MPC control of quadruped robot

Master’s Degree in Artificial Intelligence and Robotics

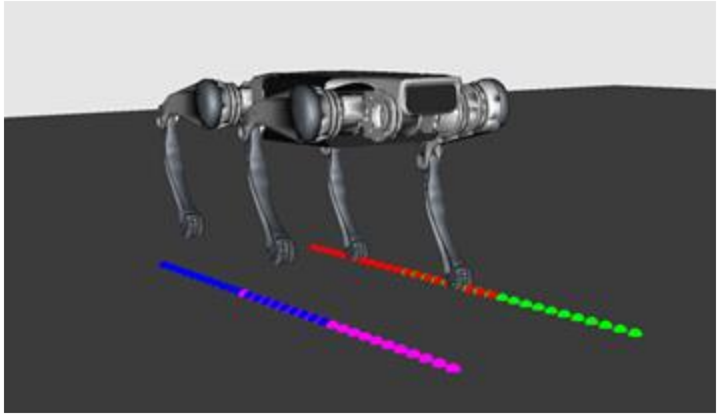
Paradiso Emiliano (1940454)
Piccione Brian (1889051)
Pisapia Vittorio (1918590)
Tedeschi Jacopo (1882789)

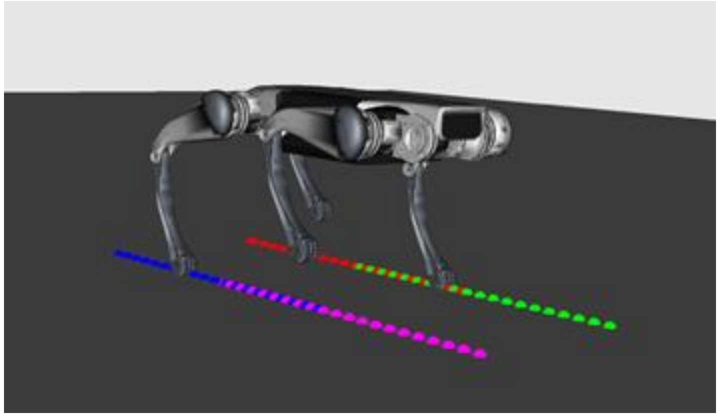
SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

Project overview
Introduction

- the project focuses on **simulating quadruped locomotion** using **Model Predictive Control**
- the control problem is formulated as a **convex quadratic program** to compute the optimal contact forces
- feasible contact sequences for stable locomotion is generated by a custom **footstep planner**
- **ground** and **swing leg controllers** apply forces computed by the MPC and execute gait phases
- multiple gait scenarios are used to assess the framework's effectiveness





Problem definition

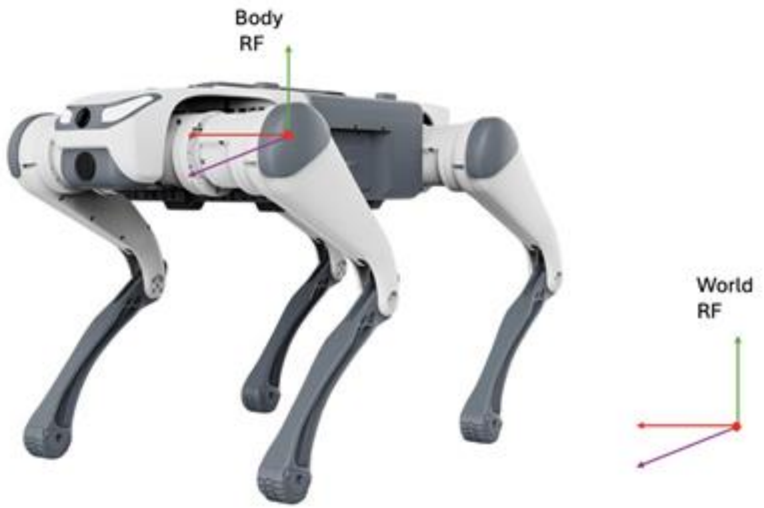
Introduction

- the analysis is carried out using the **Lite 3** quadruped robot from DeepRobotics

- the **state of the robot** is defined as $\boldsymbol{x} \in \mathrm{SE}(3) \times \mathbb{R}^6$:

$$\boldsymbol{x} = (\boldsymbol{\Theta}\ \boldsymbol{p}\ \boldsymbol{\omega}\ \dot{\boldsymbol{p}})^T$$

  - $\boldsymbol{\Theta}$: orientation expressed in ZYX Euler angles
  - $\boldsymbol{p}$: CoM position
  - $\boldsymbol{\omega}$: rigid body angular velocity
  - $\dot{\boldsymbol{p}}$: CoM linear velocity



- to enable effective walking, a reference trajectory $\boldsymbol{x_{des}}$ is properly designed by integrating predefined linear and angular velocities (v_{ref}, ω_{ref}), resulting in motion with **constant velocity**

Block scheme

Introduction

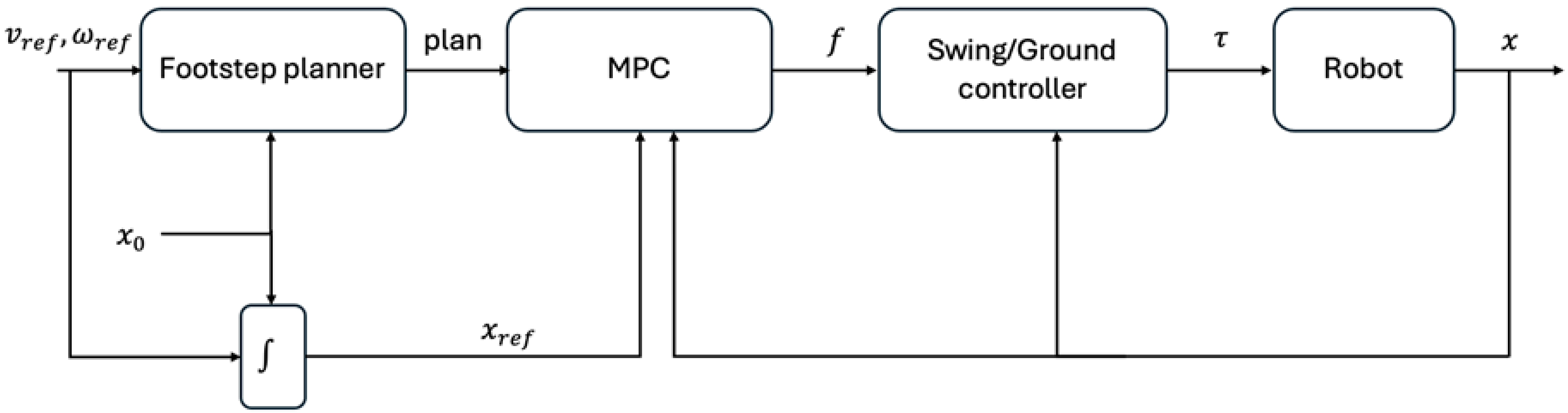


Table of Contents

Footstep planner

Overview

Footstep planner

- **input:** initial configuration, linear velocity v_{ref} ,angular velocity ω_{ref} ,number of steps, double and single support duration

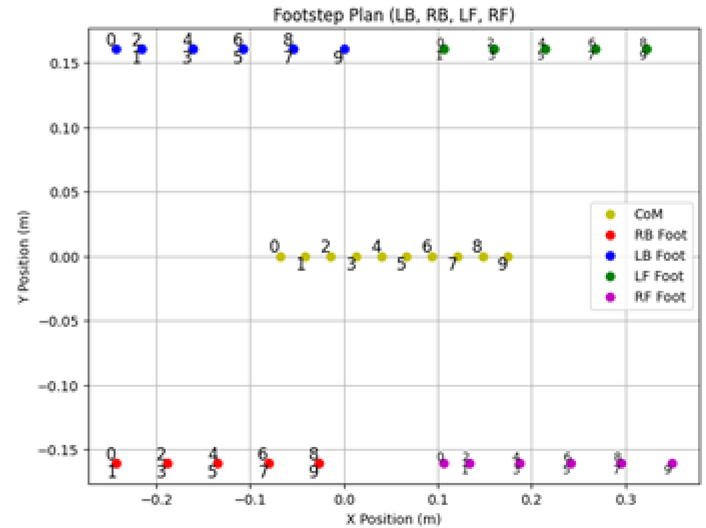
- **output**: a plan composed by a sequence of pose and the swing/support state for each leg

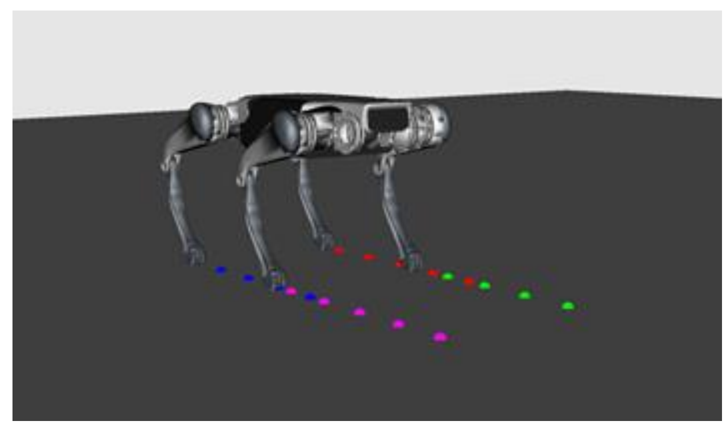
- **virtual unicycle modelling**: the robot is modelled as a virtual unicycle (x, y, θ) placed on the projection on the ground of the robot's center of mass

- **gait generation:** euler integration with $T = SS + DS$

$$\boldsymbol{p}_{\boldsymbol{com}}(t + \Delta t) = \boldsymbol{p}_{\boldsymbol{com}}(t) + \mathbf{R}(\theta) v_{ref} \cdot \Delta t$$

$$\theta(t + \Delta t) = \theta(t) + \omega_{ref} \cdot \Delta t$$

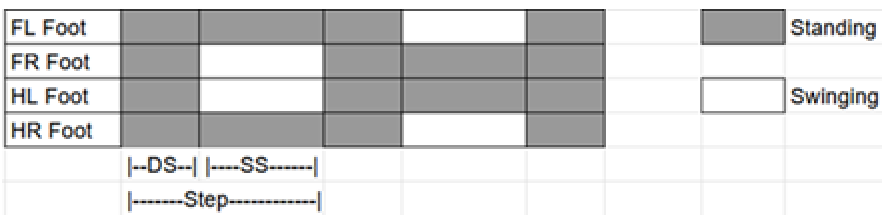




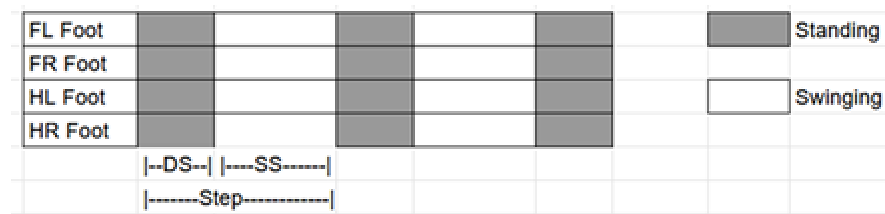
Types of gait

Footstep planner

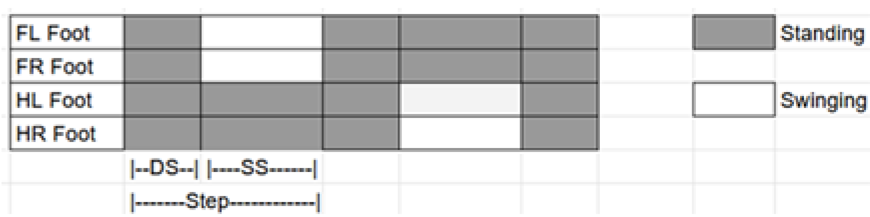
Trotting



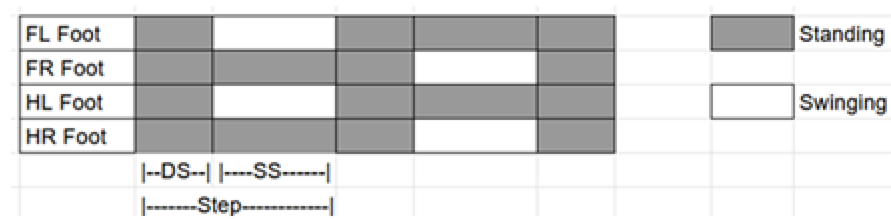
Pronking



Pseudo-gallopping



Ambling



multi-phase gait such as **galloping** can't be generated

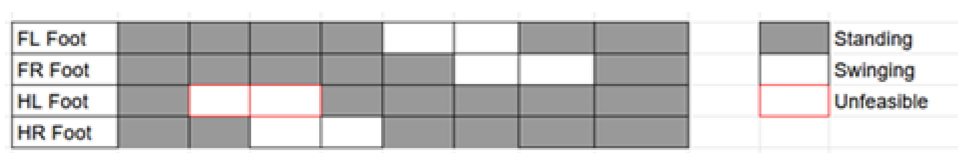


Table of Contents

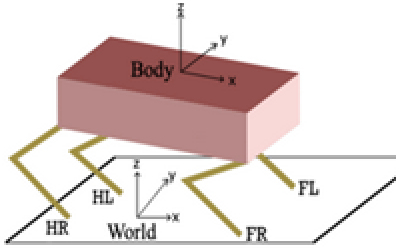
MPC

Robot's dynamics: simplification and model

MPC

The **linear system** $\boldsymbol{x}_{i+1} = \boldsymbol{A}_i \boldsymbol{x}_i + \boldsymbol{B}_i \boldsymbol{u}_i$ has been derived from the following assumptions:

- robot considered as a **rigid body** subject to external forces applied to the contact points - dynamics of the legs are neglected (low mass)
- Considering **small roll and pitch angles** and avoid near-vertical posture
- the state of the robot is then extended to include the gravitational term
- the problem is formulated as a **discrete-time linear system**.
- $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ should be computed using the future predicted position of the feet obtained through the dynamics at instant $t + i$; **desired position** of the feet are used instead



$$\begin{bmatrix} \dot{\boldsymbol{\phi}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\psi}} \end{bmatrix} \approx \boldsymbol{R}_z(\psi)\boldsymbol{\omega}$$

$$\frac{d}{dt}\begin{bmatrix} \boldsymbol{\Theta} \\ \boldsymbol{p} \\ \boldsymbol{\omega} \\ \dot{\boldsymbol{p}} \\ g \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{R}_z(\psi) & \mathbf{0}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & 0 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta} \\ \boldsymbol{p} \\ \boldsymbol{\omega} \\ \dot{\boldsymbol{p}} \\ g \end{bmatrix} + \begin{bmatrix} \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \mathbf{I}^{-1}[\mathbf{r}_1]_\times & \dots & \mathbf{I}^{-1}[\mathbf{r}_1]_\times \\ \mathbf{1}_3/m & \dots & \mathbf{1}_3/m \\ \mathbf{0}_{1\times3} & \dots & \mathbf{0}_{1\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_n \end{bmatrix}$$

Problem definition, cost function and constraints

MPC

- simplifications and linearization of the robot dynamic's leads to the following problems, formulated as **multiple-shooting**, **convex optimization problem**:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=0}^{N-1} \left\| \boldsymbol{x}_{i+1} - \boldsymbol{x}_{i+1,ref} \right\|_{\boldsymbol{Q}_i}^2 + \|\boldsymbol{u}_i\|_{\boldsymbol{R}_i}^2$$

$$\begin{aligned} \textit{subject to} \quad & \boldsymbol{x}_{i+1} = \boldsymbol{A}_i \boldsymbol{x}_i + \boldsymbol{B}_i \boldsymbol{u}_i, & i = 0, \dots, N-1 \\ & \underline{\boldsymbol{c}_i} \le \boldsymbol{C}_i \boldsymbol{u}_i \le \overline{\boldsymbol{c}_i}, & i = 0, \dots, N-1 \\ & \boldsymbol{D}_i \boldsymbol{u}_i = 0 & i = 0, \dots, N-1 \end{aligned}$$

- $\boldsymbol{x}_i$ represents the system's state at the i-th step of the horizon
- $\boldsymbol{u}_i$ is the control input at step i
- $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ represent the discrete time system dynamics
- $\boldsymbol{C}_i$ and $\boldsymbol{c}_i$ represent inequality constraints (bound on the fz component, contact forces within friction cone)
- $\boldsymbol{D}_i$ selects forces corresponding to swinging foot (whose ground forces should be zero)
- $\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$ are diagonal positive semidefinite matrices of weights

Table of Contents

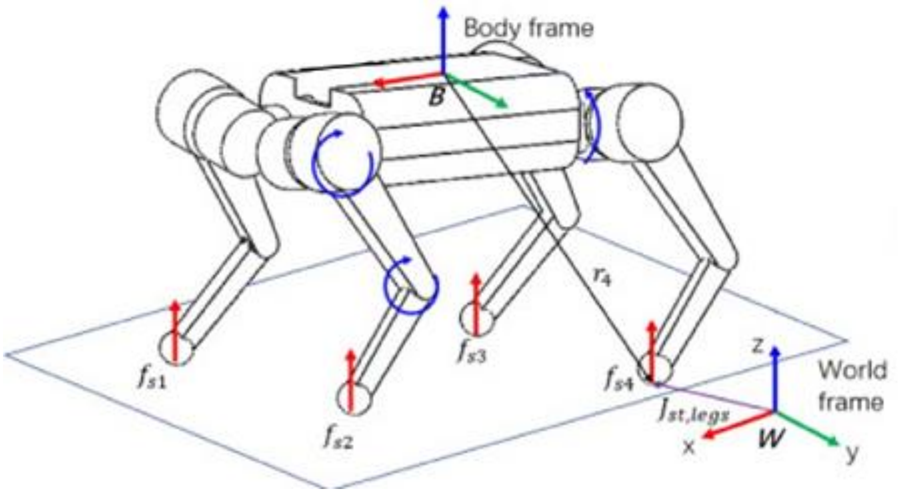
Controllers

Ground controller

- given a leg **planned to be in contact** at a specific *timestep* (*i*), is possible to map the computed **reaction force** (by the MPC) to the torque the joints of the leg should apply by simply computing:

$$\boldsymbol{\tau}_i = \boldsymbol{J}_i^T \boldsymbol{R}_i^T \boldsymbol{f}_i$$

- $\boldsymbol{J}_i^T$: i-th leg Jacobian
- $\boldsymbol{R}_i^T$: rotation from body frame to world frame
- $\boldsymbol{f}_i$: output of the MPC relative to the i-th leg



Swing trajectory definition

Controllers

- each foot of the robot should track a **predefined trajectory** for the movement during the swing phase

- using a **cubic polynomial** defined in the plane $\boldsymbol{x}-\boldsymbol{y}$ allows continuous velocity and acceleration profile and zero derivatives at the boundary:

$$\boldsymbol{p}(\boldsymbol{t}) = \boldsymbol{p}_i + (\boldsymbol{p}_f - \boldsymbol{p}_i)\left(-\mathbf{2}\left(\frac{\boldsymbol{t}}{\boldsymbol{T}}\right)^{\mathbf{3}} + \mathbf{3}\left(\frac{\boldsymbol{t}}{\boldsymbol{T}}\right)^{\mathbf{2}}\right), \quad t \in [0, T]$$

where $\boldsymbol{T}$ is the duration of the single support time

- using a **quartic polynomial** with a bell-shaped profile for the $\boldsymbol{z}$ component, allowing the liftoff and landing on 0 vertical velocity and acceleration:

$$\boldsymbol{z}(\boldsymbol{t}) = \mathbf{16}\frac{\boldsymbol{h}}{\boldsymbol{T}^4}\boldsymbol{t}^4 - \mathbf{32}\frac{\boldsymbol{h}}{\boldsymbol{T}^3}\boldsymbol{t}^3 + \mathbf{16}\frac{\boldsymbol{h}}{\boldsymbol{T}^2}\boldsymbol{t}^2, \quad t \in [0, T]$$

where the value of the ***h*** is an *hyperparameter* describing the maximum height of the step

Swing leg controller
Controllers

- simple control strategy using **feedback + feedforward** to track the reference trajectory
- commanding the torque for the swinging leg as:

$$\boldsymbol{\tau}_i = \boldsymbol{J}_i^T \left[\boldsymbol{K}_p \left(\boldsymbol{p}_{i,ref} - \boldsymbol{p}_i \right) + \boldsymbol{K}_d \left(\boldsymbol{v}_{i,ref} - \boldsymbol{v}_i \right) \right] + \boldsymbol{\tau}_{i\_ff}$$

where the $\boldsymbol{\tau}_{i\_ff}$ is the feedforward term computed taking account of the Dynamical term of the leg:

$$\boldsymbol{\tau}_{i\_ff} = \boldsymbol{J}_i^T \boldsymbol{M}_i \left(\boldsymbol{a}_{i,ref} - \dot{\boldsymbol{J}}_i \dot{\boldsymbol{q}}_i \right) + \boldsymbol{C}_i \dot{\boldsymbol{q}}_i + \boldsymbol{G}_i$$

- $\boldsymbol{J}_i^T$: Jacobian of the i-th leg
- $\dot{\boldsymbol{q}}_i$: joint velocity vector
- $\boldsymbol{M}_i$: operational space inertia matrix (apparent mass along the i-th direction)
- $\boldsymbol{C}_i$: Coriolis term for the i-th leg
- $\boldsymbol{G}_i$: gravitational term for the i-th leg

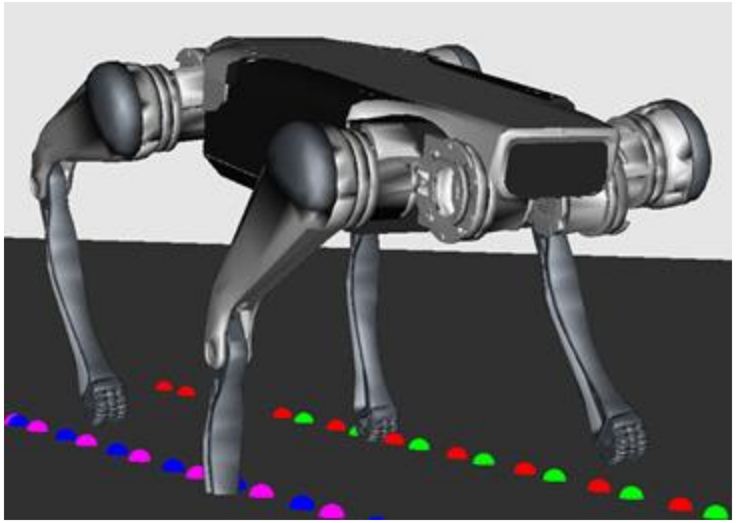
Table of Contents

Results

Simulation environment and MPC frequencies

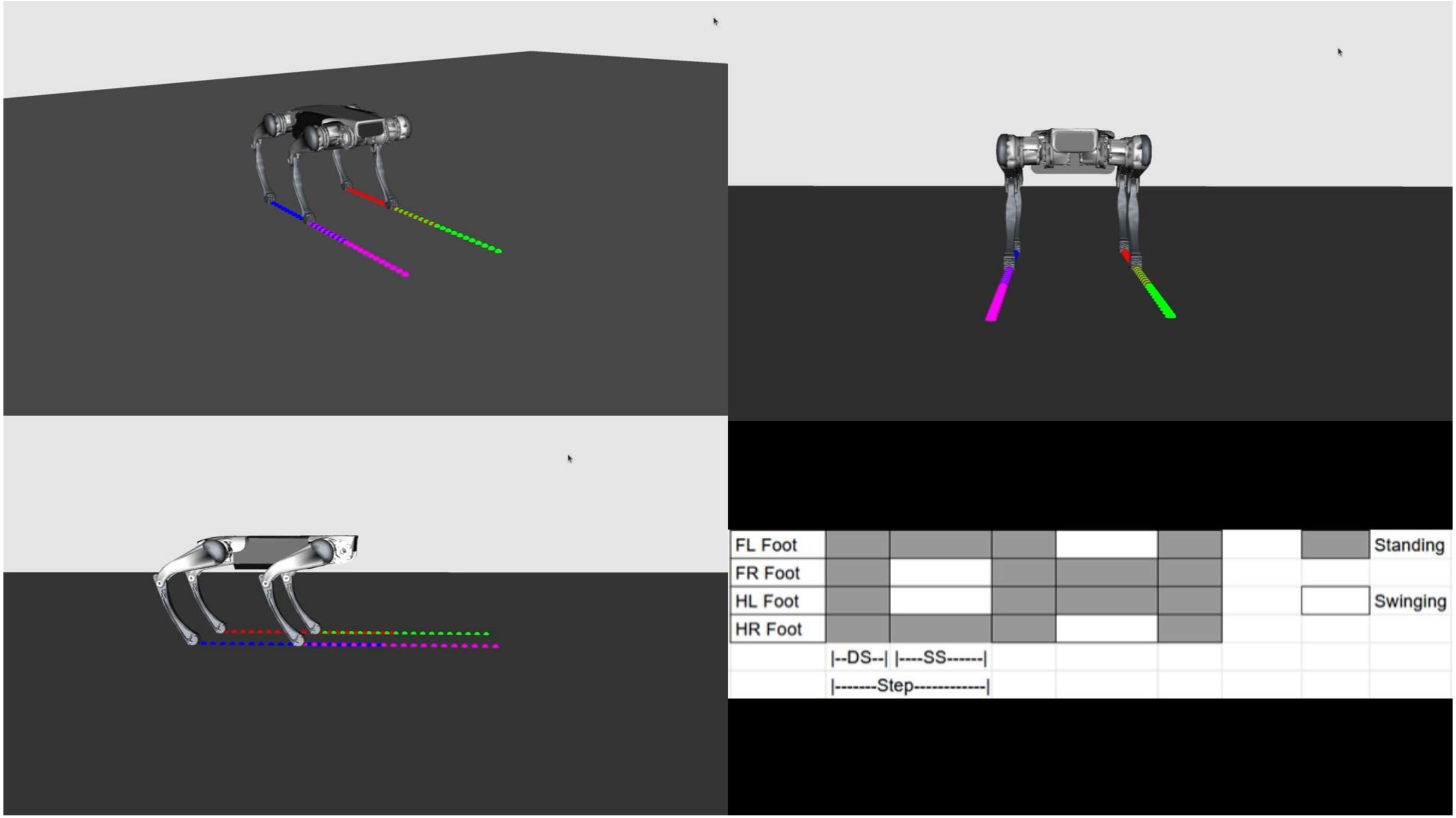
Results

- The proposed results are obtained using the python based **DARTPy simulation environment**
- MPC solver achieved solve frequencies in the range of **40-60 Hz** with horizon length of 60
- By decreasing the horizon length to 30, tracking performance were slightly worse but solve frequencies exceed **100 Hz**, making **real-time**, **on-board** implementation feasible (simulation time step of 0.01 seconds)
- One major problem was **compenetration between feet and ground**; it was solved by enlarging collision spheres of the feet and by adding small mass and inertia to them



Trotting gait - Video

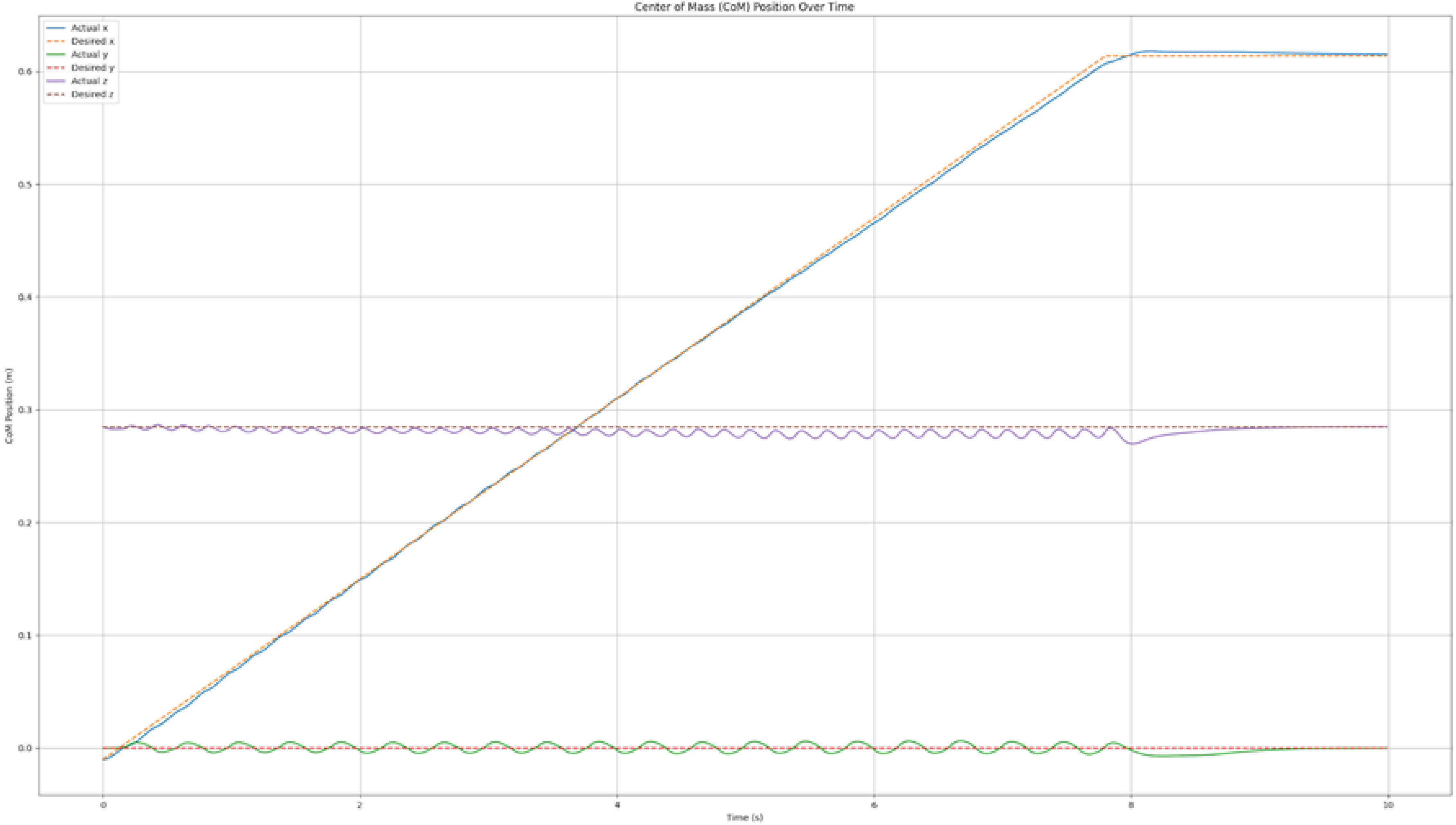
Results



$$ss\_duration = 0.1s, \quad ds\_duration = 0.1s, \quad v\_com\_ref\_x = 0.08m/s$$

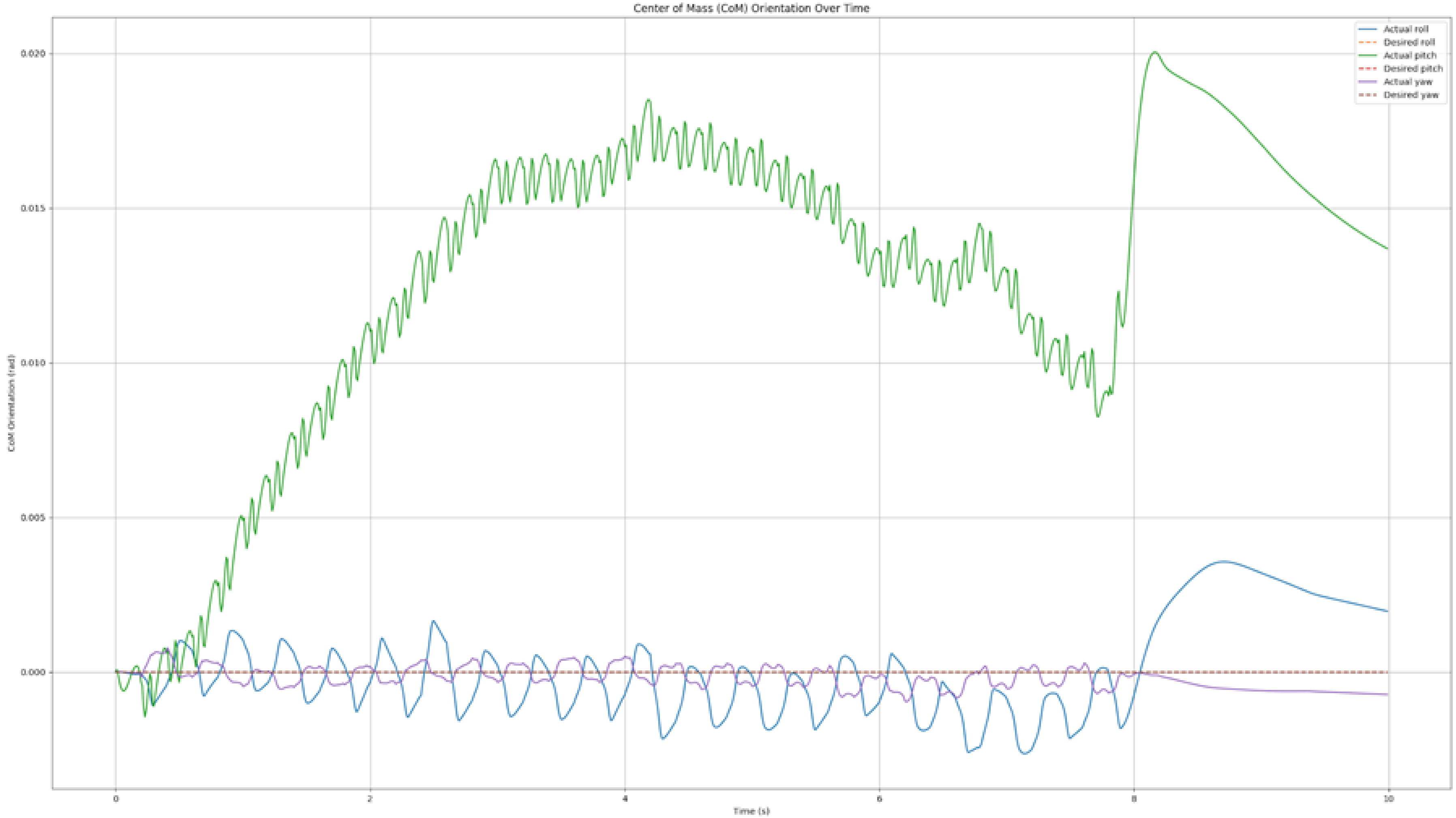
Trotting gait - Position of CoM

Results



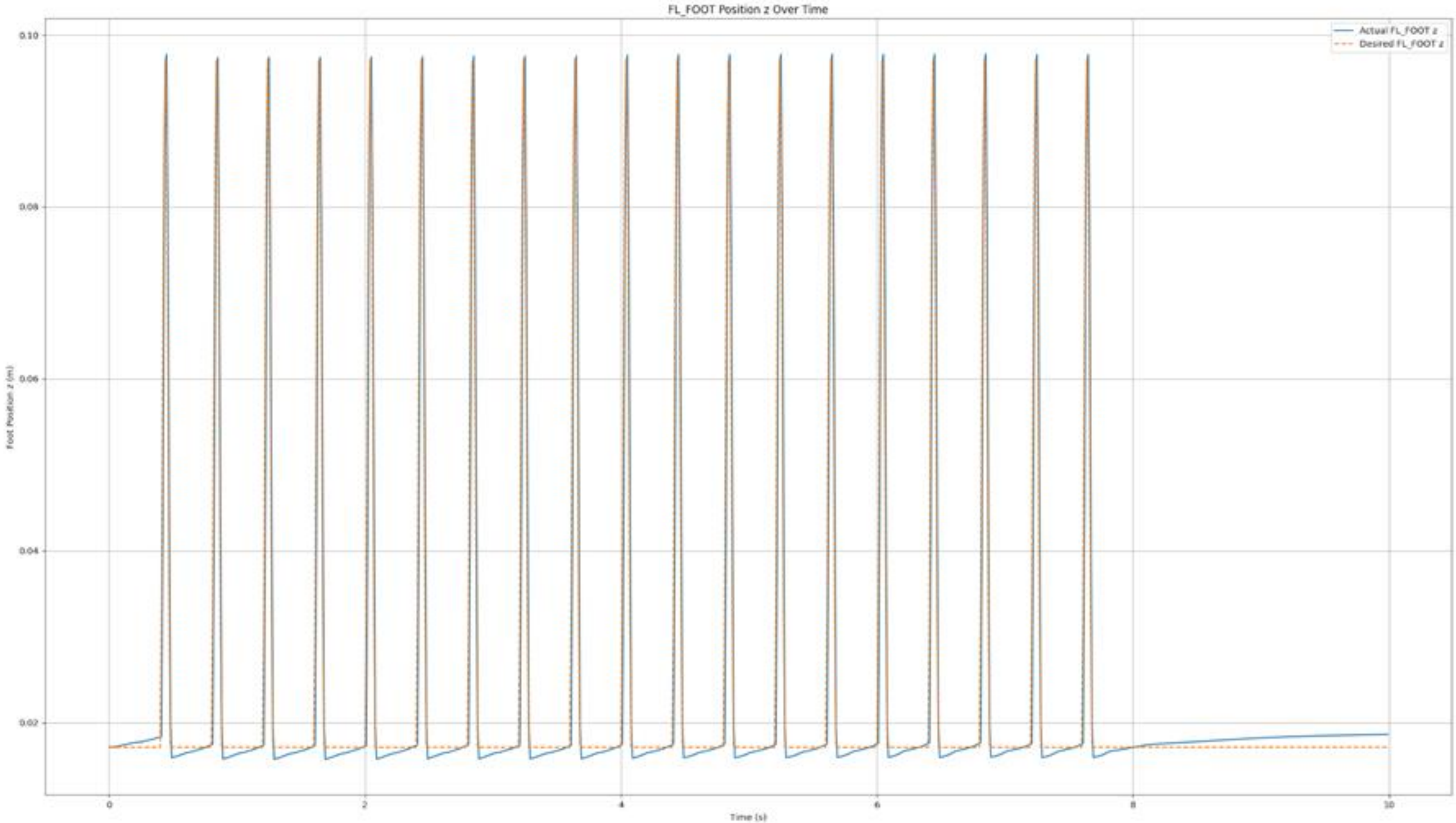
Trotting gait - Orientation of CoM

Results



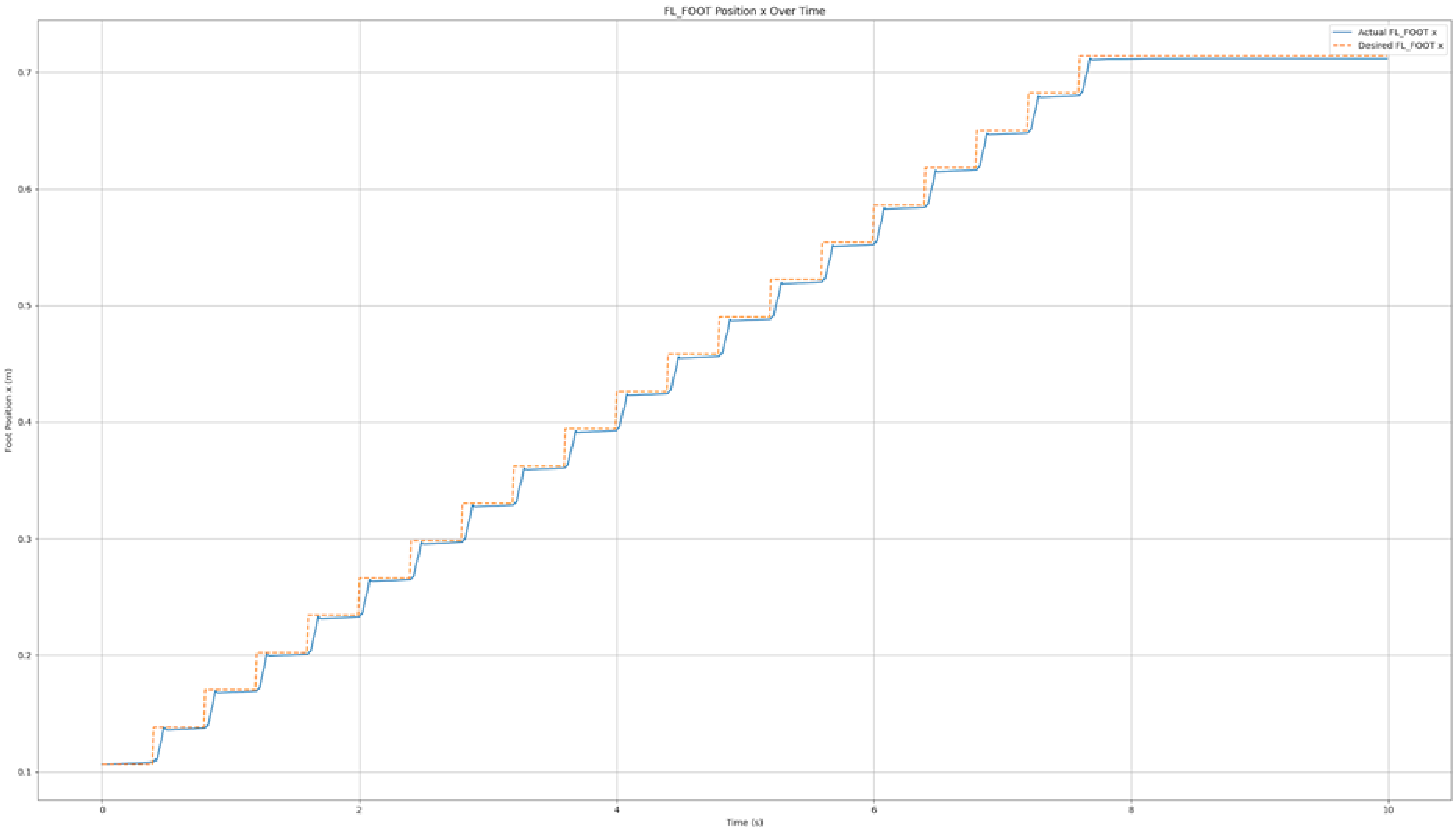
Trotting gait - FL foot z component

Results



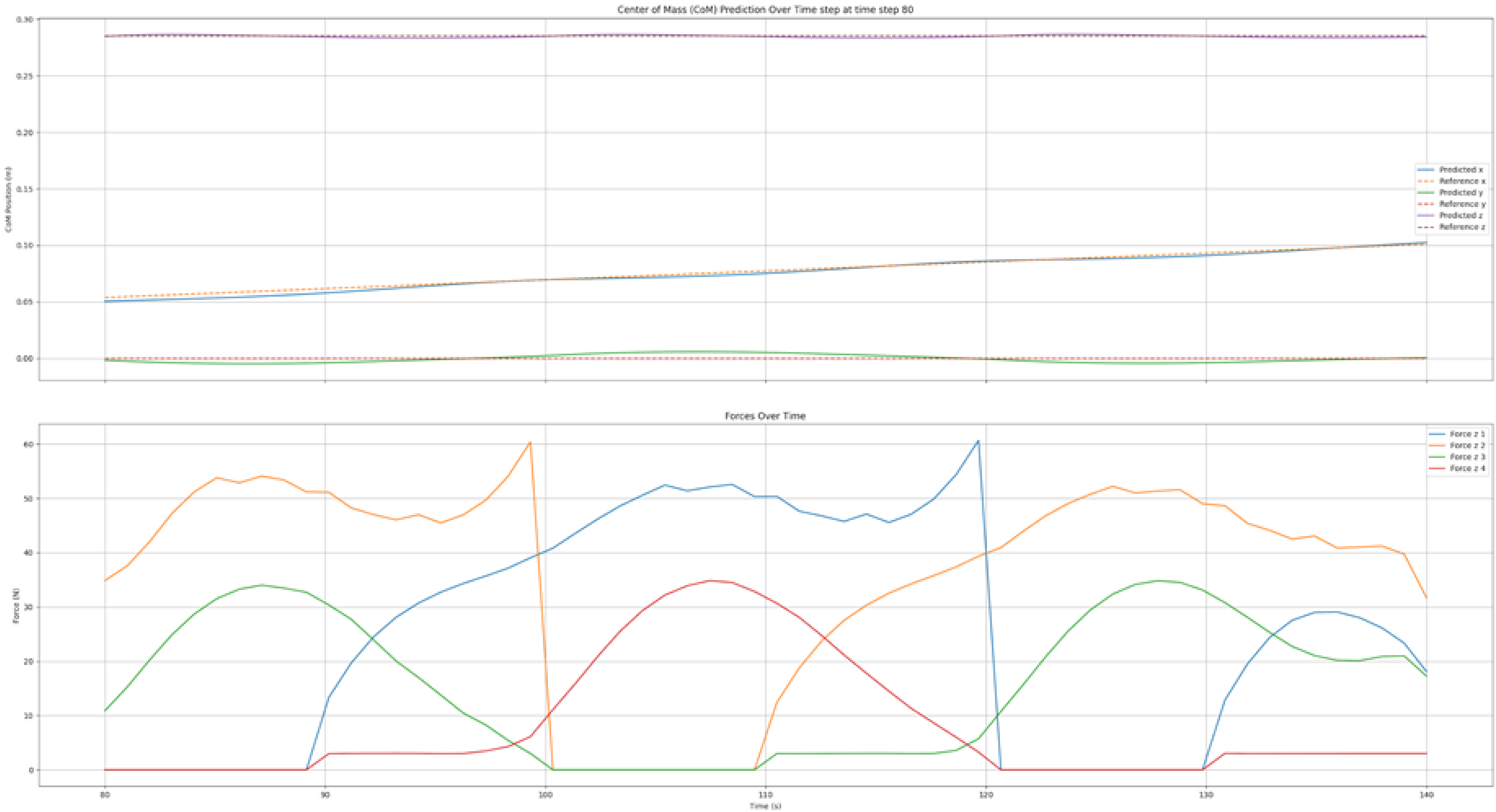
Trotting gait - FL foot x component

Results



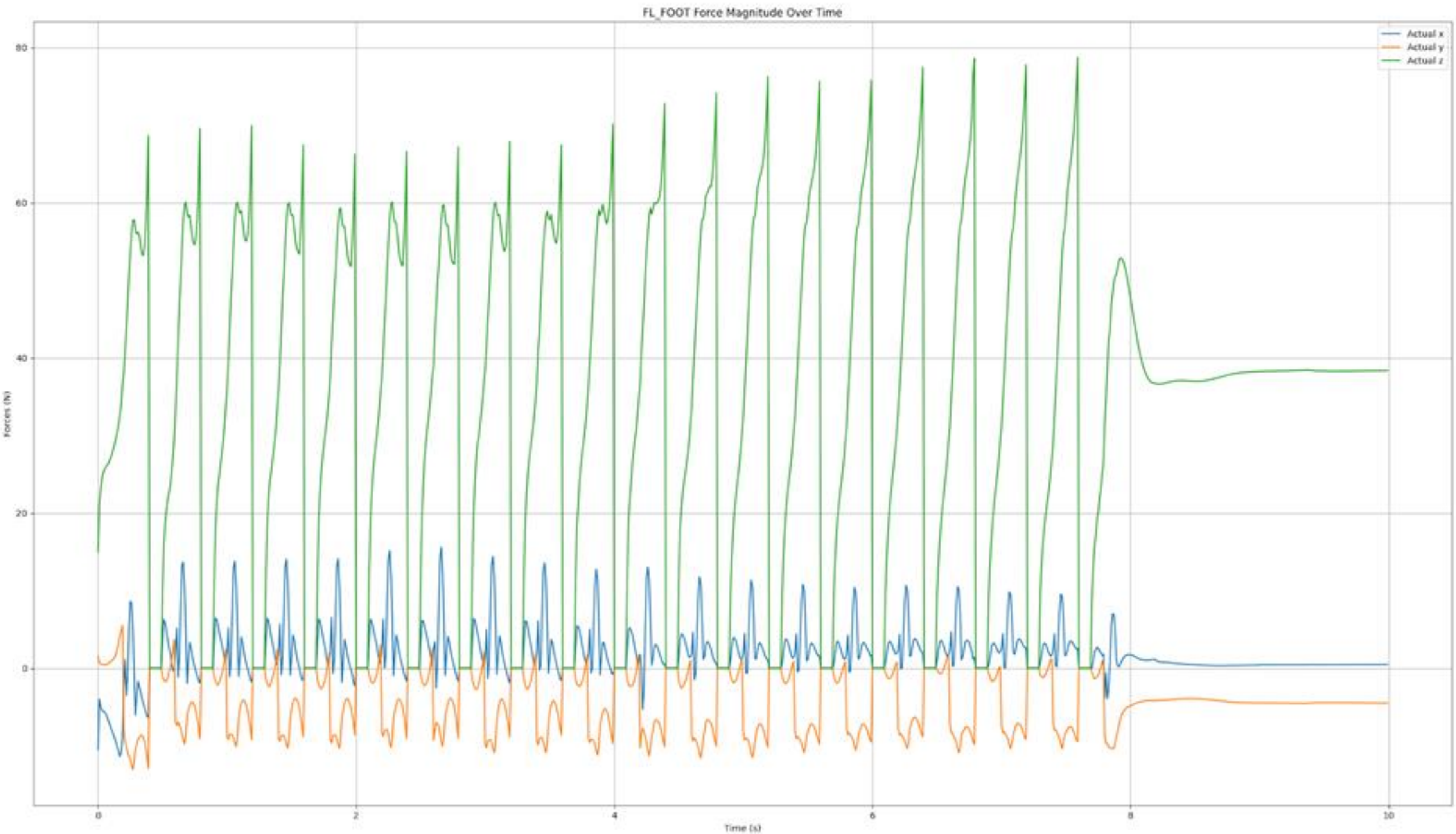
Trotting gait - MPC prediction at timestep 80

Results



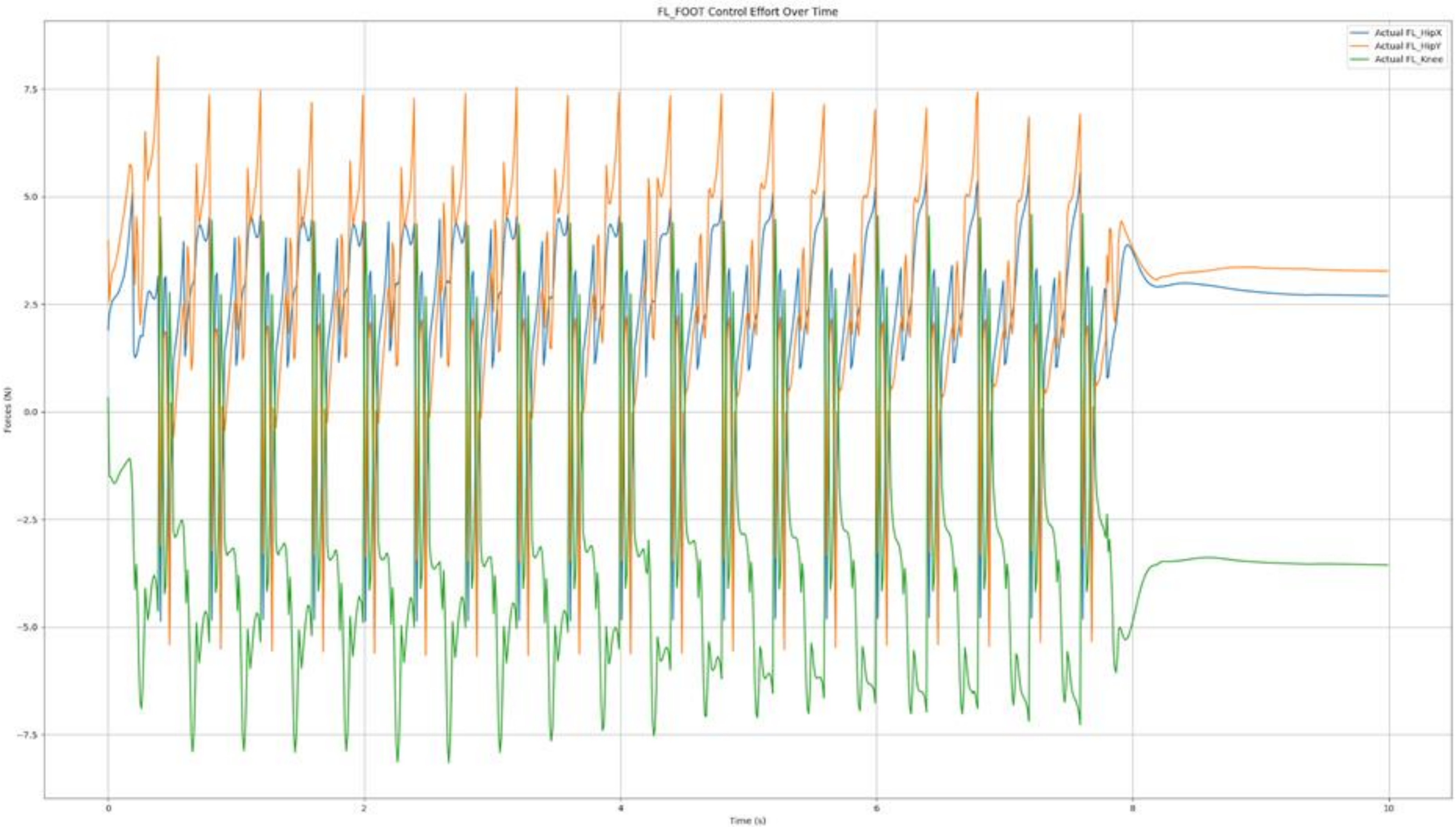
Trotting gait - FL foot forces

Results



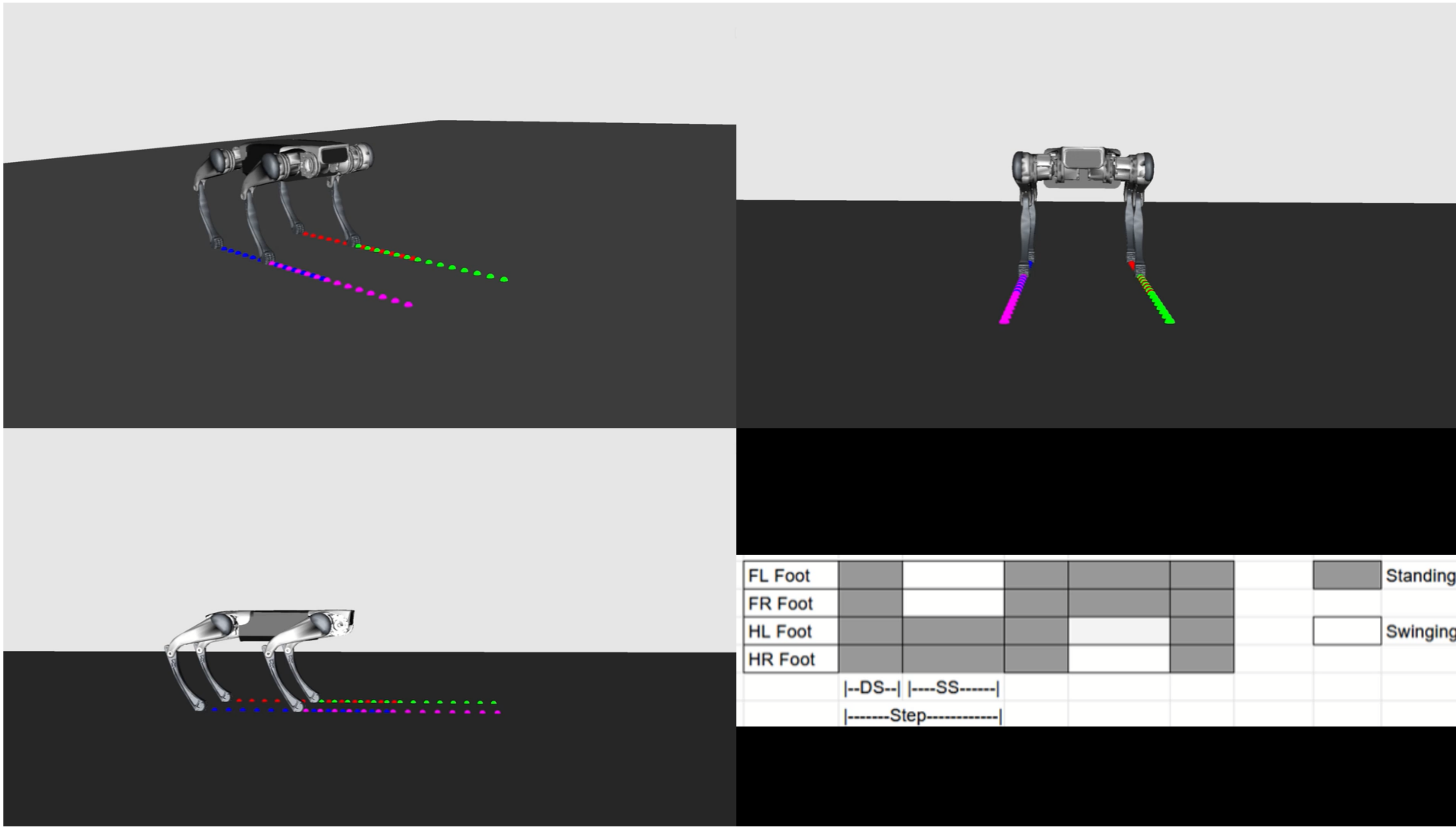
Trotting gait - FL leg control effort

Results



Pesudo-gallopping gait - Video

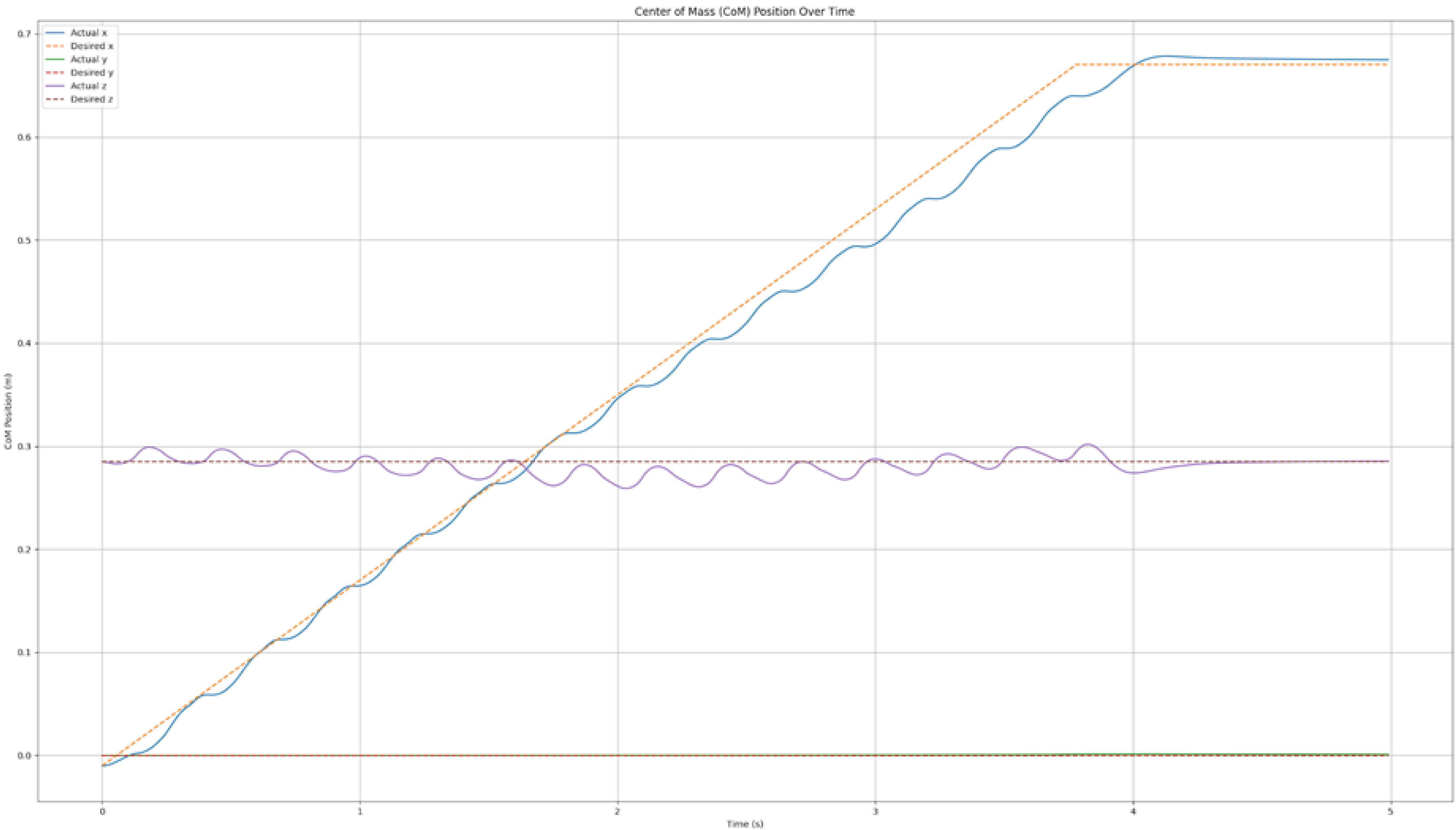
Results



$$ss\_duration = 0.1s, \quad ds\_duration = 0.04s, \quad v\_com\_ref\_x = 0.18m/s$$

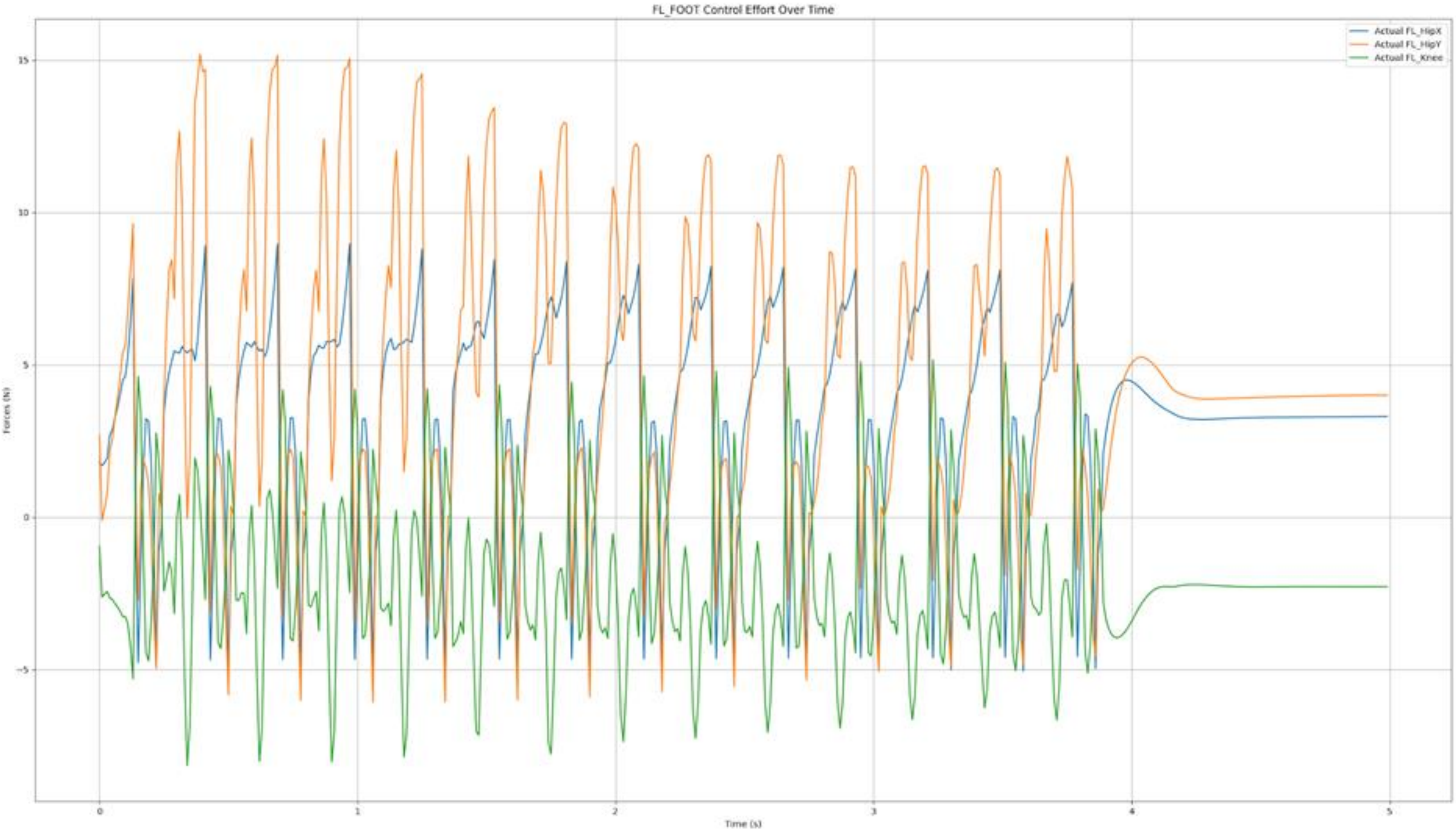
Pesudo-gallopping gait - Position of CoM

Results



Pesudo-gallopping gait - FL leg control effort

Results



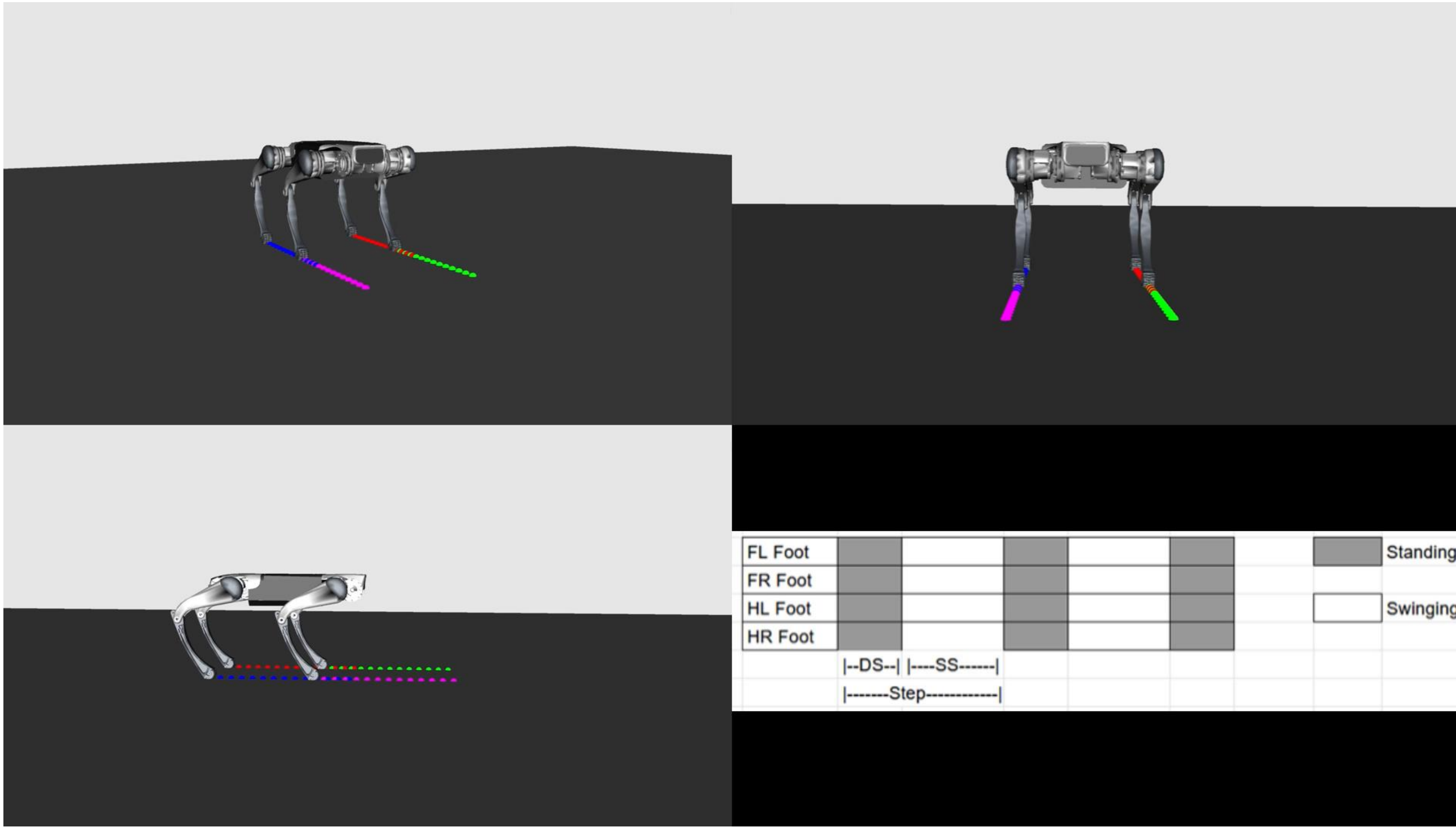
Ambling gait - Video

Results

$ss\_duration = 0.1s$, $\quad$ ds $\_duration = 0.1s$, $\quad v\_com\_ref\_x = 0.08m/s$

Pronking gait - Video

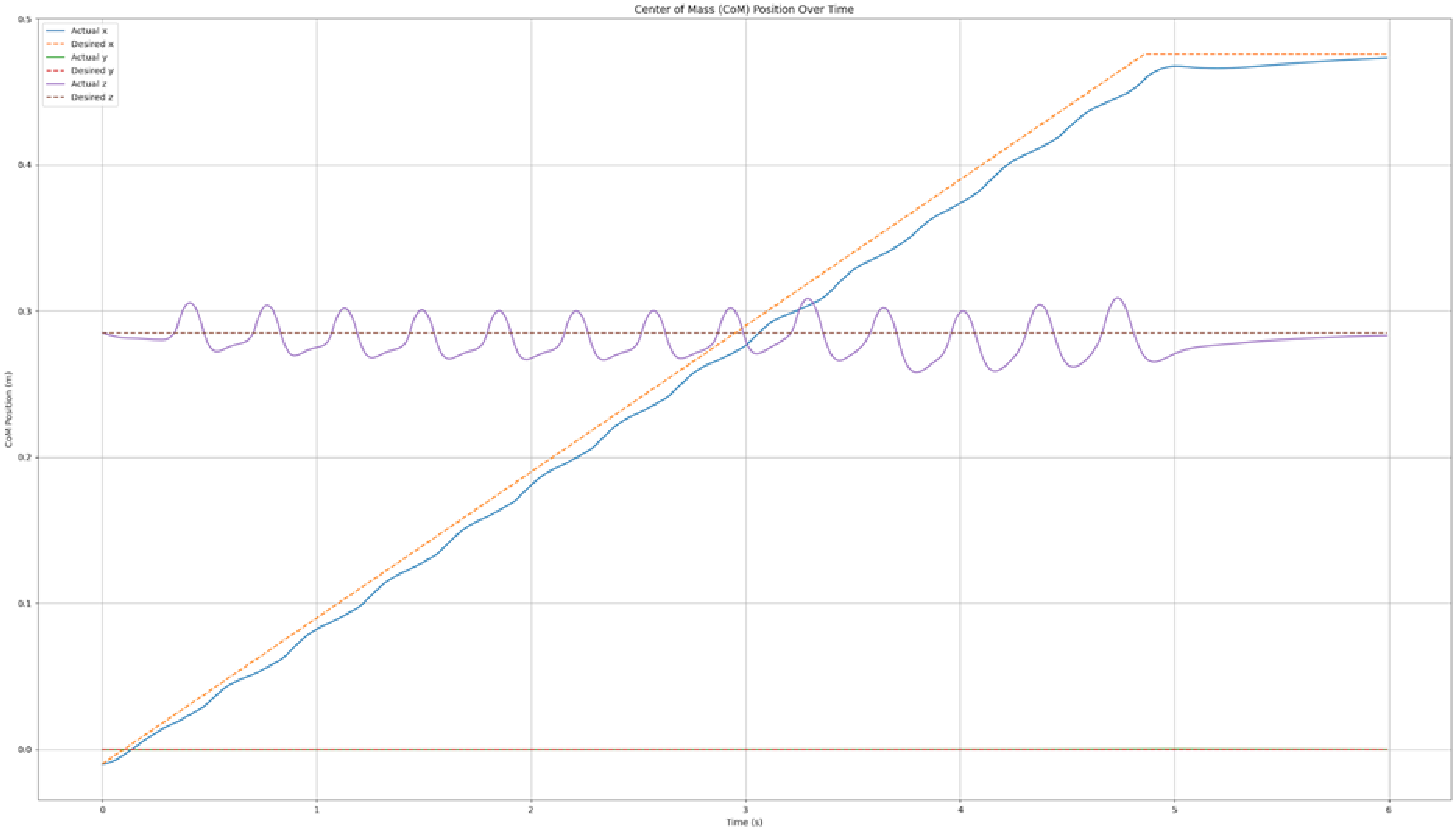
Results



$ss\_duration = 0.1s$, $\quad$ $ds\_duration = 0.08s$, $\quad$ $v\_com\_ref\_x = 0.1m/s$

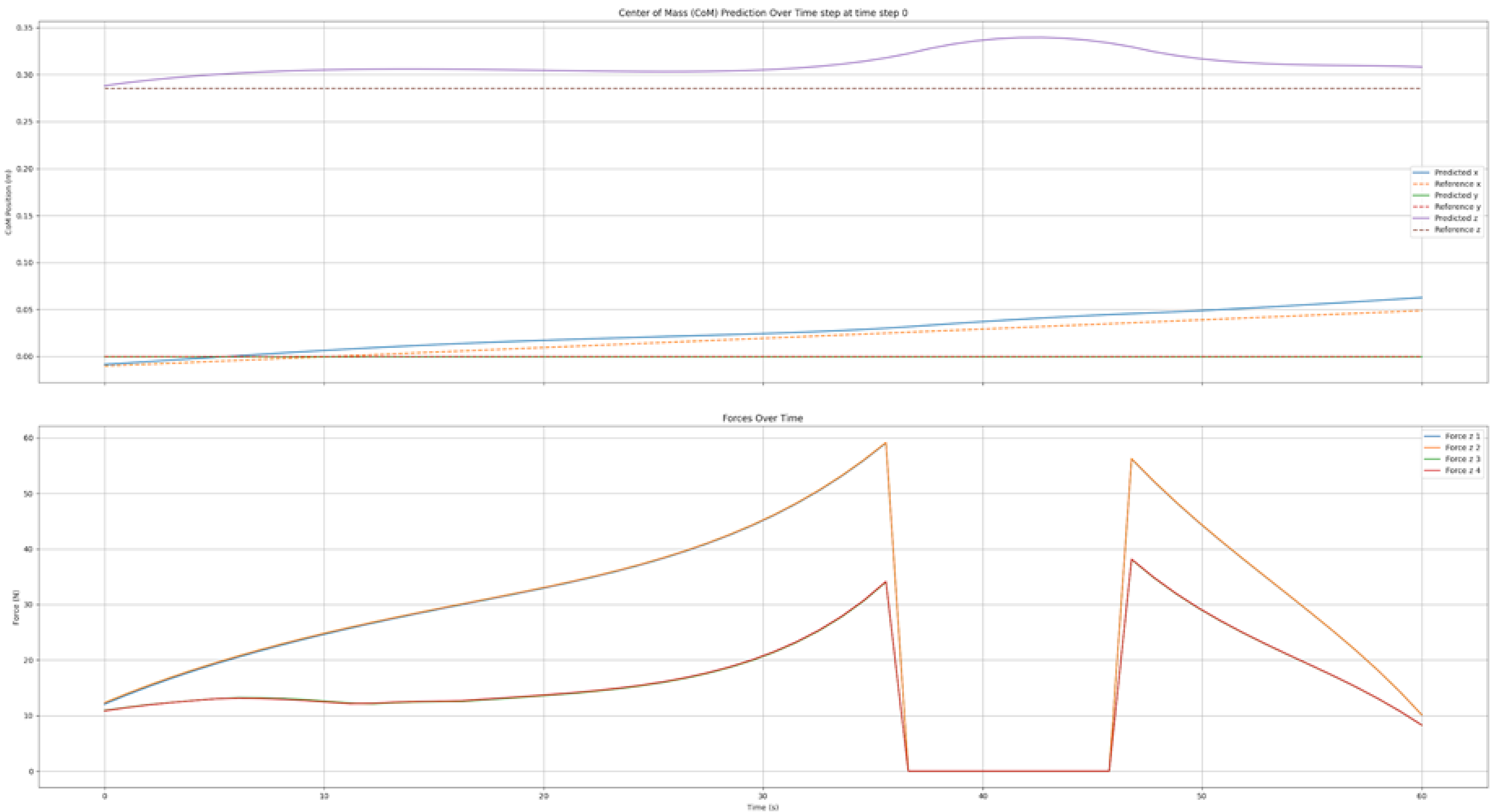
Pronking gait - Position of CoM

Results



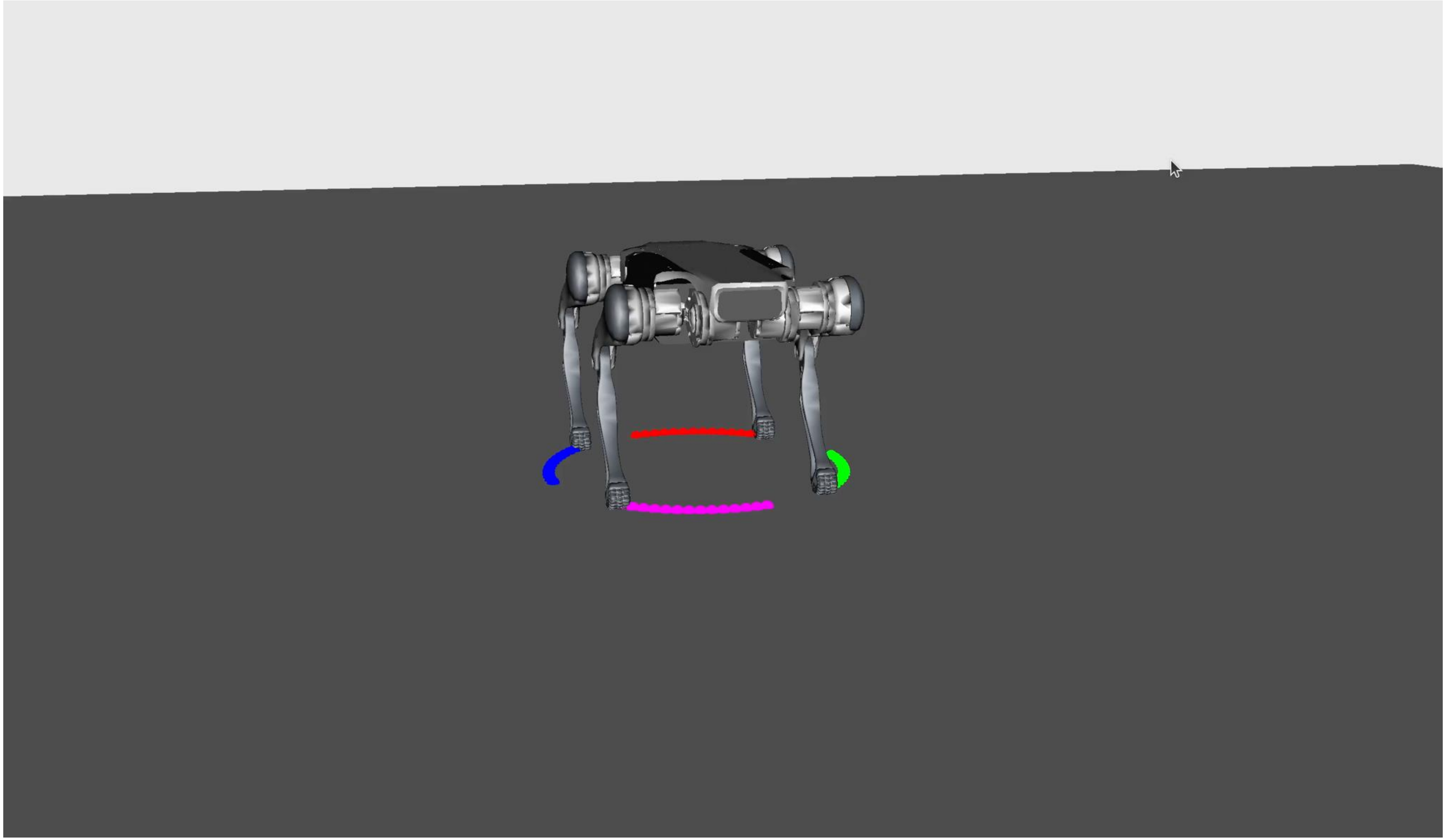
Pronking gait - MPC prediction at timestep 0

Results



Spinning in place

Results



Side-trotting

Results

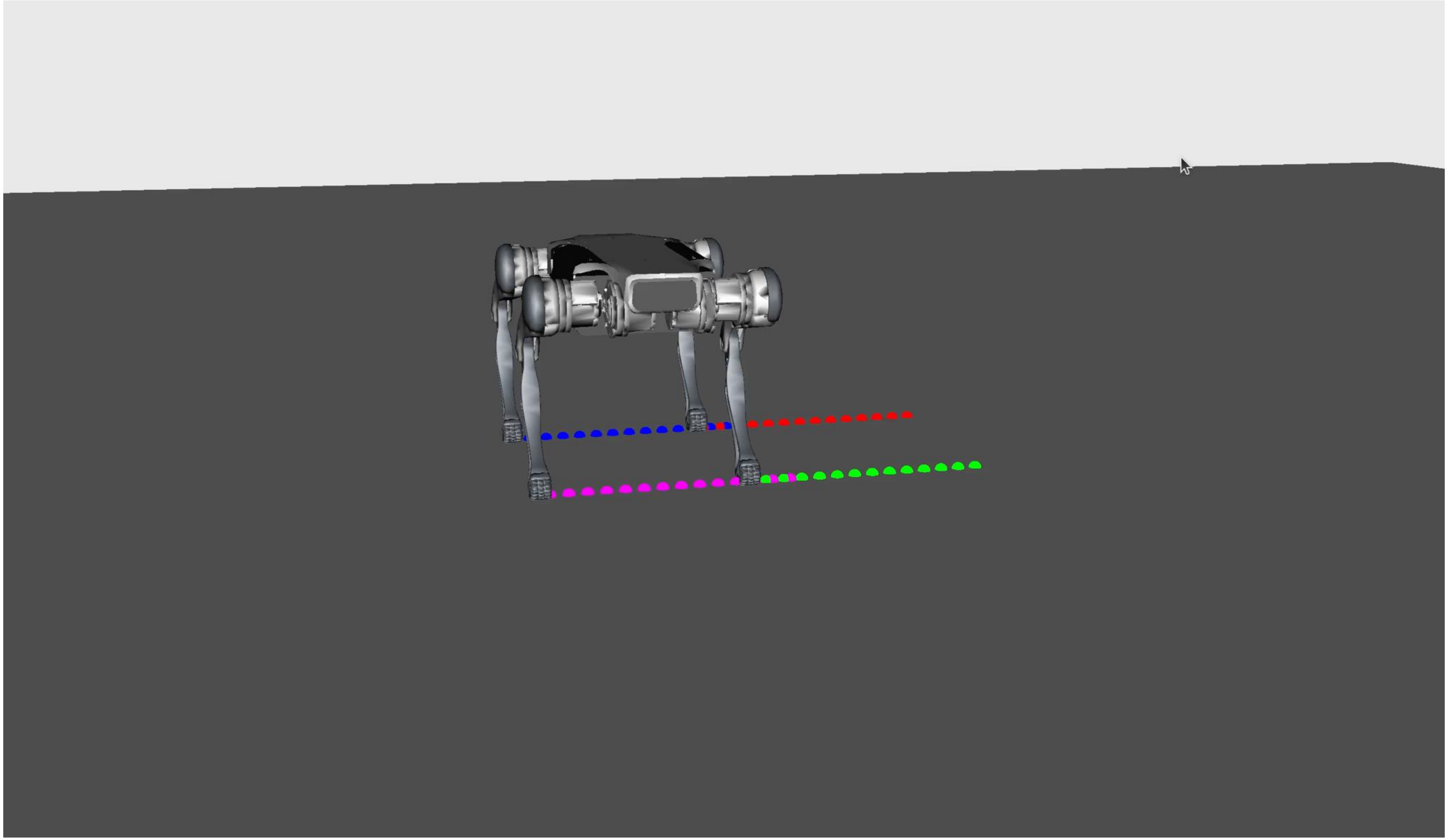


Table of Contents
Conclusion

- Introduction
- Footstep planner
- MPC
- Controllers
- Results
- Conclusion

Conclusion

Conclusion

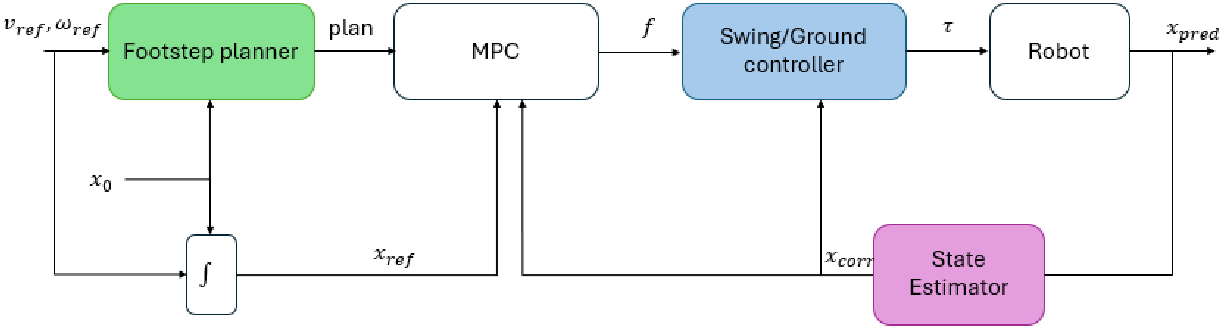
- **complete locomotion control pipeline**: implemented and validated Model Predictive Control for the Lite 3 quadruped robot
- **simulation handling of the problem**: effectively addressed the issue of ground penetration
- **integrated key components**: footstep planner, trajectory generator, ground and swing leg controllers
- **MPC convergence**: demonstrated successful convergence in simulation, with valid ground reaction forces enabling trajectory tracking.
- **smooth and stable gaits**: achieved reasonably smooth and stable locomotion with different gaits

Future works

Conclusion

Solid foundation for future work: framework sets the stage for real-world deployment and extensions to complex gaits or terrains. Some examples are:

- **multi-phase gait planner** for able the robot to handle different and more complex gaits
- **on-line planning** for real-time control
- **contact detection algorithm** for early or late contacts
- **state estimator** for real-time application



Thanks for your attention!

References

- [1] J. Di Carlo , P. M. Wensing , B. Katz , G. Bledt , and S. Kim, “Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control”, 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Madrid, Spain, October 1-5, 2018

- [2] F. Stark , J. Middelberg , D. Mronga , S. Vyas, F. Kirchner, “Benchmarking Different QP Formulations and Solvers for Dynamic Quadrupedal Walking”, arXiv:2502.01329v1 [cs.RO] 3 Feb 2025

- [3] Yijie Zhu, “Lite 3 urdf”, https://github.com/TopHillRobotics/quadruped-robot/blob/mpc-wbc/quadruped/config/lite3/lite3_robot.yaml