Dynamics & Control of a 2D Environment

Dhimiter Pikuli

August 13, 2025

1 Acceleration as Control

1.1 Dynamics

- Normal dynamics: $x_{k+1} = x_k + \dot{x}_k T + \frac{1}{2}T^2\ddot{x}_k$ and $\dot{x}_{k+1} = \dot{x}_k + T\ddot{x}_k$

- Implying: $\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} \ddot{x}_k$

- Iteratively: $\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^i \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} = \begin{bmatrix} (2i-1)/2T^2 \\ T \end{bmatrix}$ and $\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^i = \begin{bmatrix} 1 & iT \\ 0 & 1 \end{bmatrix}$

- In the following the time step 0 indicates the current step of environment that the MPC has to solve. All the equations below can be easily generalized for any step t in the episode of the environment.

- For step N: $\begin{bmatrix} x_N \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} 1 & NT \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} + \sum_{k=0}^{N-1} \begin{bmatrix} (2k-1)/2T^2 \\ T \end{bmatrix} \ddot{x}_{N-k}$

- For $X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, \ddot{X} = \begin{bmatrix} \ddot{x}_0 \\ \vdots \\ \ddot{x}_{N-1} \end{bmatrix}$ and $X_0 = \begin{bmatrix} x_0 \\ \vdots \\ x_0 \end{bmatrix}, \dot{X}_0 = \begin{bmatrix} \dot{x}_0 \\ \vdots \\ \dot{x}_0 \end{bmatrix}$:

$$X = X_0 + M_{xv} \odot \dot{X}_0 + M_{xa}\ddot{X}$$

where

$$M_{xa} = \begin{bmatrix} 1/2 & 0 & \dots & 0 \\ 3/2 & 1/2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (2N-1)/2 & (2(N-1)-1)/2 & \dots & 1/2 \end{bmatrix} T^2, M_{xv} = \begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix} T$$

- Similarly for the velocities of the horizon $\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_N \end{bmatrix}$:

$$\dot{X} = \dot{X}_0 + M_{va}\ddot{X}$$

  where

$$M_{va} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} T$$

1.2 Cost Function

- Cost function $\frac{1}{2}||X - X^{ref}||$

$$\begin{aligned}
&\Leftrightarrow \min_{\ddot{X}} \frac{1}{2}\left(X_0 - X^{ref} + M_{xv} \odot \dot{X}_0 + M_{xa}\ddot{X}\right)^\intercal \left(X_0 - X^{ref} + M_{xv} \odot \dot{X}_0 + M_{xa}\ddot{X}\right) \\
&\Leftrightarrow \min_{\ddot{X}} \frac{1}{2}\left(X_0 - X^{ref} + M_{xv} \odot \dot{X}_0\right)^\intercal M_{xa}\ddot{X} + \\
&\qquad \frac{1}{2}\ddot{X}^\intercal M_{xa}^\intercal \left(X_0 - X^{ref} + M_{xv} \odot \dot{X}_0\right) + \frac{1}{2}\left(\ddot{X}^\intercal M_{xa}^\intercal M_{xa}\ddot{X}\right) \\
&\Leftrightarrow \min_{\ddot{X}} \left(X_0 - X^{ref} + M_{xv} \odot \dot{X}_0\right)^\intercal M_{xa}\ddot{X} + \frac{1}{2}\left(\ddot{X}^\intercal M_{xa}^\intercal M_{xa}\ddot{X}\right) \\
&Q = M_{xa}^\intercal M_{xa} \\
&p = \left(X_0 - X^{ref} + M_{xv} \odot \dot{X}_0\right)^\intercal M_{xa}
\end{aligned}$$

- Solve using a QP solver: $\min_{\ddot{X}} \frac{1}{2}\ddot{X}^\intercal Q\ddot{X} + p\ddot{X}$ where $-b \leq \ddot{X} \leq b$

- Extending with a reference velocity $\frac{1}{2}||\dot{X} - \dot{X}^{ref}||$

$$\begin{aligned}
&\Leftrightarrow \min_{\ddot{X}} \frac{1}{2}\left(\dot{X}_0 - \dot{X}^{ref} + M_{va}\ddot{X}\right)^\intercal \left(\dot{X} - \dot{X}^{ref} + M_{xa}\ddot{X}\right) \\
&\Leftrightarrow \min_{\ddot{X}} \frac{1}{2}\left(\dot{X}_0 - \dot{X}^{ref}\right)^\intercal M_{va}\ddot{X} + \frac{1}{2}\ddot{X}^\intercal M_{va}^\intercal \left(\dot{X}_0 - \dot{X}^{ref}\right) + \frac{1}{2}\left(\ddot{X}^\intercal M_{va}^\intercal M_{va}\ddot{X}\right) \\
&\Leftrightarrow \min_{\ddot{X}} \left(\dot{X}_0 - X^{ref}\right)^\intercal M_{va}\ddot{X} + \frac{1}{2}\left(\ddot{X}^\intercal M_{va}^\intercal M_{va}\ddot{X}\right)
\end{aligned}$$

- Q and p become:
  $Q = M_{xa}^\intercal M_{xa} + M_{va}^\intercal M_{va}$
  $p = \left(X_0 - X^{ref} + M_{va} \odot \dot{X}_0\right)^\intercal M_{xa} + \left(\dot{X}_0 - \dot{X}^{ref}\right)^\intercal M_{va}$

1.3 Stability

- In order to ensure stability in the behaviour of the agent it is necessary to try and optimize for velocity and acceleration as well as following the reference plan

$$||X - X^{ref}|| + c||\dot{X}||,$$

with constant C for tuning the importance of each component of the cost.

- Developing the second component of the cost function gives
$\min_{\ddot{X}} \frac{1}{2}\dot{X}^\intercal \dot{X}$
$\Leftrightarrow \min_{\ddot{X}} \frac{1}{2}\left(\dot{X}_0 + M_{va}\ddot{X}\right)^\intercal \left(\dot{X} + M_{va}\ddot{X}\right)$
$\Leftrightarrow \min_{\ddot{X}} \frac{1}{2}\dot{X}_0^\intercal M_{va}\ddot{X} + \frac{1}{2}\ddot{X}^\intercal M_{va}^\intercal \dot{X} + \frac{1}{2}\ddot{X}^\intercal M_{va}^\intercal M_{va}\ddot{X}$
$\Leftrightarrow \min_{\ddot{X}} \dot{X}_0^\intercal M_{va}\ddot{X} + \frac{1}{2}\ddot{X}^\intercal M_{va}^\intercal M_{va}\ddot{X}$
$Q = M_{xa}^\intercal M_{xa} + cM_{va}^\intercal M_{va}$
$p = \left(X_0 - X^{ref} + M_{xv} \odot \dot{X}_0\right)^\intercal M_{xa} + c\dot{X}_0^\intercal M_{va}$

- Similarly if a reference velocity is given together with a reference trajectory the cost function is defined as

$$||X - X^{ref}|| + c||\dot{X} - \dot{X}^{ref}|| + d||\ddot{X}||$$

- Giving the following QP-problem:
$Q = M_{xa}^\intercal M_{xa} + cM_{va}^\intercal M_{va} + dI_{2N}$
$p = \left(X_0 - X^{ref} + M_{va} \odot \dot{X}_0\right)^\intercal M_{xa} + \left(\dot{X}_0 - \dot{X}^{ref}\right)^\intercal M_{va}$

- The values chosen in the implementation are $c = 0.2$ and $d = 0.26$ with reference velocity and $c = 0.25$ for the other cases

1.4 Dynamics in 2D

- Linearizing both dimensions from x to $p = [x, y]^\intercal$

- The horizon prediction for $P = \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ y_1 \\ \vdots \\ y_N \end{bmatrix}$, $\ddot{P} = \begin{bmatrix} \ddot{x}_0 \\ \vdots \\ \ddot{x}_{N-1} \\ \ddot{y}_0 \\ \vdots \\ \ddot{y}_{N-1} \end{bmatrix}$ and $P_0 = \begin{bmatrix} x_0 \\ \vdots \\ x_0 \\ y_0 \\ \vdots \\ y_0 \end{bmatrix}$, $\dot{P}_0 = \begin{bmatrix} \dot{x}_0 \\ \vdots \\ \dot{x}_0 \\ \dot{y}_0 \\ \vdots \\ \dot{y}_0 \end{bmatrix}$:

$$P = P_0 + M_{pv} \odot \dot{P}_0 + M_{pa}\ddot{P}$$

where

$$M_{pa} = \begin{bmatrix} M_{xa} & 0 \\ 0 & M_{ya} \end{bmatrix}, M_{pv} = \begin{bmatrix} M_{xa} \\ M_{yv} \end{bmatrix}$$

and

$$M_{xa} = M_{ya}, M_{xa} = M_{yv}$$

because of symmetry between axis x and y.

1.5 Collision Avoidance

- From (Bohorquez et al., 2016) collision avoidance through

$$||x - m_k|| \geq d$$

  where x is the center of the agent, m_k is the position of member of the crowd k and d is the safety distance between agent and people of the crowd.

- Linearize the constraints by $\hat{u}_k^\intercal(x - m_k) \geq d$ with $\hat{u}_k^\intercal = \dfrac{x - m_k}{||x - m_k||} \in \mathbb{R}^2$

- In MPC (assuming static crowd) we need to ensure $\hat{u}_k^\intercal(x_i - m_k) \geq d \quad \forall i, k$ where i are the steps of the horizon of length N

- Recall the equation for the state in the future horizon:

$$P = P_0 + M_{pv} \odot \dot{P}_0 + M_{pa}\ddot{P}$$

- Write the constraints as:

$$\hat{U}_k \left(P_0 + M_{pv} \odot \dot{P}_0 + M_{pa}\ddot{P} - M_k \right) \geq d$$

  where $\hat{U}_k = \begin{bmatrix} I_N \hat{u}_k^x & | & I_N \hat{u}_k^y \end{bmatrix}$ and $M_k = \begin{bmatrix} m_{k,1}^x \\ \vdots \\ m_{k,N}^x \\ m_{k,1}^y \\ \vdots \\ m_{k,N}^y \end{bmatrix}$

- Rewrite constraint as:

$$\hat{U}_k \left(P_0 + M_{pv} \odot \dot{P}_0 - M_k \right) + \hat{U}_k M_{pa}\ddot{P} \geq d$$

$$\hat{U}_k M_{pa}\ddot{P} \geq d - \hat{U}_k \left(P_0 + M_{pv} \odot \dot{P}_0 - M_k \right)$$

$$-\hat{U}_k M_{pa}\ddot{P} \leq -d + \hat{U}_k \left(-(M_k - P_0) + M_{pv} \odot \dot{P}_0 \right)$$

- The QP problem with collision avoidance constraints would be
$G_k = -\hat{U}_k M_{pa}$
$h_k = -d + \hat{U}_k \left(-(M_k - P_0) + M_{pv} \odot \dot{P}_0\right)$

$$\min_{\ddot{P}} \ddot{P}^{\intercal} Q \ddot{P} + p\ddot{P}$$

s.t

$$-b \leq \ddot{P} \leq b$$

and

$$G_k \ddot{P} \leq h_k \quad \forall k, \quad \dot{P}_N = 0.$$

Where in order to ensure passive safety it is necessary to include recursive feasibility by ensuring an existing braking trajectory with $\dot{P}_N = 0$. Explicitly it is hard coded in the environment that at the beginning of an episode the agent has zero velocity thus it is in a safe state.

- Another important consideration is the fact that MPC relies on a discrete representation of the world with a times-step T. The real environment is continuous and one of the problems that emerges from implementing this in the real world is that inter-step collisions are disregarded. It is possible to amend this by introducing an extra safety margin to d, see Figure 1. Originally, d is equal to the sum of the robot radius and the crowd member radius $|\overline{O_c O_r}| = r_c + r_r$ plus a small constant ϵ. Thus it is possible to find the margin and define a new value of d (another way to think of this is that we need to make h as long as $r_r + r_c$)

$$h^2 = (r_r + r_c)^2 - \left(v_{c,max} T - \frac{v_{c,max} - v_{max}}{2} T\right)^2,$$
$$d := r_r + r_c + \underbrace{r_r + r_c - h}_{\text{margin}} + \epsilon.$$

1.5.1 Static Obstacle Avoidance

- Or MPC method should also be able to constrain the agent position when necessary thus defining regions of the environment that the agent cannot access, for example the walls
- We analyse here first a simple linear constraint in the space which can be computed appropriately in order to divide the plane into two sub-planes, one designated to an obstacles and the other to the agent as viable space to move in
- We represent the constraint with the equation $ax + by + c$ as an inequality, this allows to easily represent functions like $x = c$ as well
- Putting x and y in one vector using $\ddot{P}$ the constraint looks like the following

$$M_{ai} P + C_i < 0 \Leftrightarrow M_{ai} P < -C_i$$

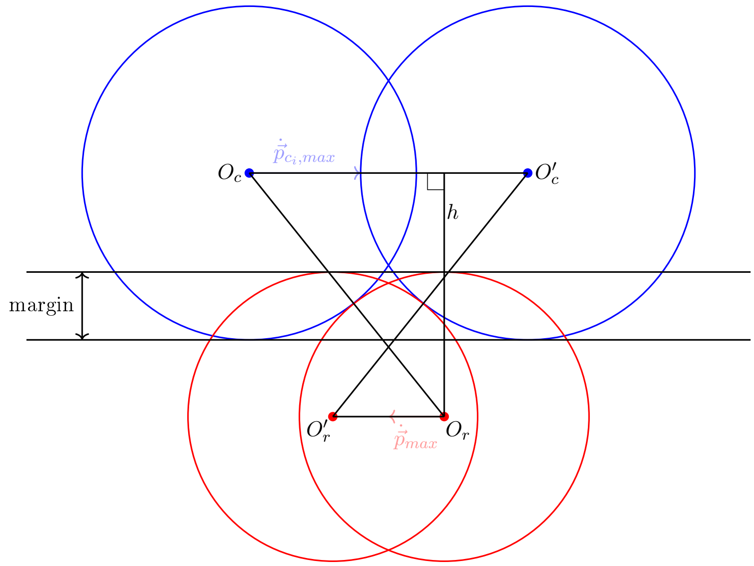

Figure 1: Due to discretization, a collision might be completely overlooked by MPC as shown above for first time-step where the robot is at O_r moving with velocity $\vec{\dot{p}}_{max}$ and crowd member at O_c moving with velocity $\vec{\dot{p}}_{c,max}$. In the second time step the robot and the crowd member are at position O'_r and O'_c respectively. Here we set the minimal distance between the agent and the crowd member which is the sum of their radii in order to compute the maximal margin.

$$\text{where} \quad M_{ai} = \begin{bmatrix} I_N a_i & | & I_N b_i \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} c_i, & \dots & c_i \end{bmatrix}^\intercal$$

where i indexes the constraint.

- Replacing with the definition above

$$M_{ai}(P_0 + M_{pv} \odot \dot{P}_0 + M_{pa}\ddot{P}) < -C_i \Leftrightarrow M_{ai}M_{pa}\ddot{P} < -M_{ai}(P_0 + M_{pv} \odot \dot{P}_0) - C_i$$

- In this case the unit we are working with is distance, so for the equation to have the unit distance a and b have no unit and c is also a distance

1.6 Limiting Acceleration and Velocity

- Defining a safe region inside the non-linear constraint of maximal speed and acceleration

- Points of the octagon are generated starting from point $A = \begin{bmatrix} 1 \\ 0 \end{bmatrix} c$, where constant c relates to the maximal value that the norm can take for the given contraint. The other point are generated by multiplying with a rotation matrix, e.g. for an octagone

$$\begin{bmatrix} \cos \pi/4 & -\sin \pi/4 \\ \sin \pi/4 & \cos \pi/4 \end{bmatrix},$$

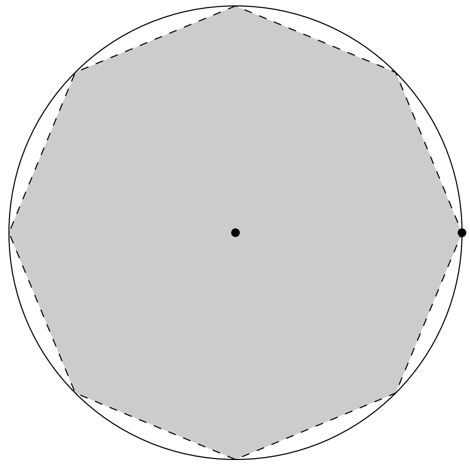

Figure 2: Linearizing constraints means also applying a harsher constraint that fits inside the non-linear constraint.

With the generated points then it is possible to compute the edges (lines) of the polygon for each two given vertices

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{and} \quad b = y_1 - mx_1.$$

For the four edges above $y = 0$ we want that $y < mx + b$ meaning $y - mx - b < 0$ while for the four edges below $y = 0$ we want $y > mx + b$ meaning $y - mx - b > 0$ or even $-y + mx + b < 0$.

- Bring constraint in linear form w.r.t. $\ddot{X}$ and $\ddot{Y}$ for each line of the polygon i

$$M_{ai}\ddot{P} - B_i < 0 \Leftrightarrow M_{ai}\ddot{P} < B_i$$

$$\text{where} \quad M_{ai} = \begin{bmatrix} -I_N m_i & | & I \end{bmatrix} \quad \text{and} \quad B_i = \begin{bmatrix} b_i, & \dots & b_i \end{bmatrix}^\intercal$$

  - In this case the unit is $[m/s^2]$ which means that m does not have a unit and b has unit $[m/s^2]$

- And similarly for velocity

$$M_{ai}(\dot{P}_0 + M_{va}\ddot{P}) - B_i < 0 \Leftrightarrow M_{ai}M_{va}\ddot{P} < B_i - M_{ai}\dot{P}_0$$

  - In this case the unit is $[m/s]$ which means that m does not have a unit and b has unit $[m/s]$

- In both cases the sign for the left and the right side of the inequality has to be changed depending on the line of the polygon

1.7 Tolerance

The prediction from the QP-solver will be only accurate only up to 10^{-8} which can cause issues on the hard constraints. If the predicted position at time $t-1$ for step t and the actual value in the environment after the update are

$$x_{t|t-1} = 0.123456781$$

$$x_t = 0.12345678,$$

it can cause violation of the distance constraints from the crowd. Since this margin of error in the prediction is negligible then it should be ignored. In *python* we use the *qpsolvers* package which offers an interface for many QP-solvers and we choose *clarabel* which has the highest accuracy. In order to combat the numerical issues described above it is possible to set tolerance values for the *clarabel* algorithm to the scale of 10^{-5} which ignores discrepancies of less then 10^{-5}. Practically this value is in a range smaller then the millimeter which seems neglectable in the natural distance we as humans walk around which would be at a minimum with an accuracy around the range of the decimeter.

1.8 Cascading Safety

In an attempt to decouple efficiency from safety it is proposed to implement a cascading stream of safety breaking trajectories instead of ensuring safety simply from the current start position of the agent. The equation representing this objective is as follows

$$\min_{\ddot{P}} ||P - P^{ref}|| \quad \text{where} \quad P = \begin{bmatrix} p_1 & \dots & p_M \end{bmatrix}^\intercal \quad \text{and} \quad p_i = b_1^i, \ddot{p}_i = \ddot{b}_1^i \quad \forall i \in \{1, \dots, M\}$$

$$\begin{aligned} \text{s.t.} \quad \dot{b}_N^i = 0 \quad \forall i \in \{1, \dots, M\} \\ G_k \ddot{B} \leq h_k \quad \forall k \in \{1, \dots, K\} \\ W\ddot{B} \leq C_j^a \quad \text{and} \quad Z\ddot{B} \leq C_j^v(\dot{P}_0) \quad \forall j \in \{1, \dots, C\}. \end{aligned}$$

In this formula M represents the horizon to plan for, N the horizon to ensure safety for and the executed control $\ddot{P}$ is made up of the first steps of the braking trajectories of each time-step of the horizon $\ddot{b}_1^i$. The first part of the constraint ensures passive safety as the agent should always have the possibility to break before anything undesired happens (represented by the constraints). The second and the third line are collision avoidance and bounding of acceleration and velocity as described in previous sections. In the definition K represents the number of the member in the crowd and C the number of linear units used to approximate the non-linear constraints on acceleration and velocity. This is shown visually in the following:

1.8.1 Changes in Dynamics

The target control computed will be linearized in the form of

$$\ddot{B} := \begin{bmatrix} \ddot{b}_0^0 & \dots & \ddot{b}_{N-1}^0 & \ddot{b}_0^1 & \dots & \ddot{b}_{N-1}^1 & \dots & \ddot{b}_0^{M-1} & \dots & \ddot{b}_{N-1}^{M-1} \end{bmatrix}^\intercal,$$

For simplicity we first look into a single dimension representation of the dynamics and then extend to two dimensions the same way as in 1.4. First we have a look at some examples of

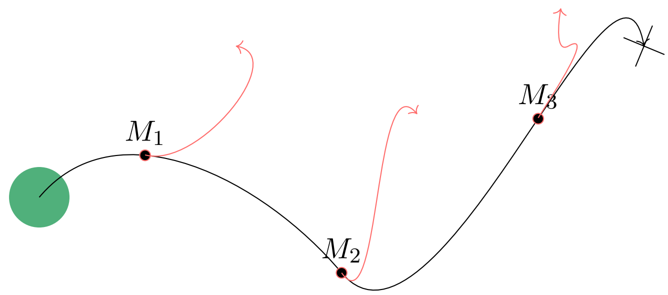

Figure 3: For the computed trajectory (black line) P at each time-stamp a braking trajectory B^i is computed. In the image only three point are sampled from $\{1, \ldots M\}$.

single step dynamics for position update equation using acceleration as control and $\{b_0, \dot{b}_0\}$ to represent the initial state of the agent

$$
\begin{aligned}
&b_1^1 = b_0 + \dot{b}_0 T + \frac{1}{2} T^2 \ddot{b}_0^0 \\
&b_2^1 = b_1^1 + \dot{b}_1^1 T + \frac{1}{2} T^2 \ddot{b}_1^0 = b_0 + 2\dot{b}_0 T + \frac{3}{2} T^2 \ddot{b}_0^0 + \frac{1}{2} T^2 \ddot{b}_1^0 \quad \text{for} \quad \dot{b}_1^1 = \dot{b}_0 + \ddot{b}_0^0 T \\
&b_1^2 = b_1^1 + \dot{b}_1^1 T + \frac{1}{2} T^2 \ddot{b}_0^1 = b_0 + 2\dot{b}_0 T + \frac{3}{2} T^2 \ddot{b}_0^0 + \frac{1}{2} T^2 \ddot{b}_0^1 \\
&b_2^2 = b_1^2 + \dot{b}_1^2 T + \frac{1}{2} T^2 \ddot{b}_1^1 = b_0 + 3\dot{b}_0 T + \frac{5}{2} T^2 \ddot{b}_0^0 + \frac{3}{2} T^2 \ddot{b}_0^1 + \frac{1}{2} T^2 \ddot{b}_1^1 \\
&\qquad\qquad \text{for} \quad \dot{b}_1^2 = \dot{b}_1^1 + \ddot{b}_1^1 T = \dot{b}_0 + \ddot{b}_0^0 T + \ddot{b}_1^1 T.
\end{aligned}
$$

The difference between b_2^1 and b_1^2 is in the control value which are $\ddot{b}_1^0$ and $\ddot{b}_0^1$ respectively. Logically, b_2^1 represents the second step of the braking trajectory at the current time instance whereas b_1^2 represents the second control step of the current time instance. In the objective of the MPC only b_1^2 and all other control inputs of the form b_1^i with $\forall i \in \{1, \ldots, M\}$ will be relevant.

Inside braking trajectories the dynamics do not change however the initial step b_1^i will be relative to all other previous steps b_1^j where $j < i$. At the same time the dynamics of steps b_1^i between themselves do not change from the original formulation. This dependency can be represented in the following manner

$$
B = b_0 + M_{b\dot{b}} \dot{b}_0 + M_{b\ddot{b}} \ddot{B}
$$

$$\text{where} \quad M_{b\dot{b}} = \begin{bmatrix} 1 \\ \vdots \\ N \\ 2 \\ \vdots \\ N+1 \\ 3 \\ \vdots \\ M \\ \vdots \\ M+N \end{bmatrix} T, \quad M_{b\ddot{b}} = \begin{pmatrix} M_{xa} & 0 & 0 & \dots & 0 \\ F(3/2) & M_{xa} & 0 & \dots & 0 \\ F(5/2) & F(3/2) & M_{xa} & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ F((2M-1)/2) & F((2M-3)/2) & \dots & F(3/2) & M_{xa} \end{pmatrix}$$

$$\text{and} \quad M_{xa} = \begin{bmatrix} 1/2 & 0 & \dots & 0 \\ 3/2 & 1/2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (2N-1)/2 & (2N-3)/2 & \dots & 1/2 \end{bmatrix} T^2, \quad F(n) = \begin{bmatrix} n & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ n & 0 & \dots & 0 \end{bmatrix} T^2.$$

The matrices above have dimensions $B \in \mathbb{R}^{MN}$, $M_{xa} \in \mathbb{R}^{N \times N}$, $F(n) \in \mathbb{R}^{N \times N}$, $M_{b\dot{b}} \in \mathbb{R}^{MN}$ and $M_{b\ddot{b}} \in \mathbb{R}^{MN \times MN}$. Similarly the dynamics for velocity are

$$\dot{B} = \dot{b}_0 + M_{\dot{b}\ddot{b}}\ddot{B}$$

$$\text{where} \quad M_{\dot{b}\ddot{b}} \begin{pmatrix} M_{va} & 0 & \dots & 0 \\ F(1/T) & M_{va} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ F(1/T) & \dots & F(1/T) & M_{va} \end{pmatrix}$$

$$\text{and} \quad M_{va} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 \end{bmatrix} T$$

1.8.2 Objective & Constraints

For the objective it is necessary to filter out all irrelevant dimensions to the reference plan P^{ref}. This is possible by computing a dot-product between a filter F_P and the vectors and a transposed dot-product between F_P and the matrices. The derivation looks like the following for the cost function $||P - P^{ref}||$
$\min_{\ddot{P}} ||P - P^{ref}|| = \min_{\ddot{B}} ||F_P \odot B - B^{ref}||$

$$\text{where in} \quad B^{ref} \quad b^{ref}_{(i-1)N+1} = p^{ref}_i \quad \text{for} \quad i \in \{1, \dots M\}$$

$$\text{and} \quad b^{ref}_j = 0 \quad \text{for} \quad j \in \{1, \dots, NM\}/\{1, N+1, 2N+1 \dots, N(M-1)+1\}$$

$$\text{where in} \quad F_P \quad f_{P,(i-1)N+1} = 1 \quad \text{for} \quad i \in \{1, \dots M\}$$
$$\text{and} \quad f_{P,j} = 0 \quad \text{for} \quad j \in \{1, \dots, NM\}/\{1, N+1, 2N+1 \dots, N(M-1)+1\}$$

Similarly as computed originally in 1.2 the minimization problem can be rewritten as
$\Rightarrow \min_{\ddot{B}} \left(F_P \odot B_0 - B^{ref} + F_P \odot M_{b\dot{b}} \odot \dot{B}_0\right)^{\intercal} F_P \odot^{\intercal} M_{b\ddot{b}} \ddot{B} + \left(\ddot{B}^{\intercal}(F_P \odot^{\intercal} M_{b\ddot{b}})^{\intercal}(F_P \odot^{\intercal} M_{b\ddot{b}})\ddot{B}\right)$
$Q = (F_P \odot^{\intercal} M_{b\ddot{b}})^{\intercal}(F_P \odot^{\intercal} M_{b\ddot{b}})$
$p = \left(F_P \odot B_0 - B^{ref} + F_P \odot M_{b\dot{b}} \odot \dot{B}_0\right)^{\intercal} F_P \odot^{\intercal} M_{b\ddot{b}}$
For the passive safety constraint we have M terminal velocities that should be zero whereas all other constraints (collision avoidance and acceleration-velocity limitation) remain the same and are applied to all steps. For passive safety we make use of another filter F_B

$$\text{where} \quad f_{B,i} = 1 \quad \text{for} \quad i \in \{1, N+1, 2N+1 \dots N(M-1)+1\}$$
$$\text{and} \quad f_{B,j} = 0 \quad \text{for} \quad j \in \{1, \dots, NM\}/\{1, N+1, 2N+1 \dots, N(M-1)+1\}$$

We can then write the constraint as

$$F_B \odot \dot{B} = 0 \Leftrightarrow F_B \odot \dot{B}_0 + F_B \odot^{\intercal} M_{\dot{b}\ddot{b}} \ddot{B} = 0.$$

1.8.3 Cascading Safety in 2D

In 2D we write position and control by appending to the first dimension the second dimension

$$B := \begin{bmatrix} bx_1^1 \\ \vdots \\ bx_N^1 \\ bx_1^2 \\ \vdots \\ bx_N^M \\ by_1^1 \\ \vdots \\ by_N^1 \\ by_1^2 \\ \vdots \\ by_N^M \end{bmatrix}, \ddot{B} := \begin{bmatrix} \ddot{bx}_0^0 \\ \vdots \\ \ddot{bx}_{N-1}^0 \\ \ddot{bx}_0^1 \\ \vdots \\ \ddot{bx}_{N-1}^{M-1} \\ \ddot{by}_0^0 \\ \vdots \\ \ddot{by}_{N-1}^0 \\ \ddot{by}_0^1 \\ \vdots \\ \ddot{by}_{N-1}^{M-1} \end{bmatrix}, B^{ref} := \begin{bmatrix} px_1^{ref} \\ 0 \\ \vdots \\ px_2^{ref} \\ 0 \\ \vdots \\ px_M^{ref} \\ 0 \\ \vdots \\ py_1^{ref} \\ 0 \\ \vdots \\ py_2^{ref} \\ 0 \\ \vdots \\ py_M^{ref} \\ 0 \\ \vdots \end{bmatrix}.$$

The matrices defined above need to be simply updated which we do by using the programming variable assignment operator = and the previous definitions

$$M_{b\dot{b}} = \begin{bmatrix} M_{b\dot{b}} \\ M_{b\dot{b}} \end{bmatrix}, \quad M_{b\ddot{b}} = \begin{bmatrix} M_{b\ddot{b}} & 0 \\ 0 & M_{b\ddot{b}} \end{bmatrix}, \quad M_{\dot{b}\ddot{b}} = \begin{bmatrix} M_{\dot{b}\ddot{b}} & 0 \\ 0 & M_{\dot{b}\ddot{b}} \end{bmatrix}, \quad F_P = \begin{bmatrix} F_P \\ F_P \end{bmatrix}, \quad F_B = \begin{bmatrix} F_B \\ F_B \end{bmatrix}.$$

2 Velocity as Control

In order to simplify the formulation and avoid numerical issues during optimization, it is possible to use directly velocity as control. We follow the same steps as described above. The intuition is given by the existing equation that puts the acceleration (our current control value) with the velocity:

$$\dot{x}_{k+1} = \dot{x}_k + T\ddot{x}_k \Leftrightarrow \ddot{x}_k = \frac{\dot{x}_{k+1} - \dot{x}_k}{T}$$

Based on this dependency it is possible to replace the acceleration control with this value and redefine all the dynamics and constraints formulated above with equal counterparts dependent only on velocity. We use the same convention as above for naming matrices, which should be considered as separate parameters for velocity control.

2.1 Dynamics

- Transition matrix representation: $\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} \frac{\dot{x}_{k+1} - \dot{x}_k}{T} \Leftrightarrow$

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2}T \\ 1 \end{bmatrix} \dot{x}_{k+1} - \begin{bmatrix} \frac{1}{2}T \\ 1 \end{bmatrix} \dot{x}_k = \begin{bmatrix} 1 & \frac{1}{2}T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2}T \\ 1 \end{bmatrix} \dot{x}_{k+1}$$

- Iteratively: $\begin{bmatrix} 1 & \frac{1}{2}T \\ 0 & 0 \end{bmatrix}^i \begin{bmatrix} \frac{1}{2}T \\ 1 \end{bmatrix} = \begin{bmatrix} T \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & \frac{1}{2}T \\ 0 & 0 \end{bmatrix}^i = \begin{bmatrix} 1 & \frac{1}{2}T \\ 0 & 0 \end{bmatrix}$

- For step N: $\begin{bmatrix} x_N \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2}T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} + \sum_{k=1}^{N-1} \begin{bmatrix} T \\ 0 \end{bmatrix} \dot{x}_{N-k} + \begin{bmatrix} \frac{1}{2}T \\ 1 \end{bmatrix} \dot{x}_N$

- For the vector representation $X = x_0 + \frac{1}{2}T\dot{x}_0 + M_{xv}\dot{X}$ where

$$M_{xv} = \begin{bmatrix} 1/2 & 0 & \dots & 0 \\ 1 & 1/2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 1 & \dots & 1 & 1/2 \end{bmatrix} T$$

- Velocity is trivial: $\dot{X} = \dot{X}$

2.2 Cost Function

- Cost function $\frac{1}{2}||X - X^{ref}||$

$$\Leftrightarrow \min_{\dot{X}} \frac{1}{2} \left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0 + M_{xv}\dot{X} \right)^\intercal \left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0 + M_{xv}\dot{X} \right)$$

$\Leftrightarrow \min_{\dot{X}} \frac{1}{2}\left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0\right)^\intercal M_{xv}\dot{X} + \frac{1}{2}\dot{X}^\intercal M_{xv}\left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0\right) + \frac{1}{2}\dot{X}^\intercal M_{xv}^\intercal M_{xv}\dot{X}$
$\Leftrightarrow \min_{\dot{X}} \left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0\right)^\intercal M_{xv}\dot{X} + \frac{1}{2}\dot{X}^\intercal M_{xv}^\intercal M_{xv}\dot{X}$
$Q = M_{xv}^\intercal M_{xv}$
$p = \left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0\right)^\intercal M_{xv}$

- Solve using a QP solver: $\min_{\dot{X}} \frac{1}{2}\dot{X}^\intercal Q\dot{X} + p\dot{X}$ where $-b \leq \dot{X} \leq b$
- For stability we can also add a term to try and minimize the amount of action applied to the environment, practically this minimizes swinging motions and enables more consistent breaking. It is the same as the addition of $||\dot{X}||$ and $||\ddot{X}||$ when using acceleration as control. In the following we show how much easier it is to apply minimize velocity control, but it would be possible to try and minimize acceleration as well given the original change in variables from $\ddot{x}_k$ to $\dot{x}_{k+1}$.
$\min_{\dot{X}} \frac{1}{2}(||X - X^{ref}|| + c||\dot{X}||)$
$\Leftrightarrow \min_{\dot{X}} \left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0\right)^\intercal M_{xv}\dot{X} + \frac{1}{2}\dot{X}^\intercal(M_{xv}^\intercal M_{xv} + cI)\dot{X}$
$Q = M_{xv}^\intercal M_{xv} + cI$
$p = \left(x_0 - X^{ref} + \frac{1}{2}T\dot{x}_0\right)^\intercal M_{xv}$
- Extending with a reference velocity $||\dot{X} - \dot{X}^{ref}||$
$\Leftrightarrow \min_{\dot{X}} \frac{1}{2}\left(\dot{X} - \dot{X}^{ref}\right)^\intercal\left(\dot{X} - \dot{X}^{ref}\right)$
$\Leftrightarrow \min_{\dot{X}} \frac{1}{2}\dot{X}^\intercal\dot{X} - \dot{X}^{ref\intercal}\dot{X}$
- Q and p become: $Q = I$ and $p = -\dot{X}^{ref\intercal}$

2.3 Dynamics in 2D

- We define the position and velocity horizon vectors as

$$P = \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ y_1 \\ \vdots \\ y_N \end{bmatrix}, \dot{P} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_N \\ \dot{y}_1 \\ \vdots \\ \dot{y}_N \end{bmatrix} \text{ and } P_0 = \begin{bmatrix} x_0 \\ \vdots \\ x_0 \\ y_0 \\ \vdots \\ y_0 \end{bmatrix}, \dot{P}_0 = \begin{bmatrix} \dot{x}_0 \\ \vdots \\ \dot{x}_0 \\ \dot{y}_0 \\ \vdots \\ \dot{y}_0 \end{bmatrix}$$

- Analogously to the case where the acceleration is used for control:

$$P = P_0 + \frac{1}{2}T\dot{P}_0 + M_{pv}\dot{P} \quad \text{with} \quad M_{pv} = \begin{bmatrix} M_{xv} & 0 \\ 0 & M_{yv} \end{bmatrix} \quad \text{where} \quad M_{xv} = M_{yv}$$

- No difference for velocity $\dot{P} = \dot{P}$

2.4 Collision Avoidance & Passive Safety

- Constraint:
$$\hat{U}_k \left(P_0 + \frac{1}{2} T \dot{P}_0 + M_{pv} \dot{P} - M_k \right) \geq d$$
where $\hat{U}_k = \begin{bmatrix} I_N \hat{u}_k^x & | & I_N \hat{u}_k^y \end{bmatrix}$ and $M_k = \begin{bmatrix} m_{k,1}^x \\ \vdots \\ m_{k,N}^x \\ m_{k,1}^y \\ \vdots \\ m_{k,N}^y \end{bmatrix}$

- Derive:
$$-\hat{U}_k M_{pv} \dot{P} \leq -d + \hat{U}_k \left(-(M_k - P_0) + \frac{1}{2} T \dot{P}_0 \right)$$

- QP problem:
$G_k = -\hat{U}_k M_{pv}$
$h_k = -d + \hat{U}_k \left(-(M_k - P_0) + \frac{1}{2} T \dot{P}_0 \right)$
$$\min_{\dot{P}} \dot{P}^\intercal Q \dot{P} + p \dot{P}$$
s.t
$$G_k \dot{P} \leq h_k \quad \forall k, \quad \dot{P}_N = 0.$$
Where we assume again that a stationary state is a passive safe state. Practically, this can be implemented by adding the constraint
$$\begin{bmatrix} 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix} \dot{P} = 0.$$
However it is possible to also hard set the last control of the horizon to be zero and thus reducing the dimension of the problem by one
$$\dot{P} = \begin{bmatrix} x_1, & \dots & x_{N-1}, & y_1, & \dots & y_{N-1} \end{bmatrix}^\intercal \quad \text{and} \quad x_N = y_N = 0$$
In this case the dynamics matrix are
$$M_{xv} = \begin{bmatrix} 1/2 & 0 & \dots & 0 \\ 1 & 1/2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1/2 \\ 1 & \dots & \dots & 1 \end{bmatrix} T \in \mathbb{R}^{N \times N-1}, \quad \text{and} \quad M_{pv} = \begin{bmatrix} M_{xa} & 0 \\ 0 & M_{ya} \end{bmatrix} \in \mathbb{R}^{2N \times 2(N-1)}$$
We use this convention in further formulas.

2.5 Static Obstacle Avoidance

- Same as in 1.5.1 we limit the regions of the agent using a linear constraint based on the function $ax + by + c$
- Resulting in:

$$M_{ai}P < -C_i \Leftrightarrow M_{ai}(P_0 + \frac{1}{2}T\dot{P}_0 + M_{pv}\dot{P}) < -C_i \Leftrightarrow M_{ai}M_{pv}\dot{P} < -M_{ai}(P_0 + \frac{1}{2}T\dot{P}_0) - C_i$$

- With again the same units as discussed in 1.5.1

2.6 Limiting Control & Stability

- Using the notation defined in 1.6 the velocity constraint looks like the following

$$M_{ai}\dot{P} - B_i < 0 \Leftrightarrow M_{ai}\dot{P} < B_i$$

$$\text{where} \quad M_{ai} = \begin{bmatrix} -I_{N-1}m_i & | & I_{N-1} \end{bmatrix} \quad \text{and} \quad B_i = \begin{bmatrix} b_i, & \dots & b_i \end{bmatrix}^\intercal$$

where m_i and b_i represent the constants defining the linear constraints.

- In order to apply more natural and smooth control it is possible to adapt the acceleration control from earlier ($\dot{P} = \dot{P}_{1:N-1}$)

$$M_{ai}\ddot{P} - B_i < 0 \Leftrightarrow M_{ai}\ddot{P} < B_i \Leftrightarrow M_{ai}\frac{\dot{P}_{1:N} - \dot{P}_{0:N-1}}{T} < B_i$$

$$\Leftrightarrow M_{ai}\begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}\dot{P} < B_i + \frac{M_{ai}}{T}\begin{bmatrix} \dot{x}_0, & 0, & \dots & \dot{y}_0, & 0, & \dots \end{bmatrix}^\intercal$$

$$\text{where} \quad D = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -1 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 0 & -1 \end{bmatrix}\frac{1}{T}.$$

The definition of $\dot{P}$ was changed to just include the first $N-1$ steps and hard set $\dot{P}_N = 0$ which makes it disappear in the derivation above. Moreover, $\dot{P}_0$ present in $\dot{P}_{0:N-1}$ is separated and written on the right-side as a vector $\begin{bmatrix} x_0, & 0, & \dots & y_0, & 0, & \dots \end{bmatrix}^\intercal$.

2.7 Cascading Safety

- Following the same pattern as explained in the previous section on cascading safety the dynamics look like the following for 1D

$$B = b_0 + \frac{1}{2}T\dot{b}_0 + M_{b\dot{b}}\dot{B}$$

$$\text{where}\qquad M_{b\dot{b}} = \begin{pmatrix} M_{xv} & 0 & \dots & 0 \\ F & M_{xv} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ F & \dots & F & M_{xv} \end{pmatrix} \in \mathbb{R}^{MN \times M(N-1)}$$

$$\text{and}\quad M_{xv} = \begin{bmatrix} 1/2 & 0 & \dots & 0 \\ 1 & 1/2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1/2 \\ 1 & \dots & \dots & 1 \end{bmatrix} T \in \mathbb{R}^{N \times N-1}, \quad F = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & \vdots & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{bmatrix} T \in \mathbb{R}^{N \times N-1}.$$

- Objective & Constraints:
  For the objective which relates to following a reference trajectory we only need to account for the relevant dimensions of B which are the first indexes of each braking trajectory. It is possible to transform B into X (as introduced earlier) with the matrix F_X

  $$F_X = \begin{bmatrix} f & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & f & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & f \end{bmatrix} \in \mathbb{R}^{M \times NM}, \quad \text{where} \quad f = \begin{bmatrix} 1, & 0, & \dots & 0 \end{bmatrix}$$

  and $\mathbf{0}$ having appropriate dimensionality $1 \times N$.
  Thus we can write our objective as

  $$\frac{1}{2} \min_{\dot{B}} ||F_P B - X^{ref}|| \Leftrightarrow \frac{1}{2} \min_{\dot{X}} ||X - X^{ref}||$$

  and not change anything from the previously defined equations.

  - The derived Q and p terms are:
    $Q = (F_X M_{b\dot{b}})^\intercal (F_X M_{b\dot{b}}) = M_{xv}^\intercal M_{xv}$
    $p = \left(b_0 - X^{ref} + \frac{1}{2} T \dot{b}_0\right)^\intercal F_X M_{b\dot{b}} = \left(x_0 - X^{ref} + \frac{1}{2} T \dot{x}_0\right)^\intercal M_{xv}$
    where $b_0 = x_0$ and $\dot{b}_0 = \dot{x}_0$
  - Passive safety is again ensured explicitly by hard setting the N-th index of each control vector to 0 which is the reason why the dynamics matrices declared above have dimensions $MN \times M(N-1)$ and $N \times N - 1$.
  - For 2D, matrices need to be stacked as shown previously.
  - Collision avoidance should be applied to all states equally which means there are no additional consideration to be made compared to the normal MPC. However, cascading needs to be taken in to account which means we need the crowd prediction for the next $M + N$ steps in a cascading manner for the while $M * N$ positions. The positions relative to the control vector $\dot{B}$ are

    $$\begin{aligned} [m^x_{k,1}, \ \dots \ m^x_{k,N}, \ m^x_{k,2}, \ \dots \ m^x_{k,N+1}, \ \dots \ m^x_{k,M-N}, \ \dots \ m^x_{k,M+N}, \\ m^y_{k,1}, \ \dots \ m^y_{k,N}, \ m^y_{k,2}, \ \dots \ m^y_{k,N+1}, \ \dots \ m^y_{k,M-N}, \ \dots \ m^y_{k,M+N}]^\intercal. \end{aligned}$$

2.8 Discussion

- In all derivations above the linear term inside the QP usually turns out to be something like $A^\intercal M\ddot{X} + \ddot{X}^\intercal M^\intercal A$ and then it is simplified to just $A^\intercal M\ddot{X}$. Is this a problem since the two components are not equal ($A^\intercal M\ddot{X} \neq \ddot{X}^\intercal M^\intercal A$)?
  These components are scalar values because we are minimizing a metric and thus also equal!

References

N. Bohorquez, A. Sherikov, D. Dimitrov, and P.-B. Wieber. Safe navigation strategies for a biped robot walking in a crowd. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 379–386, Cancun, Mexico, Nov. 2016. IEEE. ISBN 978-1-5090-4718-5. doi: 10.1109/HUMANOIDS.2016.7803304. URL `https://ieeexplore.ieee.org/document/7803304/`.