MATHÉMATIQUES VISION APPRENTISSAGE

école normale supérieure paris—saclay

Design, Implementation, and Analysis of an Intrinsically Stable MPC Framework for Bipedal Locomotion

Alexandre Mallez and Swann Cordier

Reviewed by Sylvère Bonnabel

January 19, 2026

Abstract

Abstract: This comprehensive report details the conception, mathematical formulation, and experimental analysis of a generic Model Predictive Control (MPC) framework for humanoid locomotion. Addressing the fundamental instability characterizing bipedal systems, we have developed a control architecture grounded in the Linear Inverted Pendulum Model (LIPM) and the Zero Moment Point (ZMP) stability criterion. The project followed a rigorous incremental development path, evolving from a decoupled 1D prototype to a fully coupled 2D MPC formulation capable of robust omnidirectional walking. A central contribution of this work is the rigorous implementation of the "truncated tail" terminal constraint, which theoretically guarantees infinite-time stability within a finite-horizon optimization ($N = 60$). To validate our algorithmic approach on a realistic platform, we elected to integrate the full-size humanoid robot **Talos** (95 kg) into our simulation pipeline. This choice allowed us to confront our control logic with the kinematic constraints of a 32-degree-of-freedom system. We provide an in-depth analysis of the system's robustness against external lateral perturbations (up to 150 N) and discuss the specific challenges of mapping reduced-order dynamics to whole-body Inverse Kinematics (IK).

1 Introduction

1.1 Context and Motivation

Bipedal locomotion stands as one of the most intricate and enduring challenges in the field of mobile robotics. Unlike wheeled platforms, which benefit from a wide and permanent polygon of support ensuring inherent static stability, humanoid robots are fundamentally under-actuated systems. They possess a hybrid dynamic nature, characterized by alternating phases of single and double support, and must continuously fight against gravity to maintain an upright posture. Consequently, they cannot directly command their global position; instead, they must manipulate Ground Reaction Forces (GRF) through intermittent contacts to influence the acceleration of their Center of Mass (CoM).

In this context, Model Predictive Control (MPC) has emerged as the de facto standard for generating dynamic walking motions. By optimizing a trajectory over a receding horizon, MPC allows the robot to anticipate future constraints, such as footstep locations and balance limits, and adjust its current actions accordingly.

1.2 Project Scope and Objectives

The primary objective of this project was to design, implement, and analyze a robust MPC solver for gait generation. While the core assignment focused on the algorithmic development of the controller, a theoretical solver alone is insufficient to capture the complexities of real-world robotics.

Therefore, we expanded the scope of the project by explicitly choosing to validate our algorithms on the **Talos** robot. Talos is a high-performance, torque-controlled full-size humanoid widely used in research. This choice was strategic: it allowed us to verify that our reduced-order model (MPC) could successfully drive a complex, high-dimensional kinematic structure (Whole-Body Control).

Our specific objectives were:

1. To derive and implement the Linear Inverted Pendulum dynamics from first principles.
2. To develop an online footstep generator capable of reactive steering based on velocity commands.
3. To formulate a Quadratic Program (QP) that guarantees dynamic balance and asymptotic stability.
4. To implement a Whole-Body Inverse Kinematics solver to map the optimized CoM trajectories onto the Talos joint space.

1.3 Development Roadmap

Our work draws inspiration from seminal contributions in the field, specifically the linear MPC formulation by P-B. Wieber [1], [3] and the intrinsically stable MPC frameworks proposed by Scianca et al. [2]. The development process was structured into three distinct phases:

- **Phase 1: 1D Prototyping.** Establishment of the Quadratic Programming (QP) solver basics and the visualization pipeline.
- **Phase 2: Decoupled 2D Locomotion.** Implementation of a reactive online footstep generator and extension of the control to the lateral plane using independent solvers.
- **Phase 3: Coupled Intrinsically Stable MPC.** Development of a coupled formulation enforcing a terminal stability constraint ("truncated tail") to enable robust non-linear trajectories and turns.

This report is organized as follows: Section 2 details the theoretical modeling of the biped; Section 3 describes the reactive footstep planning algorithm; Section 4 presents the core MPC formulation and the stability enforcement; Section 5 discusses the kinematic validation strategy; and Section 6 provides a critical analysis of the simulation results.

2 Dynamic Modeling: The Linear Inverted Pendulum

To render the optimal control problem computationally tractable in real-time , it is computationally prohibitive to utilize the full multi-body dynamics of the Talos robot. Therefore, we resort to a reduced-order model: the Linear Inverted Pendulum Model (LIPM).

2.1 Assumptions and Derivation

The LIPM is derived from the full robot dynamics under three major simplifying assumptions:

1. **Point Mass Approximation:** The total mass of the robot ($m = 95$ kg) is assumed to be concentrated at a single point, the Center of Mass (CoM).
2. **Constant Height Constraint:** The CoM is constrained to move on a horizontal plane at a constant height h_{com} above the ground. For Talos, we selected $h_{com} = 0.84$ m.
3. **Zero Angular Momentum:** The rate of change of angular momentum about the CoM is assumed to be negligible.

Consider the equilibrium of moments at the Zero Moment Point (ZMP), denoted as $\boldsymbol{z}$. By definition, the horizontal moments of the ground reaction forces vanish at the ZMP. This leads to the fundamental dynamic relationship linking the CoM position $\boldsymbol{p} = (x, y)$ to the ZMP:

$$\boldsymbol{z} = \boldsymbol{p} - \frac{h_{com}}{g}\ddot{\boldsymbol{p}} \tag{1}$$

where g is the gravitational acceleration vector. This second-order differential equation can be rearranged to highlight the system's dynamics:

$$\ddot{\boldsymbol{p}} = \omega^2(\boldsymbol{p} - \boldsymbol{z}) \quad \text{with} \quad \omega = \sqrt{\frac{g}{h_{com}}} \tag{2}$$

Here, ω represents the natural frequency of the inverted pendulum. This equation reveals the inherent instability of the system: the acceleration of the CoM is proportional to its distance

from the ZMP. To prevent falling, the robot must continuously "catch" its falling CoM by placing the ZMP appropriately.

2.2 Discrete-Time State Space Representation

To generate smooth, physically feasible trajectories that minimize mechanical stress on the actuators, we choose to control the **jerk** of the CoM (the time derivative of acceleration), denoted as $\dddot{\boldsymbol{p}}$. This ensures C^2 continuity of the position trajectory.

We define the state vector for a single axis (e.g., the sagittal axis x) at time step k as:

$$\boldsymbol{x}_k = \begin{bmatrix} p_k \\ \dot{p}_k \\ \ddot{p}_k \end{bmatrix} \tag{3}$$

Using a sampling period of $T = 0.025$ s, we discretize the triple integrator dynamics exactly. The discrete-time evolution is given by:

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + Bu_k \tag{4}$$

$$z_k = C\boldsymbol{x}_k \tag{5}$$

The system matrices $A \in \mathbb{R}^{3\times3}$, $B \in \mathbb{R}^{3\times1}$, and $C \in \mathbb{R}^{1\times3}$ are defined as:

$$A = \begin{pmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} T^3/6 \\ T^2/2 \\ T \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & -1/\omega^2 \end{pmatrix} \tag{6}$$

This state-space formulation allows us to predict the future evolution of the ZMP over a horizon of N steps as a linear function of the initial state and the future control sequence. Let $Z \in \mathbb{R}^N$ be the vector of predicted ZMP positions and $U \in \mathbb{R}^N$ be the vector of future jerk controls. We can write:

$$Z = P_x x_0 + P_u U \tag{7}$$

where P_x and P_u are the prediction matrices constructed by recursively applying the state equation.

3 Reactive Footstep Generation

One of the significant enhancements over basic pattern generators is our implementation of an online, reactive footstep planner. This system allows the robot to be steered dynamically via velocity commands, similar to how one would control a character in a video game.

3.1 Velocity Integration and Adaptive Timing

The planner receives a high-level velocity command vector $\boldsymbol{v}_{cmd} = [v_x, v_y, \dot{\theta}]^T$. A key feature of our implementation is **adaptive step duration**. In natural human gait, the frequency of steps

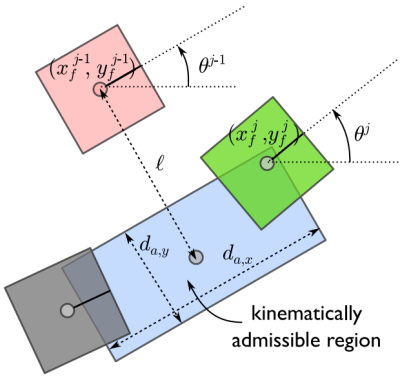

Figure 1: Kinematic projection on admissible region

increases with walking speed. To mimic this and ensure kinematic feasibility, we calculate the step duration T_{step} as:

$$T_{step} = T_{ref} \frac{\alpha + v_{ref}}{\alpha + \|\boldsymbol{v}_{cmd}\|} \tag{8}$$

With tuning parameters $\alpha = 0.1$ and nominal duration $T_{ref} = 0.3$s, this function smoothly reduces the step time as the robot accelerates, preventing the robot from taking steps that are spatially too long for its legs.

3.2 Kinematic Projection on Admissible Regions

Once the timing is determined, the planner computes the ideal landing position of the swing foot. This is done by integrating the commanded velocity relative to the current stance foot. However, the robot has strict geometric limits: it cannot cross its legs arbitrarily or stretch them beyond the limb length.

We enforce these limits by projecting the ideal footstep onto a **Kinematically Admissible Region**. This region is defined as a rectangle relative to the stance foot frame see Figure 1. For a stance foot at (x_{st}, y_{st}) with orientation θ, the next footstep coordinates $(\Delta x, \Delta y)$ in the local frame are constrained by:

- **Sagittal Reach:** $\Delta x \in [-0.25, 0.25]$ m.
- **Lateral Separation:** $\Delta y \in [l - 0.125, l + 0.125]$ m, where $l = 0.35$ m is the nominal pelvic width.

This projection logic uses 2D rotation matrices $R(\theta)$ to transform the integrated global displacement into the local frame of the stance foot, apply the clipping (clamping), and then transform it back to the global frame. This ensures that the generated reference footsteps are always reachable, regardless of the robot's orientation.

4 Coupled MPC with Intrinsic Stability

This section describes the core algorithmic contribution of our project. We transitioned from a naive decoupled implementation to a robust, coupled formulation that guarantees stability using the formulation proposed in recent literature.

4.1 The Necessity of Coupling

In our initial Phase 2, we solved two independent MPC problems for the x and y axes. While computationally efficient, this approach exhibited severe limitations during turning maneuvers.

1. **Geometric Coupling:** The ZMP constraints are defined by the support polygon of the feet. When the robot turns, the feet are rotated in the world frame with a rotation matrix R_j. A rectangular constraint aligned with a rotated foot introduces a coupling between global x and global y coordinates. Decoupled solvers cannot respect these rotated constraints accurately.

2. **Divergence Issues:** The standard LIPM is unstable. Minimizing ZMP error over a finite horizon does not guarantee that the state at the end of the horizon is viable.

4.2 Theoretical Coupled Formulation

Based on the method described in the article of N. Scianca [2], we formulate the non-linear trajectory optimization where x and y are coupled. Let δ be the sampling period (noted T in previous sections) and $\eta = \omega = \sqrt{g/h}$ be the natural frequency. The optimization problem becomes:

1. The Cost Function: We minimize the variation of the ZMP (ZMP velocity $\dot{X}_z, \dot{Y}_z$) and the tracking error of the foot placement:

$$\min_{\dot{X}_z^k, \dot{Y}_z^k} \|\dot{X}_z^k\|^2 + \|\dot{Y}_z^k\|^2 + \beta \left(\|X_f - \hat{X}_f\|^2 + \|Y_f - \hat{Y}_f\|^2 \right) \tag{9}$$

where β is a weighting factor and $\hat{X}_f, \hat{Y}_f$ are the reference foot placements.

2. ZMP Constraints (Coupled): The ZMP condition ensures the center of pressure remains inside the rotated support polygon. For step j with rotation matrix R_j, the constraint is expressed as:

$$-R_j^T \left(\begin{pmatrix} \delta \sum_{l=0}^{i} \dot{x}_z^{k+l} - x_f^j \\ \delta \sum_{l=0}^{i} \dot{y}_z^{k+l} - y_f^j \end{pmatrix} \right) \leq \frac{1}{2} \begin{pmatrix} d_{z,x} \\ d_{z,y} \end{pmatrix} - R_j^T \begin{pmatrix} x_z^k \\ y_z^k \end{pmatrix} \tag{10}$$

Here, $d_{z,x}$ and $d_{z,y}$ represent the dimensions of the foot support area.

3. Kinematic Constraints: The reachability of the next footstep relative to the previous one $(j-1)$ is governed by:

$$-R_{j-1}^T \left(\begin{pmatrix} x_f^j - x_f^{j-1} \\ y_f^j - y_f^{j-1} \end{pmatrix} \right) \leq \pm \begin{pmatrix} 0 \\ l \end{pmatrix} + \frac{1}{2} \begin{pmatrix} d_{a,x} \\ d_{a,y} \end{pmatrix} \tag{11}$$

In our implementation, as noted in our footstep generator design, this condition is **verified offline** during the step generation phase before the MPC solves the QP.

4.3 The "Truncated Tail" Stability Constraint

The most critical constraint handles the intrinsic instability of the LIPM. The unstable component of the CoM motion, denoted x_u (or Capture Point), can be expressed as an infinite sum of

future ZMP velocities. The general stability condition requires the divergent state at the end of the control horizon $(k + C)$ to be bounded:

$$x_u^{k+C} = x_z^{k+C} + \frac{1 - e^{-\eta\delta}}{\eta} e^{C\eta\delta} \sum_{i=C}^{\infty} e^{-i\eta\delta} \dot{x}_z^{k+i} \tag{12}$$

Ensuring this condition over an infinite horizon is computationally impossible in a standard QP. To resolve this, we employed the **Truncated Tail Method**. We assume that after the prediction horizon, the robot stops (or maintains a constant state), meaning the velocity becomes zero ($\dot{x}_z = 0$ for $i \geq C$).

Under this simplification, the infinite sum vanishes, and the stability constraint reduces to a simple terminal equality constraint:

$$x_u^{k+C} = x_z^{k+C} \tag{13}$$

or equivalently, matching the terminal Capture Point to the terminal ZMP position. This constraint is linear and was successfully integrated into our OSQP formulation as $A_{eq}\mathbf{U} = \boldsymbol{b}_{eq}$. This ensures that the generated trajectory always tends towards a stable equilibrium at the end of the horizon, preventing the CoM from drifting to infinity.

4.4 Quadratic Program Implementation

Combining these elements, we solve the following QP at each control cycle:

$$\begin{aligned} \min_{\mathbf{U}} \quad & \frac{Q}{2}\|Z_{pred} - Z_{ref}\|_2^2 + \frac{R}{2}\|\mathbf{U}\|_2^2 \\ \text{subject to:} \quad & G_{rot}\mathbf{U} \leq \boldsymbol{h}_{rot} \quad \text{(Coupled ZMP Constraints)} \\ & A_{tail}\mathbf{U} = \boldsymbol{b}_{tail} \quad \text{(Truncated Stability Constraint)} \end{aligned}$$

We selected weights $Q = 1.0$ and $R = 10^{-4}$, prioritizing precise ZMP tracking over control smoothness.

4.4.1 Implementation Challenge: The Sensitivity of Stability

A significant hurdle encountered during Phase 3 was a persistent numerical instability. The robot would walk successfully for several steps before the CoM would diverge exponentially. After extensive debugging, we identified an error in the construction of the stability matrix A_{eq}. Specifically, the term propagating the initial state $\boldsymbol{x}_0$ to the terminal step N was incorrectly scaled. This introduced a mathematical "bias," essentially telling the solver that the robot was falling when it was not. The solver would react by applying extreme inputs, destabilizing the system. Correcting this matrix stabilized the gait immediately, proving the effectiveness of the truncated tail method.

5 Validation and Whole-Body Control

While the MPC output provides an optimal trajectory for the center of mass, applying this to a robot requires transforming these Cartesian coordinates into joint angles. This section details the validation pipeline we established using the Talos robot model.

5.1 Kinematic Simulation Environment

We utilized the **Pinocchio** library for rigid body algorithms and **Meshcat** for 3D visualization. This setup allowed us to verify the feasibility of the generated trajectories against the robot's kinematic limits (joint ranges and limb lengths) which are not explicitly handled by the linear MPC.

5.2 Inverse Kinematics (IK) Formulation

To track the CoM trajectory, we implemented a velocity-based Inverse Kinematics solver. The solver computes the joint velocity vector $\dot{\boldsymbol{q}} \in \mathbb{R}^{32}$ that minimizes the error between the desired and actual end-effector task velocities. We employed a damped pseudo-inverse Jacobian method to handle singularities:

$$\dot{\boldsymbol{q}} = J^{\dagger}(\boldsymbol{v}_{desired} - J\dot{\boldsymbol{q}}_{current}) + (I - J^{\dagger}J)\dot{\boldsymbol{q}}_{secondary} \tag{14}$$

where $J^{\dagger}$ is the Moore-Penrose pseudo-inverse.

We defined a hierarchical stack of tasks to resolve redundancy: 1. **Feet Contact Task:** A strict constraint to keep the support feet fixed on the ground and the swing foot following the planned arc. This is crucial to prevent foot slippage. 2. **CoM Task:** Tracking the position (x, y) output by the MPC. 3. **Posture Task:** A low-priority task in the null space to maintain a natural reference posture and prevent joint drift.

5.3 The "Exorcist" Artifact and Kinematic Redundancy

During 180-degree turning maneuvers, we observed a visual artifact we termed the "Exorcist Effect." While the feet rotated correctly to follow the path, the robot's torso orientation (Yaw) was not explicitly constrained in the primary tasks. Consequently, the torso minimized its rotation in the world frame, eventually ending up facing backwards relative to the feet.

Attempting to add a strict orientation task for the torso introduced discontinuities and jitter in the IK solution due to the discrete nature of footsteps (sudden changes in reference orientation). We opted to prioritize the smoothness of the CoM motion analysis over visual perfection, accepting this artifact in the final rendering as a limitation of our simple IK scheme, independent of the MPC's quality.

6 Simulation Results and Robustness Analysis

Extensive simulations were conducted to validate the framework. The simulation parameters were set to: Horizon $N = 60$ (1.5 seconds), Time step $T = 0.025$s.

6.1 Nominal Omnidirectional Walking

The controller demonstrates stable and consistent behavior for nominal walking tasks. Straight-line, lateral, and curved motions are successfully executed at velocities up to 0.4 m/s, while the ZMP remains strictly within the support polygon at all times. The resulting CoM motion exhibits the characteristic lateral oscillations of bipedal locomotion, shifting toward the stance foot prior to each swing phase.

Figure 2 illustrates a representative linear walking sequence of the Talos robot. The corresponding lateral projection of the CoM and ZMP is shown in Figure 3. The ZMP closely follows the CoM dynamics while remaining confined within the support area, confirming the effectiveness of the balance constraints enforced by the MPC.

Finally, the top view of the trajectory in Figure 4 shows the global evolution of the motion in the horizontal plane. The smooth and continuous CoM path confirms that the generated footsteps and CoM trajectories are mutually consistent and remain within the kinematic reachability limits of the robot throughout the walk.

Figure 2: Linear walk of the robot

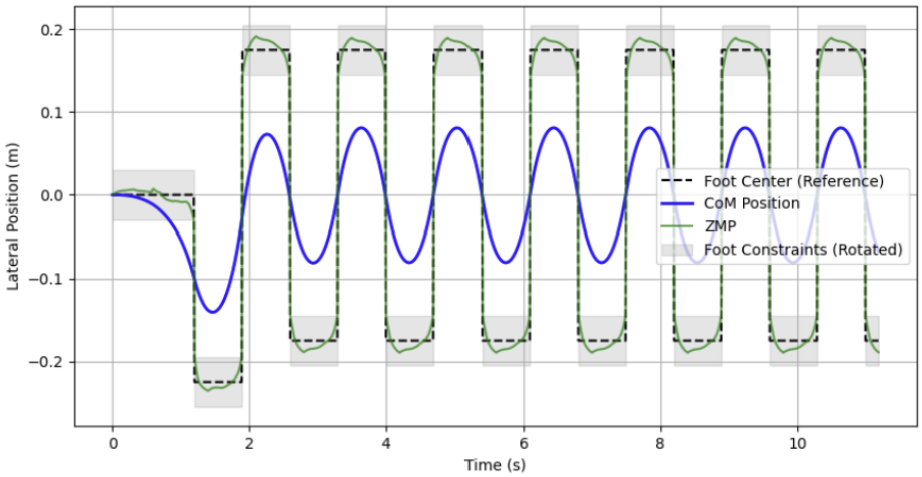

Figure 3: Lateral projection of the CoM and ZMP during the walk

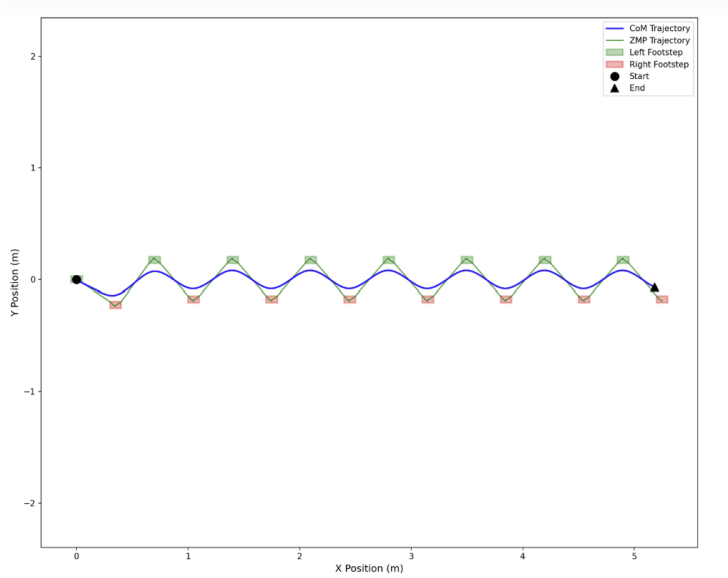

Figure 4: Top view of the robot trajectory

6.2 Walking with Coupled Coordinates

The coupled MPC formulation enables stable omnidirectional walking, including curved trajectories and changes of heading, which cannot be handled reliably with decoupled sagittal and lateral controllers. By expressing the ZMP constraints in the rotated frames of the support feet, the admissible support region naturally follows the orientation of the gait.

Figure 5 shows the lateral projection of the CoM and ZMP during a turning motion. Figure 6 represents the top view of the trajectory, which highlights the ability of the controller to generate smooth curved paths. The CoM trajectory is continuous and free of geometric distortions, confirming that the coupled optimization consistently satisfies the support constraints throughout the motion.

These results illustrate that coupling the sagittal and lateral dynamics is essential for generating geometrically consistent and stable omnidirectional walking, especially in the presence of rotations and non-rectilinear trajectories.

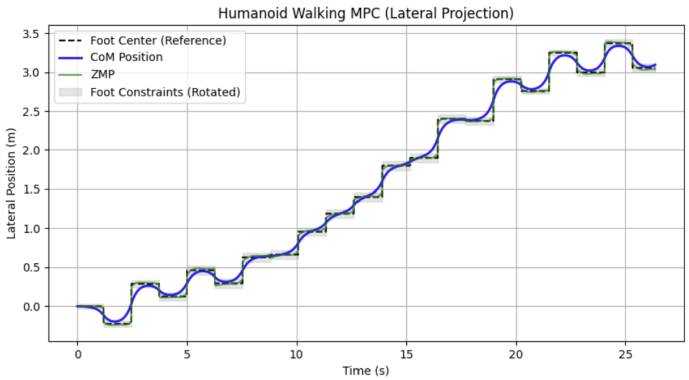

Figure 5: Lateral projection of the CoM and ZMP during a turn $\omega = 0.15\ rad/s$

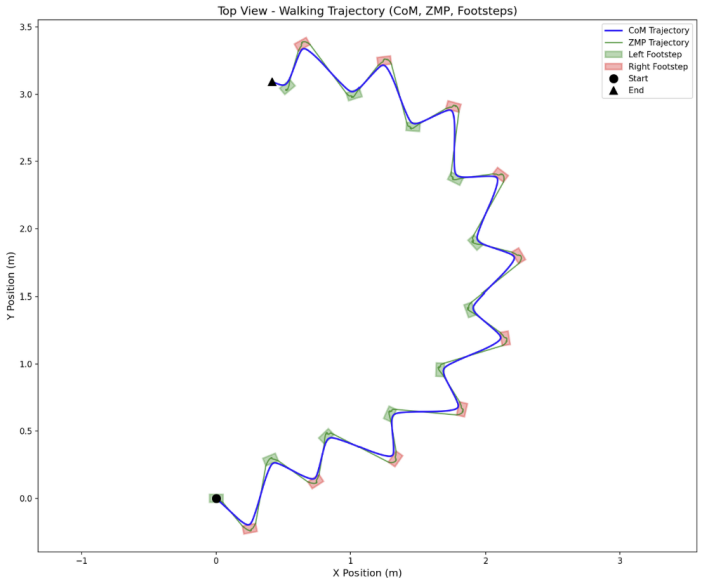

Figure 6: Top view of the robot trajectory during a turn $\omega = 0.15\ rad/s$

6.3 Robustness to External Lateral Forces

To quantify the stability margin of our controller, we subjected the robot to impulsive lateral forces applied to the CoM for a duration of 1 second at $t = 7$s. To simulate the effect of these external pushes realistically, we modified the dynamic update equations to expressly include an external force term F_k acting on the Center of Mass. The discrete-time state update and ZMP output equations become:

$$\begin{cases} \hat{x}_{k+1} = A\hat{x}_k + B\,\ddot{x}_k + B_d\frac{F_k}{m}, \\ z_k = x_k - \frac{h}{g}\ddot{x}_k + \frac{h}{g}\frac{F_k}{m}. \end{cases} \tag{15}$$

where $B_d = [T^2/2, T, 0]^T$ represents the discretization of the force's effect on position and velocity, and $m = 95$ kg is the robot mass. This formulation ensures that the ZMP shift caused by the external acceleration is physically accurate.

6.3.1 Linear Recovery Regime ($F < 90$ N)

For forces up to approximately 90 N, the controller exhibits robust recovery behavior. Upon detecting the disturbance (via the state update), the MPC instantly modifies the planned CoM trajectory. It accelerates the CoM in the direction of the force, thereby shifting the ZMP towards the edge of the support foot. This creates a restoring moment that decelerates the lateral drift. Once the force is removed, the "truncated tail" constraint ensures the CoM smoothly returns to the nominal path. This behaviour is presented in Figure 7.

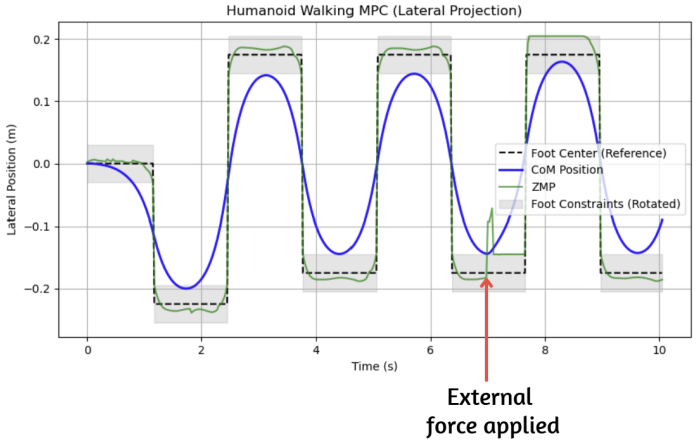

Figure 7: Stabilization of the CoM ($F = 90N$)

6.3.2 Failure Regime ($F > 150$ N)

When the external force exceeds roughly 150 N (approx. 15% of the robot's weight), the system fails. The visualizer shows the robot tipping over. Mathematically, this corresponds to the saturation of the ZMP constraints. The ZMP reaches the boundary of the support polygon, and the robot cannot generate enough torque to counteract the external force. This is illustrated in Figure 8.

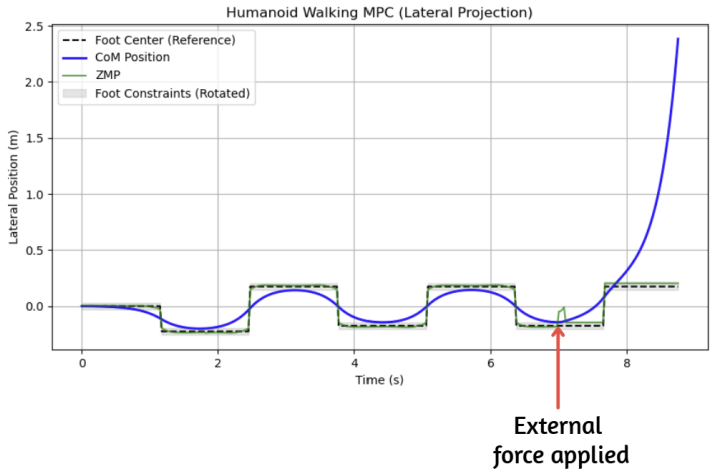

Figure 8: Divergence of the CoM ($F = 150N$)

6.4 Critical Analysis of Limitations

The observed stability limit of 90-150 N highlights a fundamental architectural limitation of our approach: the decoupling of footstep planning and CoM control. In our framework, footstep positions are computed **offline** relative to the MPC optimization loop. The MPC treats foot placements as immutable hard constraints. When a human is pushed hard, they perform a **"step recovery"** strategy: they place their foot wider than planned to expand the support polygon and catch the fall. Our robot, however, is constrained to land its foot exactly where the planner decided *before* the push occurred. It attempts to balance on a footprint that is effectively too narrow for the disturbance.

To overcome this limit, the footstep locations would need to be included as decision variables within the QP . This would, however, make the problem non-linear or require Mixed-Integer Quadratic Programming (MIQP), significantly increasing computational complexity.

7 Conclusion and Future Perspectives

This project has successfully demonstrated the design and implementation of a complete locomotion control pipeline for a full-size humanoid robot. By starting from the fundamental equations of the Linear Inverted Pendulum and progressively layering complexity, from 1D control to coupled 2D dynamics with intrinsic stability constraints, we have built a functional and robust walking controller.

While the current framework establishes a solid baseline, several avenues for improvement have been identified to bridge the gap towards deployment on a real physical platform:

7.1 Integrated Footstep Optimization (Capturability)

The primary limitation of the current system is the decoupling between footstep planning and CoM control. When the robot is pushed beyond 150 N, it fails because it tries to balance on foot placements that were decided before the push occurred. Future iterations should integrate footstep locations (x_f, y_f) as decision variables directly within the MPC. This would allow the robot to perform **"step recovery"** strategies, dynamically widening its stance to catch a fall, effectively making the support polygon adaptive to disturbances.

7.2 Explicit Force Integration in Dynamics

Currently, external forces are treated as unmodeled disturbances that the feedback controller must reject. To improve compliance, we could explicitly model external disturbances (F_{ext}) in the MPC prediction model. By augmenting the state vector with an estimator of external forces, the solver could anticipate the steady-state deviation required to counteract a push, rather than reacting solely to the resulting error.

7.3 Predictive Tail Estimation

We employed the "Truncated Tail" method, which assumes the robot stops or maintains zero velocity after the prediction horizon. A more advanced **"Predictive Tail"** could be implemented, estimating the infinite horizon value based on the assumption of a periodic gait (limit cycle). This would provide a more accurate estimation of the terminal stability condition without requiring the robot to virtually "stop" at the end of every horizon, potentially allowing for shorter prediction horizons and reduced computational load.

7.4 Enhanced Whole-Body Control

Finally, the kinematic validation revealed limitations in our Inverse Kinematics solver.

- **Torso Orientation:** The "Exorcist effect" must be addressed by implementing a prioritized task hierarchy (e.g., using a Hierarchical QP solver) that strictly enforces a forward-facing torso orientation task, preventing the upper body from drifting during turns.
- **Swing Leg Trajectories:** Instead of simple linear interpolation, the swing foot trajectory should be generated using higher-order polynomials. This would ensure better ground clearance control, vertical landing velocities to minimize impact, and collision avoidance with the stance leg.

References

[1] Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. Online Walking Motion Generation with Automatic Footstep Placement. *Advanced Robotics*, 24(5-6):719–737, January 2010. ISSN 0169-1864, 1568-5535. doi: 10.1163/016918610X493552.

[2] Nicola Scianca, Daniele De Simone, Leonardo Lanari, and Giuseppe Oriolo. MPC for Humanoid Gait Generation: Stability and Feasibility. *IEEE Transactions on Robotics*, 36(4): 1171–1188, August 2020. ISSN 1552-3098, 1941-0468. doi: 10.1109/TRO.2019.2958483.

[3] Pierre-brice Wieber. Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 137–142, University of Genova, Genova, Italy, December 2006. IEEE. ISBN 978-1-4244-0199-4 978-1-4244-0200-7. doi: 10.1109/ICHR.2006.321375.