Trajectory Optimization for Humanoids via Centroidal Dynamics

Master's Degree in Artificial Intelligence and Robotics

Autonomous and Mobile Robotics Course

Francesco Danese 1926188
Giuseppina Iannotti 1938436
Alessia Pontiggia 1892079

Professor : **Giuseppe Oriolo**

Supervisor : **Nicola Scianca**

Academic Year
2024/25

SAPIENZA
UNIVERSITÀ DI ROMA

Table Of Contents

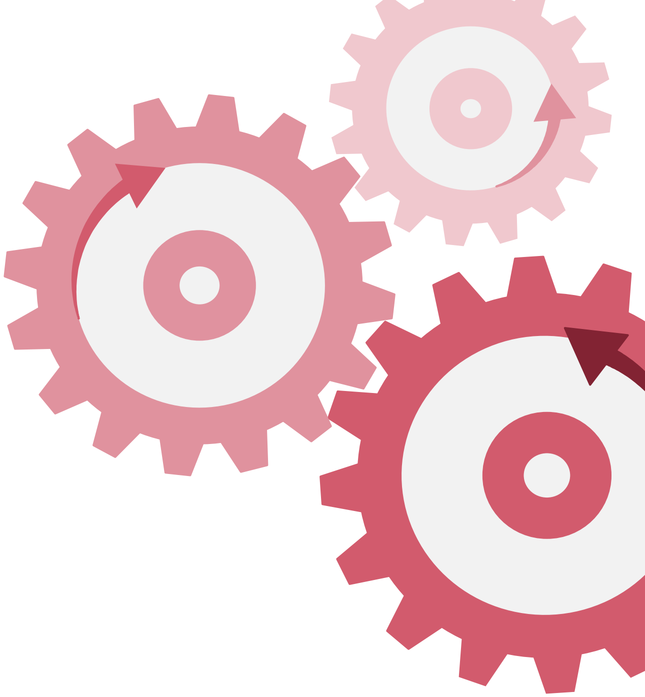
Introduction

Humanoid robots are increasingly used in dynamic tasks such as logistics and human-robot interaction.

However, generating real-time trajectories using full dynamic models is **computationally intensive**. While **reduced-order models** like LIPM [2] are simpler, they lack expressiveness, and more accurate models like SLIP [5] lack closed-form solutions.

This work builds on the **Stiffness-Based Centroidal Dynamics** (SBCD) model [1], which enables **closed-form** computation of CoM and angular momentum trajectories. The framework is implemented and evaluated for walking and balancing tasks.

Work Focus



This work builds on the SBCD framework [1], which models contact wrenches via **stiffness-based parametrization** for compact and tractable centroidal dynamics. The framework captures both **translational and rotational** motion, making it suitable for **long-horizon** trajectory planning. We implement and evaluate the SBCD approach for **walking** and **standing** tasks through simulation.

Related Works

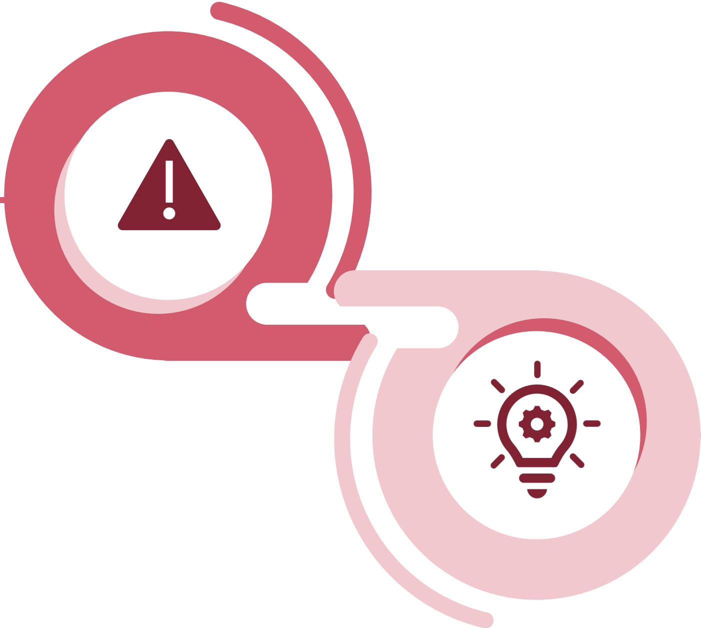
SIMPLIFIED CENTROIDAL MODELS

- **Reduced order models** simplify control in legged robots
- **LIPM[2]** : ZMP-based stabilization with constant CoM height
- **VH-LIPM[3]** : Adds vertical motion and 3D Divergent Component of Motion
- **SLIP[5]** / **ASLIP[6]** : Introduce compliant leg dynamics

INTEGRATION OF TRAJECTORY OPTIMIZATION

- TO uses **reduced order models** like LIPM[2], VH-LIPM[3], and SLIP[5]
- Ensures stability via **ZMP constraints**
- For complex motion, incorporates centroidal dynamics.
- Feasibility is enforced through **ZMP region** or **Centroidal Wrench Cone**

Limitations vs Solutions



SOLUTIONS

- Closed-form dynamics
- **mc-LIPM [8] :** Stiffness-based contact modelling
- **SBCD [1] :** Stiffness-displacement wrench formulation

LIMITATIONS

- High Computational Cost
- Limited Feasibility Assurance
- Modeling simplifications
- Reduced Applicability

Table Of Contents

Centroidal Dynamics

$$m\,\ddot{\mathbf{p}} = \mathbf{f} - m\,\mathbf{g},$$
$$\dot{\mathbf{L}} = \boldsymbol{\eta}.$$

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

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

- $\mathbf{p} \in \mathbb{R}^3$ 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., $\mathbf{g} = [0, 0, -9.81]^\top$)

Spring-like parametrization of contact wrenches

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

- λ_l^2 scales like a contact stiffness : larger λ_l implies a stronger repulsive force
- $\mathbf{p}_l + \mathbf{r}_l$ is the virtual pivot (similar to the 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 Stiffness-Based model

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

From previous slides:

$$m\,\ddot{\mathbf{p}} = \mathbf{f} \;-\; m\,\mathbf{g}$$

$$\mathbf{f} \;=\; \sum_{l=1}^{n_e} \mathbf{f}_l,$$

$$\mathbf{f}_l = m\,\lambda_l^2\Big(\mathbf{p} \;-\; (\mathbf{p}_l + \mathbf{r}_l)\Big)$$

where:

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

Derivation of Stiffness-Based model

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

where:

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

From previous slides:

$$\dot{\mathbf{L}} = \boldsymbol{\eta}$$

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

$$\boldsymbol{\eta}_l = m\,\lambda_l^2\,\hat{\boldsymbol{\eta}}_l.$$

$$\mathbf{f}_l = m\,\lambda_l^2 \Big(\mathbf{p} - (\mathbf{p}_l + \mathbf{r}_l) \Big),$$

$$m\,\ddot{\mathbf{p}} \approx m\,\bar{\lambda}^2 (\mathbf{p} - \bar{\mathbf{p}} - \bar{\mathbf{r}}),$$

Stiffness-Based Centroidal Dynamics

$$\ddot{\mathbf{p}} = \bar{\lambda}^2\big(\mathbf{p} - (\bar{\mathbf{p}} + \bar{\mathbf{r}})\big), \qquad \dot{\mathbf{L}} = m\big(\ddot{\mathbf{p}} \times \bar{\mathbf{r}} + \bar{\boldsymbol{\eta}}\big)$$

Remark (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

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 becomes

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

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

$$\mathbf{p}(t) = (\bar{\mathbf{p}}_k + \bar{\mathbf{r}}_k) \; + \; C_k(\Delta t) \left(\mathbf{p}_k - (\bar{\mathbf{p}}_k + \bar{\mathbf{r}}_k)\right) \; + \; \frac{S_k(\Delta t)}{\bar{\lambda}_k} \mathbf{v}_k$$

$$\mathbf{v}(t) = \dot{\mathbf{p}}(t) = \bar{\lambda}_k \, S_k(\Delta t) \left(\mathbf{p}_k - (\bar{\mathbf{p}}_k + \bar{\mathbf{r}}_k)\right) \; + \; C_k(\Delta t) \, \mathbf{v}_k$$

where $\Delta t = t - t_k$ and

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

Discrete-Time Equations

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

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

Notice That : 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

Analytical solution on [t_k,t_{k+1}]

Finally, substituting into the angular-momentum equation

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

and integrating from t_k to t gives

$$\boxed{\mathbf{L}(t) = \mathbf{L}_k \; + \; m\Big(\left(\mathbf{v}(t) - \mathbf{v}_k\right) \times \bar{\mathbf{r}}_k \; + \; (t - t_k)\,\bar{\boldsymbol{\eta}}_k\Big)}$$

Integration of Base-Link Rotation

In a multi-body system one shows

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

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

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

Discrete quaternion update

Over $[t_k, t_{k+1}]$, we subdivide into n_{div} equal steps

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

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

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

where $\mathbf{q}(\boldsymbol{\theta})$ is the unit quaternion corresponding to the axis-angle $\boldsymbol{\theta} \in \mathbb{R}^3$

Table Of Contents

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 characterizing the system's configuration and motion, while the control input $\mathbf{u}_k$ is defined according to the stiffness-based control strategy

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{q}_k \\ \mathbf{v}_k \\ \mathbf{L}_k \\ t_k \\ \{\mathbf{p}_{l,k}\}_{l=1,\dots,n_e} \\ \{\mathbf{q}_{l,k}\}_{l=1,\dots,n_e} \end{bmatrix}$$

(a) State Variable

$$\mathbf{u}_k = \begin{bmatrix} \tau_k \\ \{\mathbf{v}_{l,k}\}_{l=1}^{n_e} \\ \{\boldsymbol{\omega}_{l,k}\}_{l=1}^{n_e} \\ \{\lambda_{l,k}\}_{l=1}^{n_e} \\ \{\mathbf{r}_{l,k}\}_{l=1}^{n_e} \\ \{\eta_{l,k}\}_{l=1}^{n_e} \end{bmatrix}$$

(b) Control Input

Equations Update

Timestamp Update

The update of the timestamp is defined as :

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

EE Position and Orientation Update

The position and the orientation of each end are updated using basic kinematic relations :

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

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

Integrating all these elements, the system dynamics can be compactly represented by the following state transition function :

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

Task - Related Cost

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

where :

- $\mathbf{x}_k^{\text{ref}}$, $\mathbf{u}_k^{\text{ref}}$ are the reference state and input
- W_k^x , W_k^u are the weighting matrices for prioritizing deviations
- Reference trajectories are manually implemented
- Desired Stiffness values $\lambda_{l,k}^{ref}$ are computed via optimization of the following problem :

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

Inequality Constraints

Position Constraints

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

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

Range Constraints

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

$$\tau_{\min} \leq \tau \leq \tau_{\max}$$

$$0 \leq \lambda_l \leq \lambda_{\max}, \quad \forall l$$

Non - Slip Condition

Next, for each end in contact, the contact wrench must satisfy non-slip and moment conditions

$$\sqrt{f_{l,x}^2 + f_{l,y}^2} \leq \mu f_{l,z}$$

Moment Constraints

Constraints on the moments at the contact point are expressed as

$$-c_{\max,x} f_{l,z} \leq \eta_{l,x} \leq -c_{\min,x} f_{l,z}$$

$$c_{\min,y} f_{l,z} \leq \eta_{l,y} \leq c_{\max,y} f_{l,z}$$

$$-\mu_z f_{l,z} \leq \eta_{l,z} \leq \mu_z f_{l,z}$$

Log-Barrier Function

All the constraints can be compactly represented as :

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

where $g()$ is a differentiable vector-valued function, evaluated component-wise.

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

To handle these constraints, a log barrier function is introduced, with :

- n_g being the number of constraints
- g_i being the i-th constraints function
- ϵ being a small positive constant to prevent instability and avoid undefined values in the logarithm

Contact-Dependent Cost

To ensure consistent interaction between a robot's ends and the environment, a contact-dependent cost is introduced. It is defined as :

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

where :

- $\sigma_{l,k}$ denotes the contact state of the l-th end at time step k
- $\delta[*]$ is an indicator function that returns 1 if the condition inside the brackets is true, otherwise 0

Contact	Meaning
$\sigma_{l,k} = i$	Contact with the i-th surface
$\sigma_{l,k} = \emptyset$	No contact

Contact-Dependent Cost Terms

CONTACT DISTANCE CONSTRAINT

It ensures the distance between the contact end $\mathbf{p}_{l,k}$ and the surface with origin $\mathbf{o}_i$ along its normal $\boldsymbol{\eta}_i$ to be 0

$$\sum_i \delta\left[\sigma_{l,k} = i\right] \left(\boldsymbol{\eta}_i^\top (\mathbf{p}_{l,k} - \mathbf{o}_i)\right)^2$$

ZERO VELOCITY CONSTRAINT

It suppresses unwanted motion at the contact point. It penalizes linear and angular velocities to promote static behavior

$$\delta\left[\sigma_{l,k} \neq \emptyset\right] \left(\|\mathbf{v}_{l,k}\|^2 + \|\boldsymbol{\omega}_{l,k}\|^2\right)$$

ZERO STIFFNESS CONSTRAINT

It prevents spurious forces when the end-effector is not in contact. It discourages nonzero stiffness $\lambda_{l,k}$ to avoid unintended contact force during swing

$$\delta\left[\sigma_{l,k} = \emptyset\right] \lambda_{l,k}^2$$

Optimal Control Problem

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 :

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

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

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

Table Of Contents

Implementation Overview

Our work builds upon the framework provided in the **IS - MPC Repository**, which originally offers a comprehensive platform for simulating the HRP - 4 robot, including the **dynamics model** and **control architecture**. We have adapted it by providing our own implementation of the new dynamics, TO problem, and reference trajectory generation algorithms. This enables the simulation of the HRP - 4 robot performing both the **still** and **walking** tasks, allowing us to assess the robot's performance under our modified control and optimization framework.

Implementation Details

The implementation leverages the **IPOPT** solver from the **CasADi** optimization framework to compute optimal trajectories based on the **SBCD** model.

State and Input - Weights*

	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

Optimization Params

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

Table 2 : Configuration for optimization parameters

Notice That* **: For all other state components and control inputs, weights are uniformly initialized to 1.

Still Task

Task	N	Foot	Contact Sequence
Still	4	Right	0000
		Left	0000

Table 3 : Contact Sequence - Still Task

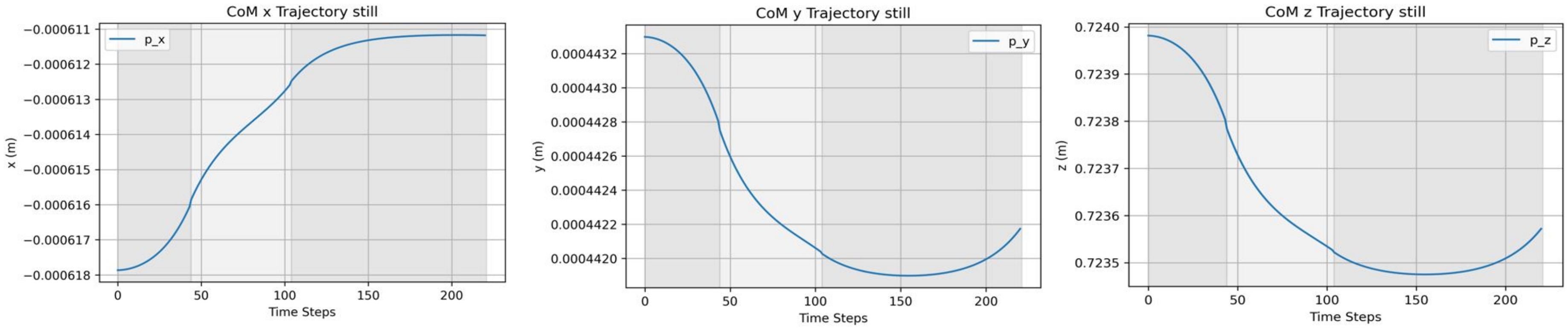
The objective of the still task is to keep the robot stationary. The contact sequence ensures that both feet remain **grounded**, establishing a condition of complete stability without locomotion.

The still task algorithm generates reference state and control trajectories aimed at **preserving** the robot's initial configuration, while **counteracting** gravitational forces through ground reaction forces.

Algorithm 1 Reference Trajectory Still Task Generation

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
Set initial CoM/feet position and orientation and $t_0 \leftarrow 0$ in $X_{ref}(0)$
for $k = 0$ to $N-1$ **do**
    Set $X_{ref}(k+1)$ as a copy of the current state $X_{ref}(k)$
    Set desired phase_duration τ_k and contact force gains λ_k in $Uref(k)$
    Update $X_{ref}(k+1)$ with $t_{k+1} \leftarrow t_k + \tau_k$
end for
return X_{ref}, U_{ref}

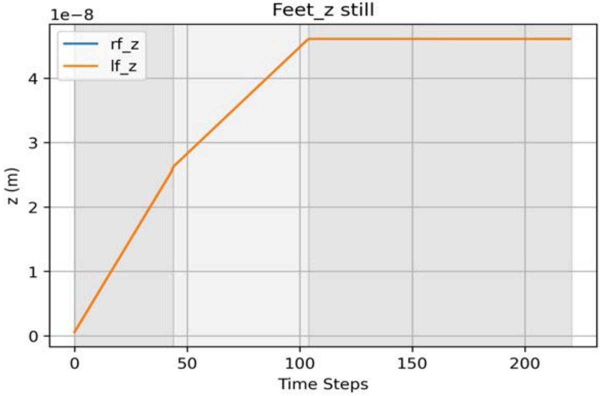
CoM - Still Task

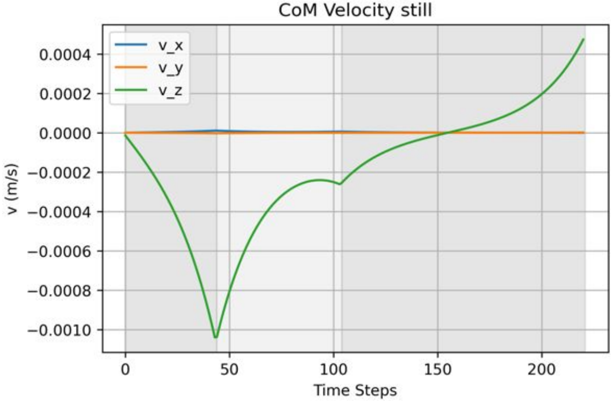


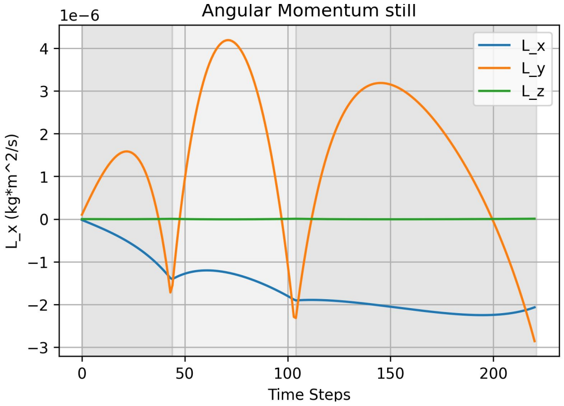
The **x** component displays slight horizontal drift, likely due to adjustments in ground reaction forces, while the **y** component briefly dips before stabilizing, indicating corrective measures to maintain balance. The **z** component initially drops and then recovers, reflecting gravitational compensation.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

Feet, CoM Vel and Momentum - Still Task

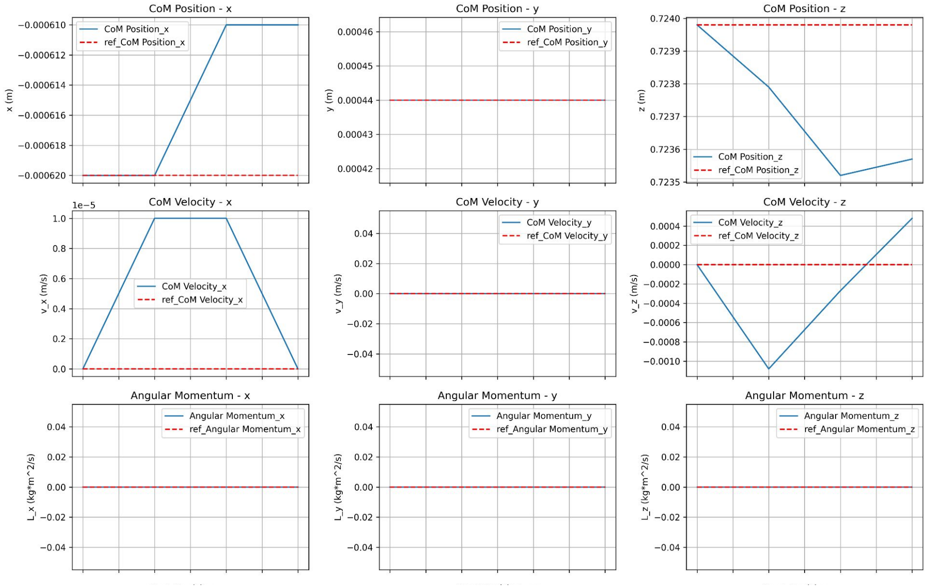
The plots illustrate the stability during the still task, with nearly constant foot positions indicating no stepping motion. CoM velocity shows minimal variation, apart from slight **y** and **z** fluctuations due to minor balance corrections. Angular momentum remains stable, with L_z constant throughout, indicating no rotation around the **z**-axis.





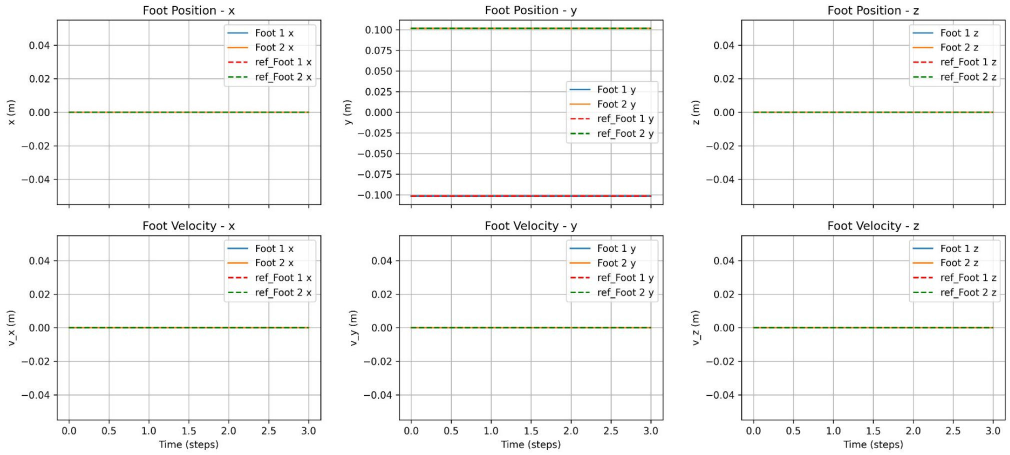


Visualization - Still Task



- 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**

Visualization - Still Task



- Feet positions and velocities remain effectively constant, confirming that the feet maintain their initial stance without significant movement.

Forces - Still Task

To maintain dynamic balance, the reaction force along the *z*-axis must counteract the gravitational force acting on the robot's mass. As indicated in Table 4, 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

Walking Task

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

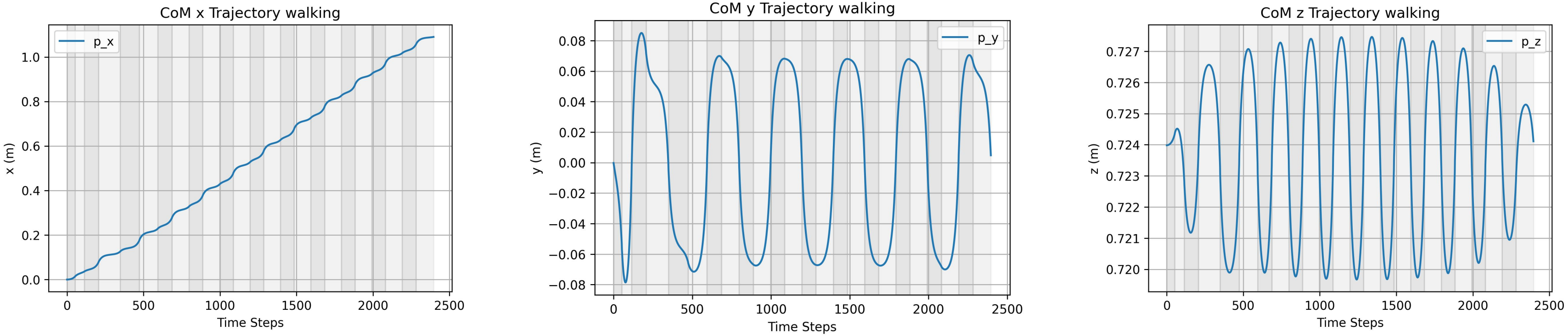
The objective of the walking task is to simulate a **natural walking pattern** by coordinating the movements of the feet and the CoM through a sequence of double-support and single-support phases, as shown in Table 5

The algorithm initializes the robot's state and calculates the CoM and feet velocities based on the transition logic between **double** and **single-support phases**. The CoM trajectory is computed by integrating velocities over the phase duration, while feet positions are updated based on contact states.

Algorithm 2 Reference Trajectory Generation - Walking Task

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
Set initial CoM/feet position and orientation and $t_0 \leftarrow 0$ in $X_{ref}(0)$
for $k = 0$ to $N-1$ **do**
    Read current and future contacts state from $\sigma(k)$ and $\sigma(k+1)$
    Set desired phase_duration τ_k and contact force gains λ_k in $Uref(k)$
    Set desired CoM/feet velocities in $X_{ref}(k), U_{ref}(k)$
    Update CoM and feet positions in $X_{ref}(k+1)$ with $p_{k+1} \leftarrow p_k + v_k \cdot \tau_k$
    Update $X_{ref}(k+1)$ with $t_{k+1} \leftarrow t_k + \tau_k$
end for
return X_{ref}, U_{ref}

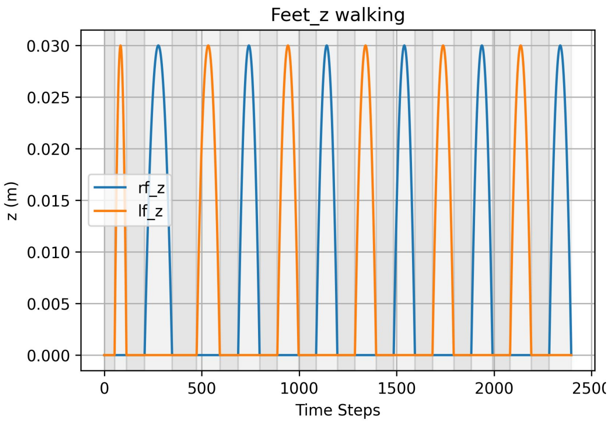
CoM - Walking Task

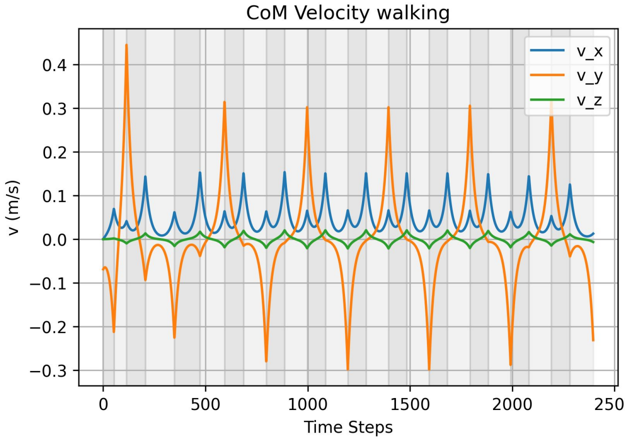


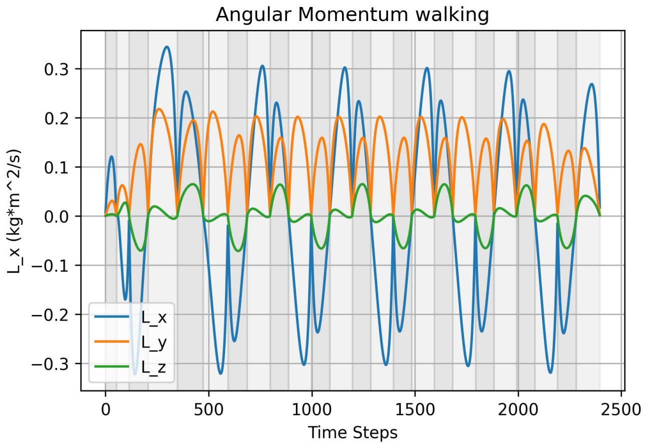
The **x** component steadily increases, indicating consistent forward motion, while the **y** component displays cyclic oscillations corresponding to the lateral sway during each step. The **z** component exhibits **periodic fluctuations**, capturing vertical displacements associated with foot lift-offs and ground contacts

Feet, CoM Vel and Momentum - Walking Task

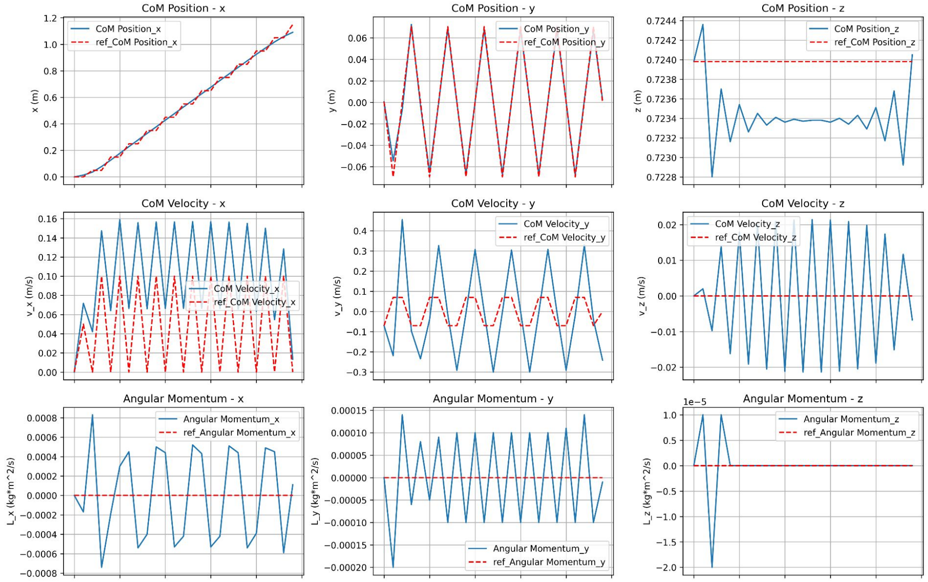
The plots illustrate the cyclic nature of walking through periodic foot trajectories, oscillations in CoM velocity, and shifts in angular momentum. The vertical foot motion marks swing and stance phases, while CoM velocity changes reflect alternating support and weight transfer. Angular momentum variations, especially in the **x** direction, indicate corrective actions for maintaining balance.





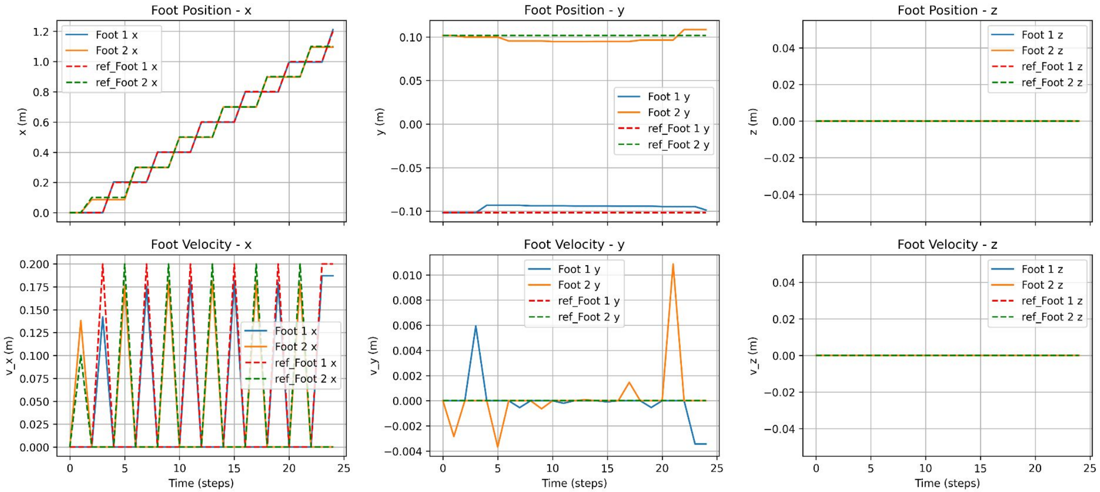


Visualization - Walking Task



- CoM position plots show a consistent **forward motion** along the **x**, **lateral oscillations** in the **y**, and minimal **vertical fluctuations** in the z - axis, indicating stable walking dynamics.
- CoM velocities have **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.

Visualization - Walking Task



- 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.

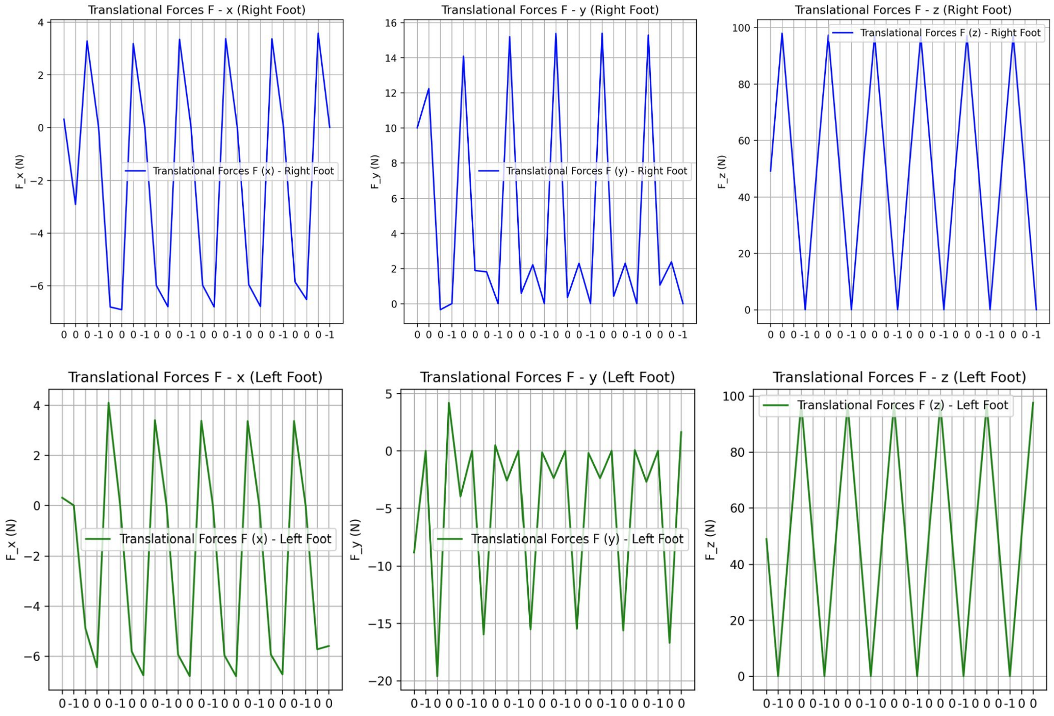
Forces - Walking Task

The sum of the contact forces along the **z** 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)
	...	...		...
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)

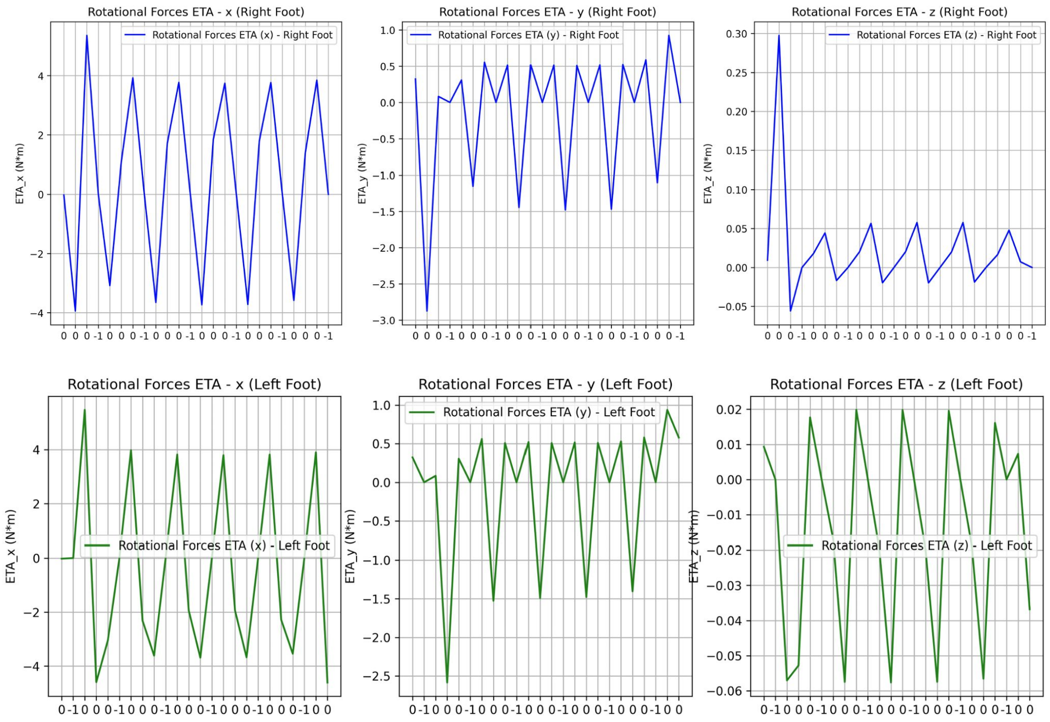
Table 7: Summary of Forces and Foot Positions per Interval - Walking Task

Translational Forces - Walking Task



- The robot mass is 10kg. Translational contact forces at the feet capture the **cyclic nature** of the gait, balancing its weight of ≈ 98.8 N on both feet during double support (49N per foot), and on one feet during single support phases (0N and 98N).

Rotational Forces - Walking Task



- Since the feet desired orientation is kept identical to the initial one at every phase, the rotational forces according to the solutions are close to 0 in each dimension, signifying minimal changes in orientation

Videos - Walking Task

Table Of Contents

Conclusions and Future Works

Challenges

Addressing the sensitivity in tuning optimization parameters and during the creation of the reference trajectories

Future Works

Extending the framework to more complex tasks like running and high jump scenarios

Implementation

Utilizing the Stiffness-Based Centroidal Dynamics (SBCD) model for trajectory optimization

Results

Successfully replicating walking and standing behaviors, validating the SBCD model's effectiveness

Table Of Contents

References

[1] Tazaki, Yuichi. "Trajectory generation for legged robots based on a closed-form solution of centroidal dynamics." *IEEE Robotics and Automation Letters* (2024).

[2] Kajita, Shuuji, and Kazuo Tani. "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode." *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 1991.

[3] Caron, Stéphane. "Biped stabilization by linear feedback of the variable-height inverted pendulum model." *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[4] Englsberger, Johannes, Christian Ott, and Alin Albu-Schäffer. "Three-dimensional bipedal walking control based on divergent component of motion." *Ieee transactions on robotics* 31.2 (2015): 355-368.

[5] Holmes, Philip, et al. "The dynamics of legged locomotion: Models, analyses, and challenges." *SIAM review* 48.2 (2006): 207-304.

[6] Poulakakis, Ioannis, 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 (2009): 1779-1793.

[7] Westervelt, Eric R., Jessy W. Grizzle, and Daniel E. Koditschek. "Hybrid zero dynamics of planar biped walkers." *IEEE transactions on automatic control* 48.1 (2003): 42-56.

[8] Tazaki, Yuichi. "Fast multi-contact motion planning based on best-neighbor search of contact sequences." *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022.

Trajectory Generation for Legged Robot Based on a Closed-form Solution of Centroidal Dynamics

Thank you for listening!
Any questions?