



SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER CONTROL AND MANAGEMENT
ENGINEERING

Trajectory Generation for Legged Robots Based on a Closed-Form Solution of Centroidal Dynamics

AUTONOMOUS AND MOBILE ROBOTICS

Professor:

Prof. Giuseppe Oriolo

Supervisor:

Nicola Scianca

Students:

Francesco Danese 1926188

Giuseppina Iannotti 1938436

Alessia Pontiggia 1892079

Contents

1	Introduction	3
2	Related Works	4
3	Proposed Method	6
3.1	Stiffness-Based Centroidal Dynamics	6
3.2	Closed-Form Solutions and Discrete-Time Equations	8
3.3	Integration of Base-Link Rotation	9
4	Trajectory Optimization	10
4.1	State Equation	10
4.2	Formulation of Optimal Control Problem	10
5	Simulation and Results	14
5.1	Implementation Details	15
5.2	Still Task	16
5.3	Walking Task	21
6	Conclusion	27
	References	28

Abstract

Recent advances in robotics have led to increasing interest in dynamic motion planning for humanoid and legged robots, which typically have many degrees of freedom. However, generating feasible trajectories in real time using full dynamic models remains computationally challenging. To address this, reduced-order models are often employed, although they can suffer from limited accuracy in complex scenarios. This project builds on the method proposed by Tazaki et al. [1], which introduces a stiffness-based parametrization of contact wrenches to derive closed-form centroidal dynamics and improve the efficiency of trajectory optimization. The resulting model, known as Stiffness-Based Centroidal Dynamics (SBCD), describes both translational and rotational motion while maintaining a compact and analytically tractable form. This structure makes it particularly well-suited for long-horizon planning and real-time control. The goal of this work is to implement and evaluate the SBCD framework through the generation of reference trajectories for walking and standing tasks. The proposed approach is tested in simulation and assessed using quantitative metrics and visual analysis.

1 Introduction

Humanoid and legged robots have seen rapid development in recent years, leading to their adoption in areas such as logistics, surveillance, and social interaction. Their many degrees of freedom allow them to perform versatile and dynamic movements, but this complexity makes real-time trajectory generation using full dynamic models computationally demanding. To reduce computational complexity, researchers have increasingly relied on reduced-order (template) models that approximate the system’s key dynamics. These models balance efficiency and accuracy, and are commonly combined with full models in trajectory optimization and model predictive control.

Among the most well-known reduced-order models are the Linear Inverted Pendulum (LIPM) [2] and its variant, the Variable-Height LIPM [3], which assume simplified linear dynamics, making them suitable for structured environments but inadequate for dynamic or uneven terrains. On the other hand, models like the Spring-Loaded Inverted Pendulum (SLIP) [4] more accurately capture the underlying physical dynamics, but lack of a closed form solutions, making optimization more complex. As a result, trajectory planning with these models often leads to large-scale problems, limiting their real-time applicability.

To address the trade-off between computational efficiency and model expressiveness, Tazaki et al. [1] propose a novel approach based on a stiffness-based parametrization of contact wrenches. This formulation simplifies centroidal dynamics and enables more efficient trajectory optimization over extended time horizons. The main contributions of the work are as follows. First, the authors introduce the Stiffness-Based Centroidal Dynamics model, which captures both translational and rotational motion components. Second, they derive closed-form expressions for these dynamics using the proposed contact wrench parametrization. Third, a trajectory optimization framework is developed, incorporating task objectives, physical constraints, and contact-related cost terms to support both static and dynamic behaviors.

This project aims to implement the proposed SBCD model and to reproduce the results presented in the original work for both walking and static balance tasks. It includes the development of reference generation algorithms tailored to these specific tasks. The algorithms are evaluated through simulation, and their performance is assessed using quantitative metrics, plots, and tables.

The organization of this report is as follows. In Section 2, related works on reduced-order modeling and trajectory optimization are reviewed. In Section 3, the derivation of the proposed SBCD model is detailed. In Section 4, the trajectory optimization problem is formulated and the adopted solution approach is presented. In Section 5, experimental results are shown, including algorithmic implementation and demonstrations of standing and walking tasks. Finally, in Section 6, the project’s main results are discussed and potential future works are outlined.

2 Related Works

Understanding Centroidal Dynamics is essential for controlling humanoid robots. Centroidal Dynamics [5] refers to the dynamics of a robot projected onto its Center of Mass (CoM), which represents the average position of all its mass and serves as the point where the overall gravitational force can be considered to act. Although humanoid robots are complex systems with high-dimensional, nonlinear dynamics, the motion of their Center of Mass can often be described using simpler, more intuitive models. Focusing on the CoM allows the global dynamics of the robot to be captured without modeling each joint and link individually, significantly reducing system complexity. This approach provides an effective framework for planning, control, and stability, enabling the development of robots with robust, agile, and human-like behavior. Thanks to these advantages, many reduced order models have been developed by simplifying CD in different ways.

Simplified Centroidal Models One of the foundational reduced-order models in humanoid locomotion is the Linear Inverted Pendulum Model (LIPM) [2]. This model treats the robot as a point mass and simplifies the equations of motion by assuming a constant height of the Center of Mass and neglecting angular momentum. These assumptions linearize the dynamics and make the model especially suitable for fast and efficient planning and control in legged robots. The model ensures stability by maintaining the Zero Moment Point (ZMP) within the support polygon defined by the feet. To improve upon LIPM and allow for more dynamic behaviors, the Variable-Height Linear Inverted Pendulum Model (VH-LIPM) [3] was introduced. This model integrates a linear feedback controller that aligns with the 3D Divergent Component of Motion (DCM) [6] under feasible conditions and leverages vertical CoM variations when the ZMP nears the edge of the support region. Another widely adopted model is the Spring-Loaded Inverted Pendulum (SLIP), which assumes a compliant leg structure and is often used to replicate running dynamics [7, 8]. An extension of this, the Asymmetric SLIP (ASLIP) model [4], combines the flexibility of SLIP with the formal guarantees of Hybrid Zero Dynamics (HZD) [9] control theory to produce stable running motions. The ASLIP includes torso dynamics that are nontrivially coupled with leg motion, further enhancing its realism.

Integration of Trajectory Optimization Reduced-order models like LIPM, VH-LIPM and SLIP are commonly embedded into trajectory optimization (TO) frameworks. TO involves computing optimal motion plans by minimizing a cost function subject to dynamic and physical constraints. In these setups, reduced models serve as simplified system representations within these optimization problems. Approaches to TO for CoM trajectories vary. Some use LIPM as the state equation

[10, 11], defining desired ZMP as a cost [12] and enforcing stability through ZMP constraints [13, 14]. When both CoM movement and base link rotation are considered, centroidal dynamics is used either as a constraint or state equation [15, 16], with stability enforced via support criteria like the ZMP region or the Centroidal Wrench Cone (CWC) [17, 18].

Limitations and Solutions Despite their flexibility, numerical trajectory optimization techniques often require small time steps for accurate integration, leading to a large number of decision variables and increased computational cost. Additionally, these methods usually guarantee feasibility only at discrete time points, leaving feasibility in between unverified. Some attempts have been made to mitigate these issues by linearizing centroidal dynamics—typically by neglecting rotational effects and fixing the CoM height [19]. While these simplifications can be effective in conventional walking scenarios, they fall short when tackling more dynamic tasks like multi-contact planning or acrobatic maneuvers. To address the limitations of discretization-heavy methods, researchers have explored closed-form solutions of centroidal dynamics. A common method involves treating contact wrenches as piecewise constant (zero-order hold), which enables larger integration intervals. However, this often induces undesirable angular momentum fluctuations unless extremely short time steps are used. In [20], the multi-contact (mc-) LIPM is proposed, expressing contact forces as functions of stiffness and the displacement between contact points and the CoM. This method allows for larger integration steps—potentially spanning entire contact phases—with significant angular momentum disturbances.

To overcome these drawbacks, a novel stiffness-based method is introduced [1]. This method derives closed-form solutions for centroidal dynamics by parameterizing contact wrenches through stiffness models. These analytical solutions are then integrated into trajectory optimization, enabling the generation of longer, dynamically feasible trajectories with significantly fewer decision variables.

3 Proposed Method

3.1 Stiffness-Based Centroidal Dynamics

We begin from the standard *centroidal dynamics* equations, which relate the motion of the robot's center of mass (CoM) to the total external wrench (force and moment) acting on the system:

$$m \ddot{\mathbf{p}} = \mathbf{f} - m \mathbf{g}, \quad (1a)$$

$$\dot{\mathbf{L}} = \boldsymbol{\eta}. \quad (1b)$$

Here:

- $\mathbf{p} \in \mathbb{R}^3$ is the position of the CoM, and $\ddot{\mathbf{p}}$ its acceleration.
- $\mathbf{L} \in \mathbb{R}^3$ is the total angular momentum about the CoM.
- $\mathbf{f} \in \mathbb{R}^3$ and $\boldsymbol{\eta} \in \mathbb{R}^3$ are, respectively, the translational and rotational components of the total external wrench.
- $m > 0$ is the total mass, and $\mathbf{g} \in \mathbb{R}^3$ is the gravity vector (e.g. $\mathbf{g} = [0, 0, -9.81]^\top$).

We assume all external wrenches to be contact wrenches at n_e end-effectors ("ends") of the robot. Denote by $\mathbf{p}_l \in \mathbb{R}^3$ the world-frame position of the l -th end, and let

$$\mathbf{f}_l \in \mathbb{R}^3, \quad \boldsymbol{\eta}_l \in \mathbb{R}^3, \quad l = 1, \dots, n_e$$

be the translational and rotational components of the contact wrench at that end. Then the total wrench is

$$\mathbf{f} = \sum_{l=1}^{n_e} \mathbf{f}_l, \quad \boldsymbol{\eta} = \sum_{l=1}^{n_e} \left[(\mathbf{p}_l - \mathbf{p}) \times \mathbf{f}_l + \boldsymbol{\eta}_l \right]. \quad (2)$$

The cross-product in (2) makes the system *bilinear* in $(\mathbf{p}, \mathbf{f}_l)$, coupling CoM motion with contact forces.

Spring-like parametrization of contact wrenches. Rather than holding $\mathbf{f}_l, \boldsymbol{\eta}_l$ constant, we introduce a *stiffness* parameter $\lambda_l \geq 0$ and a *CMP-offset* vector $\mathbf{r}_l \in \mathbb{R}^3$ for each end, plus a pure-moment direction $\hat{\boldsymbol{\eta}}_l \in \mathbb{R}^3$. Inspired by a spring model, we set:

$$\boxed{\mathbf{f}_l = m \lambda_l^2 \left(\mathbf{p} - (\mathbf{p}_l + \mathbf{r}_l) \right), \quad \boldsymbol{\eta}_l = m \lambda_l^2 \hat{\boldsymbol{\eta}}_l.} \quad (3)$$

Intuitively:

- λ_l^2 scales like a contact stiffness: larger $\lambda_l \rightarrow$ stronger repulsive force.
- $\mathbf{p}_l + \mathbf{r}_l$ is the *virtual pivot* (similar to a Centroidal Moment Pivot, CMP). The force pulls the CoM toward that point.
- $\hat{\boldsymbol{\eta}}_l$ encodes any pure moment about the CoM, scaled consistently by λ_l^2 .

Derivation of the stiffness-based model

By substituting (2) and (3) into (1a) and neglecting $O(\epsilon^2)$ terms one obtains

$$\begin{aligned}
m \ddot{\mathbf{p}} &= \sum_{l=1}^{n_e} m \lambda_l^2 (\mathbf{p} - (\mathbf{p}_l + \mathbf{r}_l)) - m \mathbf{g} \\
&\approx \sum_{l=1}^{n_e} m \lambda_l^2 (\mathbf{p} - (\mathbf{p}_l + \mathbf{r}_l)) - m \mathbf{g} + m \epsilon^2 \mathbf{p} \\
&= m \left(\sum_l \lambda_l^2 + \epsilon^2 \right) \mathbf{p} - m \left(\sum_l \lambda_l^2 (\mathbf{p}_l + \mathbf{r}_l) + \mathbf{g} \right) \\
&= m \bar{\lambda}^2 (\mathbf{p} - \bar{\mathbf{p}} - \bar{\mathbf{r}}),
\end{aligned} \tag{4}$$

where

$$\bar{\lambda}^2 = \sum_l \lambda_l^2 + \epsilon^2, \quad \bar{\mathbf{p}} = \frac{\sum_l \lambda_l^2 \mathbf{p}_l + \mathbf{g}}{\bar{\lambda}^2}, \quad \bar{\mathbf{r}} = \frac{\sum_l \lambda_l^2 \mathbf{r}_l}{\bar{\lambda}^2}.$$

Similarly, substituting into (1b) gives

$$\begin{aligned}
\dot{\mathbf{L}} &= \sum_{l=1}^{n_e} [(\mathbf{p}_l - \mathbf{p}) \times m \lambda_l^2 (\mathbf{p} - \mathbf{p}_l - \mathbf{r}_l) + m \lambda_l^2 \hat{\boldsymbol{\eta}}_l] \\
&= \sum_l [(\mathbf{p} - \mathbf{p}_l) \times m \lambda_l^2 \mathbf{r}_l] + \sum_l m \lambda_l^2 \hat{\boldsymbol{\eta}}_l \\
&= \mathbf{p} \times m \bar{\lambda}^2 \bar{\mathbf{r}} + \sum_l m \lambda_l^2 (\hat{\boldsymbol{\eta}}_l - \mathbf{p}_l \times \mathbf{r}_l) \\
&\approx (m \ddot{\mathbf{p}} + m \bar{\lambda}^2 (\bar{\mathbf{p}} + \bar{\mathbf{r}})) \times \bar{\mathbf{r}} + \sum_l m \lambda_l^2 (\hat{\boldsymbol{\eta}}_l - \mathbf{p}_l \times \mathbf{r}_l) \\
&= m (\ddot{\mathbf{p}} \times \bar{\mathbf{r}} + \bar{\boldsymbol{\eta}}),
\end{aligned} \tag{5}$$

where

$$\bar{\boldsymbol{\eta}} = \bar{\lambda}^2 (\bar{\mathbf{p}} \times \bar{\mathbf{r}}) + \sum_l \lambda_l^2 (\hat{\boldsymbol{\eta}}_l - \mathbf{p}_l \times \mathbf{r}_l).$$

Closed-form centroidal dynamics. Combining (4) and (5) yields the stiffness-based centroidal equations:

$$\ddot{\mathbf{p}} = \bar{\lambda}^2 (\mathbf{p} - (\bar{\mathbf{p}} + \bar{\mathbf{r}})), \tag{6a}$$

$$\dot{\mathbf{L}} = m (\ddot{\mathbf{p}} \times \bar{\mathbf{r}} + \bar{\boldsymbol{\eta}}). \tag{6b}$$

The aggregated parameters $\bar{\lambda}, \bar{\mathbf{p}}, \bar{\mathbf{r}}, \bar{\boldsymbol{\eta}}$ recover the same expressions as before.

Discussion and special cases.

Remark 1 (Exactness vs. flight phase). If one ignores flight (i.e. always in contact, $\sum_l \lambda_l^2 > 0$), one may set $\epsilon = 0$ such that (6a) and (6b) hold exactly. Otherwise $\epsilon > 0$ guarantees a well-defined $\bar{\lambda}$ in airborne phases.

Remark 2 (Ballistic motion). When all ends lose contact ($\lambda_l = 0$ for all l), one finds

$$\ddot{\mathbf{p}} = \epsilon^2 \mathbf{p} - \mathbf{g} \approx -\mathbf{g}, \quad \dot{\mathbf{L}} = 0,$$

recovering the usual ballistic CoM motion and conservation of angular momentum.

Remark 3 (Relation to existing models). Stiffness-based (or force-to-point) parametrization has appeared before, but typically only at the *total* wrench level. Here we assign a separate $\lambda_l, \mathbf{r}_l, \hat{\boldsymbol{\eta}}_l$ to each end, which yields a unified multi-contact description. The classical CoP and (e)CMP emerge naturally as $\bar{\mathbf{p}}$ and $\bar{\mathbf{p}} + \bar{\mathbf{r}}$, respectively.

3.2 Closed-Form Solutions and Discrete-Time Equations

We subdivide the time horizon $[0, T]$ into N consecutive intervals

$$[t_k, t_{k+1}], \quad k = 0, 1, \dots, N-1, \quad t_{k+1} = t_k + \tau_k.$$

We assume that *contact states* (i.e. which ends are in contact) change only at the boundaries t_k . Moreover, we apply a *zero-order hold* on the stiffness-based parameters

$$\{\lambda_l(t), \mathbf{r}_l(t), \hat{\boldsymbol{\eta}}_l(t)\} \mapsto \{\lambda_{l,k}, \mathbf{r}_{l,k}, \hat{\boldsymbol{\eta}}_{l,k}\} \quad \text{for } t \in [t_k, t_{k+1}),$$

meaning that each parameter is held constant over the interval.

State at the beginning of interval k . Let

$$\mathbf{p}_k = \mathbf{p}(t_k), \quad \mathbf{v}_k = \dot{\mathbf{p}}(t_k), \quad \mathbf{L}_k = \mathbf{L}(t_k),$$

and compute the aggregated quantities

$$\begin{aligned} \bar{\lambda}_k &= \sqrt{\sum_{l=1}^{n_e} \lambda_{l,k}^2 + \epsilon^2}, \quad \bar{\mathbf{p}}_k = \frac{\sum_{l=1}^{n_e} \lambda_{l,k}^2 \mathbf{p}_{l,k} + \mathbf{g}}{\bar{\lambda}_k^2}, \quad \bar{\mathbf{r}}_k = \frac{\sum_{l=1}^{n_e} \lambda_{l,k}^2 \mathbf{r}_{l,k}}{\bar{\lambda}_k^2}, \\ \bar{\boldsymbol{\eta}}_k &= \bar{\lambda}_k^2 (\bar{\mathbf{p}}_k \times \bar{\mathbf{r}}_k) + \sum_{l=1}^{n_e} \lambda_{l,k}^2 (\hat{\boldsymbol{\eta}}_{l,k} - \mathbf{p}_{l,k} \times \mathbf{r}_{l,k}). \end{aligned}$$

Analytical solution on $[t_k, t_{k+1}]$. With $\bar{\lambda}_k, \bar{\mathbf{p}}_k, \bar{\mathbf{r}}_k, \bar{\boldsymbol{\eta}}_k$ constant, the CoM-dynamics

$$\ddot{\mathbf{p}} = \bar{\lambda}_k^2 (\mathbf{p} - (\bar{\mathbf{p}}_k + \bar{\mathbf{r}}_k))$$

is a linear second-order ODE whose homogeneous+particular solution reads

$$\mathbf{p}(t) = (\bar{\mathbf{p}}_k + \bar{\mathbf{r}}_k) + C_k(\Delta t) (\mathbf{p}_k - (\bar{\mathbf{p}}_k + \bar{\mathbf{r}}_k)) + \frac{S_k(\Delta t)}{\bar{\lambda}_k} \mathbf{v}_k, \quad (7a)$$

$$\mathbf{v}(t) = \dot{\mathbf{p}}(t) = \bar{\lambda}_k S_k(\Delta t) (\mathbf{p}_k - (\bar{\mathbf{p}}_k + \bar{\mathbf{r}}_k)) + C_k(\Delta t) \mathbf{v}_k, \quad (7b)$$

where $\Delta t = t - t_k$ and

$$C_k(\Delta t) = \cosh(\bar{\lambda}_k \Delta t), \quad S_k(\Delta t) = \sinh(\bar{\lambda}_k \Delta t).$$

Finally, substituting into the angular-momentum equation

$$\dot{\mathbf{L}} = m(\ddot{\mathbf{p}} \times \bar{\mathbf{r}}_k + \bar{\boldsymbol{\eta}}_k)$$

and integrating from t_k to t gives

$$\mathbf{L}(t) = \mathbf{L}_k + m((\mathbf{v}(t) - \mathbf{v}_k) \times \bar{\mathbf{r}}_k + (t - t_k) \bar{\boldsymbol{\eta}}_k). \quad (7c)$$

Remark 4 (Zero-Order Hold). A Zero-order hold means we approximate time-varying parameters by piecewise-constant values on each interval. This yields closed-form expressions above, at the cost of not capturing high-frequency parameter variations.

3.3 Integration of Base-Link Rotation

The centroidal state $(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{L})$ does not specify the *orientation* $\mathbf{R}(t) \in SO(3)$ or the *base-link* angular velocity $\boldsymbol{\omega}(t) \in \mathbb{R}^3$. In a multi-body system one shows

$$\mathbf{L} = \underbrace{\mathbf{R} I \mathbf{R}^\top}_{I_{\text{sys}}(\mathbf{R})} \boldsymbol{\omega} + \mathbf{R} \hat{\mathbf{L}}, \quad (8)$$

where

- $I \in \mathbb{R}^{3 \times 3}$ is the composite inertia in the base-link frame,
- $\hat{\mathbf{L}}$ is the angular momentum about the base link due to internal motions.

If we fix reference values I_{ref} , $\hat{\mathbf{L}}_{\text{ref}}$ (e.g. from a nominal whole-body motion), we solve for

$$\boldsymbol{\omega}(t) = I_{\text{sys}}(\mathbf{R})^{-1}(\mathbf{L} - \mathbf{R} \hat{\mathbf{L}}_{\text{ref}}) \approx \mathbf{R} I_{\text{ref}}^{-1}(\mathbf{R}^\top \mathbf{L} - \hat{\mathbf{L}}_{\text{ref}}). \quad (9)$$

Discrete quaternion update. Let $\mathbf{q}_k \in \mathbb{H}$ be the unit-quaternion representing $\mathbf{R}(t_k)$. Over $[t_k, t_{k+1}]$ we subdivide into n_{div} equal steps

$$t_k = t'_0 < \dots < t'_i < \dots < t'_{n_{\text{div}}} = t_{k+1}, \quad \tau'_k = \frac{\tau_k}{n_{\text{div}}}, \quad t'_i = t_k + i \tau'_k.$$

At each substep we assume $\boldsymbol{\omega}$ nearly constant and update

$$\mathbf{q}_{k+1} = \underbrace{\mathbf{q}(\boldsymbol{\omega}(t'_{n_{\text{div}}-1}) \tau'_k)}_{\text{quat. for small rotation}} \cdots \mathbf{q}(\boldsymbol{\omega}(t'_1) \tau'_k) \cdot \mathbf{q}(\boldsymbol{\omega}(t'_0) \tau'_k) \cdot \mathbf{q}_k, \quad (10)$$

where $\mathbf{q}(\boldsymbol{\theta})$ is the unit quaternion corresponding to the axis-angle $\boldsymbol{\theta} \in \mathbb{R}^3$. The designer chooses n_{div} to balance integration accuracy against computational cost of gradient evaluation in trajectory optimization.

4 Trajectory Optimization

4.1 State Equation

To formally describe the evolution of the system over discrete time intervals, we define the state and input vectors at each time step k . The state vector \mathbf{x}_k includes all variables necessary to characterize the system's configuration and motion, while the control input vector \mathbf{u}_k is defined according to the stiffness-based control strategy introduced in Section 3. These vectors are structured as follows:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{q}_k \\ \mathbf{v}_k \\ \mathbf{L}_k \\ \mathbf{l}_k \\ t_k \\ \{\mathbf{p}_{l,k}\}_{l=1,\dots,n_e} \\ \{\mathbf{q}_{l,k}\}_{l=1,\dots,n_e} \end{bmatrix} \quad \mathbf{u}_k = \begin{bmatrix} \tau_k \\ \{\mathbf{v}_{l,k}\}_{l=1}^{n_e} \\ \{\boldsymbol{\omega}_{l,k}\}_{l=1}^{n_e} \\ \{\lambda_{l,k}\}_{l=1}^{n_e} \\ \{\mathbf{r}_{l,k}\}_{l=1}^{n_e} \\ \{\boldsymbol{\eta}_{l,k}\}_{l=1}^{n_e} \end{bmatrix} \quad (7)$$

By defining end-effector velocities as control inputs, contact complementarity can be enforced penalizing motion at contact points through high velocity costs.

Equations Update We now present the update equations that define the system's evolution over discrete time steps.

The update of the timestamp is defined as:

$$t_{k+1} = t_k + \tau_k \quad (8)$$

Next, the position and orientation of each end-effector are updated using basic kinematic relations:

$$\mathbf{p}_{l,k+1} = \mathbf{p}_{l,k} + \mathbf{v}_{l,k} \tau_k \quad (9)$$

$$\mathbf{q}_{l,k+1} = q(\boldsymbol{\omega}_{l,k} \tau_k) \cdot \mathbf{q}_k \quad (10)$$

Here, $q(\boldsymbol{\omega}_{l,k}, \tau_k)$ denotes the quaternion representing the angular displacement resulting from integrating the angular velocity $\boldsymbol{\omega}_{l,k}$ over the time step τ_k . Integrating all these elements, the system dynamics can be compactly represented by the following state transition function:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad (11)$$

4.2 Formulation of Optimal Control Problem

In trajectory optimization, the goal is to find a sequence of control inputs that minimize a cost function while satisfying the system dynamics and any constraints. This cost function evaluates the quality of a trajectory and typically consists of multiple terms, each reflecting a specific performance objective.

Task Related Cost In trajectory tracking tasks, it is important for the system to follow a planned or reference trajectory as closely as possible. The *task-related cost* measures how much the system’s current state and inputs deviate from their desired (reference) values. Minimizing this cost ensures the system stays close to the intended path during motion. The task-related cost function is formulated as:

$$L_{\text{task},k} = \frac{1}{2} \|W_k^x(\mathbf{x}_k - \mathbf{x}_k^{\text{ref}})\|^2 + \frac{1}{2} \|W_k^u(\mathbf{u}_k - \mathbf{u}_k^{\text{ref}})\|^2 \quad (12)$$

where $(*)^{\text{ref}}$ represents the reference (target) values for the state \mathbf{x}_k and the control input \mathbf{u}_k . The reference trajectories were manually created by the method explained in Sec. 5 and describe the desired values for the CoM, base link, and end-effectors. The desired stiffness values are computed by solving the following least-squares optimization problem at each time step k :

$$\min \left\| \sum_l \lambda_{l,k}^2 \right\|^2 \quad \text{subject to} \quad \sum_l \lambda_{l,k}^2 (\mathbf{p}_k^{\text{ref}} - \mathbf{p}_{l,k}^{\text{ref}}) = \mathbf{g} \quad (13)$$

This subproblem determines the stiffness distribution that supports the CoM against gravity. In addition, the desired values of the Centroidal Moment Pivot (CMP) offset and the moment of each end are typically set to zero unless non-zero values are specifically chosen to induce desired dynamic effects. The weighting matrices W_k^x and W_k^u are design parameters that control the importance given to state and input deviations in the cost function. Finally, when dealing with rotational variables represented by quaternions, the deviation between the actual and reference orientations is defined as :

$$\mathbf{q} - \mathbf{q}^{\text{ref}} := \omega(\mathbf{q}^{\text{ref}^{-1}} \mathbf{q}) \quad (14)$$

where $\omega(\cdot)$ maps a unit quaternion into an angle-axis vector.

Inequality Constraints In contact dynamics, physical conditions such as feasible positions, contact forces, and stiffness must be satisfied to ensure realistic motion. These are enforced through *inequality constraints*, which maintain both physical plausibility and optimization feasibility. A detailed description of these constraints follows.

The position of each end link relative to the CoM and the base link is constrained using a box formulation:

$$\mathbf{p}_{l,\text{min}} \leq \mathbf{q}^{-1}(\mathbf{p}_l - \mathbf{p}) \leq \mathbf{p}_{l,\text{max}}, \quad (15)$$

Simple range constraints are imposed on the duration of each phase and the stiffness values:

$$\tau_{\text{min}} \leq \tau \leq \tau_{\text{max}}, \quad (16)$$

$$0 \leq \lambda_l \leq \lambda_{\text{max}}, \quad \forall l. \quad (17)$$

Next, for each end in contact, the contact wrench must satisfy non-slip and moment conditions. To prevent relative motion at the contact surface, sufficient friction must be maintained. This is achieved by requiring the contact force to lie within the friction cone. Specifically, the tangential force f_t must satisfy

$$|f_t| \leq \mu f_n \implies \sqrt{f_{l,x}^2 + f_{l,y}^2} \leq \mu f_{l,z} \quad (16)$$

where μ is the static friction coefficient.

Constraints on the moments at the contact point are expressed as:

$$-c_{\max,x} f_{l,z} \leq \eta_{l,x} \leq -c_{\min,x} f_{l,z} \quad (17)$$

$$c_{\min,y} f_{l,z} \leq \eta_{l,y} \leq c_{\max,y} f_{l,z} \quad (18)$$

$$-\mu_z f_{l,z} \leq \eta_{l,z} \leq \mu_z f_{l,z} \quad (19)$$

where c_{\min} and c_{\max} define the rectangular bounds of the center-of-pressure (CoP) region, and μ_z is the coefficient of friction torque.

All the inequality constraints can be compactly represented as:

$$g(\mathbf{x}_k, \mathbf{u}_k) \geq 0, \quad (20)$$

where $g(\cdot)$ is a differentiable vector-valued function, evaluated componentwise. To handle these constraints during optimization, a log-barrier function is introduced:

$$L_{\text{limit}}(\mathbf{x}_k, \mathbf{u}_k) = \sum_{i=1}^{n_g} -\log \max(\epsilon, g_i(\mathbf{x}_k, \mathbf{u}_k)), \quad (21)$$

where n_g is the number of constraints, g_i is the i -th constraint function, and ϵ is a small positive constant used to prevent numerical instability and avoid undefined values in the logarithmic function.

Contact Dependent Cost To ensure consistent interaction between a robot's end-effectors and the environment, a contact-dependent cost is introduced. It promotes complementarity between contact forces, velocities, and stiffness, encouraging physical consistency during contact and suppressing unnecessary interaction otherwise. This supports smooth transitions between contact and non-contact phases.

The contact-dependent cost is defined as:

$$\begin{aligned}
J_{\text{compl},k} = & w_{\text{compl}}^2 \sum_l \left(\underbrace{\sum_i \delta[\sigma_{l,k} = i] (\boldsymbol{\eta}_i^\top (\mathbf{p}_{l,k} - \mathbf{o}_i))^2}_{\text{contact distance constraint}} \right. \\
& + \underbrace{\delta[\sigma_{l,k} \neq \emptyset] (\|\mathbf{v}_{l,k}\|^2 + \|\boldsymbol{\omega}_{l,k}\|^2)}_{\text{zero velocity constraint}} \\
& \left. + \underbrace{\delta[\sigma_{l,k} = \emptyset] \lambda_{l,k}^2}_{\text{zero stiffness constraint}} \right)
\end{aligned} \tag{22}$$

Here $\sigma_{l,k}$ denotes the contact state of the l -th end at time step k , with $\sigma_{l,k} = i$ indicating contact with the i -th surface, and $\sigma_{l,k} = \emptyset$ indicating no contact. The operator $\delta[*]$ is an indicator function that returns 1 if the condition inside the brackets is true, and 0 otherwise.

Each term inside the cost function has a specific physical meaning. When the l -th end is in contact with surface i , the first term (top line of Eq. (22)) penalizes the distance from the end's position $\mathbf{p}_{l,k}$ to the surface origin \mathbf{o}_i along the surface normal $\boldsymbol{\eta}_i$, enforcing proper alignment with the contact surface. The second term (middle line) becomes active whenever the end-effector is in contact with any surface and penalizes nonzero linear and angular velocities $\mathbf{v}_{l,k}$ and $\boldsymbol{\omega}_{l,k}$, thereby promoting static behavior at the contact point. Finally, when the end-effector is not in contact, the third term (bottom line) penalizes any nonzero stiffness $\lambda_{l,k}$, which prevents the generation of spurious contact forces during swing phases.

By properly tuning the weight parameter w_{compl} , these complementarity-related costs can be made negligible after optimization. A sufficiently large value of w_{compl} ensures that the physical consistency conditions are respected without significantly penalizing the quality of the optimized trajectory.

Final Cost Function and Problem Formulation After defining the task-related, limit-related, and contact-dependent costs, the overall cost function is constructed by summing these individual terms over all time steps. It is defined as :

$$J[\boldsymbol{\sigma}] = \sum_k [L_{\text{task},k} + L_{\text{limit},k} + L_{\text{compl},k}[\boldsymbol{\sigma}_k]] \tag{23}$$

The planning problem can then be formulated as the following optimal control problem:

$$\begin{aligned}
& \text{find } \mathbf{x}, \mathbf{u} \text{ that minimizes } J[\boldsymbol{\sigma}](\mathbf{x}, \mathbf{u}) \\
& \text{subject to } \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)
\end{aligned} \tag{24}$$

This formulation defines the optimal control problem to be solved for generating physically consistent and task-relevant trajectories.

5 Simulation and Results

To evaluate the effectiveness of the proposed trajectory optimization framework, we conduct simulations for two representative tasks: static balance and walking.

The analysis begins with an overview of key implementation details, focusing on the initialization of cost weights, time discretization parameters, and control bounds. These parameters define the structure of the optimization problem and facilitate the computation of feasible trajectories. Next, we outline the algorithm used to generate the desired reference trajectory for each task, establishing the target path for the robot’s motion. These trajectories serve as benchmarks, allowing us to assess the framework’s performance under different conditions.

In this framework, contact states are represented using the set $\{-, 0\}$, where $\{0\}$ indicates foot contact with the ground and $\{-\}$ denotes a swing phase. Rather than using continuous time, trajectories are structured around these discrete contact phases, ensuring consistency in representation across both tasks. This approach allows for a more intuitive interpretation of contact dynamics and simplifies the integration of state transitions.

With the reference trajectories defined, we proceed to analyze the simulation results by examining key components of the robot’s motion. First, we assess the CoM trajectories to determine how effectively the robot maintains stability and follows the intended path over time. Next, we compare the optimized state solutions with the reference trajectories at each contact step, providing insights into the effectiveness of the control strategy. Finally, we assess the contact wrenches, focusing on both translational and rotational forces to verify compliance with physical constraints and to evaluate the feasibility of the generated trajectories.

Implementation Overview Our work builds upon the framework provided in the IS - MPC Repository¹, which originally offers a comprehensive platform for simulating the HRP4 robot, including the dynamics model and control architecture. We have adapted and integrated our own implementation of the dynamics, trajectory optimization problem, and reference trajectory generation algorithm into the original repository. This integration enables the simulation of the HRP4 robot performing both the still and walking tasks, allowing us to assess the robot’s performance under our modified control and optimization framework.

¹<https://github.com/DIAG-Robotics-Lab/ismpc>

5.1 Implementation Details

Implementing the proposed SBCD model involves requires careful consideration of parameter selection, initialization, and optimization for both static and walking scenarios. Given that the parameter configuration is largely consistent across both scenarios, a unified framework is presented, with task-specific distinctions clearly specified where relevant.

Parameter Initialization and Dynamics The implementation leverages the IPOPT solver from the CasADi optimization framework to compute optimal trajectories based on the SBCD model. Both static and walking tasks are characterized by a shared set of parameters governing state initialization, dynamics, and control inputs. These parameters are systematically structured, allowing for easy adaptation between the two tasks. For both scenarios, the robot’s state is initialized with the Center of Mass positioned at a predefined reference point, with neutral feet orientations. Velocity initialization, however, is task-dependent. In the static scenario, CoM velocity is set to zero, serving as a baseline for assessing stationary behavior under SBCD control. In contrast, the walking task initialized the CoM velocity to $[0, -0.07, 0]$ (m/s), simulating an initial oscillation to the right.

Before delving into the parameter values, Table 1 outlines the specific weight configuration applied to some state components. While most parameters remain consistent, the weight for the velocity component differs between the two tasks: for the static task, it is set to 100, emphasizing minimal movement, whereas for the walking task, it is reduced to 1 to allow for more dynamic motion. For all other state components and control inputs, detailed in Eq. (7), weights are uniformly initialized to 1.

	p_y	p_z	v_k	L_k	$P_{L,k}$	$Q_{L,k}$
Static	500	500	100	1	100	0.0001
Walking	500	500	1	1	100	0.0001

Table 1: Weight configuration for state components - Still and Walking Tasks

Optimization Framework The optimization framework is designed to effectively handle contact-dependent dynamics by parameterizing state and control inputs for each contact phase. To achieve this, SBCD-based dynamics formulations are employed, enabling efficient computation of CoM trajectories and contact forces. This structured approach effectively manages transitions between single and double support phases during walking tasks.

A crucial aspect of the optimization setup is the configuration of parameters that govern the behavior of both static and walking tasks. As illustrated in Table 2, these parameters include the complementarity weight (w_{compl}), friction coefficients (μ and μ_z), and phase duration limits (τ_{min} and τ_{max}). Carefully setting these values ensures that the optimization framework remains grounded within the physical constraints of the robot’s actuators and the environmental interactions, striking a balance between control fidelity and physical feasibility.

	w_{compl}	μ	μ_z	τ_{max}	τ_{min}
Weight	1000	0.5	0.6	10	0.4

Table 2: Optimization parameters - Still and Walking Tasks

5.2 Still Task

The objective of the still task is to keep the robot stationary, preserving its initial configuration throughout the entire trajectory. As indicated in Table 3, the contact sequence ensures that both feet remain grounded, establishing a condition of complete stability without locomotion.

Task	N	Foot	Contact Sequence
Still	4	Right Left	0000 0000

Table 3: Contact sequence - Still Task

Reference Trajectory Generation The still task algorithm (Algorithm 1) generates reference state and control trajectories aimed at preserving the robot’s initial configuration throughout the task duration. By maintaining constant CoM position, orientation, and feet positions, the algorithm ensures that the robot remains stationary, counteracting gravitational forces through ground reaction forces. The algorithm begins by initializing the state X_{ref} and control U_{ref} matrices to zero. At each timestep, the CoM and feet positions are held fixed while ground reaction forces are adjusted to stabilize the robot against gravitational effects. This approach effectively prevents any unintended motion, maintaining the initial configuration without deviations.

Algorithm 1 Reference Trajectory Still Task Generation

- 1: Initialize state matrix $X_{ref} \in \mathbb{R}^{28 \times N}$ and control matrix $U_{ref} \in \mathbb{R}^{27 \times (N-1)}$ to zero
 - 2: Set initial CoM/feet position and orientation and $t_0 \leftarrow 0$ in $X_{ref}(0)$
 - 3: **for** $k = 0$ to $N - 1$ **do**
 - 4: Set $X_{ref}(k + 1)$ as a copy of the current state $X_{ref}(k)$
 - 5: Set desired phase_duration τ_k and contact force gains λ_k in $U_{ref}(k)$
 - 6: Update $X_{ref}(k + 1)$ with $t_{k+1} \leftarrow t_k + \tau_k$
 - 7: **end for**
 - 8: **return** X_{ref}, U_{ref}
-

Simulation Results Visual results are presented to illustrate the outcomes of the still task simulation, providing a comprehensive view of the system's behavior under stationary conditions.

The plots in Fig.1 illustrate the CoM evolution during the still task. The x component displays slight horizontal drift, likely due to adjustments in ground reaction forces, while the y component briefly dips before stabilizing, indicating corrective measures to maintain balance. The z component initially drops and then recovers, reflecting gravitational compensation. Despite these fluctuations, the CoM remains largely consistent, demonstrating effective stabilization.

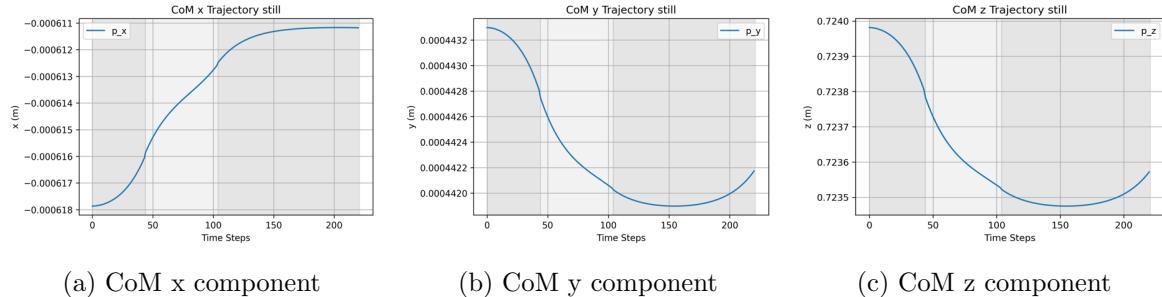


Figure 1: Evolution of CoM Position - Still Task

Fig.2 shows the evolution of feet positions along the z -axis, CoM velocity, and angular momentum during the still task. The feet positions in Fig. 2a remain nearly constant, indicating minimal vertical motion. In Fig. 2b, slight oscillations in the y and z velocity components suggest minor corrective actions to maintain stability. The angular momentum in Fig. 2c exhibits a few distinct spikes in the L_x component, indicating corrective torques applied to maintain stability, while the L_y and the L_z components remain relatively stable, reflecting a largely stable configuration.

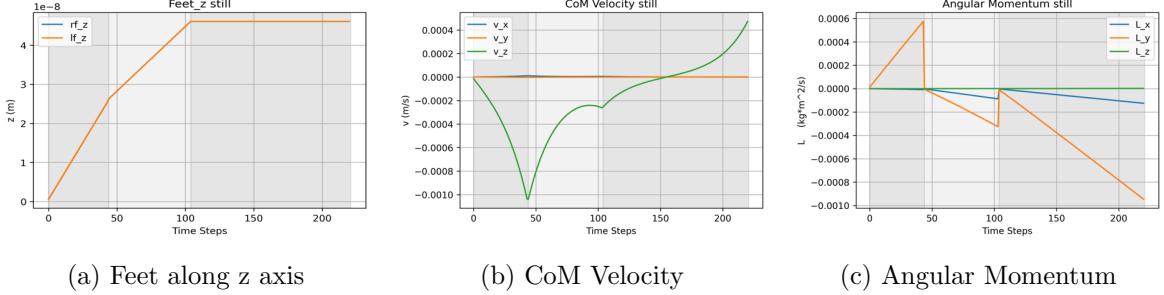


Figure 2: Feet, Com Velocity and Angular Momentum - Still Task

In Fig. 3 , the comparison between the reference and optimized trajectories for the still task is presented. The CoM position remains largely aligned with the reference values, with slight deviations observed primarily in the x and z components, reflecting minor adjustments to maintain stability. CoM velocity remains close to zero, consistent with the stationary objective, though minor oscillations indicate corrective actions against gravitational forces. Angular momentum closely follows the reference values across all components, indicating that the system remains effectively stationary with minimal rotational adjustments. Feet positions and velocities remain effectively constant, confirming that the feet maintain their initial stance without significant movement. Overall, the optimized trajectories exhibit minimal divergence from the reference, indicating effective stabilization while accounting for minor corrections.

Another key aspect of the dynamics involves the contact wrenches, which represent the mechanical forces and moments applied by contact points, such as the feet, on the robot or vice versa. To maintain dynamic balance, the reaction force along the z -axis must counteract the gravitational force acting on the robot's mass. The contact forces from the environment on both feet are summarized in Table 4. As indicated, when both feet remain grounded, the gravitational force is evenly distributed between the left and right foot, ensuring stability.

Interval	Right Foot (x,y,z)	Left Foot (x,y,z)	ΣL_k	Sum Forces (x,y,z)
0	(0.000059, 9.53808, 49.0403)	(0.000080, -9.53812, 49.046)	(0, 0)	(0.000139, -0.000038, 98.0863)
1	(-0.000233, 9.53903, 49.070)	(-0.000211, -9.53891, 49.0757)	(0, 0)	(-0.000444, 0.000127, 98.1457)
2	(-0.000143, 9.53922, 49.0531)	(-0.000121, -9.53918, 49.0587)	(0, 0)	(-0.000264, 0.000032, 98.1118)

Table 4: Summary of Forces and Foot Positions per Interval - Still Task

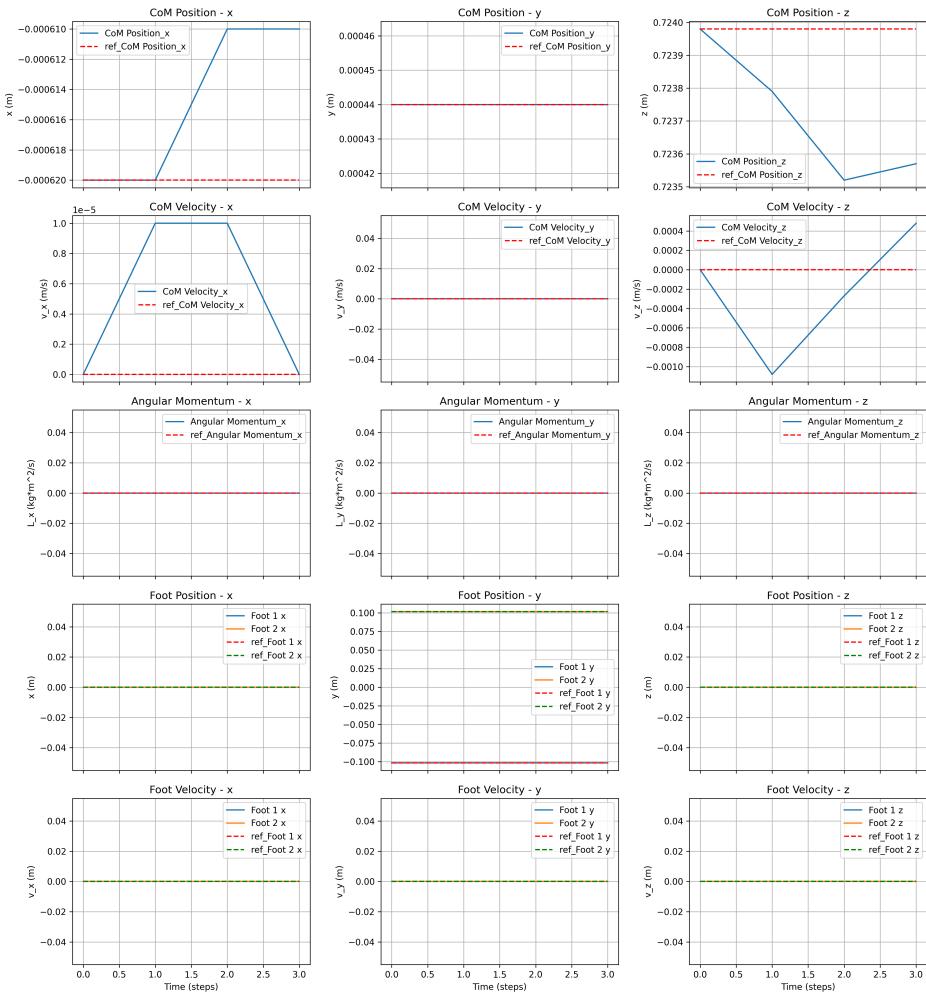


Figure 3: Trajectory vs Reference - Still Task

To conclude the analysis of the still task, Figures 4 and 5 illustrate the behavior of translational and rotational forces at the contact points for the right and left foot, respectively. Along the horizontal axes, translational forces in the x and y components remain near zero, indicating minimal lateral interaction with the ground. The vertical z component shows a pronounced peak, reflecting the compensation for gravitational forces with slight variations indicating minor adjustments. Rotational forces exhibit a more defined trend, particularly in the x and y components, with both displaying a gradual decline, suggesting a slight decrease in corrective torques over time. The z component shows a slight increase followed by a minor drop, with minor fluctuations that align the angular momentum behavior observed previously.

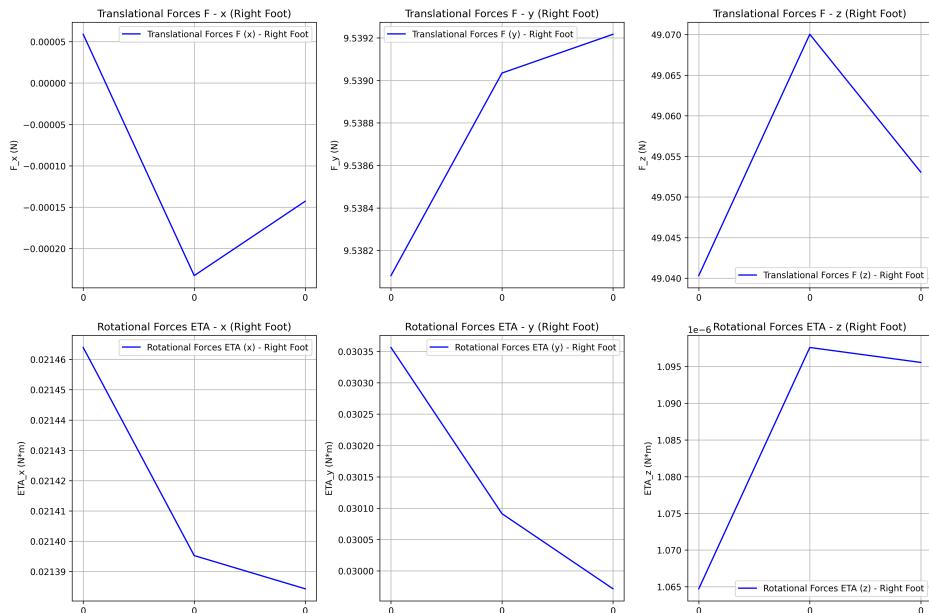


Figure 4: Translational and Rotational Forces Right Foot - Still Task

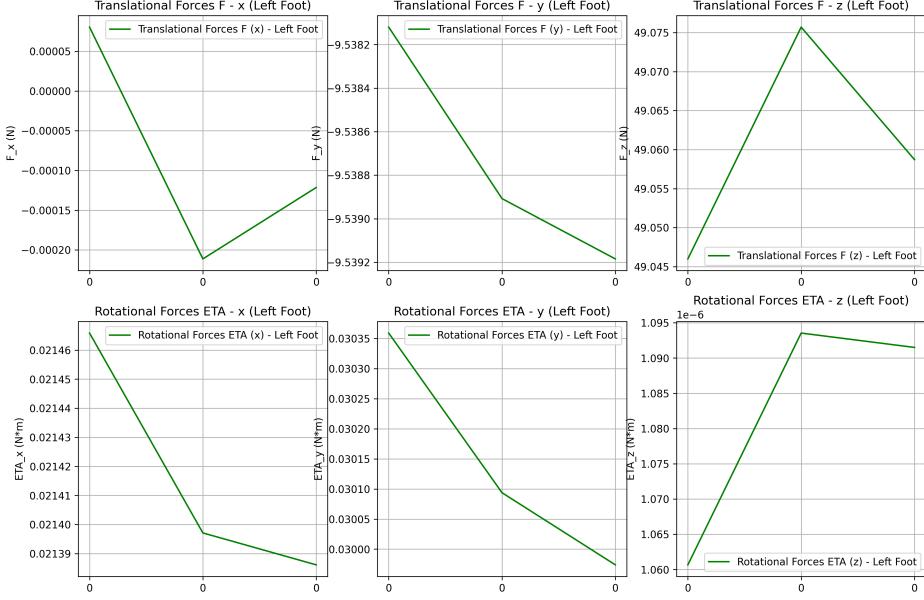


Figure 5: Translational and Rotational Forces Left Foot - Still Task

5.3 Walking Task

The objective of the walking task is to simulate a natural walking pattern by coordinating the movements of the feet and the Center of Mass (CoM) through a sequence of double-support and single-support phases. During double-support phases, both feet remain in contact with the ground, maintaining stability. In single-support phases, one foot is lifted while the other remains grounded, allowing forward motion. The CoM position update depends on the preceding phase. If the previous phase was double-support, the CoM remains stationary. Otherwise, it advances forward, ensuring consistent movement. The foot trajectory is constructed to mimic a natural step, extending the moving foot beyond the stationary one. The Z-coordinate of each foot follows a parabolic profile, ensuring smooth lift-off and descent. The contact sequence for the walking task, presented in Table 5, defines the alternating pattern of double and single-support phases, guiding the timing and transitions of each step.

Task	N	Foot	Contact Sequence
Walk	24	Right	000-000-000-000-000-0
		Left	0-000-000-000-000-000

Table 5: Contact sequence - Walking Task

Reference Trajectory Generation The reference trajectory generation algorithm for the walking task is designed to compute state and control matrices that guide the robot's CoM and foot movements throughout the walking cycle. The algorithm initializes the robot's state, reads the contact sequence to identify double and single-support phases, and calculates the CoM and feet velocities based on the

transition logic. The CoM trajectory is computed by integrating velocities over the phase duration, while feet positions are updated based on contact states. The transition logic, detailed in Table 6, specifies how velocities and displacements are adjusted for each phase.

Algorithm 2 Reference Trajectory Generation - Walking Task

- 1: Initialize state matrix $X_{ref} \in \mathbb{R}^{28 \times (N)}$ and control matrix $U_{ref} \in \mathbb{R}^{27 \times (N-1)}$ to zero
 - 2: Set initial CoM/feet position and orientation and $t_0 \leftarrow 0$ in $X_{ref}(0)$
 - 3: **for** $k = 0$ to $N - 1$ **do**
 - 4: Read current and future contacts state from $\sigma(k)$ and $\sigma(k + 1)$
 - 5: Set desired phase duration τ_k and contact force gains λ_k in $U_{ref}(k)$
 - 6: Set desired CoM/feet velocities in $X_{ref}(k), U_{ref}(k)$ (See Table 6)
 - 7: Update CoM and feet positions in $X_{ref}(k + 1)$ with $p_{k+1} \leftarrow p_k + v_k \cdot \tau_k$
 - 8: Update $X_{ref}(k + 1)$ with $t_{k+1} \leftarrow t_k + \tau_k$
 - 9: **end for**
 - 10: **return** X_{ref}, U_{ref}
-

$\sigma(k)$	$\sigma(k + 1)$	$v_{right}^x(k)$	$v_{left}^x(k)$	$v_{com}^x(k), v_{com}^y(k)$
right = 0, left = 0	right = -, left = 0	0.0	0.0	(0, 0.07)
right = 0, left = 0	right = 0, left = -	0.0	0.0	(0, -0.07)
right = -, left = 0	right = 0, left = 0	0.2	0.0	(0.1, -0.07)
right = 0, left = -	right = 0, left = 0	0.0	0.2	(0.1, 0.07)

Table 6: Transition logic for foot and CoM motion updates - Walking Task

Simulation Results As we did for the still task, here we present the simulation results for the walking task. The objective is to assess the robot’s ability to execute the walking trajectory while maintaining balance and following the reference CoM path. The results include CoM evolution, feet trajectories, and ground reaction forces.

During the walking task, as shown in Fig. 6, the CoM follows a distinct trajectory along each axis. The x component steadily increases, indicating consistent forward motion, while the y component displays cyclic oscillations corresponding to the lateral sway during each step. The z component exhibits periodic fluctuations, capturing vertical displacements associated with foot lift-offs and ground contacts. Overall, the CoM trajectory maintains stability, demonstrating effective walking dynamics.

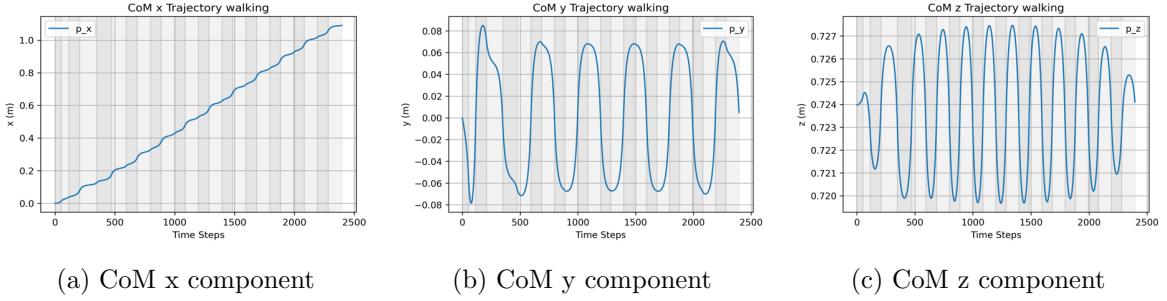


Figure 6: Evolution of CoM Position - Walking Task

The plots in Fig. 7 illustrate the dynamics of feet trajectories, CoM velocity, and angular momentum during the walking task. In Fig. 7a, the periodic variations in feet positions along the z axis capture the transition between ground clearance and contact, underscoring the cyclic nature of the walking gait. These fluctuations correspond to the CoM velocity patterns in Fig. 7b, where oscillations in the x and y components reflect the alternating support phases and weight shifts. In Fig. 7c, the angular momentum exhibits pronounced oscillations, particularly in the x component, indicating the corrective torques employed to maintain stability throughout the walking cycle.

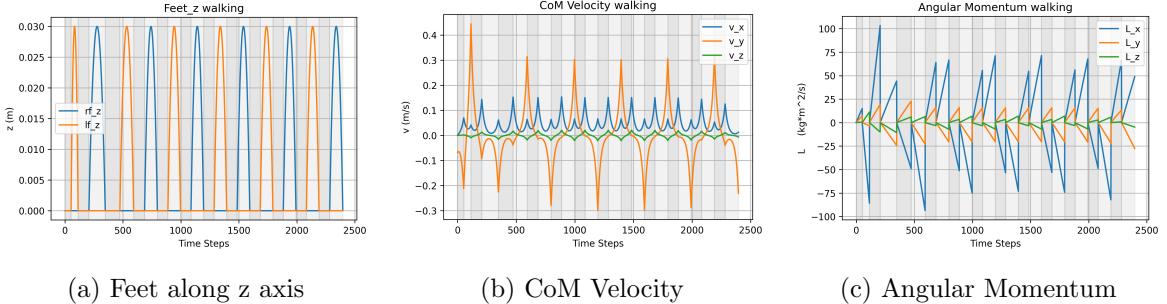


Figure 7: Feet, Com Velocity and Angular Momentum - Walking Task

The plots in Fig. 8 compare the optimized and reference trajectories for the walking task, highlighting the robot's adherence to the intended motion patterns. The CoM position plots show a consistent forward motion along the x - axis, lateral oscillations in the y - axis, and minimal vertical fluctuations in the z - axis, indicating stable walking dynamics. CoM velocities reflect these patterns, with pronounced oscillations in the x and y components and minimal z velocity, underscoring vertical stability. Angular momentum exhibits periodic variations in the x and y components during foot transitions, while the z component remains stable, suggesting controlled rotational behavior. Feet positions and velocity plots capture the cyclic nature of the gait, with distinct lift-off and ground contact phases in the z component and steady forward movement in the x component. Overall, the optimized trajectories align well with the reference paths, with minor adjustments indicating corrective actions for stability.

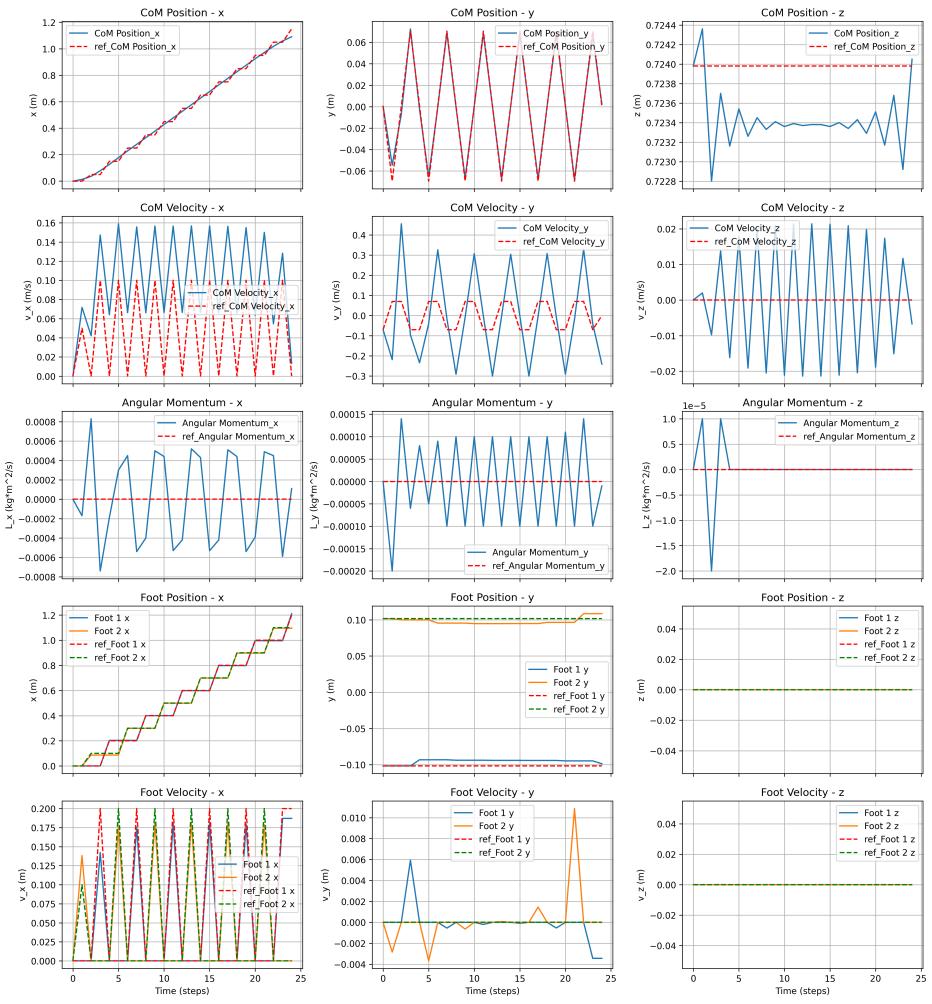


Figure 8: Trajectory vs Reference - Walking Task

As for the still task, the analysis of the contact forces during the walking task is crucial to verify that the robot maintains balance while executing the motion. As shown in Table 7, the sum of the contact forces along the z -axis consistently aligns with the gravitational force acting on the robot. This alignment indicates that the control strategy effectively regulates vertical support forces, ensuring stability throughout the gait sequence.

Interval	Right Foot (x,y,z)	Left Foot (x,y,z)	ΣL_k	Sum Forces (x,y,z)
0	(0.3098, 10.0006, 49.0692)	(0.3076, -8.8127, 49.0480)	(0, 0)	(0.6174, 1.1878, 98.1172)
1	(-2.9107, 12.2335, 97.9563)	(0, 0, 0)	(0, -1)	(-2.9107, 12.2335, 97.9563)
2	(3.2811, -0.3468, 48.5795)	(-4.8871, -19.6105, 49.9662)	(0, 0)	(-1.6060, -19.9572, 98.5457)
3	(0, 0, 0)	(-6.4470, 4.1862, 97.4919)	(-1, 0)	(-6.4470, 4.1862, 97.4919)
4	(-6.8093, 14.0802, 49.7681)	(4.0998, -3.9816, 49.0385)	(0, 0)	(-2.7095, 10.0986, 98.8066)
5	(-6.9116, 1.8871, 97.3107)	(0, 0, 0)	(0, -1)	(-6.9116, 1.8871, 97.3107)
6	(3.1803, 1.8069, 49.0314)	(-5.8173, -15.9695, 49.9272)	(0, 0)	(-2.6370, -14.1626, 98.9586)
7	(0, 0, 0)	(-6.7673, 0.4847, 97.2130)	(-1, 0)	(-6.7673, 0.4847, 97.2130)
8	(-5.9977, 15.1901, 49.9258)	(3.4081, -2.5941, 49.0988)	(0, 0)	(-2.5896, 12.5961, 99.0246)
9	(-6.7991, 0.5855, 97.1717)	(0, 0, 0)	(0, -1)	(-6.7991, 0.5855, 97.1717)
10	(3.3447, 2.2144, 49.0962)	(-5.9424, -15.5259, 49.9575)	(0, 0)	(-2.5977, -13.3115, 99.0537)
11	(0, 0, 0)	(-6.7993, -0.1305, 97.1539)	(-1, 0)	(-6.7993, -0.1305, 97.1539)
12	(-5.9816, 15.3748, 49.9533)	(3.3785, -2.3771, 49.1101)	(0, 0)	(-2.6031, 12.9977, 99.0634)
13	(-6.8083, 0.3513, 97.1498)	(0, 0, 0)	(0, -1)	(-6.8083, 0.3513, 97.1498)
14	(3.3689, 2.2945, 49.1029)	(-5.9736, -15.4697, 49.9601)	(0, 0)	(-2.6047, -13.1752, 99.0630)
15	(0, 0, 0)	(-6.8037, -0.1903, 97.1550)	(-1, 0)	(-6.8037, -0.1903, 97.1550)
16	(-5.9649, 15.3831, 49.9465)	(3.3681, -2.3898, 49.1053)	(0, 0)	(-2.5968, 12.9933, 99.0517)
17	(-6.7893, 0.4215, 97.1738)	(0, 0, 0)	(0, -1)	(-6.7893, 0.4215, 97.1738)
18	(3.3633, 2.2973, 49.0756)	(-5.9344, -15.6231, 49.9448)	(0, 0)	(-2.5711, -13.3258, 99.0204)
19	(0, 0, 0)	(-6.7306, 0.0657, 97.2207)	(-1, 0)	(-6.7306, 0.0657, 97.2207)
20	(-5.8564, 15.2809, 49.8788)	(3.3748, -2.7034, 49.0690)	(0, 0)	(-2.4816, 12.5775, 98.9477)
21	(-6.5267, 1.0544, 97.3302)	(0, 0, 0)	(0, -1)	(-6.5267, 1.0544, 97.3302)
22	(3.5736, 2.3792, 48.9388)	(-5.7275, -16.6990, 49.8343)	(0, 0)	(-2.1539, -14.3198, 98.7732)
23	(0, 0, 0)	(-5.5944, 1.6324, 97.5852)	(-1, 0)	(-5.5944, 1.6324, 97.5852)

Table 7: Summary of Forces and Foot Positions per Interval - Walking Task

To conclude the analysis of the walking task, Figures 9 and 10 depict the behavior of the translational and rotational forces at the contact points for the right and left foot, respectively. Along the horizontal axes, the translational forces exhibit periodic fluctuations, corresponding to the alternating support phases during the walking cycle. In the vertical direction (z -axis), the forces consistently counteract gravitational effects, with pronounced peaks reflecting ground contact phases and troughs indicating foot lift-off events. Rotational forces show notable oscillations in both the x and y components, particularly during transitions between single and

double support phases, indicating the corrective torques applied to maintain balance. The z component remains relatively stable, with minor oscillations that align with the angular momentum patterns previously observed.

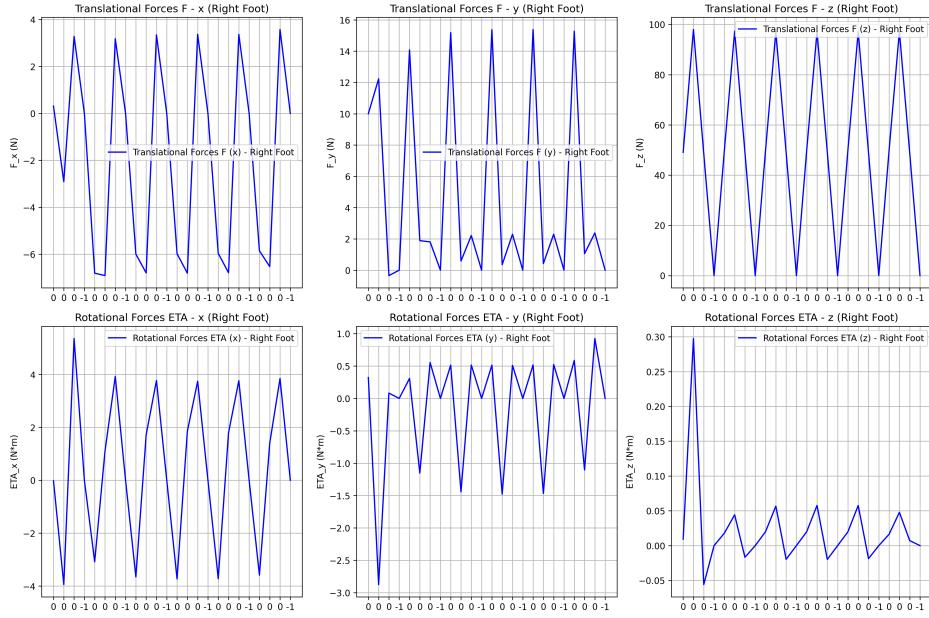


Figure 9: Translational and Rotational Forces Right Foot - Walking Task

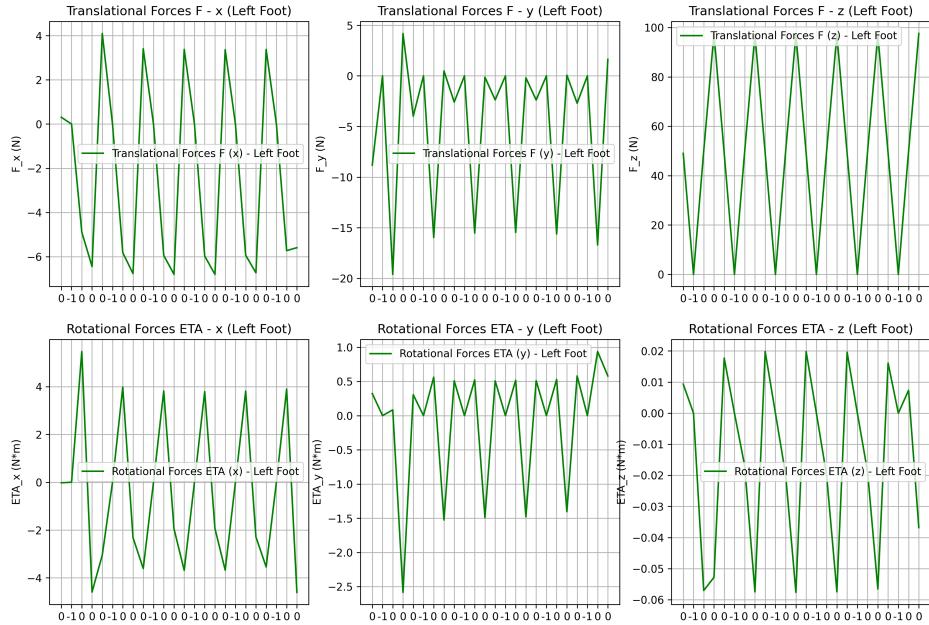


Figure 10: Translational and Rotational Forces Left Foot - Walking Task

6 Conclusion

In this project, we work with the Stiffness-Based Centroidal Dynamics (SBCD) model proposed by Tazaki [1], focusing on optimizing trajectories for both static and dynamic tasks. The SBCD framework is designed to provide accurate dynamics while remaining computationally feasible for real-time applications in legged robots. It leverages analytic integration over finite time intervals under zero-order hold assumptions, resulting in closed-form expressions for CoM trajectories and angular momentum. Additionally, we implement a method for base-link orientation updates using quaternions and nominal inertia models.

We implement and evaluate two tasks to assess the framework: a still task focused on balance control and a walking task aimed at managing contact phases while maintaining stability. The generated trajectories for both tasks adhere to contact constraints and align with expected dynamics, validating the effectiveness of the control strategy. The CoM profiles and contact force patterns further confirm the consistency with the expected physical behavior.

A significant challenge throughout this work is tuning the optimization parameters and cost function weights. The trajectory optimizer is highly sensitive to these parameters, and small adjustments can lead to drastically different outcomes. For instance, prioritizing smoothness may result in overly conservative motions that compromise task performance, while aggressive settings can cause instability or violations of physical constraints. Finding the right balance between these objectives involves a lot of trial and error, as each adjustment can introduce new trade-offs. Setting the right weights for each cost term is especially challenging, as even small changes can significantly impact the resulting motion. This process is time-consuming and requires a deep understanding of how each term influences behavior.

Despite the challenges, the final implementation successfully replicates the reference walking and standing behaviors from the original work. The generated trajectories respect contact constraints and produce plausible centroidal motion, confirming the effectiveness of the SBCD model. For future work, we plan to extend the framework to more complex scenarios, such as running and high-jump tasks, to further assess its robustness and flexibility. Additionally, testing the framework under varying model parameters and incorporating uncertainty could provide valuable insights into its applicability in real-world settings, where unexpected disturbances and variations are inevitable.

References

- [1] Yuichi Tazaki. Trajectory generation for legged robots based on a closed-form solution of centroidal dynamics. *IEEE Robotics and Automation Letters*, 2024.
- [2] Shuuji Kajita and Kazuo Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1405–1406. IEEE Computer Society, 1991.
- [3] Stéphane Caron. Biped stabilization by linear feedback of the variable-height inverted pendulum model. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9782–9788. IEEE, 2020.
- [4] Ioannis Poulakakis and Jessy W Grizzle. The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper. *IEEE Transactions on Automatic Control*, 54(8):1779–1793, 2009.
- [5] David E Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous robots*, 35:161–176, 2013.
- [6] Johannes Englsberger, Christian Ott, and Alin Albu-Schäffer. Three-dimensional bipedal walking control based on divergent component of motion. *Ieee transactions on robotics*, 31(2):355–368, 2015.
- [7] Robert J Full and Daniel E Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of experimental biology*, 202(23):3325–3332, 1999.
- [8] Philip Holmes, Robert J Full, Dan Koditschek, and John Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM review*, 48(2):207–304, 2006.
- [9] Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE transactions on automatic control*, 48(1):42–56, 2003.
- [10] Johannes Englsberger, George Mesesan, and Christian Ott. Smooth trajectory generation and push-recovery based on divergent component of motion. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4560–4567. IEEE, 2017.
- [11] Takumi Kamioka, Hiroyuki Kaneko, Toru Takenaka, and Takahide Yoshiike. Simultaneous optimization of zmp and footsteps based on the analytical solution of divergent component of motion. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1763–1770. IEEE, 2018.

- [12] Russ Tedrake, Scott Kuindersma, Robin Deits, and Kanako Miura. A closed-form solution for real-time zmp gait generation and feedback stabilization. In *2015 IEEE-RAS 15th international conference on humanoid robots (humanoids)*, pages 936–940. IEEE, 2015.
- [13] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa 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)*, volume 2, pages 1620–1626. IEEE, 2003.
- [14] Masaki Murooka, Mitsuharu Morisawa, and Fumio Kanehiro. Centroidal trajectory generation and stabilization based on preview control for humanoid multi-contact motion. *IEEE Robotics and Automation Letters*, 7(3):8225–8232, 2022.
- [15] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40:429–455, 2016.
- [16] Luca Rossini, Enrico Mingo Hoffman, Seung Hyeon Bang, Luis Sentis, and Nikos G Tsagarakis. A real-time approach for humanoid robot walking including dynamic obstacles avoidance. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.
- [17] Jiatao Ding, Chengxu Zhou, Songyan Xin, Xiaohui Xiao, and Nikolaos G Tsagarakis. Nonlinear model predictive control for robust bipedal locomotion: exploring angular momentum and com height changes. *Advanced Robotics*, 35(18):1079–1097, 2021.
- [18] Hongkai Dai and Russ Tedrake. Planning robust walking motion on uneven terrain via convex optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 579–586. IEEE, 2016.
- [19] Hervé Audren, Joris Vaillant, Abderrahmane Kheddar, Adrien Escande, Kenji Kaneko, and Eiichi Yoshida. Model preview control in multi-contact motion-application to a humanoid robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4030–4035. IEEE, 2014.
- [20] Yuichi Tazaki. Fast multi-contact motion planning based on best-neighbor search of contact sequences. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 277–284. IEEE, 2022.