A Novel Model Predictive Control Framework Using Dynamic Model Decomposition Applied to Dynamic Legged Locomotion

Junjie Shen[1] and Dennis Hong[1]

***Abstract*— Dynamic locomotion for legged robots is difficult because the system dynamics are highly nonlinear and complex, nominally underactuated and unstable, multi-input and multi-output, as well as time-variant and hybrid. One usually faces the choice between the intricate full-body dynamics which remains computationally expensive and sometimes even intractable, and the empirically simplified model which inevitably limits the locomotion capability. In this paper, we explore the legged robot dynamics from a different perspective. By decomposing the robot into the body and the legs, with interaction forces and moments connecting them, we enjoy a novel method called *Dynamic Model Decomposition* that involves lower-dimensional dynamics for each subsystem while their composition maintaining the equivalence to the original 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 leg reaction forces as inputs. The overall methodology was successfully applied to a planar five-link biped robot. The simulation results show that the robot is capable of body reference tracking, push recovery, velocity tracking, and even blind locomotion on fairly rough terrain. This suggests a promising dynamic motion control scheme in the future.**

I. Introduction

Dynamic locomotion for legged robots is extremely difficult. First, the robot movement only results from the contact of the feet with the environment. These contact forces are strongly restricted and thus need to be carefully planned to achieve the desired motion. Second, the system dynamics are highly nonlinear and complex, nominally underactuated and unstable, multi-input and multi-output, as well as time-variant and hybrid [1]. Trajectory optimization (TO) has been proven an effective method yet the descriptiveness of the dynamic model being used inevitably affects the overall system performance.

At one end, in order to exploit every single detail of the robot, the sophisticated full-body dynamics has been utilized. This approach can indeed produce nice trajectories [2], [3], [4], [5], [6]. However, due to the complexity of high-dimensional model, these problems, usually formulated as a nonlinear program (NLP), are still computationally expensive, suffer initial guess and local minima issues, and sometimes even end up to be intractable. At the other end, there exist a variety of methods using the empirically simplified models, which only focus on the most salient aspect of the system dynamics, such as the linear inverted pendulum model [7]. The assumptions of flat terrain and zero angular momentum, along with zero-moment point substituting contact forces as the stability criteria [8], greatly simplify the dynamics so that model predictive control (MPC) can be implemented efficiently [9], [10], [11]. Nevertheless, the formulation requires variations on rough terrain [12]. Besides, the robot is forced to mimic the oversimplified models, which are not applicable to more complicated maneuvers.

Between these two extremes, another dynamic model commonly used in TO is the centroidal dynamics [13], which projects the robot dynamics at its center of mass (CoM) position and only considers the external interaction as input, provided that joint torque is always sufficient. A wide range of dynamic motions have been successfully demonstrated with this approach [14], [15], [16], [17], but real-time implementation remains open due to the nonlinearity of the angular momentum. By further assuming massless legs and small body angular velocity which is valid for certain quadrupeds, the centroidal dynamics can be linearly simplified and applied in an MPC fashion [18], [19], [20]. However, ignoring leg dynamics has drastic effects if the mass ratio of the body to the leg is not sufficiently high [21], not to mention humanoid robots with heavy limbs.

Another interesting line of research is dynamic decomposition. The idea is that, without sacrificing the model fidelity, the problem becomes more tractable with several subsystems of lower dimension. A 6-degree-of-freedom (DOF) manipulator was decomposed into two 3-DOF submodels with adaptive control successfully tested [22]. A wheel-leg robot was decomposed into the torso and the wheel-legs for whole-body control [23]. A quadrupedal robot was decomposed into two biped subsystems so as to rapidly optimize locomotion gaits even with NLP [24].

Inspired by the previous works, we explore the robot dynamics from a different perspective. We consider robot locomotion as the problem how to move the robot body from point A to point B with the help of its mobility mechanism interacting with the environment, e.g., propellers for drones, wheels for vehicles, legs for legged robots. Therefore, we think it is reasonable to decompose a legged robot into the body and the legs, 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 original full-order robot model. We call this philosophy of interpreting dynamics *Dynamic Model Decomposition* (DMD), which of course is not just applicable to legged robots. For example, we can decompose a wheeled inverted pendulum mobile robot into the body and the

[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

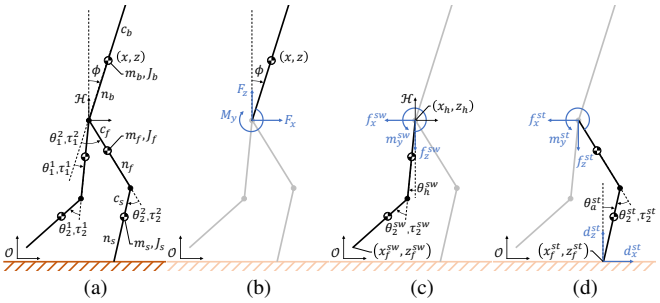


Fig. 1. Dynamic Model Decomposition. The (a) planar five-link biped robot is decomposed into three components, (b) body, (c) swing leg sw, and (d) stance leg st, with interaction forces and moments connecting them.

wheels; we can also separate the end effector from the rest of a manipulator. Based on that, we further propose a corresponding MPC framework. The goal is to regulate the body motion via real-time TO, which can be formulated into a favorable quadratic program (QP). Specifically, we consider linearly approximated body dynamics subject to constrained leg reaction forces as inputs. These forces need to respect the leg dynamics, actuator capability, terrain condition, etc.

We will demonstrate the proposed methodology for the locomotion of the planar five-link biped robot as an example. The rest of this paper is organized as follows. Using DMD, Section II decomposes the robot into three components, the body, the stance leg, and the swing leg. Section III details the locomotion MPC framework. To evaluate the performance, different experiments, e.g., body reference tracking, dynamic standing, push recovery, velocity tracking, blind locomotion on rough terrain, were successfully tested and simulation results are discussed in Section IV. Section V concludes the paper with potential future directions.

II. Robot Modeling

The robot of interest is a planar five-link biped robot with limp ankle, as shown in Fig. 1a. It has three main parts, the body and two identical legs. Each leg has two links, femur and shin. The parameters m_i, J_i, c_i, and n_i, with the index $i = b, f, s$, are the masses, moments of inertia, and CoM locations of the body, femur, and shin, respectively. Their corresponding values are listed in Table I, modeled after a 50th percentile adult male [25].

Define the vector of generalized coordinates

$$\boldsymbol{q} := \left[\boldsymbol{p}^\top, \left(\boldsymbol{\theta}^1\right)^\top, \left(\boldsymbol{\theta}^2\right)^\top\right]^\top = \left[x, z, \phi, \theta_1^1, \theta_2^1, \theta_1^2, \theta_2^2\right]^\top, \quad (1)$$

where $\boldsymbol{p} := [x, z, \phi]^\top$, $[x, z]^\top$ is the CoM position of the body, ϕ is the body pitch angle, and $\boldsymbol{\theta}^j := [\theta_1^j, \theta_2^j]^\top$ describes the configuration of leg j, with the index $j = 1, 2$. Note that the variables related to leg j are denoted by a succeeding j superscript. This j superscript may be replaced by st/sw if the variables are specifically referring to the stance/swing leg. The equations of motion of the robot take the form as

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C} = \boldsymbol{T} + \sum_{st \in \mathbb{T}} \left(\boldsymbol{J}^{st}\right)^\top \boldsymbol{d}^{st}, \quad (2)$$

where $\boldsymbol{M}(\boldsymbol{q})$ stands for the inertia matrix, the vector $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ captures the Coriolis, centrifugal, and gravitational forces,

TABLE I
Robot Parameters

i	body (b)	femur (f)	shin (s)
m_i [kg]	48	7.5	4.5
J_i [kg·m²]	9	0.47	0.19
c_i [m]	0.4	0.2	0.2
n_i [m]	0.4	0.3	0.3

the vector $\boldsymbol{T} := [0, 0, 0, \tau_1^1, \tau_2^1, \tau_1^2, \tau_2^2]^\top$ with $\boldsymbol{\tau}^j := [\tau_1^j, \tau_2^j]^\top$ gives the actuation torques, the stance foot Jacobian matrix $\boldsymbol{J}^{st}(\boldsymbol{q})$ transforms ground reaction forces $\boldsymbol{d}^{st} := [d_x^{st}, d_z^{st}]^\top$ into generalized forces for stance leg st, and $\mathbb{T}$ denotes the set of stance legs. In addition, a kinematics constraint is imposed to fix stance foot st on the ground before lift-off:

$$\boldsymbol{J}^{st}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}^{st}\dot{\boldsymbol{q}} = \boldsymbol{0}. \quad (3)$$

Finally, an impact takes place when the swing leg sw touches the ground. By assuming an impulsive and perfectly plastic collision, the touch-down impact model can be formulated according to [26] as

$$\boldsymbol{M}\left(\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^-\right) = \left(\boldsymbol{J}^{sw}\right)^\top \delta\boldsymbol{d}^{sw}, \quad (4a)$$

$$\begin{bmatrix} \boldsymbol{J}^{sw} & \boldsymbol{J}^{st} \end{bmatrix} \dot{\boldsymbol{q}}^+ = \boldsymbol{0}, \quad (4b)$$

where $\boldsymbol{J}^{sw}(\boldsymbol{q})$ is the swing foot Jacobian matrix, $\delta\boldsymbol{d}^{sw}$ is the touch-down impact force, and the superscript $+/-$ denotes the post/pre-impact generalized velocities.

Using DMD, the entire robot is further decomposed into three components, the body, the swing leg, and the stance leg, to be implemented in the proposed MPC framework.

A. Body Modeling

The robot body is modeled as a single rigid body subject to the forces and moments created by the leg dynamics, as shown in Fig. 1b. The body state vector is defined as

$$\boldsymbol{x} := \left[\boldsymbol{p}^\top, \dot{\boldsymbol{p}}^\top\right]^\top = \left[x, z, \phi, \dot{x}, \dot{z}, \dot{\phi}\right]^\top. \quad (5)$$

The net external wrench is defined as

$$\boldsymbol{u} := \left[F_x, F_z, M_y\right]^\top = \sum_{j=1}^{2} \left[f_x^j, f_z^j, m_y^j\right]^\top =: \sum_{j=1}^{2} \boldsymbol{\nu}^j, \quad (6)$$

where $\boldsymbol{f}^j := [f_x^j, f_z^j]^\top$ and m_y^j are the interaction force and moment exerted at the hip location with leg j. Note that

$$m_y^j = \tau_1^j \quad (7)$$

essentially since only the hip actuators are responsible for generating this pitch moment. The body dynamics can thus be written as

$$m_b\ddot{x} = F_x, \quad (8a)$$

$$m_b\ddot{z} = F_z - m_b g, \quad (8b)$$

$$J_b\ddot{\phi} = M_y - F_x n_b \cos\phi + F_z n_b \sin\phi, \quad (8c)$$

where g is the gravitational acceleration. For the sake of MPC viability, the nonlinear rotational dynamics (8c) is desired to be simplified. Different linear approximation approaches have been proven effective for 3D single rigid body model

[18], [19], [20], [27]. For the planar case, assuming ϕ will not change vigorously under well-controlled locomotion, we can thus consider it as a constant, e.g., $\phi(t) = \phi(t_c) := \phi_c$ in (8c), 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 presumably be either estimated or directly measured. As a result, the body dynamics (8) can be approximated as a continuous-time linear time-invariant system as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{g} \tag{9}$$

for a short time horizon, where the matrix $\boldsymbol{B}$ will depend on ϕ_c, i.e., $\boldsymbol{B}(\phi_c)$. (9) can be further discretized as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d\boldsymbol{u}_k + \boldsymbol{g}_d \tag{10}$$

with a fixed discrete time interval of Δt for MPC implementation later, where the index $k \in \mathbb{N}$.

B. Swing Leg Modeling

The swing leg sw is modeled from the hip to the foot, as shown in Fig. 1c. The goal is to control the swing foot to follow some trajectory. Thus, the hip is considered as the base. Define the vector of generalized coordinates

$$\boldsymbol{s}^{sw} := \left[\boldsymbol{p}_h^\top, (\boldsymbol{\theta}_s^{sw})^\top\right]^\top = \left[x_h, z_h, \theta_h^{sw}, \theta_2^{sw}\right]^\top, \tag{11}$$

where $\boldsymbol{p}_h := [x_h, z_h]^\top$ describes the hip position and $\boldsymbol{\theta}_s^{sw} := [\theta_h^{sw}, \theta_2^{sw}]^\top$ describes the leg configuration. The equations of motion of the swing leg take the form as

$$\underbrace{\begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{12}^\top & \boldsymbol{M}_{22} \end{bmatrix}}_{\boldsymbol{M}_s} \underbrace{\begin{bmatrix} \ddot{\boldsymbol{p}}_h \\ \ddot{\boldsymbol{\theta}}_s^{sw} \end{bmatrix}}_{\ddot{\boldsymbol{s}}^{sw}} + \underbrace{\begin{bmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \end{bmatrix}}_{\boldsymbol{C}_s} = \begin{bmatrix} -\boldsymbol{f}^{sw} \\ \boldsymbol{\tau}^{sw} \end{bmatrix}, \tag{12}$$

where $\boldsymbol{M}_s(\boldsymbol{s}^{sw})$ gives the inertia matrix and $\boldsymbol{C}_s(\boldsymbol{s}^{sw}, \dot{\boldsymbol{s}}^{sw})$ captures the Coriolis, centrifugal, and gravitational forces. (12) can be further decomposed as

$$\boldsymbol{J}_s^\top \boldsymbol{\Lambda}_{12}\left({}^{\mathcal{H}}\ddot{\boldsymbol{p}}_f^{sw} - \dot{\boldsymbol{J}}_s\dot{\boldsymbol{\theta}}_s^{sw}\right) + \boldsymbol{M}_{11}\ddot{\boldsymbol{p}}_h + \boldsymbol{C}_1 = -\boldsymbol{f}^{sw}, \tag{13a}$$

$$\boldsymbol{J}_s^\top \boldsymbol{\Lambda}_{22}\left({}^{\mathcal{H}}\ddot{\boldsymbol{p}}_f^{sw} - \dot{\boldsymbol{J}}_s\dot{\boldsymbol{\theta}}_s^{sw}\right) + \boldsymbol{M}_{12}^\top\ddot{\boldsymbol{p}}_h + \boldsymbol{C}_2 = \boldsymbol{\tau}^{sw}, \tag{13b}$$

in operational space, where

$$\boldsymbol{\Lambda}_{12} = \left(\boldsymbol{J}_s\boldsymbol{M}_{12}^{-1}\boldsymbol{J}_s^\top\right)^{-1}, \ \boldsymbol{\Lambda}_{22} = \left(\boldsymbol{J}_s\boldsymbol{M}_{22}^{-1}\boldsymbol{J}_s^\top\right)^{-1}, \tag{13c}$$

are the operational space inertia matrices, $\boldsymbol{p}_f^{sw}$ denotes the swing foot position, and $\boldsymbol{J}_s(\boldsymbol{\theta}_s^{sw})$ is the Jacobian matrix for the swing foot, i.e., $\boldsymbol{J}_s\dot{\boldsymbol{\theta}}_s^{sw} = {}^{\mathcal{H}}\dot{\boldsymbol{p}}_f^{sw}$. Note that the variables with respect to the hip frame are denoted by a preceding $\mathcal{H}$ superscript while those without it are relative to the inertia frame $\mathcal{O}$.

C. Stance Leg Modeling

The stance leg st is modeled from the foot to the hip, as shown in Fig. 1d. The goal is to find the relationship between the actuation torques and the desired interaction force as well as moment with the body. Therefore, the foot is treated as the base. Define the vector of generalized coordinates

$$\boldsymbol{r}^{st} := \left[\left(\boldsymbol{p}_f^{st}\right)^\top, \left(\boldsymbol{\theta}_r^{st}\right)^\top\right]^\top = \left[x_f^{st}, z_f^{st}, \theta_a^{st}, \theta_2^{st}\right]^\top, \tag{14}$$

where $\boldsymbol{p}_f^{st} := [x_f^{st}, z_f^{st}]^\top$ describes the foot position and $\boldsymbol{\theta}_r^{st} := [\theta_a^{st}, \theta_2^{st}]^\top$ describes the leg configuration. The equations of motion of the stance leg take the form as

$$\boldsymbol{M}_r\ddot{\boldsymbol{r}}^{st} + \boldsymbol{C}_r = \boldsymbol{T}_r + \boldsymbol{J}_d^\top\boldsymbol{d}^{st} + \boldsymbol{J}_\nu^\top\left(-\boldsymbol{\nu}^{st}\right), \tag{15}$$

where $\boldsymbol{M}_r(\boldsymbol{r}^{st})$ is the inertia matrix, the vector $\boldsymbol{C}_r(\boldsymbol{r}^{st}, \dot{\boldsymbol{r}}^{st})$ captures the Coriolis, centrifugal, and gravitational forces, the vector $\boldsymbol{T}_r := [0, 0, 0, \tau_2^{st}]^\top$ gives the actuation torques, the Jocabian matrices $\boldsymbol{J}_d$ and $\boldsymbol{J}_\nu(\boldsymbol{r}^{st})$ transform ground reaction force $\boldsymbol{d}^{st}$ and body reaction force $-\boldsymbol{\nu}^{st}$ into generalized forces, respectively. Note that $\boldsymbol{J}_d$ is a constant matrix. In addition, we have the kinematics constraint

$$\ddot{\boldsymbol{p}}_f^{st} = \boldsymbol{J}_d\ddot{\boldsymbol{r}}^{st} + \cancel{\dot{\boldsymbol{J}}_d}\dot{\boldsymbol{r}}^{st} = \boldsymbol{0} \tag{16}$$

imposed to fix the stance foot on the ground before lift-off.

III. Control Framework

This section illustrates the proposed MPC framework for the planar five-link biped robot walking on flat ground.

A. Walking State Machine

A simple walking state machine is implemented for each leg in order for proper transition between swing leg control and stance leg control modes. Our locomotion strategy utilizes a combination of alternating single support phase (SSP, i.e., one stance leg in contact with the ground and one swing leg) and double support phase (DSP, i.e., two stance legs in contact with the ground). Assuming the information of the entire robot can be accessed, the transition from the swing state to the stance state is made simply when the leg touch-down is detected; on the other hand, the transition from the stance state to the swing state happens when the nominal duration of DSP T_{ds} runs out or the leg lift-off is detected before the end of T_{ds}.

B. Swing Leg Control

When the stance leg transitions to the swing state, swing leg control mode is thus applied to move the leg from the current footstep location $\boldsymbol{p}_{f,n}^{sw} := [x_{f,n}^{sw}, z_{f,n}^{sw}]^\top$ to the desired next footstep location $\boldsymbol{p}_{f,d}^{sw} := [x_{f,d}^{sw}, z_{f,d}^{sw}]^\top$.

1) Footstep Decision: This desired footstep location $\boldsymbol{p}_{f,d}^{sw}$ is essentially responsible for determining the velocity that the robot will travel at. In particular, it is projected onto an assumed horizontal ground plane, i.e., $z_{f,n}^{sw} = z_{f,d}^{sw}$, based on the current footstep location of the stance leg $\boldsymbol{p}_{f,n}^{st} := [x_{f,n}^{st}, z_{f,n}^{st}]^\top$, similar to the technique used in [28], [29], which gives in the x direction

$$x_{f,d}^{sw} = x_{f,n}^{st} + v_d T_s + k_1\left(\dot{x}_c - v_d\right) + k_2\left(\dot{x}_c - \dot{x}_{-1}\right), \tag{17}$$

where $\boldsymbol{p}_c := [x_c, z_c]^\top$ is the robot CoM position, v_d is the desired robot forward velocity, $\dot{x}_{-1}$ is the CoM touch-down speed at the previous step, k_1 and k_2 are two positive gains determining the contribution of each term, and T_s is the nominal duration of stance phase, i.e., $T_s = T_{ss} + T_{ds}$ with T_{ss} the nominal duration of SSP. To have a better performance in terms of stabilization, e.g., overcoming velocity perturbation against pushing, the desired next footstep location is constantly updated after mid-swing.

2) Trajectory Generation: Cycloidal functions, with nice properties including easy implementation, continuous derivatives, and slow accelerations at the beginning and end [21], are used to generate swing foot reference trajectory $\boldsymbol{p}^{sw}_{f,ref}$.

3) Operational Space Control: The Cartesian swing leg reference trajectory is tracked using an operational space formulation [30]. Specifically, it is a PD feedback controller with a feedforward reference based on the swing leg dynamics (13b), which gives the control law

$$\boldsymbol{\tau}^{sw}_c = \boldsymbol{J}^\top_s \boldsymbol{\Lambda}_{22} \left(\boldsymbol{y}_{ff} + \boldsymbol{y}_{fb} - \dot{\boldsymbol{J}}_s \dot{\boldsymbol{\theta}}^{sw}_s \right) + \boldsymbol{C}_2, \tag{19a}$$

$$\boldsymbol{y}_{ff} = {}^{\mathcal{H}}\ddot{\boldsymbol{p}}^{sw}_{f,ref}, \tag{19b}$$

$$\boldsymbol{y}_{fb} = \boldsymbol{K}_p \big({}^{\mathcal{H}}\boldsymbol{p}^{sw}_{f,ref} - {}^{\mathcal{H}}\boldsymbol{p}^{sw}_{f} \big) + \boldsymbol{K}_d \big({}^{\mathcal{H}}\dot{\boldsymbol{p}}^{sw}_{f,ref} - {}^{\mathcal{H}}\dot{\boldsymbol{p}}^{sw}_{f} \big), \tag{19c}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are diagonal positive definite proportional and derivative gain matrices. Note that the contribution of the hip acceleration is assumed small under well-controlled locomotion and thus the term $\boldsymbol{M}^\top_{12}\ddot{\boldsymbol{p}}_h$ is neglected here.

4) Reaction Force Prediction: At the beginning of each MPC iteration, given the swing foot reference trajectory and control law, we can then simulate the swing leg dynamics and predict its resultant reaction force and moment on the body using (7) and (13), which gives

$$\widehat{\boldsymbol{f}}^{sw} = \widehat{\boldsymbol{J}}^\top_s \widehat{\boldsymbol{\Lambda}}_{12} \left(\widehat{\dot{\boldsymbol{J}}}_s \widehat{\dot{\boldsymbol{\theta}}}^{sw}_s - \boldsymbol{y}_{ff} \right) - \widehat{\boldsymbol{C}}_1, \tag{20a}$$

$$\widehat{m}^{sw}_y = \begin{bmatrix} 1 & 0 \end{bmatrix} \widehat{\boldsymbol{\tau}}^{sw}_c, \tag{20b}$$

$$\widehat{\boldsymbol{\tau}}^{sw}_c = \widehat{\boldsymbol{J}}^\top_s \widehat{\boldsymbol{\Lambda}}_{22} \left(\boldsymbol{y}_{ff} + \widehat{\boldsymbol{y}}_{fb} - \widehat{\dot{\boldsymbol{J}}}_s \widehat{\dot{\boldsymbol{\theta}}}^{sw}_s \right) + \widehat{\boldsymbol{C}}_2, \tag{20c}$$

where the hat notation indicates the variables are simulation results. Note that the hip acceleration terms are ignored here as well. These approximations will be later taken into consideration for stance leg control during SSP, which is critical and necessary when the swing leg dynamics cannot be neglected, nor simply treated as disturbances.

C. Stance Leg Control

The control of stance leg, either during SSP or DSP, is formulated as a mathematical TO problem, in light of the robot model and physical constraints. Starting from the current state, the goal is to determine an optimal joint 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. Since we can formulate this TO problem into a QP, which can be solved efficiently, it can be implemented in an MPC fashion.

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

$$\boldsymbol{\chi} = \{ \boldsymbol{x}_N, \boldsymbol{x}_k, \boldsymbol{\nu}^j_k, \boldsymbol{d}^j_k, \boldsymbol{\tau}^j_k, \ddot{\boldsymbol{r}}^j_k \,|\, j = 1, 2, \; k = 1, \dots, N-1 \}, \tag{21}$$

where the body states $\boldsymbol{x}_k = [\boldsymbol{p}^\top_k, \dot{\boldsymbol{p}}^\top_k]^\top$ from (5), the interaction forces and moments $\boldsymbol{\nu}^j_k = [f^j_{x,k}, f^j_{z,k}, m^j_{y,k}]^\top$ from (6), the ground reaction forces $\boldsymbol{d}^j_k$ from (2), the actuation torques $\boldsymbol{\tau}^j_k = [\tau^j_{1,k}, \tau^j_{2,k}]^\top$ from (2), and the generalized accelerations $\ddot{\boldsymbol{r}}^j_k$ from (15). This would be the most complete situation for our MPC framework. Nevertheless, most of the time, many decision variables turn out to be irrelevant:

(a) During SSP, it is clear that the decision variables related to the swing leg sw can be removed, except for $\boldsymbol{\nu}^{sw}_k$ and $\tau^{sw}_{1,k}$, which will be explained later.

(b) During DSP, since we have a fixed nominal duration of DSP T_{ds}, we can actually take full advantage of MPC essence, i.e., keeping future time slots in account so as to act in advance. Specifically, supposing the stance leg $st = 1$ is about to swing, given the remaining time $T_r \leq T$, the decision variables related to the stance leg 1 can be treated just like a swing leg as the previous condition, but only after time step $k = N_r := [T_r/\Delta t]$. The effect is that the stance leg 1 would naturally push off the ground before starting to swing, thus enhancing the locomotion performance. Essentially, the stance leg 1 contributes to the body control during DSP but becomes a burden afterwards during SSP, so the system is able to take it into consideration and prepare ahead of time.

Note that other variables involved later, if not directly related to the predefined body reference $\boldsymbol{x}_{ref}$, are simply considered constants throughout the entire prediction horizon T, which are actually updated at the beginning of each MPC iteration. The effect is that the robot model will always be correct for the first time step, preventing it from divergence due to this rough approximation.

2) Cost Function: For reference tracking, the very common quadratic function

$$J = \sum_{k=1}^{N} \boldsymbol{e}^\top_k \boldsymbol{Q}_k \boldsymbol{e}_k + \sum_{k=1}^{N-1} \begin{bmatrix} \boldsymbol{\tau}^1_k \\ \boldsymbol{\tau}^2_k \\ \ddot{\boldsymbol{r}}^1_k \\ \ddot{\boldsymbol{r}}^2_k \end{bmatrix}^\top \boldsymbol{R}_k \begin{bmatrix} \boldsymbol{\tau}^1_k \\ \boldsymbol{\tau}^2_k \\ \ddot{\boldsymbol{r}}^1_k \\ \ddot{\boldsymbol{r}}^2_k \end{bmatrix} \tag{22}$$

is selected, where $\boldsymbol{e}_k = \boldsymbol{x}_{ref,k} - \boldsymbol{x}_k$ while $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the weighting matrices. As a result, J will be minimized in terms of overall tracking errors and control efforts in the least-squares sense. In addition, the generalized accelerations are also penalized to avoid aggressive leg motions.

3) Constraints: All the following constraints are linear in terms of the decision variables (21).

(a) *Initial Condition Constraint*
The robot body states for the very first time step should coincide with the current measurements:

$$\boldsymbol{x}_1 = \boldsymbol{x}(t_c). \tag{23}$$

(b) *Body Dynamics Constraint*
The robot body states need to obey its system dynamics (10) subject to the external wrench:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d \boldsymbol{x}_k + \boldsymbol{B}_d \sum_{j=1}^{2} \boldsymbol{\nu}^j_k + \boldsymbol{g}_d. \tag{24a}$$

Meanwhile, we also need to respect (7):

$$m^j_{y,k} = \tau^j_{1,k}. \tag{24b}$$

Note that (24) are applied for leg $j = 1, 2$ with time step $k = 1, \dots, N-1$.

(c) *Stance Leg Constraint*

For stance leg st, the body reaction forces, joint torques, and accelerations need to first satisfy the stance leg dynamics (15):

$$\boldsymbol{M}_r \ddot{\boldsymbol{r}}_k^{st} + \boldsymbol{C}_r = \boldsymbol{T}_{r,k} + \boldsymbol{J}_d^\top \boldsymbol{d}_k^{st} - \boldsymbol{J}_\nu^\top \boldsymbol{\nu}_k^{st}, \tag{25a}$$

as well as the kinematics constraint (16):

$$\boldsymbol{J}_d \ddot{\boldsymbol{r}}_k^{st} = \boldsymbol{0}. \tag{25b}$$

Meanwhile, the ground reaction forces need to be within the friction cone to prevent slippage:

$$\begin{bmatrix} 0 & -1 \\ 1 & -\mu \\ -1 & -\mu \end{bmatrix} \boldsymbol{d}_k^{st} \leq \boldsymbol{0}, \tag{25c}$$

where μ is the ground friction coefficient. Also, the joint actuator cannot exceed its capability in terms of actuation torque and velocity:

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau}_k^{st} \leq \boldsymbol{\tau}_{\max}, \tag{25d}$$

$$\dot{\boldsymbol{\theta}}_{\min} \leq \boldsymbol{J}_b^{-1} \dot{\boldsymbol{p}}_k \leq \dot{\boldsymbol{\theta}}_{\max}, \tag{25e}$$

where $\boldsymbol{J}_b(\boldsymbol{\theta}_b^{st})$ is the body Jacobian matrix, i.e., $\boldsymbol{J}_b \dot{\boldsymbol{\theta}}_b^{st} = \dot{\boldsymbol{p}}$, $\boldsymbol{\theta}_b^{st} := [\theta_a^{st}, \theta_2^{st}, \theta_1^{st}]^\top$. This Jacobian matrix is invertible as long as the stance leg configuration is non-singular. Lastly, since the stance ankle is limp, i.e., zero joint actuation torque, the interaction forces between the body and stance leg are limited by

$$\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{J}_\nu^\top \boldsymbol{\nu}_k^{st} = 0. \tag{25f}$$

Note that (25a-f) are applied for stance leg $st = 1$ with time step $k = 1, \ldots, N_r$ if stance leg 1 is about to swing after N_r, while for the other stance leg $st = 2$ with time step $k = 1, \ldots, N-1$. In addition, for stance leg 1 with time step $k = N_r+1, \ldots, N-1$, it will become a burden to the body and the leg reaction forces are set to

$$\boldsymbol{\nu}_k^1 = \begin{bmatrix} 0 & (m_f + m_s)\, g & 0 \end{bmatrix}^\top \tag{25g}$$

in consideration of the leg weight only, but good enough for the system to act in advance.

(d) *Swing Leg Constraint*

The previous stance leg constraint needs to be applied either during SSP or DSP. In particular, during SSP, the decision variables related to the swing leg sw can be removed except for $\tau_{1,k}^{sw}$ subject to (24b) and the swing leg reaction forces, which can be predicted based on the simulation results from (20):

$$\boldsymbol{\nu}_k^{sw} = \begin{bmatrix} \left(\widehat{\boldsymbol{f}}_k^{sw}\right)^\top & \widehat{m}_{y,k}^{sw} \end{bmatrix}^\top. \tag{26}$$

Note that (26) is applied for swing leg sw with time step $k = 1, \ldots, N-1$.

D. Summary

The central idea of the proposed MPC framework is to regulate the body motion via real-time TO, which can be formulated into a QP, with stance leg reaction forces as control inputs while swing leg reaction forces as forced inputs. The stance leg reaction forces need to respect its leg dynamics, actuator capability, and terrain condition, whereas the swing leg reaction forces can be predicted based on the leg dynamics, swing trajectory, and control law.

IV. Simulation Results

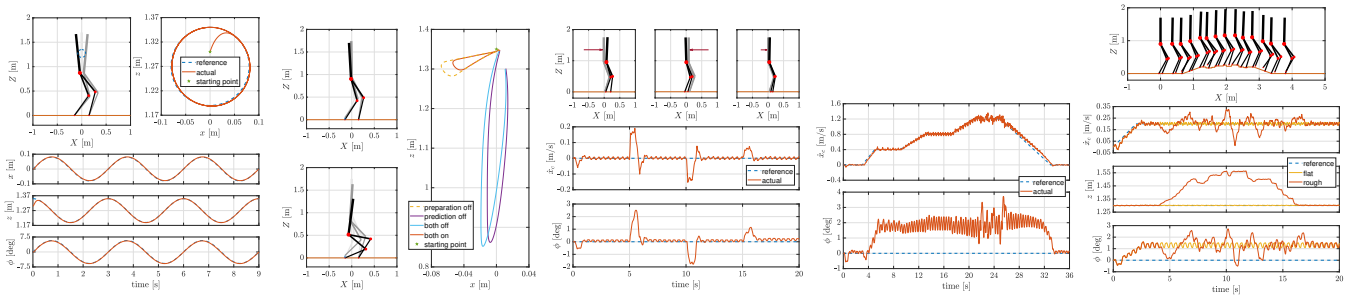
The proposed MPC framework was implemented on the planar five-link biped robot introduced in Section II, simulated using MATLAB's *ode45* function, with $g = 9.81$ m/s^2 and $\mu = 0.8$. The swing leg reaction force prediction (20) was carried out based on forward Euler method. The QP for the stance leg MPC was solved using MATLAB's *quadprog* function at a fixed frequency of 100 Hz. The optimal joint torque solution of the first time step is used directly on the robot. Different experiments were tested to evaluate the MPC performance, demonstrated in the video attachment.

A. Static Standing & Body Reference Tracking

Without considering SSP, the robot is able to stand still by tracking a constant body reference trajectory with one foot on each side of the CoM, thus called static standing. Note that the robot would be guaranteed stable as long as the friction cone constraint (25c) was satisfied [8]. To go further, we demonstrated in Fig. 2a that the proposed MPC framework can also be applied in the field of manipulators. Starting with static standing, the robot body was commanded to track sinusoidal references $\boldsymbol{x}_{ref} := [\boldsymbol{p}_{ref}, \dot{\boldsymbol{p}}_{ref}]^\top$ with $\boldsymbol{p}_{ref}(t) = [c_x + r\sin(wt), c_z + r\cos(wt), A\sin(wt)]^\top$, where the angular frequency $w = 2\pi/3$ rad/s, the radius $r = 0.08$ m, the center $[c_x, c_z]^\top = [0, 1.27]^\top$ m, and the pitch amplitude $A = 0.1$ rad $= 18/\pi$ deg. With larger weights penalizing position tracking errors, the robot body was able to tack the sinusoidal references pretty well.

B. Effect of Swing Preparation & Swing Force Prediction

As previously mentioned, there are two main points worthy of attention in our MPC framework, i.e., swing preparation during DSP and swing leg reaction force prediction during SSP. With the former, the system is able to act in advance before a stance leg starts to swing; with the latter, the system is able to compensate the burden of swing leg dynamics. To evaluate their effects, we compared the results with and without involving (25g) and (26) in the QP, as shown in Fig. 2b. Starting with static standing (slightly off the equilibrium), the robot was commanded to lift its left foot off the ground for a small portion and then get back, with the swing foot reference trajectory in blue. The simulation results indicate that the swing force prediction significantly improves body control, while swing preparation, though not as much as the former, still helps a lot. Notably, with swing preparation on, the stance leg naturally pushed off the ground before starting to swing and the body moved upward a little



(a) (b) (c) (d) (e)

Fig. 2. Simulation results. (a) Static standing and body reference tracking. The top left shows the screenshots at initial (light gray) and two random instants (dark gray and black), with the body reference in blue. The top right compares the body reference and actual trajectories. The bottom shows the time histories of the body position and pitch angle. (b) Effect of swing preparation and swing leg reaction force prediction. The right compares the body trajectories. The top left shows the screenshots at initial (light gray), mid-swing (dark gray), and touch-down (black) with both on. The bottom left shows the screenshots with both off. (c) Dynamic standing and push recovery. The top shows the screenshots right before each push (dark gray) and right after finishing two steps (black). The arrow indicates direction and magnitude of each push. The bottom shows the time histories of the robot CoM forward velocity and body pitch angle. (d) Velocity tracking on flat ground. The figure compares the robot reference and actual CoM forward velocities as well as body pitch angles. (e) Blind locomotion on rough ground. The top shows the screenshots on rough ground at random instants. The rest compare the robot reference and actual CoM forward velocities, body heights, and body pitch angles on both flat and rough grounds.

bit accordingly. As a result, the body trajectory ended up with an overall smaller deviation from the starting point.

C. Dynamic Standing & Push Recovery

For static standing, the robot is always standing still in DSP. Via alternating DSP and SSP, the robot is also able to hold position by stepping in place with $v_d = 0$ m/s, called dynamic standing. As seen in Fig. 2c, starting with static standing and then commanded to switch to dynamic standing, the robot was able to keep its balance by adjusting its footstep locations, and quickly converge to the steady state. Later, to gauge the overall system robustness in terms of disturbance rejection, a push recovery test was conducted. Specifically, an external force of 120 N was applied to the body for 100 ms in the positive X direction at $t = 5$ s and another one with the same magnitude in the negative X direction at $t = 10$ s. The push was forceful enough to immediately accelerate the robot CoM over 0.15 m/s, but the robot was able to recover within the following 2 s. Finally, a continuous external force of 20 N was exerted on the body in the positive X direction right after $t = 15$ s and the robot was still able to adapt to it by moving the footstep locations ahead of robot CoM, so that the centroidal angular momentum was not evolving.

D. Velocity Tracking on Flat Ground

As aforementioned, velocity tracking with varying v_d is done mainly through footstep placement (17). As seen in Fig. 2d, on flat ground, starting with static standing, the robot was initially switched to dynamic standing, and then it was commanded to track velocity reference by increment of 0.4 m/s up to 1.2 m/s, until it was ramped back down to 0 m/s. With a smooth change in commanded velocity, the robot was able to track the reference well. It is interesting to observe that the body naturally bent in the moving direction even with a zero reference against ground friction. Furthermore, for constant v_d on flat ground, the system showed convergence to some limit cycle attractor, while it became more and more chaotic when it was reaching the limit, i.e., maximum feasible forward velocity around 1.2 m/s.

E. Blind Locomotion on Rough Ground

Finally, to demonstrate the overall system robustness in terms of terrain rough condition, the robot was tested on irregular ground without any perception information. As seen in Fig. 2e, starting with static standing on flat ground, the robot was immediately commanded to speed up to 0.2 m/s in 2 s. Later, it succeeded in first climbing up a slope of 20 deg, then walking over an uneven terrain with ground height rapidly changing, and afterwards going downhill with a slope of -20 deg, before walking on flat ground again. Note that in order to adapt to the ground height difference, the apex height for the swing foot trajectory was intentionally increased; the body height was intentionally lowered to avoid leg singularity; swing foot collision was not considered here; if the swing foot was still in the air after $1.1T_{ss}$ in case of unexpected gaps, it was simply commanded to step down as fast as possible.

V. Conclusion

The descriptiveness of the robot dynamic model being used in a model predictive control (MPC) framework inevitably affects the overall system performance. Instead of the full-body dynamics which remains computationally expensive, or the empirically simplified model which limits the capability, we interpret the robot dynamics from a different perspective. By decomposing a legged robot into the body and the legs, with interaction forces and moments connecting them, we enjoy a novel method called *Dynamic Model Decomposition* that involves lower-dimensional dynamics for each subsystem while their composition maintaining the equivalence to the original full-order model. Based on that, we further designed a corresponding MPC framework, which considers linearly approximated body dynamics with constrained leg reaction forces as inputs. The proposed methodology was successfully applied to the dynamic locomotion of a planar five-link biped robot. We are looking forward to generalizing it to three-dimensional scenarios as well as applying it to different kinds and applications of robots.

REFERENCES

[1] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[2] K. Mombaur, "Using optimization to create self-stable human-like running," *Robotica*, vol. 27, no. 3, p. 321–330, 2009.

[3] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[4] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3346–3351, 2015.

[5] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[6] J. Shen, Y. Liu, X. Zhang, and D. Hong, "Optimized jumping of an articulated robotic leg," in *2020 17th International Conference on Ubiquitous Robots (UR)*, pp. 205–212, 2020.

[7] S. Kajita, K. Tani, and A. Kobayashi, "Dynamic walk control of a biped robot along the potential energy conserving orbit," in *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pp. 789–794 vol.2, 1990.

[8] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots - adios zmp," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 1976–1983, 2006.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, pp. 1620–1626 vol.2, 2003.

[10] P. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 137–142, 2006.

[11] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[12] J. Hooks and D. Hong, "Implementation of a versatile 3d zmp trajectory optimization algorithm on a multi-modal legged robotic platform," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3777–3782, 2018.

[13] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 10 2013.

[14] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 295–302, 2014.

[15] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Auton. Robots*, vol. 40, p. 429–455, Mar. 2016.

[16] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, "A versatile and efficient pattern generator for generalized legged locomotion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3555–3561, 2016.

[17] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[18] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4102–4109, 2017.

[19] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252, 2018.

[20] Y. Ding, A. Pandala, and H. Park, "Real-time model predictive control for versatile dynamic motions in quadrupedal robots," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8484–8490, 2019.

[21] J. R. Hooks, *Real-Time Optimization for Control of a Multi-Modal Legged Robotic System*. PhD thesis, UCLA, 2019.

[22] S. Tzafestas, G. Stavrakakis, and A. Zagorianos, "Dynamic decomposition and adaptive control of a six-degrees-of-freedom robotic manipulator," *IFAC Proceedings Volumes*, vol. 20, no. 5, Part 3, pp. 145 – 150, 1987. 10th Triennial IFAC Congress on Automatic Control - 1987 Volume III, Munich, Germany, 27-31 July.

[23] Y. Xin, H. Chai, Y. Li, X. Rong, B. Li, and Y. Li, "Speed and acceleration control for a two wheel-leg robot based on distributed dynamic model and whole-body control," *IEEE Access*, vol. 7, pp. 180630–180639, 2019.

[24] W. L. Ma and A. D. Ames, "From bipedal walking to quadrupedal locomotion: Full-body dynamics decomposition for rapid gait generation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4491–4497, 2020.

[25] P. M. Wensing, *Optimization and Control of Dynamic Humanoid Running and Jumping*. PhD thesis, Ohio State University, 2014.

[26] Y. Hurmuzlu and D. B. Marghitu, "Rigid body collisions of planar kinematic chains with multiple contact points," *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, 1994.

[27] M. Focchi, A. Del Prete, I. Havoutis, R. Featherstone, D. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, vol. 41, pp. 259–272, 05 2016.

[28] S. Rezazadeh, C. Hubicki, M. Jones, A. Peekema, J. Why, A. Abate, and J. Hurst, "Spring-mass walking with atrias in 3d: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot," in *ASME 2015 Dynamic Systems and Control Conference*, p. V001T04A003, 10 2015.

[29] S. Rezazadeh and J. W. Hurst, "Control of atrias in three dimensions: Walking as a forced-oscillation problem," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 774–796, 2020.

[30] J. Hooks, M. S. Ahn, J. Yu, X. Zhang, T. Zhu, H. Chae, and D. Hong, "Alphred: A multi-modal operations quadruped robot for package delivery applications," *IEEE Robotics and Automation Letters*, vol. 5, pp. 5409–5416, Oct 2020.