Model Predictive Control Using Dynamic Model Decomposition Applied to Two-Wheeled Inverted Pendulum Mobile Robot

Junjie Shen[1] and Dennis Hong[1]

Abstract— **In this paper, we discuss the locomotion control for the two-wheeled inverted pendulum (TWIP) mobile robot. The robot in consideration involves two independent driving wheels sharing the same axle as well as one inverted pendulum in the middle acting as the main body. Instead of considering the entire TWIP mobile robot as a whole, following the idea of *Dynamic Model Decomposition*, we decompose the robot into the body and the two wheels, with interaction forces and moments connecting them. The effect is that we can thus enjoy lower-dimensional dynamics for each subsystem while their composition maintaining the equivalence to the full-order robot model. Based on that, we further propose a corresponding model predictive control framework via quadratic programming, which considers linearly approximated body dynamics with constrained wheel reaction forces as inputs. The overall methodology was successfully implemented on a TWIP mobile robot in the simulation environment. The simulation results show that the robot is capable of station keeping, disturbance rejection, velocity tracking, and path following.**

I. Introduction

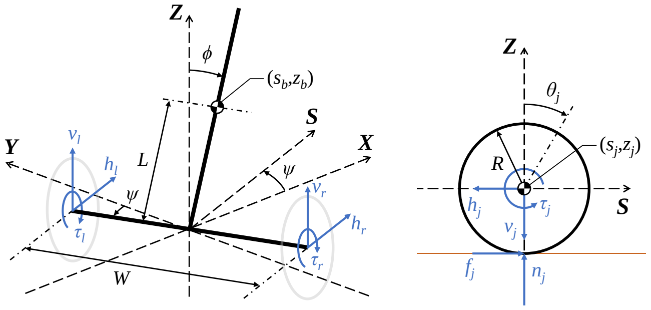
The intelligent robot system investigated in this paper is a dynamically-stable wheeled mobile robot with two driving wheels and an inverted pendulum acting as the body, thus the so-called two-wheeled inverted pendulum (TWIP) system. It has been widely applied in many fields due to its compact structure, great maneuverability, high energy efficiency, etc. In terms of a mobility mechanism, the robot needs to be capable of some basic movement actions, e.g., advance, reverse, and steering, and most importantly in the meanwhile, the robot must always keep its balance from falling down. The robot is a typical underactuated system since it only has two control inputs, i.e., the left and right wheel actuators, while it has three degrees of freedom (DoFs) under the condition of pure rolling without slipping, e.g., the left and right wheel rotation angles as well as the body tilt angle. The characteristics of underactuated system, nonholonomic constraint, nonlinear dynamics, etc., challenge the researchers to come up with a sound locomotion control strategy for the TWIP mobile robot system.

While classic control theories are no more effective for the TWIP system with such complexity, numerous studies of advance control approaches have been undertaken. The dynamics of the TWIP system is highly nonlinear. However, it is often possible to obtain a linearly approximated model around the operating point, where the deviations of the states and controls are assumed small. Several linear controllers were proposed accordingly, e.g., a linear stabilizing controller based on Routh–Hurwitz stability criterion was designed in [1], motion control using linear-quadratic regulator was proposed in [2], and the design technique of pole placement was used in [3]. There also exist significant efforts devoted to many other control strategies, e.g., partial feedback linearization [4], sliding mode control [5], fuzzy control [6], neural network-based motion control [7], and adaptive backstepping control [8].

Within robotics, the model predictive control (MPC) technique has been placed under the spotlight recently due to its great power to plan and stabilize complex dynamic motions [9], [10]. It usually involves solving a trajectory optimization (TO) problem in real-time, which determines the control sequences over a receding prediction horizon into the future. As a result, the overall system is able to act in advance. Another particular advantage of MPC is its capability of addressing various physical constraints, e.g., torque limit and contact friction, which is critical in practice but cannot be handled with the above-mentioned methods. Some MPC approaches to the TWIP system have been reported. [11] managed to balance the TWIP robot based on the linearly approximated model around the upright configuration. [12] was further able to perform trajectory tracking with feedback linearization. However, the framework is either too simple to explore robot full capability or unnecessarily complicated.

The descriptiveness of the dynamic model being used in MPC inevitably affects the system performance. On one hand, the sophisticated full-body dynamics suffers solving time issue. On the other hand, the oversimplified model may not be applicable to more complicated maneuvers. Lately, a novel idea has been proposed to explore the robot dynamics from a different perspective, i.e., *Dynamic Model Decomposition* (DMD) [13]. Instead of considering the entire robot as a whole, DMD decomposes the robot into the body and the rest, with interaction forces and moments connecting them. The effect is that the problem becomes more tractable since we can enjoy lower-dimensional dynamics for each subsystem while their composition is still equivalent to the full-order model. This approach has already been proven effective in MPC for dynamic legged locomotion. In order to investigate its generality, in this paper, we apply the same methodology to the TWIP robot. Specifically, we first decompose the robot into the body and the two wheels. Based on that, we design a corresponding MPC framework, which considers linearly approximated body dynamics with constrained wheel reaction forces as inputs. These forces need to respect the wheel kinematics and dynamics, actuator

[1] Junjie Shen and Dennis Hong are with the Robotics and Mechanisms Laboratory, the Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA 90095, USA {junjieshen, dennishong}@ucla.edu



(a) (b)

Fig. 1. The TWIP mobile robot is decomposed into (a) the body and (b) the two wheels, with interaction forces and moments connecting them.

TABLE I

TWIP Mobile Robot Parameters

Symbol	Parameter	Value [Unit]
m_b	Body mass	15 [kg]
m_w	Wheel mass	2.5 [kg]
I_{xx}	Body x principal moment of inertia	1.56 [kg·m²]
I_{yy}	Body y principal moment of inertia	1.26 [kg·m²]
I_{zz}	Body z principal moment of inertia	0.33 [kg·m²]
I_ψ	Body yaw moment of inertia	Varying
I_w	Wheel axle moment of inertia	0.013 [kg·m²]
W	Distance between two wheels	0.5 [m]
L	Distance between body and wheel axle	0.5 [m]
R	Wheel radius	0.1 [m]

capability, terrain condition, etc. The robot reference tracking TO problem can then be formulated into a favorable quadratic program (QP), which can be solved efficiently and thus implemented in real time.

The rest of this paper is organized as follows. Following DMD, Section II decomposes the TWIP mobile robot into the body and the two wheels. Section III elaborates the proposed locomotion MPC framework. To evaluate the performance, several simulations, e.g., station keeping, disturbance rejection, velocity tracking, and path following, were successfully tested and simulation results are discussed in Section IV. Section V concludes the paper with potential future work.

II. Robot Modeling

As shown in Fig. 1, a typical TWIP mobile robot has three main parts, two identical wheels and an intermediate body. The body involves a chassis carrying an actuator to drive each wheel. All the parameters are listed in Table I.

Assuming the two wheels are always rolling purely on the flat ground without slipping, the TWIP system has essentially three DoFs subject to the nonholonomic constraints. Let us define the vector of generalized coordinates as follows:

$$\boldsymbol{q} := \left[\phi, \theta_r, \theta_l\right]^\top, \tag{1}$$

where ϕ is the body pitch angle and θ_j is the rotation angle of the wheel with the index $j = r$ or l, indicating the wheel on the right or left. The equations of motion of the TWIP mobile robot take the form of

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{B}\boldsymbol{\tau}, \tag{2}$$

where $\boldsymbol{M}(\boldsymbol{q})$ stands for the inertia matrix, the vector $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ captures Coriolis and centrifugal forces, $\boldsymbol{F}(\dot{\boldsymbol{q}})$ represents the vector of viscous friction forces, $\boldsymbol{G}(\boldsymbol{q})$ describes the vector of gravitational forces, and the constant matrix $\boldsymbol{B}$ defines how the actuation torques $\boldsymbol{\tau} := \left[\tau_r, \tau_l\right]^\top$ enter the model.

The entire robot is further decomposed into three subsystems, the body and the two wheels, to be implemented in the proposed MPC framework via DMD.

A. Body Modeling

Considering negligible cornering forces [3], the free body diagram of the TWIP body can be developed subject to the interaction forces and moments with the wheels, as shown in Fig. 1a. Let us define the body state vector as

$$\boldsymbol{x}_b := \left[s_b, z_b, \psi, \phi, \dot{s}_b, \dot{z}_b, \dot{\psi}, \dot{\phi}\right]^\top, \tag{3}$$

where s_b is the straight-line position of the body center of mass (CoM), z_b describes the body CoM height, ψ is the yaw angle. Specifically, we have the following kinematic relationships involved:

$$\psi = \frac{R}{W}\left(\theta_r - \theta_l\right), \tag{4a}$$

$$z_b = L\cos\phi, \tag{4b}$$

$$s_b = \frac{R}{2}\left(\theta_r + \theta_l\right) + L\sin\phi, \tag{4c}$$

if we assume s_b, ψ, θ_r, and θ_l all start with zero. Define the vector of external forces and moments acted on the body as

$$\boldsymbol{u}_b := \left[h_r, h_l, v_r, v_l, \tau_r, \tau_l\right]^\top, \tag{5}$$

where h_j and v_j are the components of the interaction force in the horizontal and vertical directions with the wheel j, respectively. The body dynamics can thus be written as

$$m_b\ddot{s}_b = \mathcal{H}, \tag{6a}$$

$$m_b\ddot{z}_b = \mathcal{V} - m_b g, \tag{6b}$$

$$I_\psi\ddot{\psi} = \frac{W}{2}\left(h_r - h_l\right), \tag{6c}$$

$$I_{yy}\ddot{\phi} = \mathcal{V}L\sin\phi - \mathcal{H}L\cos\phi + \mathcal{T}, \tag{6d}$$

where $\mathcal{H} = h_r + h_l$, $\mathcal{V} = v_r + v_l$, $\mathcal{T} = \tau_r + \tau_l$, and g is the gravitational acceleration. Note that the body moment of inertia in the yaw direction I_ψ depends on the body pitch angle ϕ:

$$I_\psi\left(\phi\right) = I_{xx}\sin^2\phi + I_{zz}\cos^2\phi. \tag{7}$$

For the sake of MPC viability, the nonlinear rotational dynamics (6c) and (6d) is desired to be linearly simplified. Assuming ϕ will not change substantially under well-controlled locomotion, we can thus consider it as a constant, e.g., $\phi(t) = \phi(t_c) := \phi_c$, for a short period over the MPC prediction horizon, where ϕ_c is the body pitch angle at current time t_c, which will be updated for each MPC iteration and can be presumably estimated or directly measured. As a result, the body dynamics (6) can be approximated as a continuous-time linear time-invariant system as follows:

$$\dot{\boldsymbol{x}}_b = \boldsymbol{A}_b \boldsymbol{x}_b + \boldsymbol{B}_b \boldsymbol{u}_b + \boldsymbol{G}_b \tag{8}$$

for a short time horizon, where the matrix $\boldsymbol{B}_b$ depends on ϕ_c, i.e., $\boldsymbol{B}_b(\phi_c)$. (8) can be further discretized as

$$\boldsymbol{x}_b[k+1] = \boldsymbol{\mathcal{A}}_b \boldsymbol{x}_b[k] + \boldsymbol{\mathcal{B}}_b \boldsymbol{u}_b[k] + \boldsymbol{\mathcal{G}}_b \tag{9}$$

for MPC implementation later, where the matrices

$$\boldsymbol{\mathcal{A}}_b = e^{\boldsymbol{A}_b \Delta t}, \tag{10a}$$

$$\boldsymbol{\mathcal{B}}_b = \left(\int_0^{\Delta t} e^{\boldsymbol{A}_b \tau} \mathrm{d}\tau \right) \boldsymbol{B}_b, \tag{10b}$$

$$\boldsymbol{\mathcal{G}}_b = \left(\int_0^{\Delta t} e^{\boldsymbol{A}_b \tau} \mathrm{d}\tau \right) \boldsymbol{G}_b, \tag{10c}$$

the index $k \in \mathbb{N}$, and Δt is the sampling period.

B. Wheel Modeling

Based on the free body diagram of the wheel as shown in Fig. 1b, let us define the state vector of the wheel j as

$$\boldsymbol{x}_j := \left[s_j, z_j, \theta_j, \dot{s}_j, \dot{z}_j, \dot{\theta}_j \right]^\top, \tag{11}$$

where s_j is the straight-line position of the wheel CoM, z_j describes the wheel CoM height, and specifically, we have the following kinematic relationships involved:

$$z_j = 0, \tag{12a}$$

$$s_j = R\theta_j. \tag{12b}$$

Let us now define the vector of external forces and moments acted on the wheel j as

$$\boldsymbol{u}_j := \left[h_j, v_j, \tau_j, f_j, n_j \right]^\top, \tag{13}$$

where f_j and n_j are the components of the ground reaction force in the tangential and normal directions, respectively. The dynamics of the wheel j can thus be written as

$$m_w \ddot{s}_j = f_j - h_j, \tag{14a}$$

$$m_w \ddot{z}_j = n_j - v_j - m_w g, \tag{14b}$$

$$I_w \ddot{\theta}_j = -f_j R - \tau_j. \tag{14c}$$

(14) is linear and it can be easily converted into the state-space form as follows:

$$\dot{\boldsymbol{x}}_j = \boldsymbol{A}_j \boldsymbol{x}_j + \boldsymbol{B}_j \boldsymbol{u}_j + \boldsymbol{G}_j, \tag{15}$$

which can be further discretized as

$$\boldsymbol{x}_j[k+1] = \boldsymbol{\mathcal{A}}_j \boldsymbol{x}_j[k] + \boldsymbol{\mathcal{B}}_j \boldsymbol{u}_j[k] + \boldsymbol{\mathcal{G}}_j, \tag{16}$$

similar to (9) and (10).

III. Model Predictive Control

This section elaborates the proposed MPC framework for the TWIP mobile robot. Starting from the current state, the goal is to determine an optimal actuation torque control strategy over a finite time horizon while satisfying the constraints on the states and controls, so as to guide the robot body along the reference trajectory. We are able to formulate this TO problem into a QP, which can be solved efficiently and thus implemented in an MPC fashion.

A. Decision Variables

Given the prediction horizon T, the total number of time steps $N = 1 + T/\Delta t$, define the set of decision variables as

$$\boldsymbol{\chi} := \{ \boldsymbol{x}_b[N], \boldsymbol{x}_b[k], \boldsymbol{u}_b[k], \boldsymbol{x}_j[N], \boldsymbol{x}_j[k], f_j[k], n_j[k] \mid j = r, l,\ k = 1, \ldots, N-1 \}, \tag{17}$$

where $\boldsymbol{x}_b[k]$ and $\boldsymbol{u}_b[k]$ are the body state vector and corresponding external force input, while $\boldsymbol{x}_j[k]$, $f_j[k]$, and $n_j[k]$ are the state vector of the wheel j and corresponding ground reaction force components at the kth time step.

B. Cost Function

For robot reference tracking problem, the very common quadratic function

$$J = \sum_{k=1}^{N} \boldsymbol{e}_b^\top[k] \boldsymbol{Q}[k] \boldsymbol{e}_b[k] + \sum_{k=1}^{N-1} \boldsymbol{\tau}^\top[k] \boldsymbol{R}[k] \boldsymbol{\tau}[k] \tag{18}$$

is used, where the body error state $\boldsymbol{e}_b[k] := \boldsymbol{x}_b^{\text{ref}}[k] - \boldsymbol{x}_b[k]$ and $\boldsymbol{x}_b^{\text{ref}}[k]$ is the body state reference, while $\boldsymbol{Q}[k]$ and $\boldsymbol{R}[k]$ are the diagonal positive semi-definite weighting matrices. As a result, J will be minimized in terms of overall tracking errors and control efforts in the least-squares sense.

The direct application of whole-body controllers to non-minimum phase systems, in fact, can be problematic, as pointed out in [14]. For example, in terms of the TWIP mobile robot, in order to move forwards, the wheels first need to accelerate backwards to achieve a certain body pitch angle and should only then speed up forwards. In its standard form, however, the proposed MPC would fail to reproduce this behavior since it will not minimize the cost function (18). In order to overcome the issue, a similar strategy used in [15] is applied to regulate the wheel motion. In particular, the body pitch angle reference is automatically generated based on the difference between the reference and actual body straight-line positions and velocities:

$$\phi^{\text{ref}}[k] = k_p \left(s_b^{\text{ref}}[k] - s_b(t_c) - \dot{s}_b(t_c) \cdot (k-1)\Delta t \right) + k_d \left(\dot{s}_b^{\text{ref}}[k] - \dot{s}_b(t_c) \right) \tag{19}$$

for $k = 1, \ldots, N$, where k_p and k_d are the nonnegative proportional and derivative coefficients, respectively, which can be experimentally tuned. As a consequence, the overall system will be able to respond with the desired motion, e.g., first moving backwards for a certain body pitch angle and then driving forwards.

C. Constraints

Note that all the following constraints are linear in terms of the decision variables (17).

1) Initial Condition Constraint: The TWIP mobile robot body and wheel states at the first time step should coincide with the current measurements:

$$\boldsymbol{x}_b[1] = \boldsymbol{x}_b(t_c), \tag{20a}$$

$$\boldsymbol{x}_j[1] = \boldsymbol{x}_j(t_c), \tag{20b}$$

for $j = r, l$.

2) Dynamics Constraint: The TWIP mobile robot body and wheel states need to obey their corresponding system dynamics (9) and (16):

$$\boldsymbol{x}_b[k+1] = \boldsymbol{\mathcal{A}}_b \boldsymbol{x}_b[k] + \boldsymbol{\mathcal{B}}_b \boldsymbol{u}_b[k] + \boldsymbol{\mathcal{G}}_b, \tag{21a}$$

$$\boldsymbol{x}_j[k+1] = \boldsymbol{\mathcal{A}}_j \boldsymbol{x}_j[k] + \boldsymbol{\mathcal{B}}_j \boldsymbol{u}_j[k] + \boldsymbol{\mathcal{G}}_j, \tag{21b}$$

for $j = r, l$ and $k = 1, \ldots, N-1$.

3) Kinematics Constraint: Several kinematics constraints, in terms of both position and velocity, need to be imposed to limit as well as connect the body and wheel states. Since the wheels are assumed to always have contact with the flat ground, (12a) gives

$$z_j[k] = 0, \tag{22a}$$

$$\dot{z}_j[k] = 0, \tag{22b}$$

for $j = r, l$ and $k = 1, \ldots, N$.

The wheels are also supposed to have pure rolling motion without slipping, and thus (12b) suggests

$$s_j[k] = R\theta_j[k], \tag{22c}$$

$$\dot{s}_j[k] = R\dot{\theta}_j[k], \tag{22d}$$

for $j = r, l$ and $k = 1, \ldots, N$.

The body yaw angle and wheel rotation angles are connected by (4a), which induces

$$\psi[k] = \frac{R}{W}\left(\theta_r[k] - \theta_l[k]\right), \tag{22e}$$

$$\dot{\psi}[k] = \frac{R}{W}\left(\dot{\theta}_r[k] - \dot{\theta}_l[k]\right), \tag{22f}$$

for $k = 1, \ldots, N$.

The body CoM height and pitch angle are constrained by the nonlinear equation (4b). It can be optimally linearized at $\phi = \phi_c$ using OLQP [16], which results in

$$z_b[k] = L\cos\phi_c + a^*\left(\phi[k] - \phi_c\right), \tag{22g}$$

$$\dot{z}_b[k] = a^*\dot{\phi}[k], \tag{22h}$$

for $k = 1, \ldots, N$, where a^* is the optimal linear gain. Note that OLQP linear approximation works for a larger region of interest than the conventional Jacobian linearization method.

The body straight-line position and wheel rotation angles need to satisfy the nonlinear equation (4c). Similarly, it can be optimally linearized at $\phi = \phi_c$ via OLQP, which yields

$$s_b[k] = \frac{R}{2}(\theta_r[k] + \theta_l[k]) + L\sin\phi_c + b^*(\phi[k] - \phi_c), \tag{22i}$$

$$\dot{s}_b[k] = \frac{R}{2}\left(\dot{\theta}_r[k] + \dot{\theta}_l[k]\right) + b^*\dot{\phi}[k], \tag{22j}$$

for $k = 1, \ldots, N$, where b^* is the optimal linear gain.

Note that in practice, the velocity and position kinematics constraints are equivalent if we assume the current robot states for the first time step always satisfy both of them. As a result, only one of them needs to be imposed. Besides, according to our experience, the MPC can still work well without considering (22i) and (22j), which essentially connect the body and the two wheels kinematically. Although the MPC model would deviate from the actual one without them imposed, it is always correct at the first time step, thus the overall system is prevented from divergence.

4) Other Constraints: Several other constraints also need to be taken care of so as to meet the physical requirements. To prevent the wheels from slipping, the static friction and the unilateral normal force should satisfy

$$n_j[k] \geq 0, \tag{23a}$$

$$|f_j[k]| \leq \mu_s n_j[k] \Rightarrow -\mu_s n_j[k] \leq f_j[k] \leq \mu_s n_j[k], \tag{23b}$$

for $j = r, l$ and $k = 1, \ldots, N-1$, where μ_s is the coefficient of static friction between the wheel and ground. Note that (23a) is already embedded in (23b) and thus can be removed.

The actuator cannot exceed its capability in terms of actuation torque and velocity:

$$\tau_{\min} \leq \tau_j[k] \leq \tau_{\max} \tag{23c}$$

for $j = r, l$ and $k = 1, \ldots, N-1$, as well as

$$\omega_{\min} \leq \dot{\theta}_j[k] - \dot{\phi}[k] \leq \omega_{\max} \tag{23d}$$

for $j = r, l$ and $k = 1, \ldots, N$.

D. Complete Formulation

The complete QP formulation of the proposed MPC framework can thus be summarized as follows:

$$\begin{aligned} \underset{\mathcal{X}}{\text{minimize}} \quad & \textit{Cost Function } (18) \\ \text{subject to} \quad & \textit{Initial Condition Constraint } (20) \\ & \textit{Dynamics Constraint } (21) \\ & \textit{Kinematics Constraint } (22) \\ & \textit{Other Constraints } (23) \end{aligned} \tag{24}$$

The central idea of the proposed MPC framework is to regulate the robot body motion via real-time TO, which can be formulated into a favorable QP, with constrained wheel reaction forces and moments as inputs. These forces and moments need to respect the wheel kinematics and dynamics, actuator capability, as well as terrain condition.

IV. Simulation Results

The proposed MPC framework was implemented on the TWIP mobile robot introduced in Section II, simulated using MATLAB's *ode45* function with $g = 9.81$ m/s^2 and $\mu_s = 0.8$. The QP (24) was solved using MATLAB's *quadprog* function at a fixed conservative frequency of 100 Hz with $N = 6$, $\Delta t = 0.1$ s, $\tau_{\max} = -\tau_{\min} = 12$ N·m, and $\omega_{\max} = -\omega_{\min} = 35$ rad/s. Note that the weighting matrices in (18) might vary for different applications. The optimal solution of the actuation torques at the first time step, i.e., $\tau_r^*[1]$ and $\tau_l^*[1]$, was used directly on the robot. Several simulations were tested to evaluate the MPC performance, demonstrated in the video attachment.

A. Station Keeping & Disturbance Rejection

In this test, the robot was commanded to balance in place. As seen in Fig. 2, the robot was released slightly out of balance but was able to quickly correct its body orientation via its wheel motion. Later to gauge the overall system robustness in terms of disturbance rejection, starting from $t = 2$ s, an external force with a magnitude of 120 N and

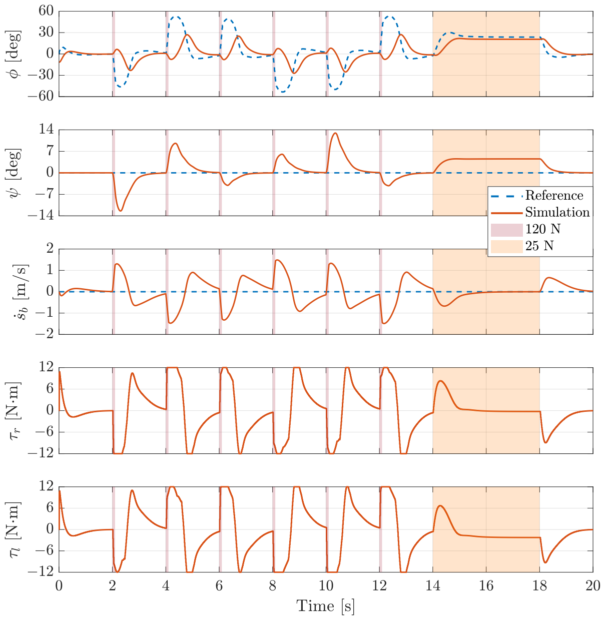


Fig. 2. Station keeping and disturbance rejection. The figure compares the robot reference and actual body pitch angles, yaw angles, and straight-line velocities, as well as shows the commanded actuation torques. The shaded areas indicate both the magnitude and duration of the disturbances.

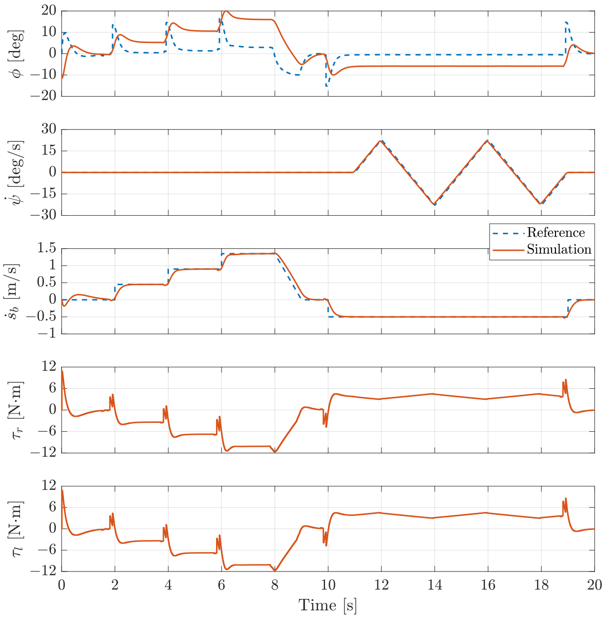


Fig. 4. Velocity tracking. The figure compares the robot reference and actual body pitch angles, yaw angle rates, and straight-line velocities, as well as shows the commanded actuation torques. Note that the steady-state pitch angle is larger than the reference in order to compensate the friction.

a duration of 100 ms was applied to the top of the body at some random location for every two seconds, until $t = 12$ s. The push was forceful enough to immediately speed up the body velocity over 1 m/s, but the robot was able to quickly regain its balance and recover within around two seconds. Finally, a continuous external force of 25 N was exerted on the body from $t = 14$ s to $t = 18$ s, the robot was still able to keep balance by leaning its body against the push, as shown in Fig. 3. Although the ground is considered to be always horizontal in the MPC model, the ability to resist a constant external disturbance indicates the overall system is also capable of operating on an inclined plane, where the tangential component of the gravitational force can be just treated as a constant external disturbance.

B. Velocity Tracking

In this test, the robot was commanded to track both body straight-line velocity as well as yaw angle rate. As seen in Fig. 4, at the beginning, the robot was commanded to track some straight-line velocity step reference by increment of 0.45 m/s up to 1.35 m/s at a spell of 2 s until it was ramped back down to zero from $t = 8$ s within one second. Later starting from $t = 11$ s, with a desired constant straight-line velocity of -0.5 m/s, the robot was further commanded to track some yaw angle rate triangular reference for a duration of 8 s. The robot was able to track the reference well and it is first interesting to observe that its body naturally bent in the moving direction even with a constant zero reference in order to fight against the friction. Furthermore, in response to (19), in order to move in some direction, the robot however started with a quick opposite movement, causing its body to lean towards the goal direction due to inertia, and then reversed its wheels and accelerated. Lastly, the nature of MPC was able to make the robot act in advance for an overall better tracking performance. Note that a velocity calibration needs to be performed in order to compensate for the steady-state error if the corresponding position reference is not considered.

C. Path Following

Having demonstrated the capability of reference tracking, we finally tested if the overall system was able to follow some predefined path. As seen in Fig. 5, a fairly tough path

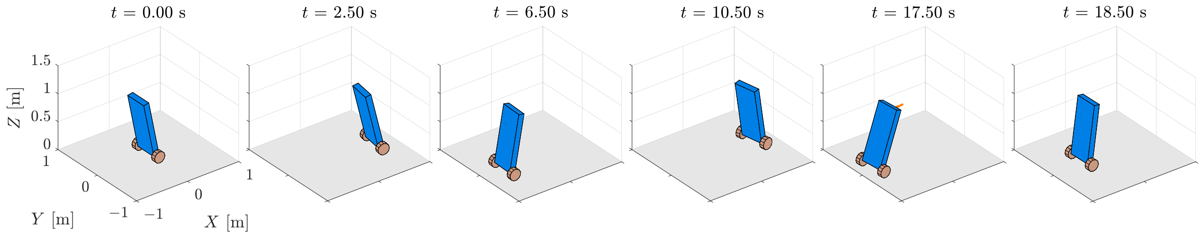


Fig. 3. Screenshots of station keeping and disturbance rejection.

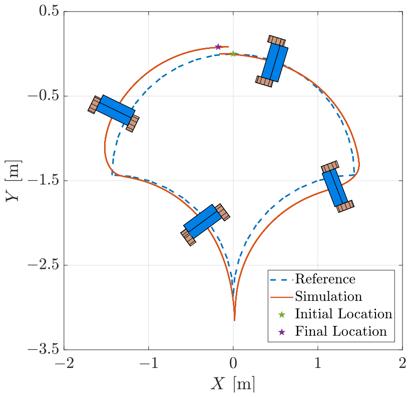


Fig. 5. Robot reference and actual paths with random screenshots.

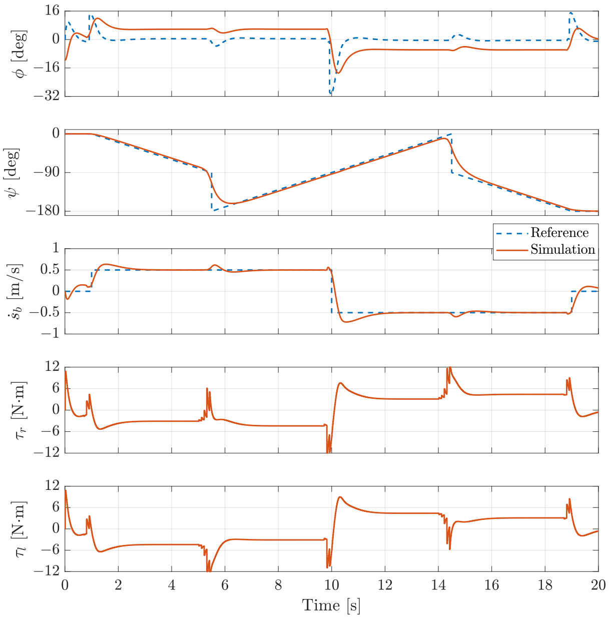


Fig. 6. Path following. The figure compares the robot reference and actual body pitch angles, yaw angles, and straight-line velocities, as well as shows the commanded actuation torques.

was interested, composed of four quarter circles with the same radius of 1.43 m. With a desired constant body straight-line speed of 0.5 m/s, this Cartesian path was then converted into the yaw angle reference, as shown in Fig. 6. The robot was commanded to follow the path autonomously starting from $t = 1$ s and the overall path following performance was considered good enough, in the light of the sharp corners in the path and thus the sudden changes in the reference. Note that there were no extrinsic sensors accounted so no robot location information was used.

V. Conclusion

In this paper, a novel model predictive control (MPC) framework is designed for the two-wheeled inverted pendulum (TWIP) mobile robot. Specifically, using *Dynamic Model Decomposition*, the TWIP mobile robot is decomposed into three lower-dimensional subsystems, the body and the two wheels with interaction forces and moments connecting them. Based on that, an MPC framework can be formed via quadratic programming, which considers linearly approximated body dynamics with constrained wheel reaction forces as inputs. The proposed methodology was successfully verified on a TWIP mobile robot in the simulation environment. The robot managed to resist external disturbances while balancing, track velocity reference, as well as follow some predefined path. The future work includes implementation on the real hardware platforms.

References

[1] K. Yamafuji and T. Kawamura, "Postural control of a monoaxial bicycle," *Journal of the Robotics Society of Japan*, vol. 7, no. 4, pp. 338–343, 1989.

[2] Y. Ha and S. Yuta, "Trajectory tracking control for navigation of self-contained mobile inverse pendulum," in *1994 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1875–1882, 1994.

[3] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "Joe: a mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107–114, 2002.

[4] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505–513, 2005.

[5] J. Huang, H. Wang, T. Matsuno, T. Fukuda, and K. Sekiyama, "Robust velocity sliding mode control of mobile wheeled inverted pendulum systems," in *2009 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2983–2988, 2009.

[6] C. Huang, W. Wang, and C. Chiu, "Design and implementation of fuzzy control on a two-wheel inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 2988–3001, 2011.

[7] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2004–2016, 2014.

[8] R. Cui, J. Guo, and Z. Mao, "Adaptive backstepping control of wheeled inverted pendulums models," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 501–511, 2015.

[9] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong, "Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[10] J. Shen and D. Hong, "Convex model predictive control of single rigid body model on so(3) for versatile dynamic legged motions," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[11] N. Minouchehr and S. K. Hosseini-Sani, "Design of model predictive control of two-wheeled inverted pendulum robot," in *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, pp. 456–462, 2015.

[12] M. Yue, C. An, and J. Sun, "An efficient model predictive control for trajectory tracking of wheeled inverted pendulum vehicles with various physical constraints," *International Journal of Control, Automation and Systems*, vol. 16, pp. 265–274, 2018.

[13] J. Shen and D. Hong, "A novel model predictive control framework using dynamic model decomposition applied to dynamic legged locomotion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4926–4932, 2021.

[14] V. Klemm, A. Morra, L. Gulich, D. Mannhart, D. Rohr, M. Kamel, Y. de Viragh, and R. Siegwart, "Lqr-assisted whole-body control of a wheeled bipedal robot with kinematic loops," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3745–3752, 2020.

[15] J. Shen and D. Hong, "Omburo: A novel unicycle robot with active omnidirectional wheel," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8237–8243, 2020.

[16] J. Shen and D. Hong, "Optimal linearization via quadratic programming," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4572–4579, 2020.