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

Academic Year 2024/2025

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 Trajectory Generation 14

5.2 Implementation Details 16

5.3 Still Task: results 18

5.4 Walking Task: results 21

6 Conclusion **26**

References **27**

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 (SBCD) 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 traejctory 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 (CoM) 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—without 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:

$$m\,\ddot{\mathbf{p}} = \mathbf{f} \;-\; m\,\mathbf{g}, \tag{1a}$$
$$\dot{\mathbf{L}} = \boldsymbol{\eta}. \tag{1b}$$

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 are 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

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

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

$$\mathbf{f} \;=\; \sum_{l=1}^{n_e} \mathbf{f}_l, \qquad \boldsymbol{\eta} \;=\; \sum_{l=1}^{n_e} \Big[(\mathbf{p}_l - \mathbf{p}) \times \mathbf{f}_l \;+\; \boldsymbol{\eta}_l\Big]. \tag{2}$$

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:

$$\boxed{\mathbf{f}_l = m\,\lambda_l^2\Big(\mathbf{p} \;-\; (\mathbf{p}_l + \mathbf{r}_l)\Big), \qquad \boldsymbol{\eta}_l = m\,\lambda_l^2\,\hat{\boldsymbol{\eta}}_l.} \tag{3}$$

Intuitively:

- λ_l^2 scales like a contact stiffness: larger $\lambda_l \to$ 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

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

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

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

where

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

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

$$
\begin{aligned}
\ddot{\mathbf{p}} &= \bar{\lambda}^2\big(\mathbf{p} - (\bar{\mathbf{p}} + \bar{\mathbf{r}})\big), \qquad (6a) \\
\dot{\mathbf{L}} &= m\big(\ddot{\mathbf{p}} \times \bar{\mathbf{r}} + \bar{\boldsymbol{\eta}}\big). \qquad (6b)
\end{aligned}
$$

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 $\epsilon = 0$ and (6) 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

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

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

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

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

$$\bar{\lambda}_k = \sqrt{\sum_{l=1}^{n_e} \lambda_{l,k}^2 + \epsilon^2}, \quad \bar{\mathbf{p}}_k = \frac{\sum_{l=1}^{n_e} \lambda_{l,k}^2\, \mathbf{p}_{l,k} + \mathbf{g}}{\bar{\lambda}_k^2}, \quad \bar{\mathbf{r}}_k = \frac{\sum_{l=1}^{n_e} \lambda_{l,k}^2\, \mathbf{r}_{l,k}}{\bar{\lambda}_k^2},$$

$$\bar{\boldsymbol{\eta}}_k = \bar{\lambda}_k^2\,(\bar{\mathbf{p}}_k \times \bar{\mathbf{r}}_k) \;+\; \sum_{l=1}^{n_e} \lambda_{l,k}^2\big(\hat{\boldsymbol{\eta}}_{l,k} - \mathbf{p}_{l,k} \times \mathbf{r}_{l,k}\big).$$

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

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

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

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

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

where $\Delta t = t - t_k$ and

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

Finally, substituting into the angular-momentum equation

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

and integrating from t_k to t gives

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

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

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

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}, $\hat{\mathbf{L}}_{\mathrm{ref}}$ (e.g. from a nominal whole-body motion), we solve for

$$\boldsymbol{\omega}(t) = \; I_{\mathrm{sys}}(\mathbf{R})^{-1}\big(\mathbf{L} - \mathbf{R} \, \hat{\mathbf{L}}_{\mathrm{ref}}\big) \approx \mathbf{R} \, I_{\mathrm{ref}}^{-1}\big(\mathbf{R}^\top \mathbf{L} - \hat{\mathbf{L}}_{\mathrm{ref}}\big). \tag{9}$$

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

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

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

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

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:

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

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:

$$t_{k+1} = t_k + \tau_k \tag{8}$$

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

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

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

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:

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

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:

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

where $(*)^{\text{ref}}$ represents the reference (target) values for the state $\mathbf{x}_k$ and the control input $\mathbf{u}_k$. A waypoint-tracking task is considered, where a series of intermediate waypoints are specified for the CoM, base link, and the ends. Desired positions and velocities along a smooth path connecting these waypoints are generated using spline curves. The desired stiffness values are computed by solving the following least-squares optimization problem at each time step k:

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

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 :

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

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:

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

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

$$\tau_{\text{min}} \leq \tau \leq \tau_{\text{max}}, \tag{16}$$

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

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

$$|f_t| \le \mu f_n \Longrightarrow \sqrt{f_{l,x}^2 + f_{l,y}^2} \le \mu f_{l,z} \tag{18}$$

where μ is the static friction coefficient.
Constraints on the moments at the contact point are expressed as:

$$-c_{\max,x} f_{l,z} \le \eta_{l,x} \le c_{\max,x} f_{l,z}, \tag{19}$$

$$c_{\min,y} f_{l,z} \le \eta_{l,y} \le c_{\max,y} f_{l,z}, \tag{20}$$

$$-\mu_z f_{l,z} \le \eta_{l,z} \le \mu_z f_{l,z}, \tag{21}$$

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:

$$g(\mathbf{x}_k, \mathbf{u}_k) \ge 0, \tag{22}$$

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

$$L_{\text{limit}}(\mathbf{x}_k, \mathbf{u}_k) = \sum_{i=1}^{n_g} -\log \max(\epsilon, g_i(\mathbf{x}_k, \mathbf{u}_k)), \tag{23}$$

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:

$$
\begin{aligned}
J_{\text{compl},k} = w_{\text{compl}}^2 \sum_l \Bigg(& \underbrace{\sum_i \delta\left[\sigma_{l,k} = i\right] \left(\boldsymbol{\eta}_i^\top (\mathbf{p}_{l,k} - \mathbf{o}_i)\right)^2}_{\text{contact distance constraint}} \\
& + \underbrace{\delta\left[\sigma_{l,k} \neq \emptyset\right] \left(\|\mathbf{v}_{l,k}\|^2 + \|\boldsymbol{\omega}_{l,k}\|^2\right)}_{\text{zero velocity constraint}} \\
& + \underbrace{\delta\left[\sigma_{l,k} = \emptyset\right] \lambda_{l,k}^2}_{\text{zero stiffness constraint}} \Bigg)
\end{aligned}
\tag{24}
$$

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. 24) 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 :

$$
J[\boldsymbol{\sigma}] = \sum_k \left[L_{\text{task},k} + L_{\text{limit},k} + L_{\text{compl},k}[\boldsymbol{\sigma}_k]\right] \tag{25}
$$

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

$$
\begin{aligned}
&\text{find} \quad \mathbf{x}, \mathbf{u} \quad \text{that minimizes} \quad J[\boldsymbol{\sigma}](\mathbf{x}, \mathbf{u}) \\
&\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)
\end{aligned}
\tag{26}
$$

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

5 Simulation and Results

This section presents the results obtained using the proposed method. We begin by describing the reference trajectory used for each task. Subsequently, relevant plots, tables, and simulations are provided to illustrate system behavior and performance. Before introducing the implemented trajectories, we define the contact state notation:

- **0** indicates the foot is in contact with the ground.
- **-** indicates the foot is not in contact (i.e., lifted).

Note that the trajectories are described with respect to contact phases, not continuous time.

5.1 Trajectory Generation

Still Task

The objective of the still task is for the robot to remain stationary, maintaining its initial configuration throughout the entire duration of the trajectory. In this scenario, both feet stay in continuous contact with the ground, ensuring complete stability and zero locomotion. The contact sequence used is shown in Table 1, where the first row represents the right foot and the second row the left foot.

Task	**N**	**Contact Sequence**
Still	4	0 0 0 0 0 0 0 0

Table 1: Contact sequence for still task

The reference trajectory for the "Still Task" is generated and updated in a step-by-step process. First, the system initializes two main components: the reference state vector, which represents the robot's position and motion, and the reference input vector, which contains the control inputs. The dimensions of these vectors are defined based on the number of phases in the task. Next, the initial values for the robot's Center of Mass (CoM), foot positions, and orientations are set in the reference state vector. At this point, time is set to zero

Algorithm 1 Reference Trajectory Initialization and Update for Still Task

Initialize $X_{\text{ref}} \in \mathbb{R}^{28\times(N+1)}$, $U_{\text{ref}} \in \mathbb{R}^{27\times N}$
Set initial CoM state, feet positions, and orientations in X_{ref}
time $\leftarrow 0$
for $t = 0$ to $N - 1$ **do**
    Read current contact state from σ
    Set phase duration τ and contact force gains λ
    Update U_{ref} with the above
    Set CoM and feet position, velocities, and orientation as the initial state values
    time $\leftarrow$ time+ phase duration
    Update $X_{\text{ref}}(t + 1)$ with the above
end for
return $X_{\text{ref}}, U_{\text{ref}}$

Walking Task

For the walking task, the trajectory is constructed based on the following principles: During double-support phases, where both feet are on the ground ([0, 0]), the Center of Mass (CoM) remains stationary. Movement along the X-axis occurs during single-support phases, when one foot is lifted ([-, 0] or [0, -]). If the preceding phase is double support, the CoM position remains unchanged; if the preceding phase is single support, the CoM advances by a specified displacement. The Z-coordinate (height) of each foot is zero at the beginning of each phase, as the feet are in contact with the ground. The X-coordinate increases alternately, depending on which foot was lifted in the previous phase. The foot in motion follows a trajectory that surpasses the stationary foot, simulating a natural walking pattern. Each foot's intra-phase height position is interpolated using a parabolic profile, starting and ending at the positions determined by the phase solutions, rather than employing fixed inputs (zero-hold). The contact sequence used is presented in Table 2.

Table 2: Contact sequence for walking task

Task	**N**	**Right Foot**	**Left Foot**
Walk	24	000-000-000-000-000-0	0-000-000-000-000-000

It starts by setting up matrices for the reference trajectory states (X_ref for positions and U_ref for control inputs) and defining the initial conditions for the center of mass (CoM), foot positions, and orientations. Then, for each step (from 0 to N-1), the algorithm reads the contact states (which indicate whether the foot is in contact with the ground), sets the phase duration (the length of time each walking phase lasts), and adjusts the control parameters accordingly.

Algorithm 2 Reference Trajectory Initialization and Update for Walking Task

Initialize $X_{\text{ref}} \in \mathbb{R}^{28\times(N+1)}$, $U_{\text{ref}} \in \mathbb{R}^{27\times N}$
Set initial CoM state and feet state in X_{ref}
time $\leftarrow 0$
for $t = 0$ to $N-1$ **do**
    Read contact states from σ
    Set phase duration and contact force gains λ
    Update U_{ref} with the above
    Set CoM velocity and feet velocities, according to which foot is about to move
    Set CoM and feet positions based on velocity and duration ($p = v * t$)
    time = time + phase duration
    Update $X_{\text{ref}}(t+1)$ with the above
end for
return $X_{\text{ref}}, U_{\text{ref}}$

5.2 Implementation Details

In this section, we provide a summary of the key parameters and choices made in the implementation, particularly distinguishing between the still and walking scenarios. These parameters govern the state initialization, the dynamics, and the optimization settings for the robot's movement.
The solution for both tasks is computed with ipopt solver from Casadi optimizer. Both the reference and solution are contact-dependent, meaning that X and U contain values for each contact phase, thus to find the time dependent solution we used the dynamics. Let's first define the parameters and values that are used both fot still and walking tasks.

Parameter	**Value**	**Units**
Initial CoM Position	(-0.00062, 0.00044, 0.724)	m
Initial Feet Positions	(0, -0.102, 0)) (0, 0.102, 0))	m
Initial Feet Orientations	(1, 0, 0, 0) (1, 0, 0, 0)	-
Feet Length	0.1	m
Gravity Factor (g)	$[0, 0, 9.81]$	$\mathrm{m/s^2}$

Table 3: Initial Parameters

Still Task

For the still scenario, the robot begins with a fixed initial state, with the Center of Mass (CoM) positioned at a specific point, neutral orientations for the feet, and zero velocities. The system is initialized with parameters that ensure no significant movement, providing a static reference for comparison with dynamic cases. The different parameter for the still scenario concerns initial velocity, which is set to 0. In this setup, the robot is essentially in a static position, with minimal movement.

Cost and Weighting Matrices

All parameters weights are set to 1, except the following

State Component	**Weight**
p_y	500
p_z	500
v_k	300
L_k	0.0001
$P_{L,k}$	100
$Q_{L,k}$	0.0001

Table 4: State Weight Matrix Still ($W_{x,k}$)

The inputs weights are all set to one.

Walking Task

In contrast, for the walking scenario, the robot's state is dynamic, with the CoM moving in response to lifting one foot while the other remains grounded. The robot's motion is controlled by optimizing the contact forces and foot movements according to the phases of walking. Here, the CoM velocity is initialized in the negative y-direction to simulate walking forward. The different parameters for the walking scenario concerns initial velocity which is set to [0, -0.07, 0]. In the walking scenario, the optimization process adjusts the robot's foot positions and orientations, ensuring that the forces exerted during the gait cycle are balanced, with the robot transitioning between phases of double and single support. This approach is essential for dynamic stability during motion.

Cost and Weighting Matrices

All parameters state weights are set to 1, except the following

State Component	Weight
p_y	500
p_z	500
v_k	3
L_k	0.0001
$P_{L,k}$	100
$Q_{L,k}$	0.0001

Table 5: State Weight Matrix Walking ($W_{x,k}$)

The parameters inputs weights are all set to one.

Optimization Parameters

Finally, optimization parameters, used for both tasks, such as the complementarity weight (w_{compl}), the friction coefficients (μ and μ_z), and the maximum and minimum values for the phase duration ($\tau_{\min}$ and $\tau_{\max}$) are set. These values ensure that the optimization is constrained by the physical limits of the robot's actuators and the forces exerted by the environment.

Parameter	Value
Complementarity Weight (w_{compl})	1000
Friction Coefficient (μ)	0.5
Friction Coefficient in Z (μ_z)	0.6
Maximum Phase Duration ($\tau_{\max}$)	10
Minimum Phase Duration ($\tau_{\min}$)	0.4

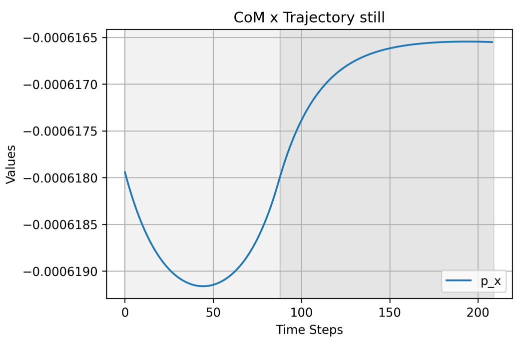
Table 6: Optimization Parameters

5.3 Still Task: results

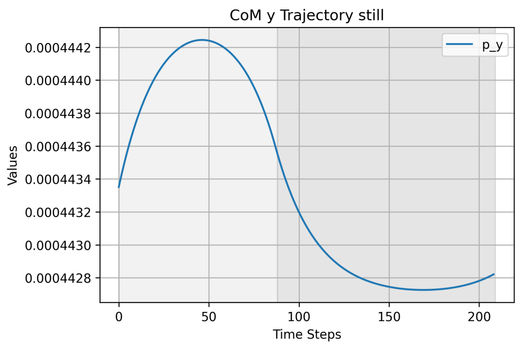
In this case the state vector has dimension 3, and the input vector has dimension 2.

CoM Plots

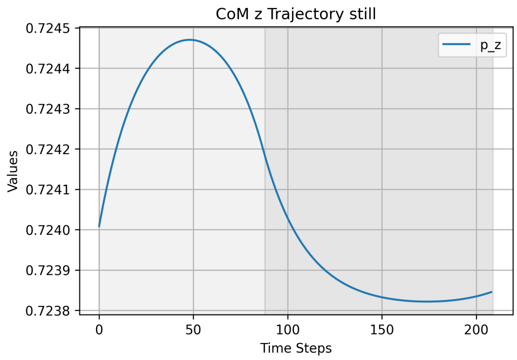
The following plots report the CoM trajectory in time. The alternation between light and dark grey on the background suggests the switch between phases. As observed, there is negligible motion in any direction, confirming the robot's stationary state.



(a) Com X Trajectory 1



(b) Com Y Trajectory 1



(c) Com Z Trajectory 1

Figure 1: Com Trajectory

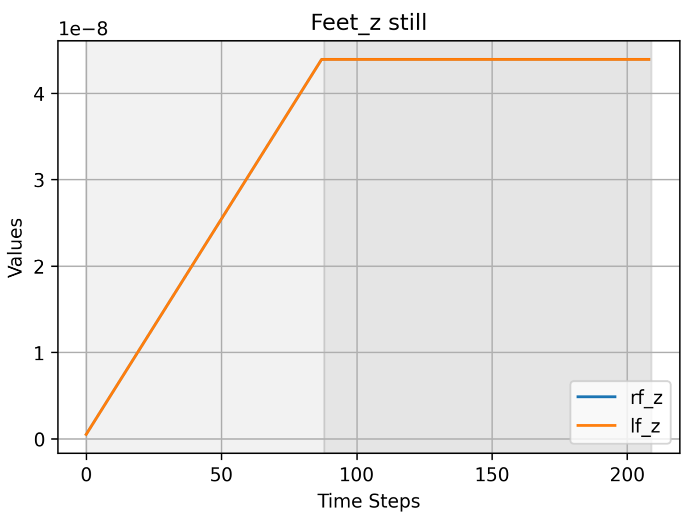


Figure 2: Feet along Z (scale is 1e-8)

State and Input Contact Values

In the following plots, the main components of the state and input vectors solutions are compared with the reference at each contact step.

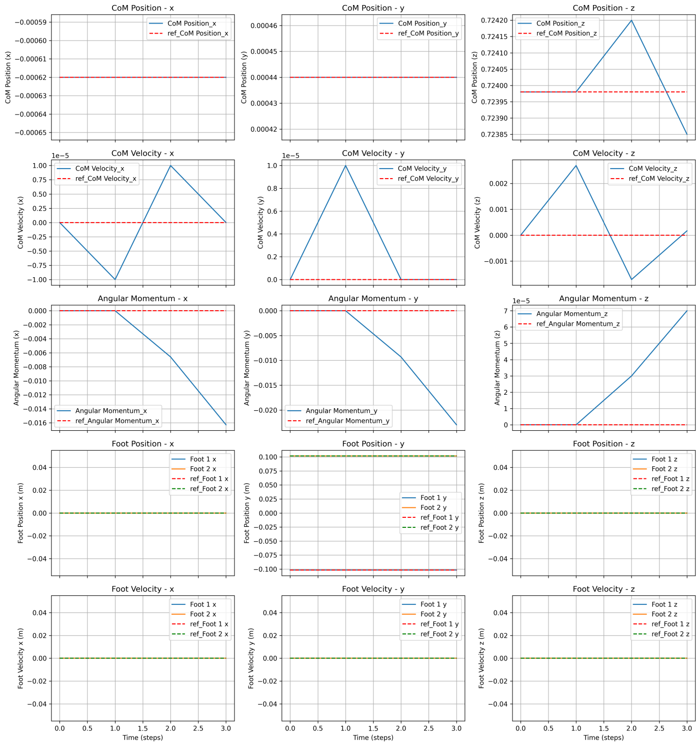


Figure 3: Trajectory vs Reference: state vector

Contact Forces

Another important dynamic aspect pertains to the forces involved. The input vector includes the contact wrench, which describes the mechanical influence that a contact point (such as a foot or hand) exerts on the robot, or vice versa. In order to satisfy the dynamic balance laws, the environment should exert a reaction force along the Z-axis that is equal to the gravitational force multiplied by the robot's mass. Table 7 below lists the contact forces exerted by the environment on both feet. As shown in the table, when both feet are on the ground, the gravitational force is evenly distributed between the left and right foot.

Right Foot Z	Left Foot Z	Σ_L^k
49.0644	49.0531	[0., 0.]
48.8976	49.9925	[0., 0.]
48.0847	49.0895	[0., 0.]

Table 7: Z-axis gravity force and Σ_L^k values

Components of contact wrench, rotational and translational forces, are graphically expressed in the following plots:

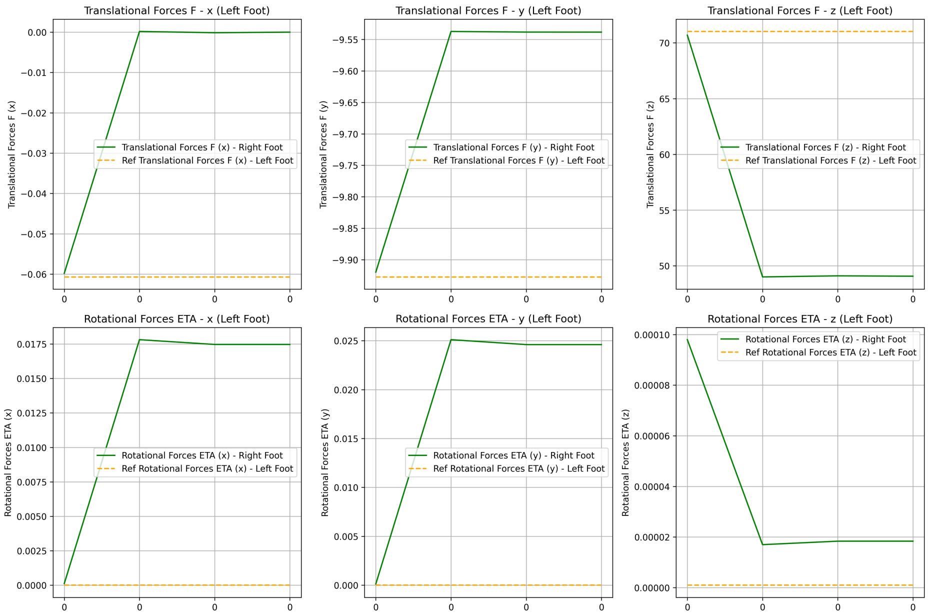


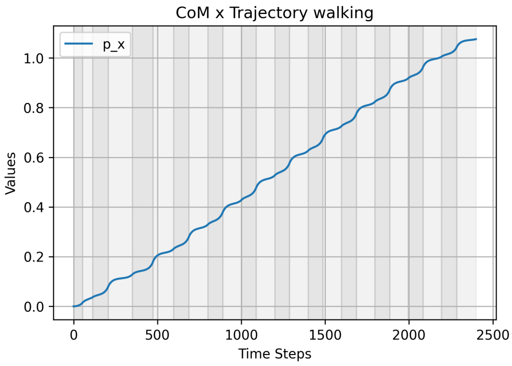
Figure 4: Trajectory vs Reference: Forces

5.4 Walking Task: results

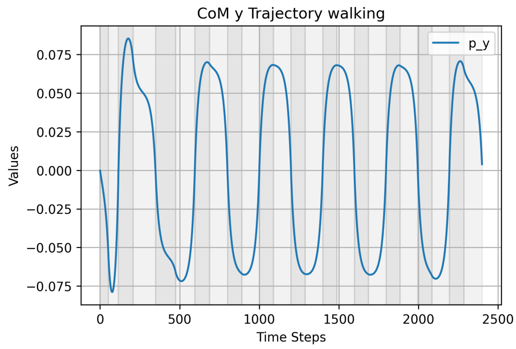
In this case the state vector has dimension 24 and the input vector has dimension 23.

CoM Plots

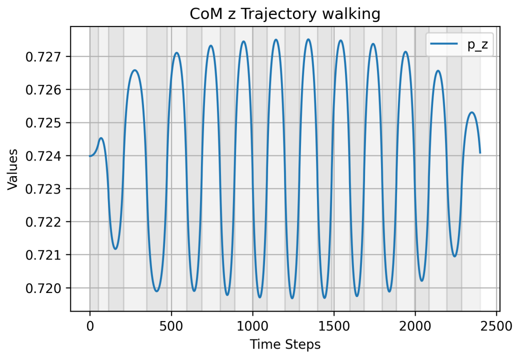
The following plots illustrate the trajectory of the Center of Mass (CoM) over time. The alternating light and dark grey backgrounds in the plots indicate the transitions between different phases of the walking motion. As shown in Fig. **??**, the CoM exhibits linear motion along the x-axis. In the y-direction, as illustrated in Fig. **??**, the CoM moves toward the foot that remains on the ground while the other foot is lifted. Specifically, if we consider the first two contact phases: during the first phase, the robot is in double support (both feet on the ground), while in the second phase, the left foot is expected to lift. At the end of the first phase, the CoM has shifted towards the right foot, demonstrating the transition in weight distribution. This pattern of alternating between double and single support phases continues throughout the walking cycle, with the CoM constantly adjusting its position. In the z-direction, shown in Fig. **??**, the CoM exhibits a slight up-and-down motion, with a relatively small range of displacement. This vertical motion is consistent with the natural bobbing motion of the body during walking, providing necessary stability and balancing forces. This pattern of CoM movement is crucial for maintaining dynamic stability as the robot progresses through each phase of the gait. In summary, the CoM trajectory provides key insights into the robot's walking mechanics, including the shift in weight distribution during transitions between support phases and the vertical adjustments required for balance. These movements are essential for achieving stable locomotion.



(a) Com X Trajectory 1



(b) Com Y Trajectory 1



(c) Com Z Trajectory 1

Figure 5: Com Trajectory

Furthermore, the image below represent the foot position along the Z-axis in time. As one can see, they follow a parabolic profile.

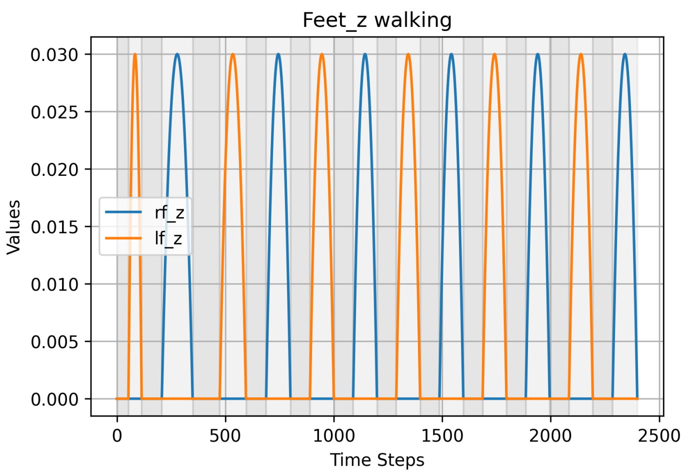


Figure 6: Feet along Z

State and Input Contact Values

As previously said, in the following plots, the main components of the state and input vectors solutions are compared with the reference at each contact step.

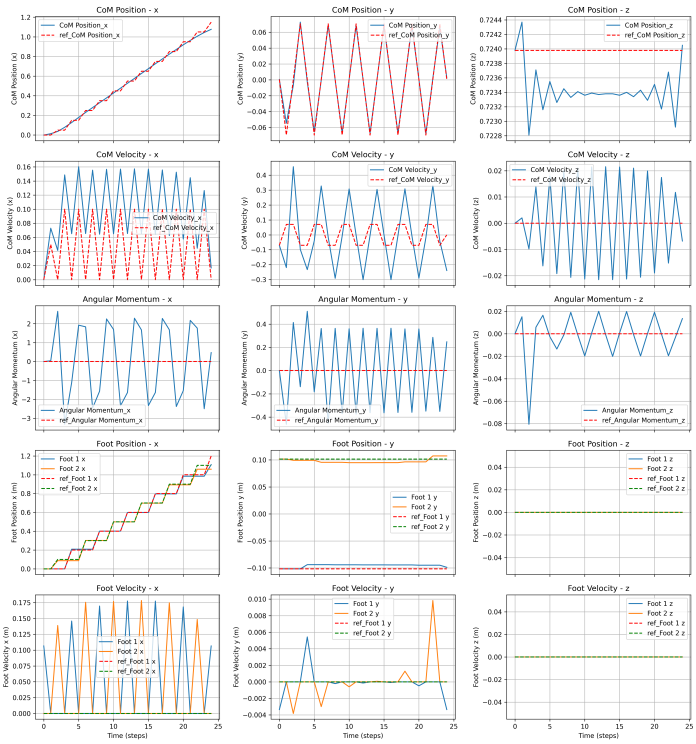


Figure 7: Trajectory vs Reference: state vector

Contact Forces

Another key dynamic aspect, which we have already discussed, concerns the forces involved. Specifically, the input vector includes the contact wrench, which captures the mechanical interaction between the robot and contact points, such as the feet or hands. To maintain dynamic balance, the environment must exert a reaction force along the Z-axis equal to the gravitational force, which is the robot's mass multiplied by gravity. Table 8 below lists the contact forces exerted by the environment on both feet. As shown, when both feet are on the ground, the gravitational force is evenly distributed between the left and right foot, while during single support, the force is applied only to the foot that remains on the ground.

Right Foot Z	Left Foot Z	Σ_L^k
49.0644	49.0531	[0., 0.]
97.9551	0	[0., -]
48.8973	49.65	[0., 0.]
0	97.4904	[-, 0.]
49.6528	49.1563	[0., 0.]
97.3081	0	[0., -]
49.2399	49.7241	[0., 0.]
0	97.2091	[-, 0.]
49.7388	49.293	[0., 0.]
97.1669	0	[0., -]
49.3007	49.7613	[0., 0.]
0	97.1486	[-, 0.]
49.762	49.31	[0., 0.]
97.1443	0	[0., -]
49.3062	49.7655	[0., 0.]
0	97.1495	[-, 0.]
49.7557	49.3045	[0., 0.]
97.1685	0	[0., -]
49.2815	49.7467	[0., 0.]
0	97.216	[-, 0.]
49.7003	49.254	[0., 0.]
97.3264	0	[0., -]
49.1286	49.6485	[0., 0.]
0	97.5828	[-, 0.]

Table 8: Z-axis gravity force and Σ_L^k values

Again, components of contact wrench, rotational and translational forces, are graphically expressed in the following plots:

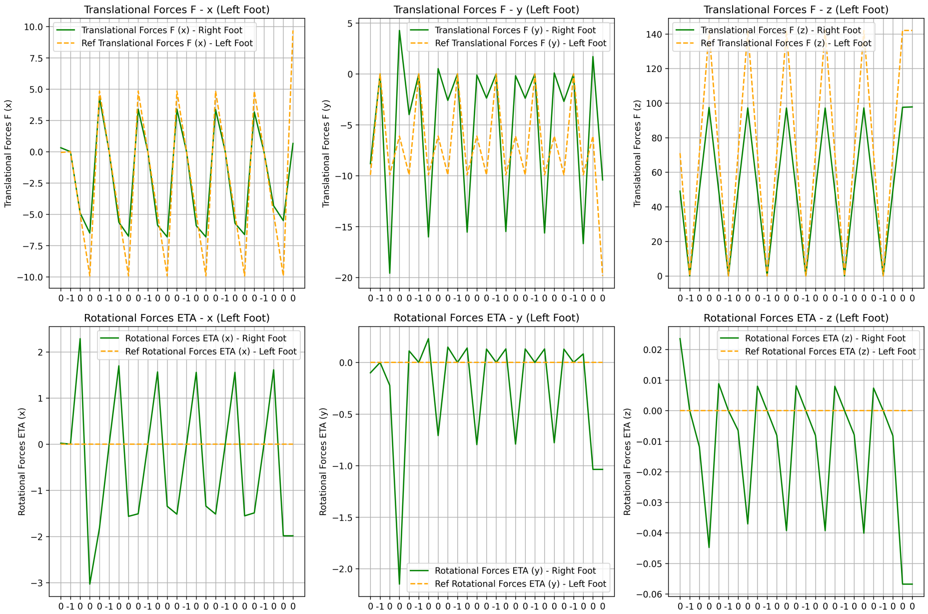


Figure 8: Trajectory vs Reference: Forces

Robot representation

In this section it is possible to visualize the path followed by the CoM, right and left foot in time both for still (Fig. 9) and walking (Fig. 10) tasks:

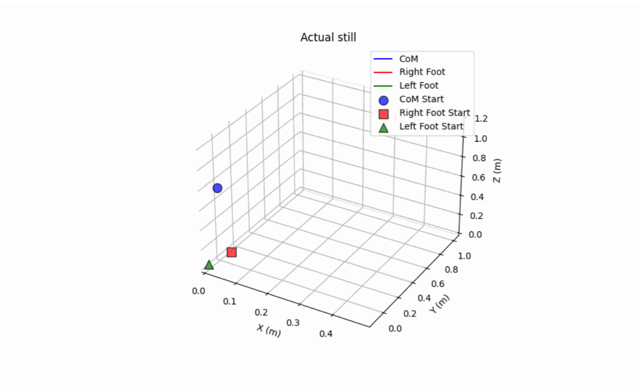


Figure 9: Still

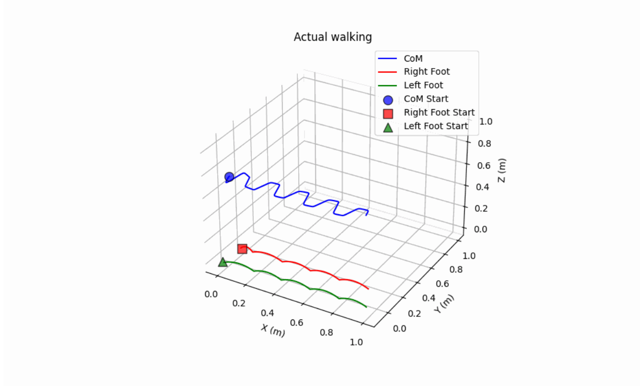


Figure 10: Walking

6 Conclusion

In this project, we implemented and evaluated the Stiffness-Based Centroidal Dynamics (SBCD) model proposed by Tazaki, with a focus on trajectory optimization for both static balancing and dynamic walking tasks. Overall, the SBCD framework presents a compelling alternative to existing reduced-order models, offering a balance between physical fidelity and computational tractability that is well-suited for real-time trajectory planning in legged robots. Infact, this formulation allows for analytic integration over finite time intervals under zero-order hold assumptions and yields closed-form expressions for CoM trajectory and angular momentum evolution. Furthermore, we outlined a practical method for integrating base-link orientation from the centroidal state using quaternion updates and nominal inertia models.

This work presented also a comprehensive simulation framework for generating and analyzing motion trajectories in both static and dynamic scenarios for a bipedal robot. Two primary tasks were examined: a still task, emphasizing balance and immobility, and a walking task, showcasing locomotion with dynamically varying contact phases. The trajectory generation methods effectively incorporated contact state information and phase-dependent parameters to produce physically consistent reference motions.

At the end, simulation results demonstrated that the robot successfully maintained static equilibrium in the still task and achieved stable locomotion in the walking task. For instance, the CoM trajectories profiles and contact forces values confirmed the physical plausibility and effectiveness of the control strategy, while the optimization parameters ensured robust convergence and adherence to realistic constraints.

One of the main challenges encountered during the implementation was the selection and tuning of optimization parameters and cost function weights. Since the behavior of the trajectory optimizer is highly sensitive to these weights, we observed that small variations could lead to drastically different motion outcomes — ranging from overly conservative to physically unstable behaviors. This tuning process required extensive trial and error, as well as manual adjustments to balance competing objectives such as smooth motion, adherence to physical constraints, and task performance.

Despite these difficulties, the final implementation demonstrated successful reproduction of the reference walking and standing behaviors presented in the original work. The generated trajectories respected contact constraints and achieved plausible centroidal motion, validating the effectiveness of the SBCD model.

These results validate the proposed approach and establish a foundation for extending the framework to more complex maneuvers and real-world implementation.

Future work could extend this approach by incorporating model uncertainty, testing the framework on hardware, and exploring more complex multi-contact scenarios. Additionally, the integration of learning-based components for adaptation and robustness in unstructured environments represents a promising direction.

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.