



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

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	7
5	Simulation and Results	8
5.1	Trajectory Generation	8
5.1.1	Still Task	8
5.1.2	Walking Task	9
5.2	Solutions	10
5.3	Still Task: results	10
5.3.1	CoM Plots	11
5.3.2	State and Input Contact Values	12
5.3.3	Contact Forces	12
5.3.4	Robot representation	13
5.4	Walking Task: results	13
5.4.1	CoM Plots	14
5.4.2	State and Input Contact Values	14
5.4.3	Contact Forces	16
5.4.4	Robot representation	17

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

4.1.1 Task Related Cost

4.1.2 Inequality Constraints

4.1.3 Contact Dependent Cost

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

```
1: Initialize  $X_{\text{ref}} \in \mathbb{R}^{28 \times (N+1)}$ ,  $U_{\text{ref}} \in \mathbb{R}^{27 \times N}$ 
2: Set initial CoM state, feet positions and orientations in  $X_{\text{ref}}$ 
3: time  $\leftarrow 0$ , sig_idx  $\leftarrow 0$ 
4: for  $t = 0$  to  $N - 1$  do
5:   Read contact states from  $\sigma$ 
6:   Set phase duration and contact force gains  $\lambda$ 
7:   Update  $U_{\text{ref}}$  with phase duration and  $\lambda$ 
8:   sig_idx  $\leftarrow$  sig_idx + 2
9: end for
10: for  $t = 1$  to  $N$  do
11:   Set CoM velocity and feet step velocities as the initial state values (zeros)
12:   Update CoM and feet positions based on velocity and duration
13:   Update time and orientations in  $X_{\text{ref}}$  as initial state
14:   sig_idx  $\leftarrow$  sig_idx + 2
15: end for
16: 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

```

1: Initialize  $X_{\text{ref}} \in \mathbb{R}^{28 \times (N+1)}$ ,  $U_{\text{ref}} \in \mathbb{R}^{27 \times N}$ 
2: Set initial CoM state, feet positions and orientations in  $X_{\text{ref}}$ 
3: time  $\leftarrow 0$ , sig_idx  $\leftarrow 0$ 
4: for  $t = 0$  to  $N - 1$  do
5:   Read contact states from  $\sigma$ 
6:   Set phase duration and contact force gains  $\lambda$ 
7:   Update  $U_{\text{ref}}$  with phase duration and  $\lambda$ 
8:   sig_idx  $\leftarrow$  sig_idx + 2
9: end for
10: for  $t = 1$  to  $N$  do
11:   Read current and previous contact states
12:   Set CoM velocity ( $v_x, v_y$ ) and feet step velocities
13:   Update CoM and feet positions based on velocity and duration
14:   Update time and orientations in  $X_{\text{ref}}$ 
15:   sig_idx  $\leftarrow$  sig_idx + 2
16: end for
17: 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.

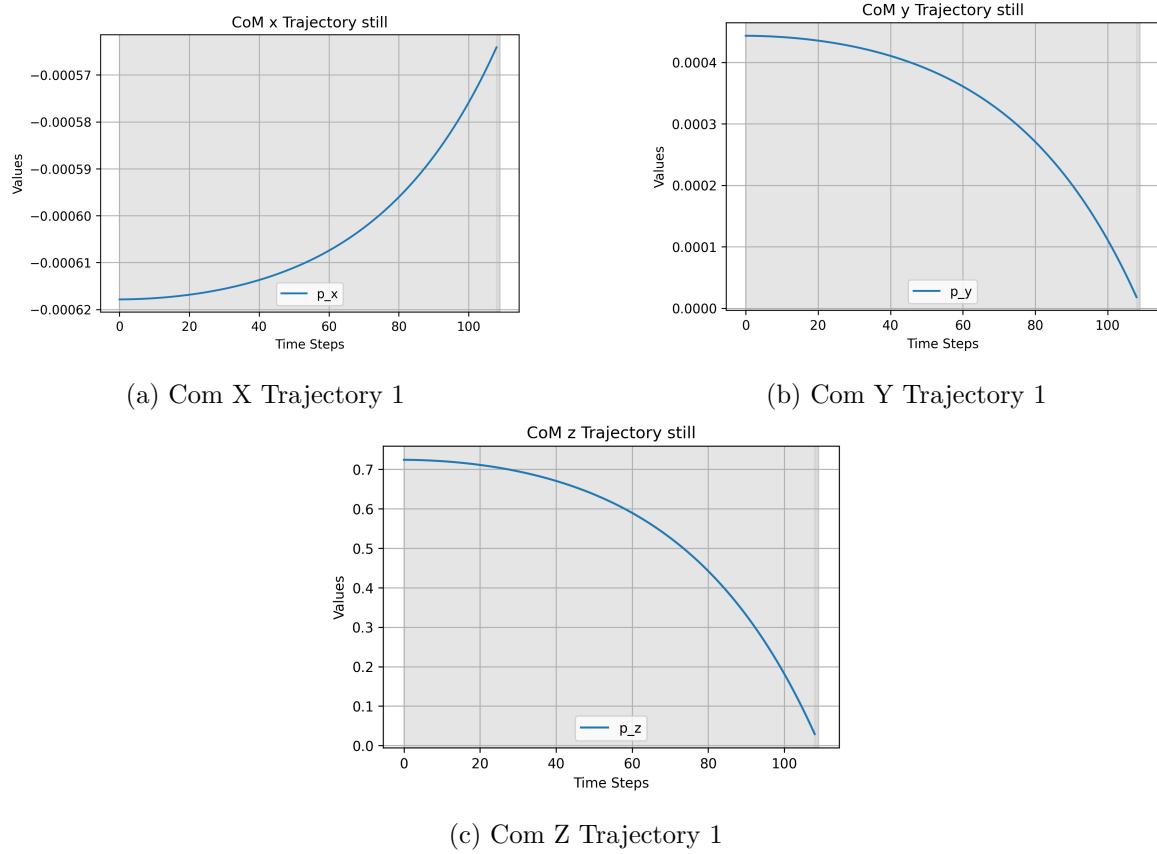


Figure 1: Com Trajectory

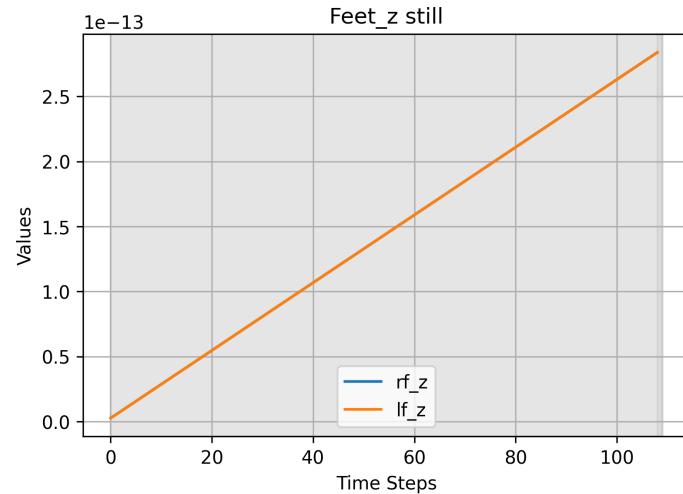


Figure 2: Feet along Z

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.

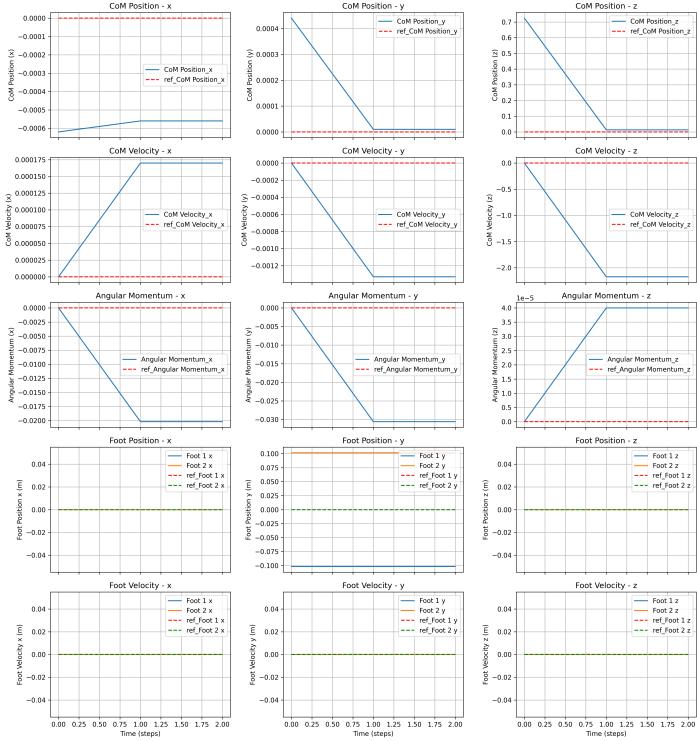


Figure 3: Trajectory vs Reference: state vector

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:

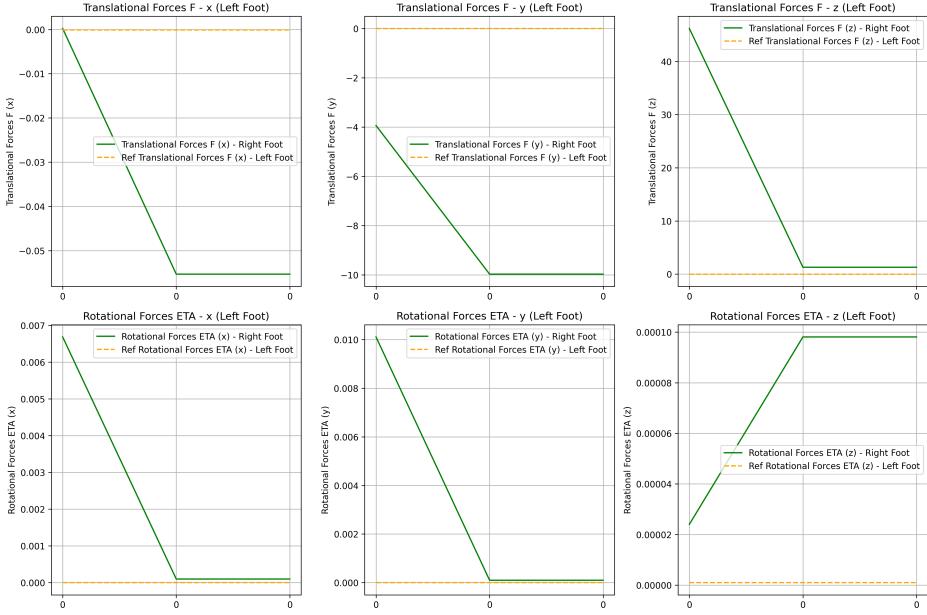


Figure 4: Trajectory vs Reference: Forces

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:

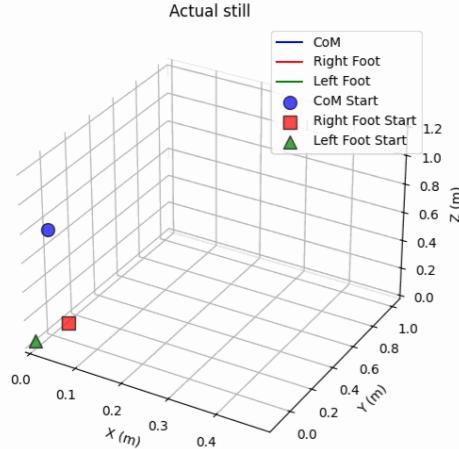


Figure 5: Still

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.

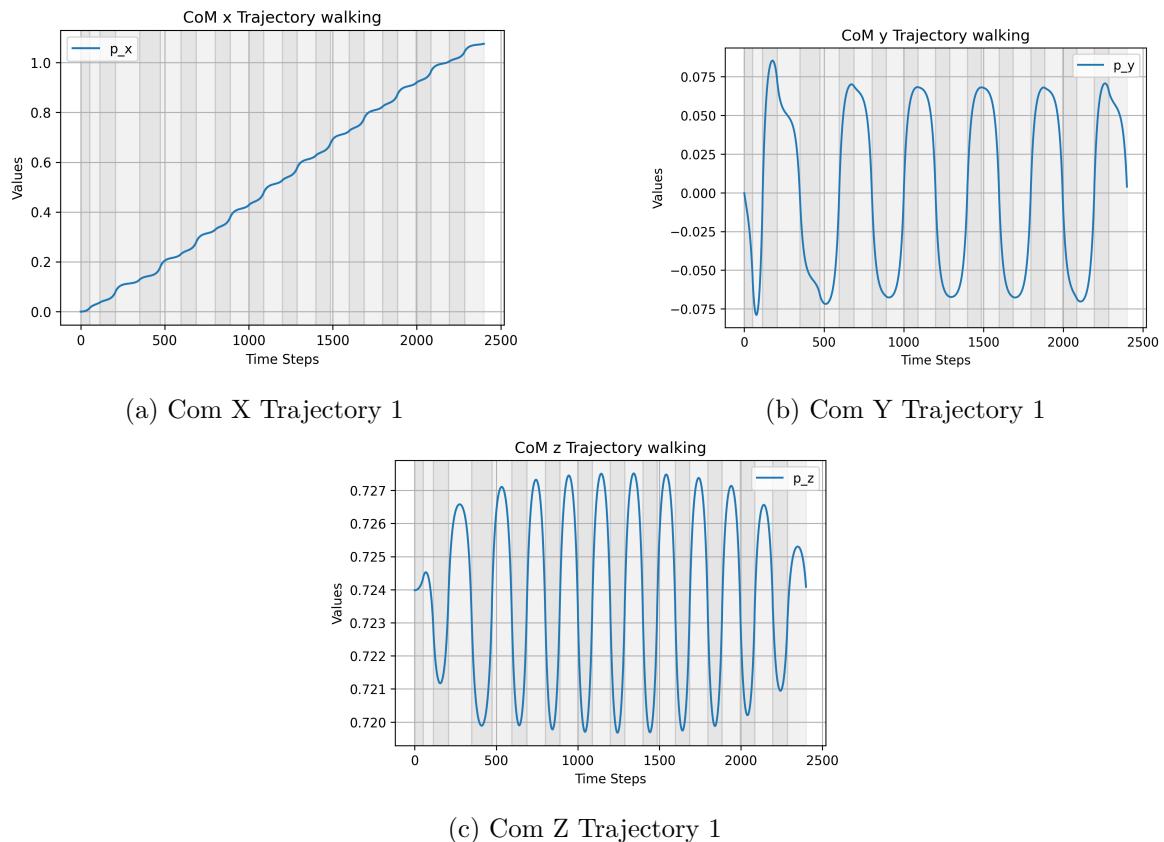


Figure 6: Com Trajectory

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.

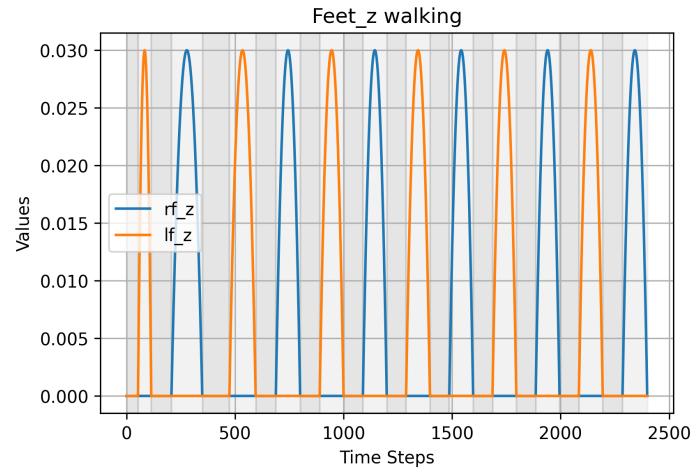


Figure 7: Feet along Z

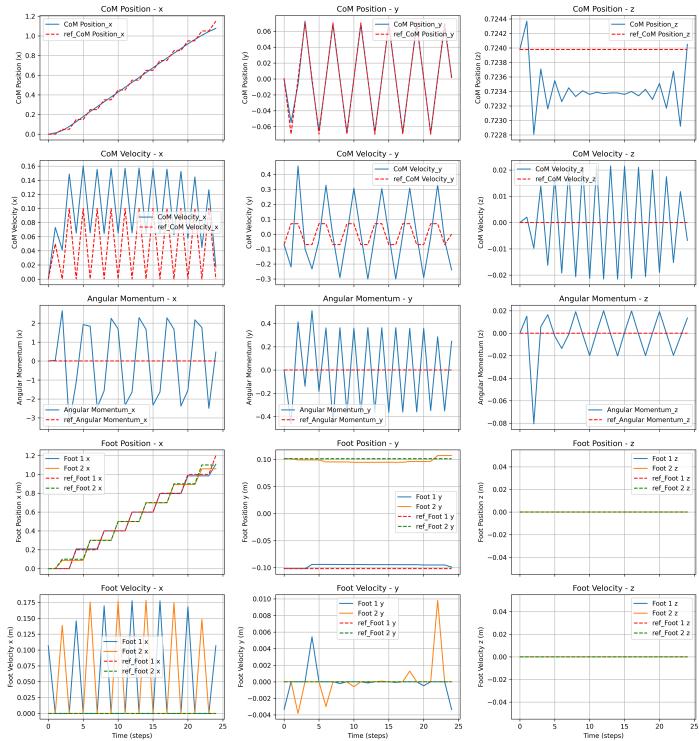


Figure 8: Trajectory vs Reference: state vector

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:

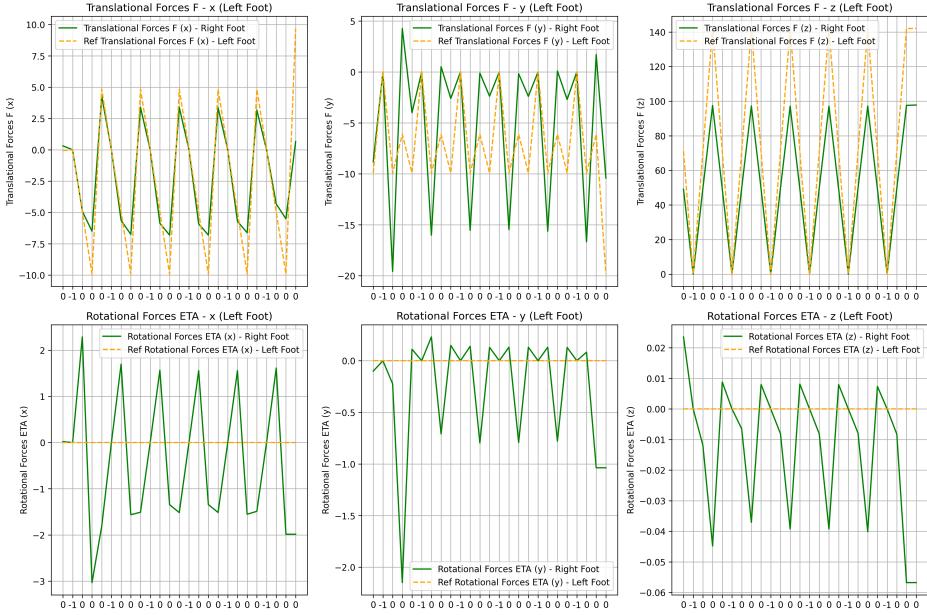


Figure 9: Trajectory vs Reference: Forces

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:

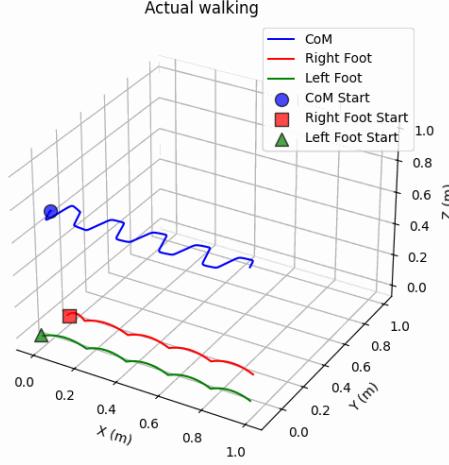


Figure 10: Walking

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

