinear dynamic model was established with nonholonomic rolling constraints and null-space projection. A slope-dependent equilibrium angle and a steady-state compensation torque were then introduced. These terms allow the control target to reflect the influence of the slope gravity component on the balance posture and the driving torque demand.

Based on this model, a hybrid control framework combining NMPC and LQR was developed. NMPC provides nonlinear predictive recovery capability in the large-deviation region, whereas LQR provides local stabilization near the slope-dependent equilibrium point. A smoothstep weighting function continuously blends the two controller outputs within the transition region. The comparison with hard switching confirms that the smooth weighting strategy substantially attenuates switching-induced dynamic discontinuities without degrading the overall stabilization performance.

The parameter-sensitivity analysis further examined the effects of the minimum NMPC weight, equilibrium-neighborhood threshold, and transition width. Under the tested 8° slope condition, the minimum NMPC weight of 0.4 produced the shortest pitch settling time and the smallest final-1-s mean absolute pitch error among the tested values, while retaining accurate uphill-velocity tracking. It was therefore selected as the nominal empirical setting. Thresholds from 9° to 11° produced similar steady-state pitch and velocity performance, while the 10° threshold gave the shortest settling time. Increasing the transition width from 0.5° to 2° reduced the maximum absolute pitch angular velocity during the initial transition. A 3° width was excluded because the initial absolute pitch error would lie inside the corresponding transition region, causing the controller to start with a reduced NMPC weight. The 2° width is therefore the largest evaluated width that preserves the initial full-NMPC recovery region and produces the smoothest transition among the admissible tested cases. Overall, the proposed method combines slope-dependent dynamic modeling, nonlinear predictive control, local linear stabilization, and continuous weight blending.

The simulation results show that the proposed method improves pitch stability and uphill velocity tracking. The added wheel-end torque histories also show bounded actuator commands without sustained saturation under the tested hybrid-controller conditions. Compared with standalone LQR, the proposed controller avoids downhill motion and maintains the desired uphill velocity. Compared with standalone NMPC, the proposed controller produced a smaller pitch error and lower pitch fluctuation in the stable 4° case. It also remained within the operational fall boundary on the 6° and 8° slopes, whereas standalone NMPC became unrecoverable under the implemented settings. The SMC benchmark also achieved stable uphill motion, but the proposed controller provided smaller steady-state pitch errors, more accurate velocity tracking, and faster pitch stabilization on the 6° and 8° slopes under the tested settings.

The additional tests characterize the torque-disturbance recovery and switching behavior of the proposed method under the tested nominal conditions. The controller recovered after the tested short positive, short negative, and time-varying random torque disturbances. Under a sustained torque disturbance, pitch stability was retained, but velocity regulation degraded. These results are limited to the implemented additive torque disturbances and do not establish robustness to model-parameter uncertainty, slope-estimation error, measurement noise, or delay.

https://doi.org/10.3390/electronics15184110

The computational measurements showed that the added LQR and smooth-weighting calculations introduced negligible overhead. However, the current MATLAB/Simulink and CasADi implementation was dominated by NMPC optimization and did not satisfy the computation-time requirement for real-time implementation under the present simulation settings. Future work will first optimize the NMPC implementation through persistent solver construction, warm starting, compiled code generation, real-time iteration, and prediction-horizon optimization. Subsequent validation will include structured variations in occupant mass, center-of-mass location, equivalent pendulum length, friction, and actuator characteristics. Slope-estimation errors, sensor noise, quantization, sensing and actuation delays, and estimator dynamics will also be evaluated. Processor-in-the-loop, hardware-in-the-loop, and physical-prototype tests will be conducted after the real-time computation requirement has been satisfied.

Author Contributions: Conceptualization, Y.G. and H.T.; methodology, Y.G., H.S., J.X. and X.Z.; software, Y.G.; validation, Y.G., H.S., J.X. and X.Z.; formal analysis, Y.G. and H.S.; investigation, Y.G., H.S., J.X. and X.Z.; data curation, Y.G., H.S., J.X. and X.Z.; writing—original draft preparation, Y.G.; writing—review and editing, Y.G., H.S., J.X., X.Z., Q.M., H.Y. and H.T.; visualization, Y.G. and H.S.; supervision, H.T.; project administration, H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by National Natural Science Foundation of China (62403320), and Shanghai High-Quality Development Program for Science and Technology Industry–Elderly Care Technology Support Project (grant no. 25YL1900900).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Chan, R.P.M.; Stol, K.A.; Halkyard, C.R. Review of Modelling and Control of Two-Wheeled Robots. *Annu. Rev. Control* **2013**, *37*, 89–103. [CrossRef]
2. Grasser, F.; D'Arrigo, A.; Colombi, S.; Rufer, A.C. JOE: A Mobile, Inverted Pendulum. *IEEE Trans. Ind. Electron.* **2002**, *49*, 107–114. [CrossRef]
3. Takahashi, Y.; Tsubouchi, O. Modern Control Approach for Robotic Wheelchair with Inverse Pendulum Control. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Warsaw, Poland, 8–10 September 2005*; IEEE: New York, NY, USA, 2005; pp. 364–369. [CrossRef]
4. Ahmad, S.; Tokhi, M.O. Forward and Backward Motion Control of Wheelchair on Two Wheels. In *Proceedings of the 2008 3rd IEEE Conference on Industrial Electronics and Applications, Singapore, 3–5 June 2008*; IEEE: New York, NY, USA, 2008; pp. 461–466. [CrossRef]
5. Aula, A.; Altalmas, T.M.; Ahmad, S.; Akmeliawati, R.; Sidek, S.N.; Tokhi, M.O. Integrated design, modelling and analysis of two-wheeled wheelchair for disabled. In *Proceedings of the Nature-Inspired Mobile Robotics*; WORLD SCIENTIFIC: Sydney, Australia, 2013; pp. 141–152. [CrossRef]
6. Kim, S.; Kwon, S. Dynamic Modeling of a Two-Wheeled Inverted Pendulum Balancing Mobile Robot. *Int. J. Control Autom. Syst.* **2015**, *13*, 926–933. [CrossRef]
7. Delgado, S.; Gajbhiye, S.; Banavar, R.N. Reduced Equations of Motion for a Wheeled Inverted Pendulum. *IFAC-PapersOnLine* **2015**, *48*, 328–333. [CrossRef]
8. Jamin, N.F.; Abdul Ghani, N.M.; Ibrahim, Z.; Nasir, A.N.K.; Rashid, M.; Tokhi, M.O. Stabilizing Control of Two-Wheeled Wheelchair with Movable Payload Using Optimized Interval Type-2 Fuzzy Logic. *J. Low Freq. Noise Vib. Act. Control* **2021**, *40*, 1585–1606. [CrossRef]
9. Razali, N.A.A.; Abdul Ghani, N.M.; Bari, B.S. Lifting and Stabilizing of Two-Wheeled Wheelchair System Using Interval Type-2 Fuzzy Logic Control Based Spiral Dynamic Algorithm. *Bull. EEI* **2021**, *10*, 3019–3031. [CrossRef]
10. Ahmad, S.; Tokhi, M.O. Linear Quadratic Regulator (LQR) Approach for Lifting and Stabilizing of Two-Wheeled Wheelchair. In *Proceedings of the 2011 4th International Conference on Mechatronics (ICOM), Kuala Lumpur, Malaysia, 17–19 May 2011*; IEEE: New York, NY, USA, 2011; pp. 1–6. [CrossRef]

11. Pathak, K.; Franch, J.; Agrawal, S.K. Velocity and Position Control of a Wheeled Inverted Pendulum by Partial Feedback Linearization. *IEEE Trans. Robot.* **2005**, *21*, 505–513. [CrossRef]
12. Kanjanawanishkul, K. LQR and MPC Controller Design and Comparison for a Stationary Self-Balancing Bicycle Robot with a Reaction Wheel. *Kybernetika* **2015**, *51*, 173–191. [CrossRef]
13. Yue, M.; An, C.; Sun, J.-Z. An Efficient Model Predictive Control for Trajectory Tracking of Wheeled Inverted Pendulum Vehicles with Various Physical Constraints. *Int. J. Control Autom. Syst.* **2018**, *16*, 265–274. [CrossRef]
14. Albert, K.; Phogat, K.S.; Anhalt, F.; Banavar, R.N.; Chatterjee, D.; Lohmann, B. Structure-Preserving Constrained Optimal Trajectory Planning of a Wheeled Inverted Pendulum. *IEEE Trans. Robot.* **2020**, *36*, 910–923. [CrossRef]
15. Andersson, J.A.E.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A Software Framework for Nonlinear Optimization and Optimal Control. *Math. Prog. Comp.* **2019**, *11*, 1–36. [CrossRef]
16. Liu, C.; Qin, K.; Xin, G.; Li, C.; Wang, S. Nonlinear Model Predictive Control for a Self-Balancing Wheelchair. *IEEE Access* **2024**, *12*, 28938–28949. [CrossRef]
17. Nasrallah, D.S.; Michalska, H.; Angeles, J. Controllability and Posture Control of a Wheeled Pendulum Moving on an Inclined Plane. *IEEE Trans. Robot.* **2007**, *23*, 564–577. [CrossRef]
18. Peng, K.; Ruan, X.; Zuo, G. Dynamic Model and Balancing Control for Two-Wheeled Self-Balancing Mobile Robot on the Slopes. In *Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China, 6–8 July 2012*; IEEE: New York, NY, USA, 2012; pp. 3681–3685. [CrossRef]
19. Liang, D.; Sun, N.; Wu, Y.; Fang, Y. Modeling and Motion Control of Self-Balance Robots on the Slope. In *Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, Hubei Province, China, 11–13 November 2016*; IEEE: New York, NY, USA, 2016; pp. 93–98. [CrossRef]
20. Kim, D.; Choi, D. Control of a Longitudinally Extended Two-Wheeled Inverted Pendulum Robot with a Sliding Mechanism. *J. Comput. Des. Eng.* **2025**, *12*, 118–132. [CrossRef]
21. 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]
22. Hirata, K.; Kamatani, M.; Murakami, T. Advanced Motion Control of Two-Wheel Wheelchair for Slope Environment. In *Proceedings of the IECON 2013—39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10–13 November 2013*; IEEE: New York, NY, USA, 2013; pp. 6436–6441. [CrossRef]
23. Dinale, A.; Hirata, K.; Zoppi, M.; Murakami, T. Parameter Design of Disturbance Observer for a Robust Control of Two-Wheeled Wheelchair System. *J. Intell. Robot. Syst.* **2015**, *77*, 135–148. [CrossRef]
24. Huang, J.; Guan, Z.-H.; Matsuno, T.; Fukuda, T.; Sekiyama, K. Sliding-Mode Velocity Control of Mobile-Wheeled Inverted-Pendulum Systems. *IEEE Trans. Robot.* **2010**, *26*, 750–758. [CrossRef]
25. Huang, K.; Wang, M.; Zhen, S. Udwadia-Kalaba Equation Based Adaptive Robust Control for Two-Wheeled Inverted Pendulum System: Underactuation and Uncertainty. *Int. J. Control Autom. Syst.* **2021**, *19*, 2420–2430. [CrossRef]
26. 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]
27. Kim, Y.; Kwon, S. Robust Stabilization of Underactuated Two-Wheeled Balancing Vehicles on Uncertain Terrains with Nonlinear-Model-Based Disturbance Compensation. *Actuators* **2022**, *11*, 339. [CrossRef]
28. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained Model Predictive Control: Stability and Optimality. *Automatica* **2000**, *36*, 789–814. [CrossRef]
29. Khani, F.; Haeri, M. Smooth Switching in a Scheduled Robust Model Predictive Controller. *J. Process Control* **2015**, *31*, 55–63. [CrossRef]
30. Florez, A.; Astudillo, A.; Decré, W.; Swevers, J.; Gillis, J. IMPACT: A Toolchain for Nonlinear Model Predictive Control Specification, Prototyping, and Deployment. *IFAC-PapersOnLine* **2023**, *56*, 3164–3169. [CrossRef]
31. Verschueren, R.; Frison, G.; Kouzoupis, D.; Frey, J.; Van Duijkeren, N.; Zanelli, A.; Novoselnik, B.; Albin, T.; Quirynen, R.; Diehl, M. Acados—A Modular Open-Source Framework for Fast Embedded Optimal Control. *Math. Prog. Comp.* **2022**, *14*, 147–183. [CrossRef]

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.