Overview of MPC problems

December 3, 2025

Contents

1 Whole body controller

The whole-body controller is formulated as a quadratic program, defined by the following optimization problem:

$$\mathbf{x} = \begin{pmatrix} \ddot{\mathbf{q}}_{fb} \\ \ddot{\mathbf{q}}_j \\ \mathbf{F}_{c_l} \\ \mathbf{F}_{c_r} \end{pmatrix} = \begin{pmatrix} \ddot{\mathbf{q}} \\ \mathbf{F}_{c_l} \\ \mathbf{F}_{c_r} \end{pmatrix}$$

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T H \mathbf{x} + \mathbf{f}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b}, \\ & d_{\min} \leq C\mathbf{x} \leq d_{\max}. \end{aligned}$$

The optimization function includes:

- $||\ddot{\mathbf{q}}||^2$
- $||\mathbf{F_{ci}}||^2$
- tasks in the form:

$$\sum_j w_j \, ||J_j \, \ddot{q} + \dot{J}_j \, \dot{q} - a_j^d - k_p e_j - k_d \dot{e}_j||^2$$

For

  - com position tracking
  - r/l wheel center position + orientation tracking
  - orientation of the base link
  - joints regulation
- angular momentum
- **equality constraint:**
  - floating–base unactuated (u) dynamic constraint:

$$M_u\,\ddot{\mathbf{q}} + \mathbf{n_u} - J_{c,l,u}^T\,\mathbf{w_{c,l}} - J_{c,r,u}^T\,\mathbf{w_{c,r}} = 0$$

    $\mathbf{w_{c,i}}$ represents the contact wrench, given that the contact surface is modeled as a small patch approximated by a line segment whose length equals the wheel width.
  - pure rolling constraint at acceleration level (for $i = l, r$):

$$J_{ci}^{lin}\ddot{\mathbf{q}} = -\dot{J}_{ci}^{\;lin}\dot{\mathbf{q}} + [0 + K_d(0 - J_{ci}^{lin}\dot{\mathbf{q}})]$$

- **inequality constraint:**
  - contact wrench constraint (for $i = l, r$):

$$-\bar{\mu}\,F_{c_{iz}}^c \;\leq\; F_{c_{ix}}^c \;\leq\; \bar{\mu}\,F_{c_{iz}}^c,$$
$$-\bar{\mu}\,F_{c_{iz}}^c \;\leq\; F_{c_{iy}}^c \;\leq\; \bar{\mu}\,F_{c_{iz}}^c.$$

    The forces here are expressed in then contact frame.
  - unilaterality constraint (for $i = l, r$):

$$F_{c_{iz}}^c \;\geq\; 0.$$

  - joint limits:

$$\mathbf{q_j}^{min} \leq \mathbf{q_j} \leq \mathbf{q_j}^{max}$$
$$\dot{\mathbf{q}}_{\mathbf{j}}^{min} \leq \dot{\mathbf{q}}_{\mathbf{j}} \leq \dot{\mathbf{q}}_{\mathbf{j}}^{max}$$

The solution is used to compute the inverse dynamics for the actuated (a) joints:

$$\tau_{\mathbf{a}} = M_a\,\ddot{\mathbf{q}} + \mathbf{n_a} - J_{c,l,a}^T\,\mathbf{w_{c,l}} - J_{c,r,a}^T\,\mathbf{w_{c,r}}$$

2 MPC

The MPC is used to solve the optimization problem that produces the feasible trajectories to be tracked by the WBC.

2.1 The working version

The implemented MPC is based on the Variable Height Inverted Pendulum (VHIP) model and uses a Differential Dynamic Programming (DDP) solver to compute the optimal trajectories.
The VHIP dynamics are described by:

$$m(\ddot{\mathbf{p}}_{\mathbf{com}} - \mathbf{g}) = \mathbf{f}$$

$$(\mathbf{p}_{\mathbf{zmp}} - \mathbf{p}_{\mathbf{com}}) \times \mathbf{f} = 0$$

The MPC generates the following trajectories:

- $p_{com}(t), \dot{p}_{com}(t), \ddot{p}_{com}(t)$
- $p_{zmp}(t), \dot{p}_{zmp}(t), \ddot{p}_{zmp}(t)$

The original problem is split into two sequential stages that are solved separately over the same control horizon.
This decomposition allows the original non-linear optimization problem to be reformulated as two subproblems, each written as a Quadratic Program (QP).
Removing the nonlinearity appears to have helped the DDP solver handle this class of problems.

2.1.1 QP-z

The sate and inputs: $\mathbf{x} = \begin{pmatrix} z_{com} \\ \dot{z}_{com} \end{pmatrix}, \quad u = F_{cz}$
The dynamic is:

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ 1/m \end{pmatrix} u + \begin{pmatrix} 0 \\ -g \end{pmatrix}$$

The problem is formulated as:

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N} l(x_k, u_k) + l_N(x_N, u_N) \\ \text{s.t.} \quad & x_{k+1} = x_k + \Delta(Ax_k + Bu_k + c) \\ & x_0 = x_{current} \end{aligned}$$

The running cost is:

$$l(x_k, u_k) = w_h||z_{com} - z_{com}^{ref}||^2 + w_{vh}||\dot{z}_{com} - \dot{z}_{com}^{ref}||^2 + w_{Fcz}||F_{cz} - mg||^2$$

The terminal cost is analogous (without the input).

2.1.2 QP-xy

The sate and inputs: $\mathbf{x} = \begin{pmatrix} x_{com} \\ y_{com} \\ \dot{x}_{com} \\ \dot{y}_{com} \\ x_{zmp} \\ y_{zmp} \\ \dot{x}_{zmp} \\ \dot{y}_{zmp} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \ddot{x}_{zmp} \\ \ddot{y}_{zmp} \end{pmatrix}$

The dynamics is:

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & I_2 & 0 & 0 \\ \lambda_k I_2 & 0 & -\lambda_k I_2 & 0 \\ 0 & 0 & 0 & I_2 \\ 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0_2 & 0_2 \\ 0_2 & 0_2 \\ 0_2 & 0_2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{u}$$

Since the dynamic is linear time varying the solution to the previous problem is used to reconstruct λ_k at each time step, according to:

$$\lambda_k = \frac{\ddot{z}^k_{com} + g}{z^k_{com} - z^k_c}$$

To preserve ground contact, the ZMP height z_c is assumed to be constant and equal to zero.
The optimization problem is formulated as:

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N} l(x_k, u_k) + l_N(x_N, u_N) \\ \text{s.t.} \quad & x_{k+1} = x_k + \Delta(A_k x_k + B u_k) \\ & x_0 = x_{current} \\ & h_{ter}(x) = 0 \end{aligned}$$

The running cost is:

$$\begin{aligned} l(x_k, u_k) = {} & w_{\text{com}} \left\| p^{x,y}_{\text{com}} - p^{x,y,\text{ref}}_{\text{com}} \right\|^2 + w_{\text{vcom}} \left\| \dot{p}^{x,y}_{\text{com}} - \dot{p}^{x,y,\text{ref}}_{\text{com}} \right\|^2 \\ & + w_z \left\| p^{x,y}_{\text{zmp}} - p^{x,y,\text{ref}}_{\text{zmp}} \right\|^2 + w_{zd} \left\| \dot{p}^{x,y}_{\text{zmp}} - \dot{p}^{x,y,\text{ref}}_{\text{zmp}} \right\|^2 \\ & + w_{\text{acc}} \left\| u \right\|^2 \end{aligned}$$

The terminal cost is analogous (without the input).
The terminal constraint is

$$h_{ter}(x) = \begin{pmatrix} x^N_{com} - x^N_{zmp} \\ y^N_{com} - y^N_{zmp} \end{pmatrix} = 0$$

This version of the MPC adopts a simplified formulation in which the ZMP is treated as a free-flying point, in order to validate the controller under a reduced problem setting. The non-holonomic constraints of the wheels are not enforced in this case, and the ZMP trajectory generated by the MPC is subsequently used to construct the desired trajectories for the left and right wheel centers as follows:

- $x_{l\_wheel}(t) = x_{zmp}, \qquad x_{r\_wheel}(t) = x_{zmp}$
- $y_{l\_wheel}(t) = -y\_offset, \qquad y_{r\_wheel}(t) = y\_offset$
- $z_{l\_wheel}(t) = R, \qquad z_{r\_wheel}(t) = R$

Where y_offset is a constant lateral displacement, computed empirically as the horizontal distance between the CoM and one wheel at the initial step of the simulation. R denotes the wheel radius, the wheel is assumed to be in contact with the ground vertically aligned, in this case the vertical coordinate of its center remains constant and equal to R.

2.2 Attempts with DDP

2.2.1 Momentum balancing equality constraint (open loop only)

The sate and inputs: $\mathbf{x} = \begin{pmatrix} x_{com} \\ y_{com} \\ z_{com} \\ \dot{x}_{com} \\ \dot{y}_{com} \\ \dot{z}_{com} \\ x_{zmp} \\ y_{zmp} \\ \dot{x}_{zmp} \\ \dot{y}_{zmp} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \ddot{x}_{zmp} \\ \ddot{y}_{zmp} \\ F_{cx} \\ F_{cy} \\ F_{cz} \end{pmatrix}$

In this case the dynamics of the system are linear and equal to:

$$\dot{\mathbf{x}} = \begin{pmatrix} 0_{3x3} & I_3 & 0_{3x2} & 0_{3x2} \\ 0_{3x3} & 0_{3x3} & 0_{3x2} & 0_{3x2} \\ 0_{2x3} & 0_{2x3} & 0_{2x2} & I_2 \\ 0_{2x3} & 0_{2x3} & 0_{2x2} & 0_{2x2} \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0_{3x2} & 0_{3x3} \\ 0_{3x2} & 1/m \cdot I_3 \\ 0_{2x2} & 0_{2x3} \\ I_2 & 0_{2x3} \end{pmatrix} \mathbf{u} + \begin{pmatrix} 0_3 \\ \mathbf{g} \\ 0_2 \\ 0_2 \end{pmatrix}$$

And the non linear constrained optimization problem is formulated as:

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N} l(x_k, u_k) + l_N(x_N, u_N) \\ \text{s.t.} \quad & x_{k+1} = x_k + \Delta(Ax_k + Bu_k + c) \\ & x_0 = x_{current} \\ & h(x,u) = 0 \\ & g(x,u) \leq 0 \\ & h_{ter}(x) = 0 \end{aligned}$$

The running cost is:

$$\begin{aligned} l(x_k, u_k) = {} & w_{\text{com}}^{x,y} \left\| p_{\text{com}}^{x,y} - p_{\text{com}}^{x,y,\text{ref}} \right\|^2 + w_{\text{vcom}}^{x,y} \left\| \dot{p}_{\text{com}}^{x,y} - \dot{p}_{\text{com}}^{x,y,\text{ref}} \right\|^2 \\ & + w_{\text{h}} \left\| z_{\text{com}} - z_{\text{com}}^{\text{ref}} \right\|^2 + w_{\text{vh}} \left\| \dot{z}_{\text{com}} - \dot{z}_{\text{com}}^{\text{ref}} \right\|^2 \\ & + w_z \left\| p_{\text{zmp}}^{x,y} - p_{\text{zmp}}^{x,y,\text{ref}} \right\|^2 + w_{zd} \left\| \dot{p}_{\text{zmp}}^{x,y} - \dot{p}_{\text{zmp}}^{x,y,\text{ref}} \right\|^2 \\ & + w_{\text{acc}} \left\| u_{acc} \right\|^2 \\ & + w_{\text{fxy}} \left\| u_{F_{cx,y}} \right\|^2 \\ & + w_{\text{fz}} \left\| u_{F_{cz}} - mg \right\|^2 \end{aligned}$$

The terminal cost is analogous (without the input).
The equality constraint is:

$$h(x, u) = (p_{zmp} - p_{com}) \times F_c = 0$$

The inequality constraint is:

$$g(x, u) = -F_{cz} \leq 0$$

The terminal constraint is

$$h_{ter}(x) = \begin{pmatrix} x_{com}^N - x_{zmp}^N \\ y_{com}^N - y_{zmp}^N \end{pmatrix} = 0$$

Recently I noted a problem in the warmstart of the DDP solver. Now the algorithm for the DDP in the pseudocode version is:

Algorithm 1 Differential Dynamic Programming (DDP)

Input: initial state x_{init}, horizon N, initial control sequence $\{u_k\}_{k=0}^{N-1}$
Output: improved control sequence $\{u_k^\star\}$ and state trajectory $\{x_k^\star\}$
Initialization
Initialize the trajectories initial guess: $x_{k+1} = x_{init}, u_k = u_0$ for $k = 0, \ldots, N-1$
Repeat until SOLVER_MAXITER or convergence is reached:
1. Backward Pass
2. Forward Pass
Update $\{u_k\}, \{x_k\}$ and check convergence.

After correcting this issue, the solver was able to return a solution even when the non-linear constraint was enabled (which had never occurred before, when the warmstart was not set correctly). However, during a series of experiments I observed that the resulting trajectory is strongly influenced by the warmstart choice.
As an illustrative example, consider the MPC executed in open-loop mode (i.e., at each iteration the next measured state is set equal to the first predicted state of the previous MPC step). For the following initial condition:

$$x_{\text{init}} = \Big[0.002,\ 0.0003,\ 0.39,\ 0.01,\ -0.02,\ -0.1,\ 0.0003,\ -0.0002,\ 0.0,\ 0.0\Big],$$

The references are the constant positions:

$$p_{com}^{ref} = \Big(0.0,\ 0.0,\ 0.4\Big)^T, \qquad p_{zmp}^{x,y,ref} = \Big(0.0,\ 0.0\Big)^T$$

the warmstart used at the first MPC iteration is:

$$x_{\text{initial\_guess}} = x_{\text{init}},$$

$$u_{\text{initial\_guess}} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{F}^{\text{init}} \end{pmatrix}$$

for the entire horizon.
Here, $\mathbf{F}^{\text{init}}$ denotes an initial contact force chosen to be collinear with the COM–ZMP segment, so as to satisfy the non-linear constraint

$$(p_{\text{zmp}}^{\text{init}} - p_{\text{com}}^{\text{init}}) \times \mathbf{F}^{\text{init}} = 0,$$

with $p_{\text{zmp}}^{\text{init}}$ and $p_{\text{com}}^{\text{init}}$ computed consistently from x_{init}. $\mathbf{F}^{init}$ is rescaled in order to have the z-component equal to mg.
note: If instead the warmstart for $u_{\text{initial\_guess}}$ is $\begin{pmatrix} 0 & 0 & 0 & 0 & mg \end{pmatrix}^T$ the solver is not able to find a solution.
In this experiment, the solver performs only **one** DDP iteration per control cycle (SOLVER_MAXITER = 1).
The results obtained under these conditions are shown in Figure 1 (prediction at time step $t = 0$) and in Figure 2 (prediction at time step $t = 4$). Once the solver reaches a plausible solution (as in Figure 2) it continues to produce valid and consistent predictions in the subsequent cycles.

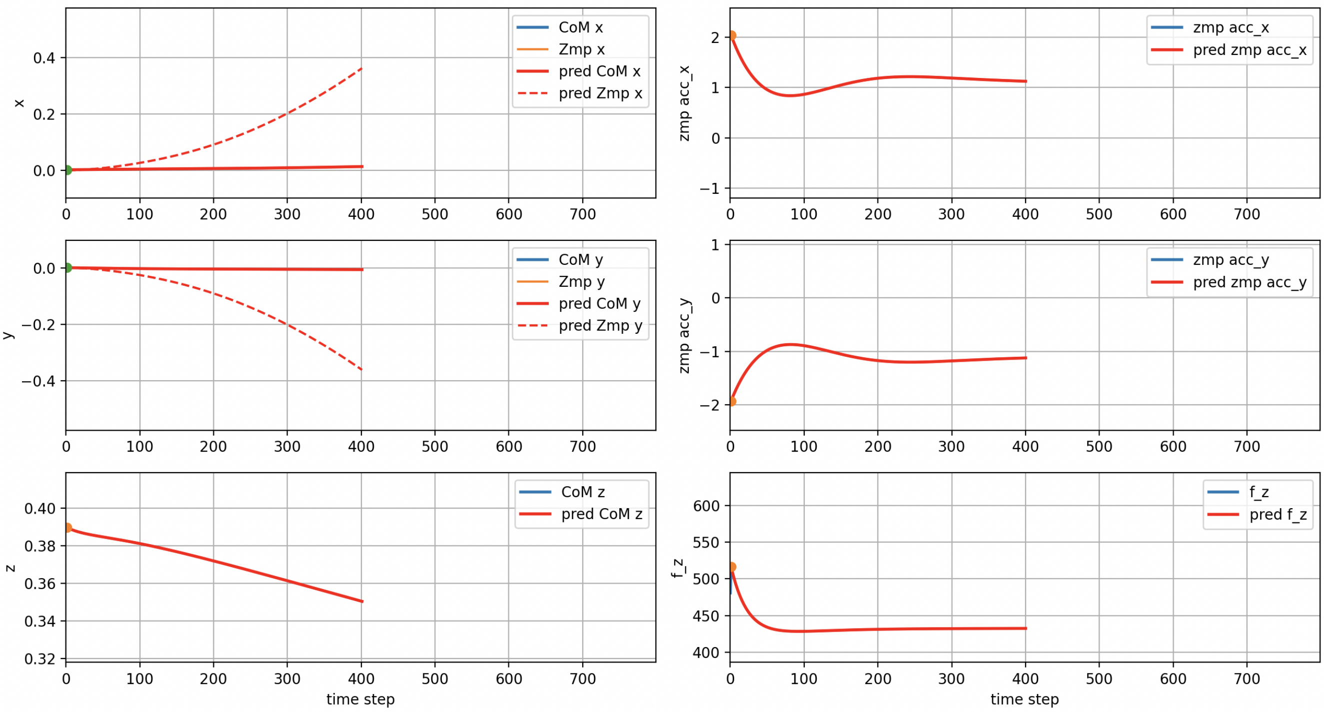

Figure 1: MPC history and predictions at time step t=0. with SOLVER_MAXITER=1.

As a consequence, if the maximum number of DDP iterations is increased to 10, the MPC solution computed at time step $t = 0$ (shown in Figure 3) correctly satisfies the cross-product constraint.
The drawback is a reduction in the achievable control frequency.
Despite the fact that the warmstart issue has been resolved and the DDP solver is now capable of producing feasible solutions that satisfy the constraints (in a limited case of experiments), the experiments reveal a significant sensitivity to the choice of cost weights

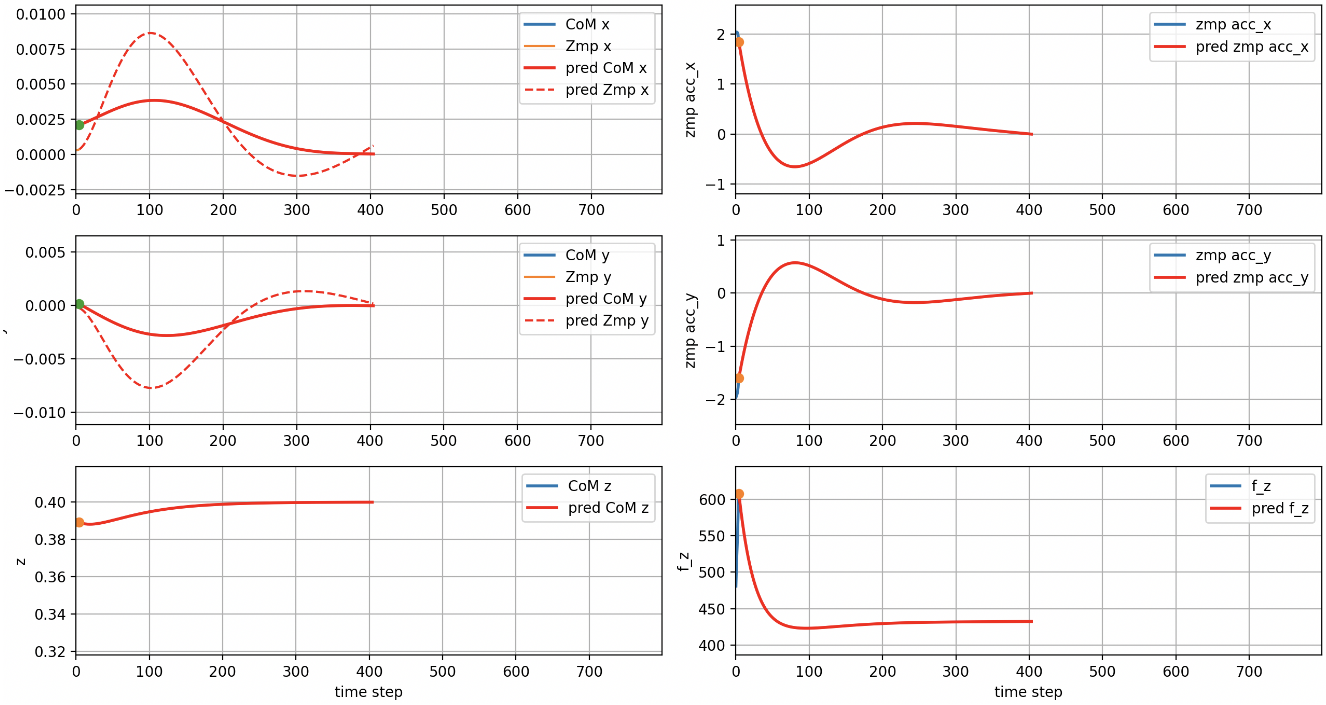

Figure 2: MPC history and predictions at time step t=4 with SOLVER_MAXITER=1.

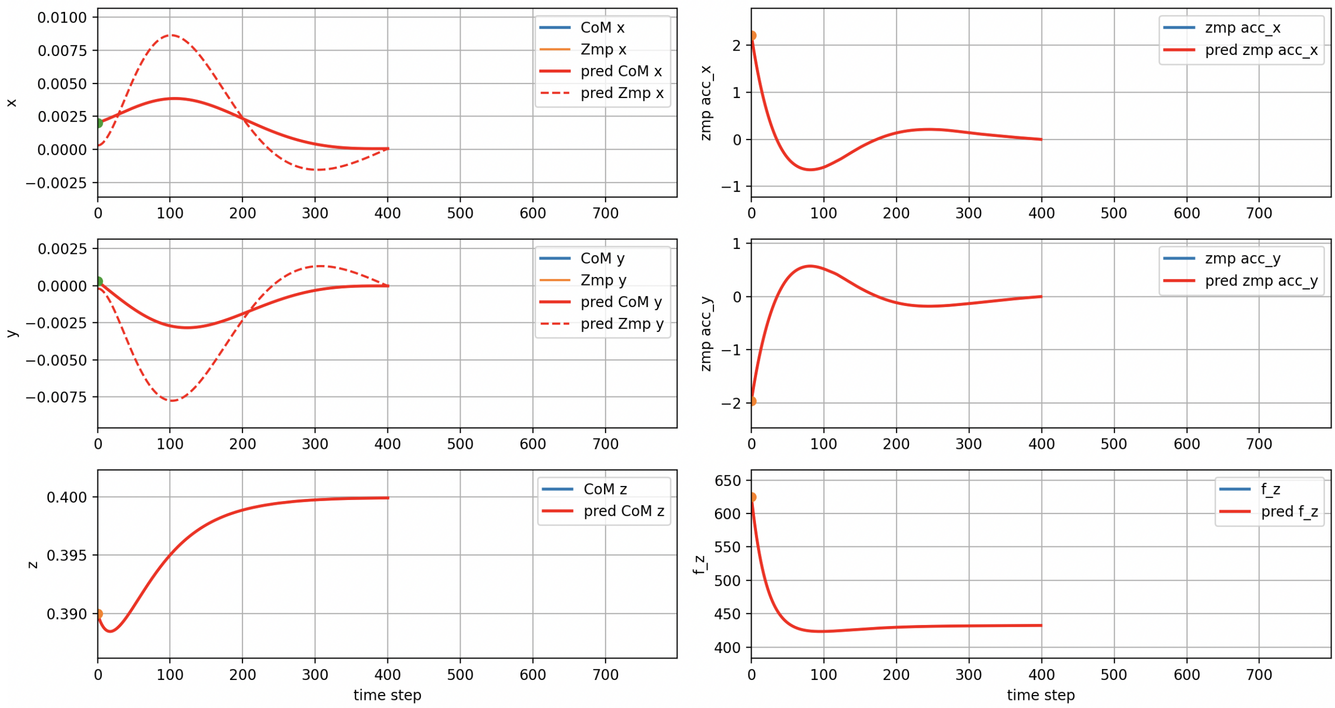

Figure 3: MPC history and predictions at time step $t = 0$ with `SOLVER_MAXITER` $= 10$.

and to the warmstart initialization. This sensitivity is particularly critical in the first MPC iterations, where the solver may fail to find a valid solution if the initial guess is not sufficiently close to the correct solution.

2.2.2 Non holonomic constraint (open loop only)

The sate and inputs: $\mathbf{x} = \begin{pmatrix} x_{com} \\ y_{com} \\ z_{com} \\ \dot{x}_{com} \\ \dot{y}_{com} \\ \dot{z}_{com} \\ x_{zmp} \\ y_{zmp} \\ v \\ \theta \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} a \\ \omega \\ F_{cx} \\ F_{cy} \\ F_{cz} \end{pmatrix}$

The dynamics of the system become non linear:

$$\begin{aligned} \ddot{\mathbf{p}}_{\mathbf{com}} &= 1/m\mathbf{F_c} + \mathbf{g} \\ \dot{x}_{zmp} &= v \cdot cos(\theta) \\ \dot{y}_{zmp} &= v \cdot sin(\theta) \\ \dot{v} &= a \\ \dot{\theta} &= \omega \end{aligned}$$

And the non linear constrained optimization problem is formulated as:

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N} l(x_k, u_k) + l_N(x_N, u_N) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k) \\ & x_0 = x_{current} \\ & h(x, u) = 0 \\ & g(x, u) \leq 0 \\ & h_{ter}(x) = 0 \end{aligned}$$

The running cost is:

$$\begin{aligned} l(x_k, u_k) = {} & w^{x,y}_{\text{com}} \left\| p^{x,y}_{\text{com}} - p^{x,y,\text{ref}}_{\text{com}} \right\|^2 + w^{x,y}_{\text{vcom}} \left\| \dot{p}^{x,y}_{\text{com}} - \dot{p}^{x,y,\text{ref}}_{\text{com}} \right\|^2 \\ & + w_{\text{h}} \left\| z_{\text{com}} - z^{\text{ref}}_{\text{com}} \right\|^2 + w_{\text{vh}} \left\| \dot{z}_{\text{com}} - \dot{z}^{\text{ref}}_{\text{com}} \right\|^2 \\ & + w_z \left\| p^{x,y}_{\text{zmp}} - p^{x,y,\text{ref}}_{\text{zmp}} \right\|^2 \\ & + w_v \left\| v - v^{ref} \right\|^2 \\ & + w_\theta \left\| \theta - \theta^{ref} \right\|^2 \\ & + w_\omega \left\| \omega \right\|^2 \\ & + w_{\text{acc}} \left\| a \right\|^2 \\ & + w_{\text{fxy}} \left\| u_{F_{cx,y}} \right\|^2 \\ & + w_{\text{fz}} \left\| u_{F_{cz}} - mg \right\|^2 \end{aligned}$$

The terminal cost is analogous (without the input).
The equality constraint is:

$$h(x, u) = (p_{zmp} - p_{com}) \times F_c = 0$$

The inequality constraint is:

$$g(x, u) = -F_{cz} \leq 0$$

The terminal constraint is

$$h_{ter}(x) = \begin{pmatrix} x_{com}^N - x_{zmp}^N \\ y_{com}^N - y_{zmp}^N \end{pmatrix} = 0$$

The experiment is still conducted in open loop settings. In this case, the motion of the ZMP is constrained to follow unicycle dynamics, so instantaneous lateral motions are not allowed.
As an example, consider the following experiment:

$$x_{\text{init}} = \begin{bmatrix} 0.0003, \ 0.002, \ 0.39, \ -0.01, \ 0.01, \ -0.1, \ 0.0001, \ 0.0003, \ 0.0, \ 1.7 \end{bmatrix},$$

The references are the constant positions:

$$p_{com}^{ref} = \begin{pmatrix} 0.0, \ 0.0, \ 0.4 \end{pmatrix}^T, \qquad p_{zmp}^{x,y,ref} = \begin{pmatrix} 0.0, \ 0.0 \end{pmatrix}^T$$

The warmstart is the same used in Section 2.2.1.
As illustrated in Figure 4, the corrected DDP formulation is able to handle both the nonlinear dynamics and the nonlinear constraint in some experiments (still the first few iterations are similar to Figure 1). However, in other cases, with different initial conditions or references, the solver fails to find a solution, for reasons likely similar to those discussed in the previous section. Moreover, in this case even in successful runs, the x-component of the COM trajectory does not converge exactly to the reference, exhibiting a steady-state error.

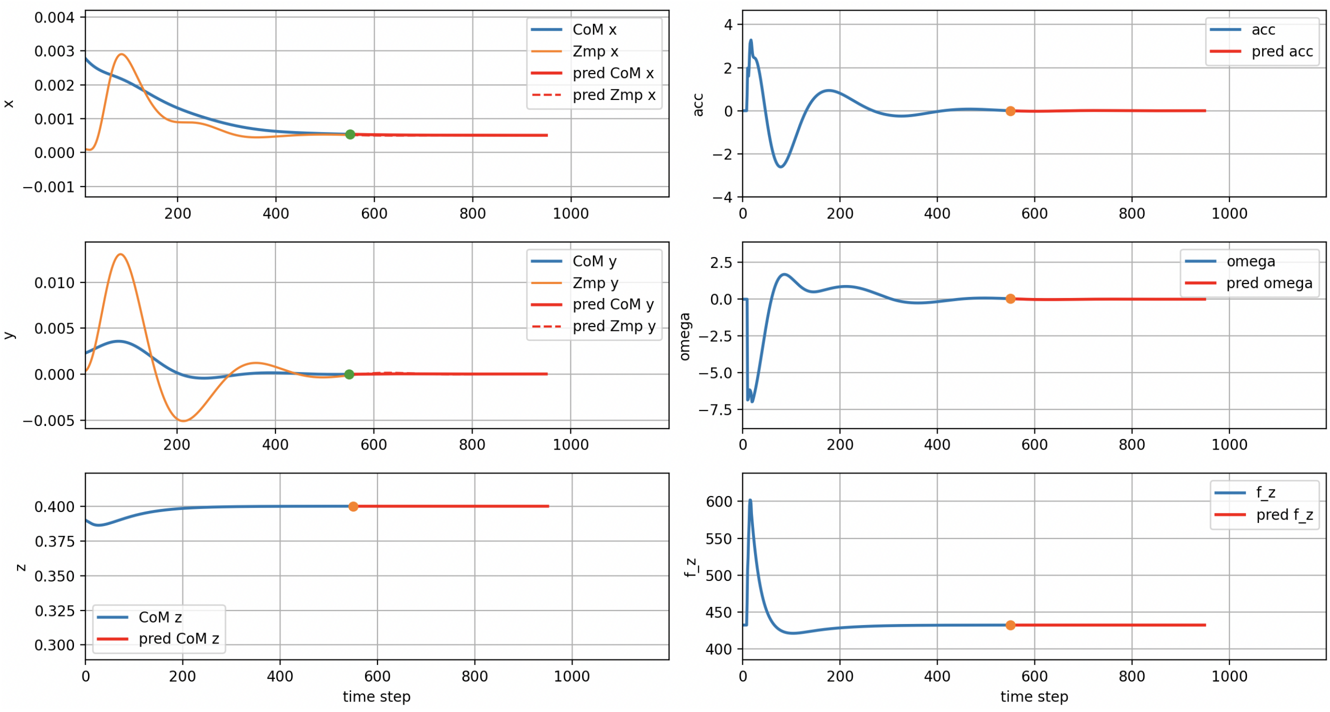

Figure 4: MPC history and predictions at time step $t = 160$.

2.3 Attempt with IPOPT

2.3.1 Momentum balancing equality constraint (open loop only)

I attempted to formulate the same problem described in Section 2.2.1, this time using the IPOPT solver. Although this comes at the cost of a significantly larger average control period ($\approx 24ms$), the resulting optimization proves to be substantially more robust with respect to both the warmstart initialization and the tuning of the cost weights. For instance, IPOPT is able to return a valid solution already at timestep $t = 0$, even when the warmstart is set to $x_{initial\_guess} = 0, u_{initial\_guess} = 0$ over the entire prediction horizon.
Figure 5 illustrates the solution obtained at a generic timestep.
It is also worth noting that IPOPT performs many more iterations than the SOLVER_MAXITER =1 used for DDP.
However, increasing the number of iterations in DDP does not necessarily guarantee convergence: in several cases the solver is still unable to find a feasible solution, whereas IPOPT remains considerably more reliable.

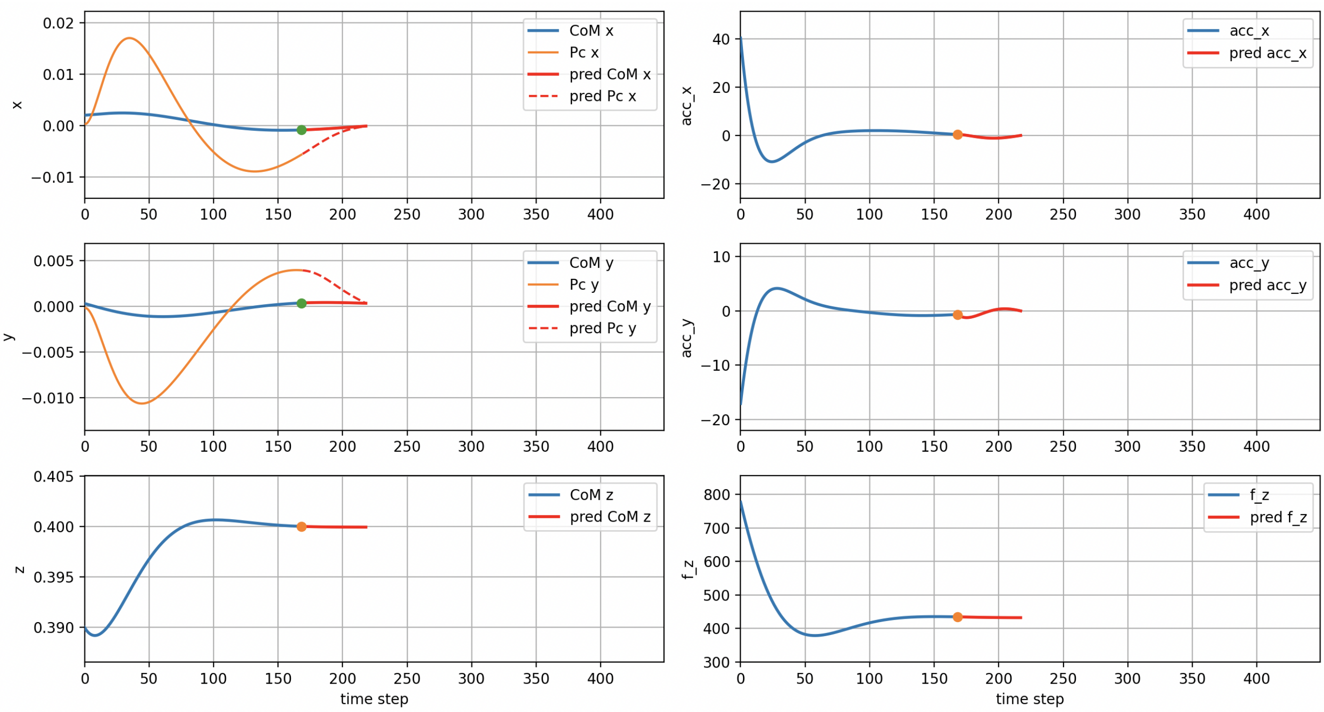

Figure 5: MPC history and predictions at time step $t = 160$.