SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER CONTROL AND MANAGEMENT

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

AUTONOMOUS AND MOBILE ROBOTICS

Professors:
Giuseppe Oriolo
Nicola Scianca

Students:
Francesco Danese,
Giuseppina Iannotti,
Alessia Pontiggia

Academic Year 2024/2025

Contents

1 Introduction **2**

2 Related Works **3**

2.1 Centroidal Dynamics Basics 3

2.2 Reduced Order Models for Trajectory Generation 3

2.2.1 Linear Inverted Pendulum Mode 3

2.2.2 Variable-Height Linear Inverted Pendulum Model 3

2.2.3 Spring-Loaded Inverted Pendulum Model 3

2.3 Integration of Trajectory Optimization 4

2.4 Limitations 4

3 Proposed Method **6**

3.1 Equations 6

3.2 Closed Form Solution and Discrete Time Equations 6

3.3 Integration of Base Link Rotation 6

4 Formulation of Trajectory Optimization Problem **7**

4.1 Costs 7

4.1.1 Task Related Cost 7

4.1.2 Inequality Constraints 7

4.1.3 Contact Dependent Cost 9

4.1.4 Final Cost Function and Problem Formulation 10

5 Simulation and Results **11**

5.1 Trajectory Generation 11

5.1.1 Still Task 11

5.1.2 Walking Task 12

5.2 Solutions 13

5.3 Still Task: results 13

5.3.1 CoM Plots 14

5.3.2 State and Input Contact Values 15

5.3.3 Contact Forces 15

5.3.4 Robot representation 16

5.4 Walking Task: results 16

5.4.1 CoM Plots 17

5.4.2 State and Input Contact Values 17

5.4.3 Contact Forces 19

5.4.4 Robot representation 20

1 Introduction

2 Related Works

2.1 Centroidal Dynamics Basics

The purpose of this section is to explore the applications and importance of Centroidal Dynamics of Humanoid Robots [**?**].
Centroidal Dynamics is the dynamics of a humanoid robot projected at its Center of Mass (CoM), a uniquely important point in its dynamics representing the effective location of the total mass of the robot and the point through which the resultant gravity force acts.
A humanoid robot can be seen as a multi-link chain where the dynamics of individual parts can be complex, while the motion of the CoM can be easily described.

2.2 Reduced Order Models for Trajectory Generation

For the advantages explained in the previous section, various reduced order models and control algorithms contain the CoM and therefore the Centroidal Dynamics as integral components.

2.2.1 Linear Inverted Pendulum Mode

The Linear Inverted Pendulum Model (LIPM) [**?**] is a point mass model that focuses on the translational dynamics of a legged robot's locomotion. It was the reduced model most applied in humanoid and quadruped robots during the 2000's and 2010's. Since Centroidal Dynamics is not linear, LIPM makes two assumptions to linearize the dynamics: (1) there is no angular momentum around the center of mass, (2) the center of mass keeps a constant height. Stability is achieved by ensuring the Zero Moment Point (ZMP) stays within the support polygon which is the foot region.

2.2.2 Variable-Height Linear Inverted Pendulum Model

The Variable-Height Inverted Pendulum Model (VH-LIPM) [**?**] extends the classic Linear Inverted Pendulum Model by allowing the center of mass (CoM) to move vertically rather than being constrained to a fixed height.

2.2.3 Spring-Loaded Inverted Pendulum Model

The Spring-Loaded Inverted Pendulum Model (SLIP) [**?**] models locomotion as a point mass on top of a massless, elastic spring, representing the compliant behavior of legs during running, hopping, and bouncing gaits. Unlike the rigid inverted pendulum, SLIP captures energy storage and release through the spring mechanism, enabling dynamic movements with minimal energy loss. The SLIP model can express the vertical motion of the CoM required for running.

2.3 Integration of Trajectory Optimization

Trajectory optimization is a mathematical approach used to compute the best possible path or motion for a system while satisfying given constraints and optimizing a performance objective.
For instance, in order to define a trajectory optimization problem, we need state variables, control inputs, an objective function, and constraints. Reduced order models defined in Section 2.2 integrate trajectory optimization to solve the motion generation of legged robots.

Reduced order models are incorporated into trajectory optimization problems in various ways. For example, some methods use the LIPM as the system's state equation, define the desired ZMP as a cost term, or enforce ZMP-based stability criteria as constraints. More advanced approaches that include both CoM motion and base link rotation typically rely on full centroidal dynamics, which can be integrated either as state equations or constraints. These formulations often impose stability through conditions like the support polygon or the centroidal wrench cone.

However, the use of numerical integration and collocation techniques generally requires small time steps to maintain accuracy, which leads to a large number of decision variables. Moreover, trajectory feasibility is typically enforced only at discrete keypoints, even though some studies address feasibility between those points.

If a closed-form solution of the dynamic model is available, it allows for the trajectory to be expressed using far fewer decision variables while ensuring feasibility at all time instants, as long as control constraints like ZMP or contact wrench bounds are satisfied. Some studies have attempted to linearize or analytically integrate centroidal dynamics, often by making simplifying assumptions such as ignoring vertical-axis rotations or enforcing a constant CoM height. While useful in some practical scenarios, these assumptions limit applicability to more complex motions such as multi-contact or acrobatic maneuvers.

One analytical integration method involves applying a zero-order hold to contact wrenches, but this induces undesirable variations in angular momentum unless the integration time step is kept very small. To overcome this, a previously proposed multi-contact LIPM (mc-LIPM) expresses contact forces in terms of stiffness and displacement between the contact point and CoM. This formulation allows for larger integration time steps—up to a full contact phase—without inducing significant angular momentum variation. However, it still lacks full expressiveness for modeling rotational dynamics.

2.4 Limitations

All the reduced order models discussed above introduce several limitations. The LIPM was originally designed for horizontal motion only, and both LIPM and VH-LIPM rely

on strong assumptions to linearize the inherently nonlinear dynamics. In contrast, the SLIP model maintains the nonlinear characteristics but lacks a closed-form solution, which complicates its analytical handling.

Furthermore, when integrating trajectory optimization, these models often require a high number of decision variables to achieve acceptable accuracy. Feasibility is generally guaranteed only at discrete keyframes or nodes, which may lead to discontinuities or infeasibility between those points.

Although these simplifications may be acceptable in basic locomotion scenarios, they are insufficient for more complex tasks such as multi-contact planning, dynamic manipulation, or acrobatic motion. In such contexts, higher fidelity and time-continuous feasibility are required.

Moreover, the absence of closed-form solutions for many trajectory optimization problems leads to high computational costs and makes real-time applications challenging. A method that reduces the number of decision variables while ensuring feasibility at every time instant would therefore significantly improve efficiency and applicability.

In this approach, a novel stiffness-based method is introduced. It provides a closed-form solution to the trajectory optimization problem, enabling the generation of long, feasible motion trajectories with fewer decision variables and better computational performance.

3 Proposed Method

3.1 Equations

3.2 Closed Form Solution and Discrete Time Equations

3.3 Integration of Base Link Rotation

4 Formulation of Trajectory Optimization Problem

4.1 Costs

The cost function is a crucial component of the trajectory optimization problem, as it quantifies the performance of a given trajectory. The cost function is typically composed of several terms, each representing different aspects of the system's performance.

4.1.1 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:

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

where $(*)^{\text{ref}}$ represents the reference (target) values for the state x_k and the control input u_k. In this work, we focus on a waypoint-tracking task, where a series of intermediate waypoints are specified for the center of mass (CoM), base link, and limb endpoints. Desired positions and velocities along a smooth path connecting these waypoints are generated using spline curves. The system's stiffness parameters are computed by solving the following least-squares optimization problem at each time step k:

$$\min \left\| \sum_l \lambda_{l,k}^2 \right\|^2 \quad \text{subject to} \quad \sum_l \lambda_{l,k}^2 (p_k^{\text{ref}} - p_{l,k}^{\text{ref}}) = g \tag{2}$$

This optimization distributes stiffness values to maintain support for the CoM against gravity. In addition, the reference values for the Centroidal Moment Pivot (CMP) offset and the moments at the end effectors 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 calculated by transforming the quaternion difference into an equivalent angle-axis representation:

$$q - q^{\text{ref}} := \omega(q^{\text{ref}^{-1}} q) \tag{3}$$

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

4.1.2 Inequality Constraints

In physical systems involving contact dynamics, it is essential to ensure that certain physical conditions—such as feasible positions, contact forces, and proper stiffness—are

always satisfied. These conditions are formulated as *inequality constraints* and play a key role in maintaining the physical realism of the motion and feasibility of the optimization problem. In this framework, inequality constraints are used to regulate the position of each contact point relative to the CoM and the base link, enforce non-slipping conditions, and bound parameters like phase duration and stiffness values. First, the box constraint on the position of each end link relative to the CoM and the base link is expressed as:

$$p_{\mathrm{l,min}} \leq q^{-1}(p - p_l) \leq p_{\mathrm{l,max}}, \tag{4}$$

where p is the CoM position, p_l is the position of the l-th end, and q^{-1} denotes the inverse transformation to the local frame. Simple range constraints are imposed on the duration of each phase and the stiffness values:

$$\tau_{\min} \leq \tau \leq \tau_{\max}, \tag{5}$$

$$0 \leq \lambda_l \leq \lambda_{\max}, \quad \forall l. \tag{6}$$

Next, for each end in contact, the contact wrench must satisfy non-slip and moment conditions. If we want to avoid relative motion of the contact surfaces, we must ensure sufficient friction. Thus, to achieve no slipping, the force must be within the friction cone, i.e., the tangential force f_t must satisfy

$$|f_t| \leq \mu f_n \Longrightarrow \sqrt{f_{l,x}^2 + f_{l,y}^2} \leq \mu f_{l,z} \tag{7}$$

where μ is the static friction coefficient, and $(f_{l,x}, f_{l,y}, f_{l,z})$ are the force components at the contact. Constraints on the moments at the contact point are given as:

$$-c_{\max,x} f_{l,z} \leq \eta_{l,x} \leq c_{\max,x} f_{l,z}, \tag{8}$$

$$c_{\min,y} f_{l,z} \leq \eta_{l,y} \leq c_{\max,y} f_{l,z}, \tag{9}$$

$$-\mu_z f_{l,z} \leq \eta_{l,z} \leq \mu_z f_{l,z}, \tag{10}$$

where $c_{\min}$ and $c_{\max}$ define the rectangular bounds of the center-of-pressure (CoP) region, and μ_z is the friction coefficient for the moment. All the inequality constraints can be compactly represented as:

$$g(x_k, u_k) \geq 0, \tag{11}$$

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

$$L_{\mathrm{limit}}(x_k, u_k) = \sum_{i=1}^{n_g} -\log \max(\epsilon, g_i(x_k, u_k)), \tag{12}$$

where n_g is the number of constraints, g_i is the i-th constraint function, and ϵ is a small positive constant that ensures numerical stability and prevents the log function from becoming undefined.

4.1.3 Contact Dependent Cost

Modeling the interaction between a robot's ends and the environment is crucial in dynamic contact systems. The contact-dependent cost is introduced to enforce the complementarity between the contact forces, end velocities, and stiffness values. This ensures that if an end effector is in contact, its motion and interaction forces behave physically, and if it is not in contact, unnecessary forces or stiffness values are suppressed. By minimizing these costs, the system ensures physically meaningful transitions between contact and non-contact states throughout the trajectory.
The contact-dependent cost is defined as:

$$
\begin{aligned}
J_{\text{compl},k} = w_{\text{compl}}^2 \sum_l \Bigg(& \underbrace{\sum_i \delta\left[\sigma_{l,k} = i\right] \left(\eta_i^\top (p_{l,k} - o_i)\right)^2}_{\text{contact distance constraint}} \\
& + \underbrace{\delta\left[\sigma_{l,k} \neq \emptyset\right] \left(\|v_{l,k}\|^2 + \|\omega_{l,k}\|^2\right)}_{\text{zero velocity constraint}} \\
& + \underbrace{\delta\left[\sigma_{l,k} = \emptyset\right] \lambda_{l,k}^2}_{\text{zero stiffness constraint}} \Bigg)
\end{aligned}
\tag{13}
$$

where $\sigma_{l,k}$ denotes the contact state of the l-th end at time step k. In particular,

- $\sigma_{l,k} = i$ if the l-th end is in contact with the i-th contact surface.
- $\sigma_{l,k} = \emptyset$ if it is not in 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:

- The first term enforces that if the l-th end is in contact with surface i, namely the distance in the normal direction (defined by normal vector η_i) from the end's position $p_{l,k}$ to the contact surface origin o_i must be zero.
- The second term requires that if the l-th end is in contact with any surface, its linear velocity $v_{l,k}$ and angular velocity $\omega_{l,k}$ must also be zero, representing static contact.
- The third term ensures that if the l-th end is not in contact, the associated stiffness $\lambda_{l,k}$ must be zero. Physically, this prevents generating unnecessary contact forces when there is no actual contact.

By properly tuning the weight parameter w_{compl}, these complementarity errors can be made acceptably small after optimization. A sufficiently large value of w_{compl} is necessary to enforce these physical consistency conditions without overly penalizing the overall optimization performance.

4.1.4 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. This aggregated cost captures all important objectives and physical constraints of the motion planning task.
The overall cost function is defined as:

$$J[\sigma] = \sum_{k} \left[L_{\text{task},k} + L_{\text{limit},k} + L_{\text{compl},k}[\sigma_k] \right] \tag{14}$$

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

$$\begin{aligned} &\text{find} \quad x, u \quad \text{that minimizes} \quad J[\sigma](x, u) \\ &\text{subject to} \quad x_{k+1} = f(x_k, u_k) \end{aligned} \tag{15}$$

Here, x and u denote the state and control input trajectories, respectively. The function $f(x_k, u_k)$ represents the discrete-time system dynamics, ensuring that the state evolution is dynamically feasible.

5 Simulation and Results

This section aims to visualize the results obtained with the proposed method. In the following, there is first defined the trajectory used for the task reference. Then, there are shown plots, tables and simulations. Before defining the trajectories implemented, let's denote with:

- 0, when the foot is in contact with the surface
- -, when the foot is off contact with the surface

The trajectories are defined with respect to contact point, not time.

5.1 Trajectory Generation

5.1.1 Still Task

In this task the robot should not move at any instant of time and thus keeping the same initial position. The contact sequence used is reported in Table **??** where the first row concerns the right foot and the second row the left foot.

Task	**N**	**Contact Sequence**
Still	3	0 0 0 0 0 0

Table 1: Contact sequence for still task

The trajectory is simply defined as follows:

Algorithm 1 Reference Trajectory Initialization and Update

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 $\leftarrow 0$, sig_idx $\leftarrow 0$
for $t = 0$ to $N - 1$ **do**
    Read contact states from σ
    Set phase duration and contact force gains λ
    Update U_{ref} with phase duration and λ
    sig_idx $\leftarrow$ sig_idx $+ 2$
end for
for $t = 1$ to N **do**
    Set CoM velocity and feet step velocities as the initial state values (zeros)
    Update CoM and feet positions based on velocity and duration
    Update time and orientations in X_{ref} as initial state
    sig_idx $\leftarrow$ sig_idx $+ 2$
end for
return $X_{\text{ref}}, U_{\text{ref}}$

5.1.2 Walking Task

For the walking task, the trajectory used relies on the following concepts: during the double-contact phases, in which both feet are on the ground, namely [0, 0], the COM does not move forward. Instead, it moves forward (along the X axis) only during single-support phases (a step), namely when right/left foot is lifting ([-, 0] or [0, -] respectively). Therefore if the preceding phase is a double support, the current phase COM keeps the same position as before, while if the preceding phase is of single support, the current phase COM has the previous position + some displacement; The Z (height) of each foot is always zero since at the beginning of each phase the feet are on the ground, while the X increases in an alternating manner depending on which foot was lifted in the previous phase. Also, the feet moves simulating a real walk, where the foot that is moving "surpasses" the one in contact, touching the ground beyond it, instead of positioning just next to the contact foot like before. Lastly, each foot is interpolated in the intra-phase height position by following a parabolic profile that starts and arrives at the corresponding positions given by the phases solutions, instead of using the fixed inputs (zero-hold). The contact sequence used is reported in Table **??** where the first row concerns the right foot and the second row the left foot.

Task	N	Contact Sequence
Walk	24	000-000-000-000-000-0 0-000-000-000-000-000

Table 2: Contact sequence for walking task

The algorithm used for this task is shown below:

Algorithm 2 Reference Trajectory Initialization and Update

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 $\leftarrow 0$, sig_idx $\leftarrow 0$
for $t = 0$ to $N - 1$ **do**
    Read contact states from σ
    Set phase duration and contact force gains λ
    Update U_{ref} with phase duration and λ
    sig_idx $\leftarrow$ sig_idx $+ 2$
end for
for $t = 1$ to N **do**
    Read current and previous contact states
    Set CoM velocity (v_x, v_y) and feet step velocities
    Update CoM and feet positions based on velocity and duration
    Update time and orientations in X_{ref}
    sig_idx $\leftarrow$ sig_idx $+ 2$
end for
return $X_{\text{ref}}, U_{\text{ref}}$

5.2 Solutions

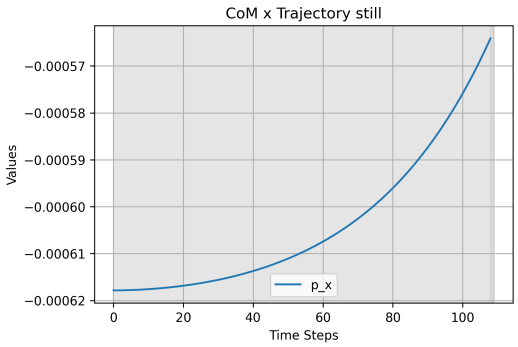
Solutions of those problems and tasks are tuples (X, U) where X is the state vector with dimension N , and U is the input vector with dimension N-1.
The solution is computed with ipopt solver from Casadi optimizer. Both the reference and solution are contact-dependent, meaning that X and U contain values for each contact phase, thus to find the time dependent solution we used the dynamics.

5.3 Still Task: results

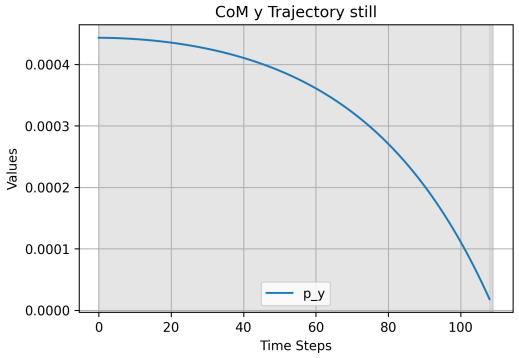
In this case the state vector has dimension 3, and the input vector has dimension 2.

5.3.1 CoM Plots

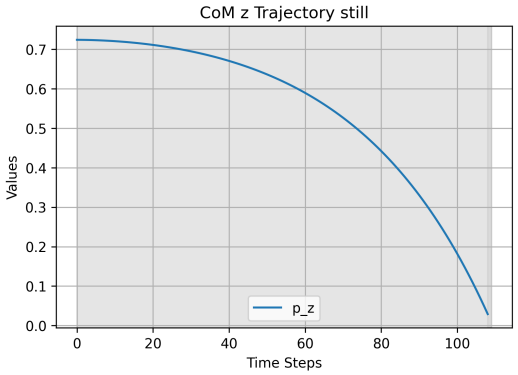
The following plots report the CoM trajectory in time. The alternation between light and dark grey on the background suggests the switch between phases. As plots suggest there's no significant motion in any direction.



(a) Com X Trajectory 1



(b) Com Y Trajectory 1



(c) Com Z Trajectory 1

Figure 1: Com Trajectory

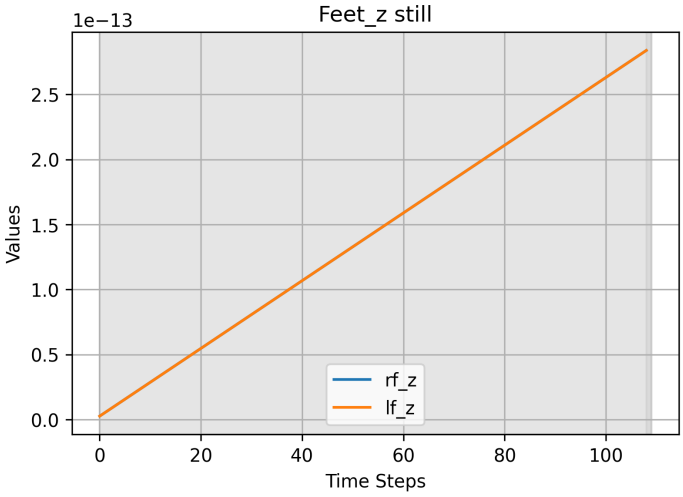


Figure 2: Feet along Z

5.3.2 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.

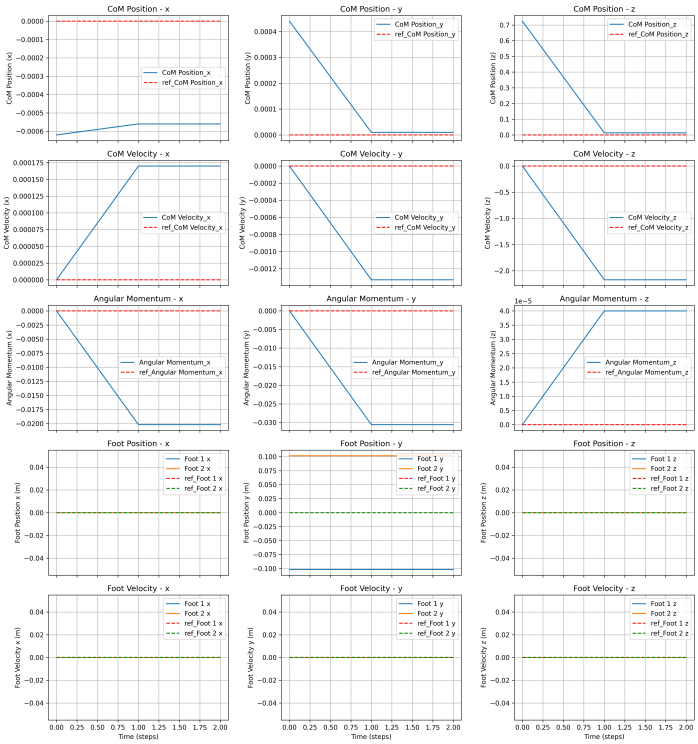


Figure 3: Trajectory vs Reference: state vector

5.3.3 Contact Forces

Another important dynamic aspect regards forces. In the input vector, the contact wrench that describes all the mechanical influence that a contact point (like a foot or hand) exerts on the robot or vice versa. To balance dynamic laws, along the Z-axis the environment should exert a reaction force equal to the gravity factor multiplied by the mass of the robot. In the table **??** below there are listed the contact forces exerting from the environment to both feet. As the table shows, when both feet are on the ground the gravity force is equally distributed on left and right foot.

Right Foot Z	Left Foot Z	Σ_L^k
49.0644	49.0531	[0., 0.]
48.8973	49.65	[0., 0.]

Table 3: Z-axis gravity force and Σ_L^k values

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

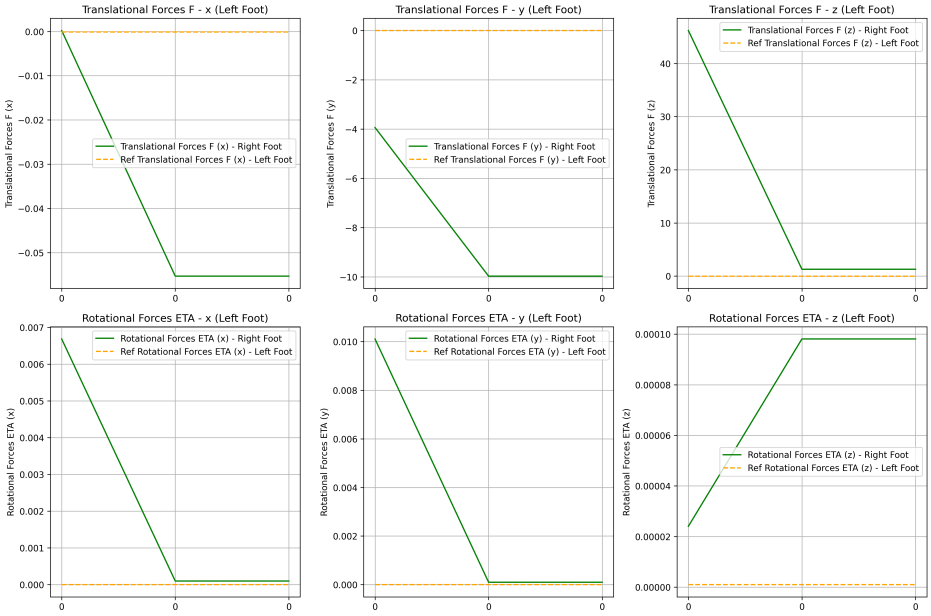


Figure 4: Trajectory vs Reference: Forces

5.3.4 Robot representation

In this section it is possible to visualize first the path followed by the CoM, right and left foot in time:

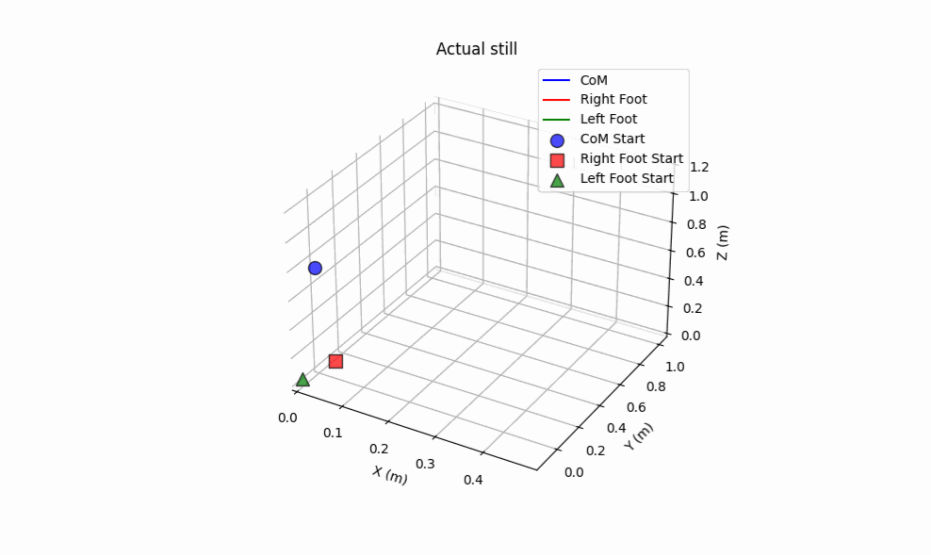


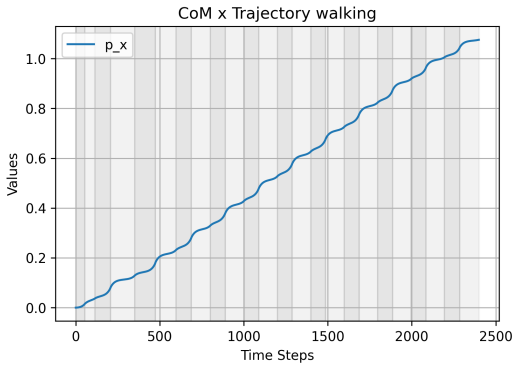
Figure 5: Still

5.4 Walking Task: results

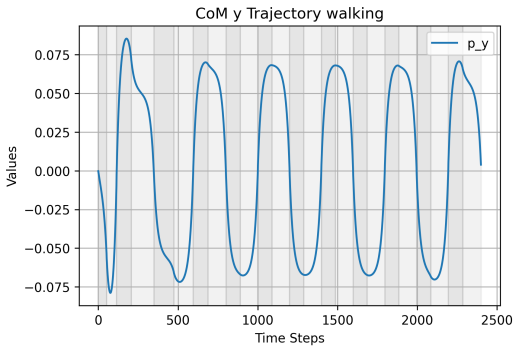
In this case the state vector has dimension 24 and the input vector has dimension 23.

5.4.1 CoM Plots

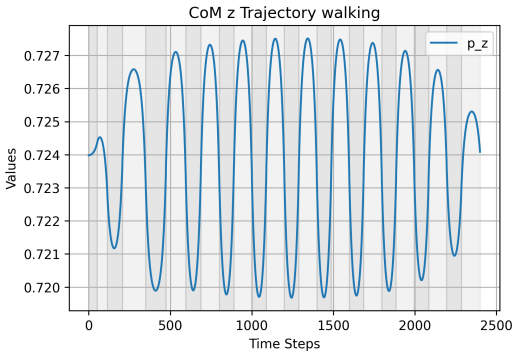
The following plots report the CoM trajectory in time. The alternation between light and dark grey on the background suggests the switch between phases. As Fig. **??** shows, the CoM exhibits linear motion in the x-direction. Along the y-axis **??** the CoM goes towards the foot staying on the ground when lifting the other. Let's consider the first two contacts: in the first one, the robot is in double support (both feet are on the ground), while the second one expects the left foot to raise. Infact, the CoM at the end of the first phase has moved towards the right foot. This mechanism is repeated through all the alternances between double support and single support. In the z-direction **??**, the CoM goes up and down in a small range of motion.



(a) Com X Trajectory 1



(b) Com Y Trajectory 1



(c) Com Z Trajectory 1

Figure 6: 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.

5.4.2 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.

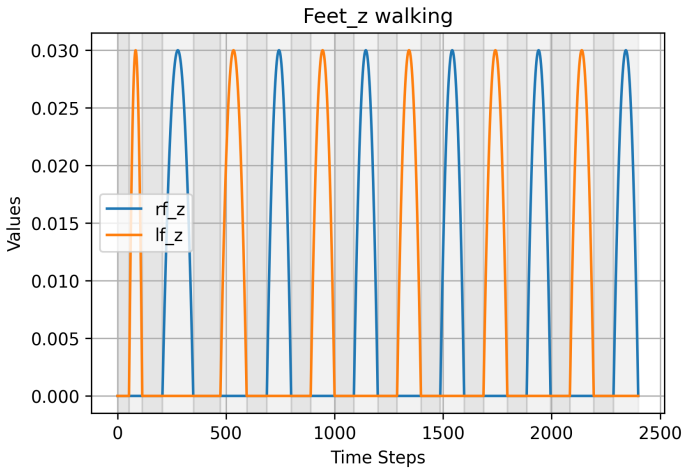


Figure 7: Feet along Z

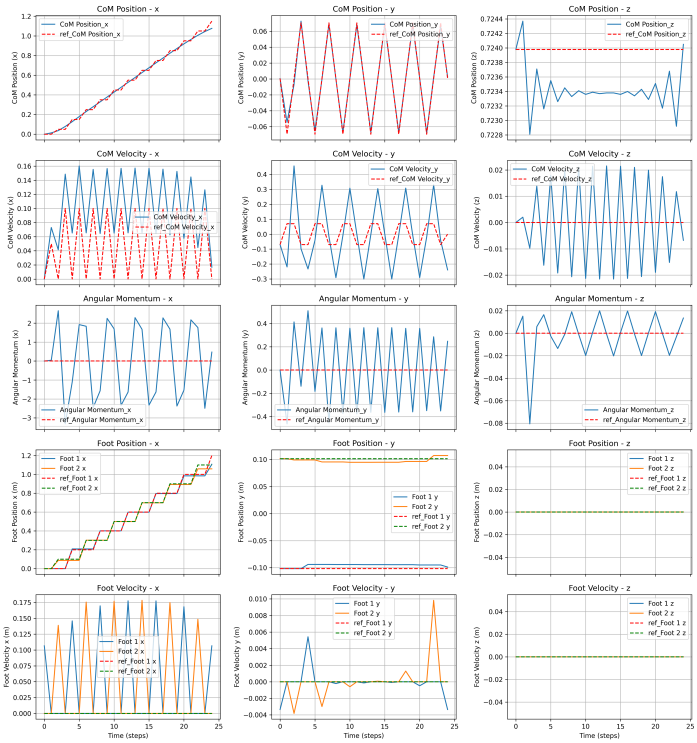


Figure 8: Trajectory vs Reference: state vector

5.4.3 Contact Forces

Another important dynamic aspect regards forces. In the input vector, the contact wrench that describes all the mechanical influence that a contact point (like a foot or hand) exerts on the robot or vice versa. To balance dynamic laws, along the Z-axis the environment should exert a reaction force equal to the gravity factor multiplied by the mass of the robot. In the table **??** below there are listed the contact forces exerting from the environment to both feet. As the table shows, when both feet are on the ground the gravity force is equally distributed on left and right foot, while when one foot lifts the gravity force acts only on the foot staying 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 4: Z-axis gravity force and Σ_L^k values

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

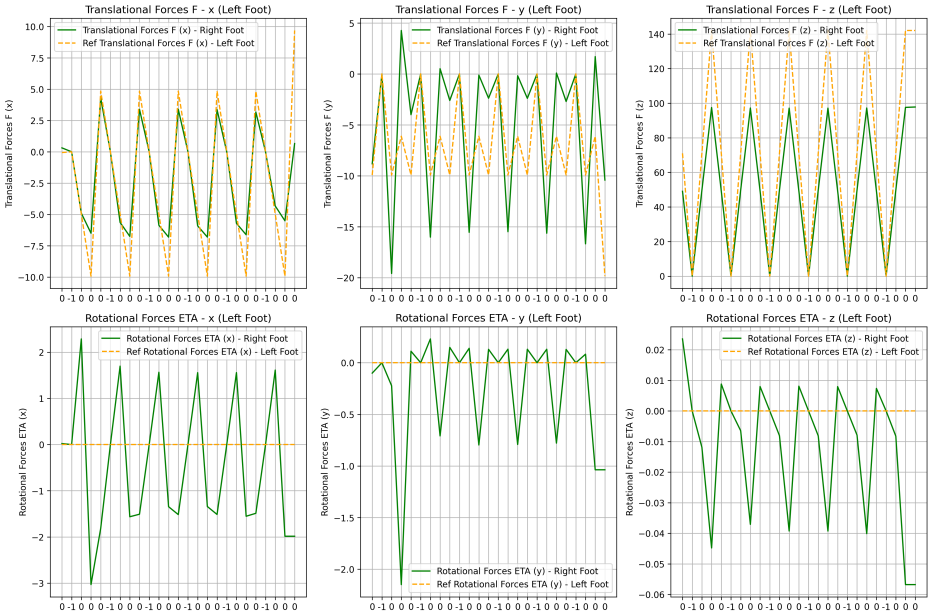


Figure 9: Trajectory vs Reference: Forces

5.4.4 Robot representation

In this section it is possible to visualize first the path followed by the CoM, right and left foot in time:

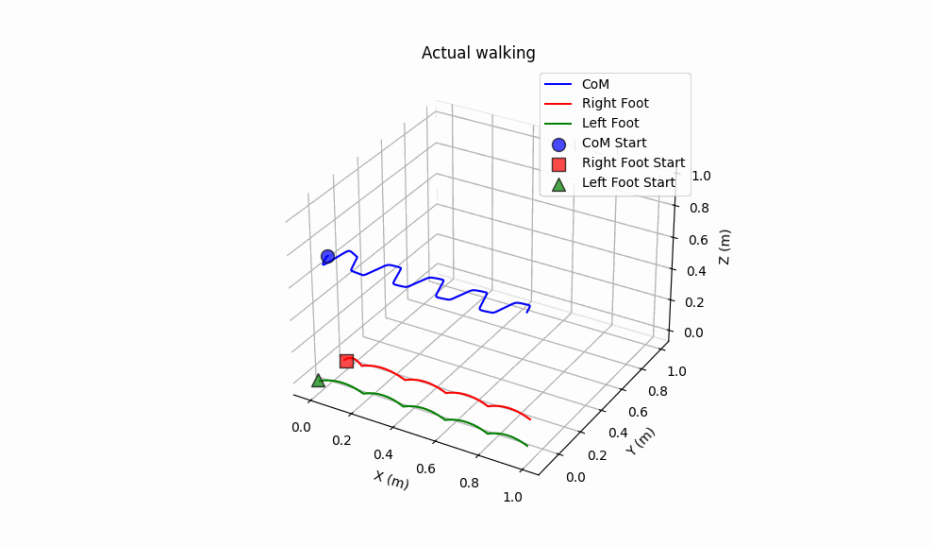


Figure 10: Walking

The CoM solution and reference foot are also passed to the URDF Skeleton into the Dart Simulation environment.