Final Project
Quadruped Gait Genration using IS-MPC

Course of Autonomous and Mobile Robotics

Master in Control Engineering
Department of Computer, Control and Management Engineering
La Sapienza, Rome, Italy

S. Sabatucci (M1745198), L. Salustri (M1762556), F. Scotti (M1758391)

Academic Year 2020/2021

Contents

1 Introduction

This report is focused on extending the work presented by N. Scianca [1] , in order to achieve a quadruped gait generation for walking and trotting. The report is organized as follow. In Section II the reason why we have chosen to extend the work previously mentioned instead of following a more natural problem formulation is explained. Section III provides a description for the considered quadruped and reference. Section IV describes a "Trotting" gait, while Section V is focused on the "Walking" one. Section VI describes the overall behaviour of quadruped gait generation in the IS-MPC framework considering. In Section VII the QP for the foot replacement, both in a trotting and walking scenarios, is investigated. Sections VIII-IX present the simulation performed in MATLAB and DART, while at the end in Section X there are the conclusions.

2 Approach to the problem

In the recent years, with the developement of computational power, the MPC has become very popular even in the robotics field. In fact it is now possible to address even nonlinear optimization (NL-MPC) in real time. Even though the model of a quadruped is intrinsically nonlinear, in many works such as [4] the use a simplified model like the inverted pendulum as preview model for the control algorithm is preferred. In general the objective function of the optimization problem involves the square of the CoM jerk to have boundedness of the CoM velocities and the minimization of the error in the footstep location with respect to a footstep sequence defined a priori or by some other module of the control architecture, often based on vision and perception. The minimization of the jerk, however, does not guarantee the stability of the CoM, this is why in our work, as we will explain in section VI, we are using the Intrinsically Stable MPC (IS-MPC). Moreover, in the previously mentioned work by Shi et al. it is clear how the interest in extending the work made on bipedal locomotion to quadrupedal locomotion is vivid, since there is a wide range of references in literature about the gait generation for biped, and this is another aspect that we have taken into consideration for the definition of our approach, which is based on a biped gait, as we will explain later. Another approach is the one used in [6], where the authors decided to tackle the entire problem solving different optimization problems at the several stages of the control architecture. The first module of the architecture is the MPC which uses the linear inverted pendulum (LIP) model to describe as usual the CoM dynamics, and solves a QP problem defining the ZMP trajectories and consequentely the footstep placements. Through interpolation the trajectory for the robot base and for the feet are generated and tracked using an LQR approach, which gives as output directly the torques for the joints. One of the main problem of defining a gait for a quadruped, which is not taken into account in the aforementioned approach is the dynamic balance so, usually, as we can see in works like [5], the idea is to use the MPC for the gait generation guaranteeing a dynamically stable condition of the robot enforcing the so called ZMP constraints: the zero moment point is always kept inside the support polygon. Obviously its shape will change depending on the type of gait chosen for the quadruped and this translate with an increase of number for ZMP nonlinear contraints. For example if we consider

a trotting gait, we will have two phases: the quad support phase, in which the shape of the polygon is a trapezoid, and the double support phase, during which the other legs will swing to reach the next footstep positions.
The ZMP trajectory is used to generate the CoM trajectory, that will be tracked by the inverse kinematic module, since the robot can be commanded in velocity. For this reason the ZMP velocities together with the footstep positions are actually the decision variables obtained from the MPC module. At this point it is clear that the constraints for the dynamic balance of the ZMP are not defined by "box" constraints if we don't make some simplifying assumption: to define the admissible region we need to bound the ZMP position using lines with different angular coefficients, and this results in nonlinear constraints in the decision variables. Choosing for example a control horizon $C = 160$, a prediction horizon $P = 320$ and a number of predicted footstep $F = 3$ we would have a total of $2C + 2FN = 332$ decision variables within a nonlinear quadratic programming, where N is the number of legs that are moving depending on the chosen gait ($N = 2$ for the trotting case). This approach will become very computationally intensive, both due to the high number of variables and the nonlinearity of the constraints. This is why we have considered a different idea (scheme in Figure 1).

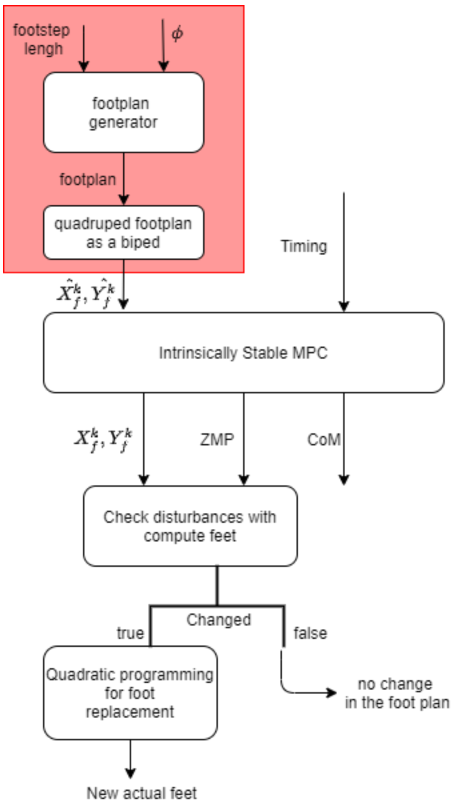


Figure 1: Block scheeme of the proposed MPC framework for gait generation

First of all, with our approach we can preserve the linearity considering a fictitious square around the ZMP that will be the box constraint for the ZMP

at all times. It's clear that this is a restrictive solution because the area of this square is smaller than the actual support polygon in the various phases of the gait, but the fact that it is a box allows us to formulate the problem with linear constraints. The main idea depicted schematically in Figure 1 is in primis to compute once for all, the initial footplan for the quadruped. Then thanks to some considerations that we will explan later, we translate that footplan to a footplan for the ideal biped whose steps coincide exactly with the fictitious square relative to the constraints on the quadruped ZMP. This is depicted as the red square in the previous scheme.
Then we perform the IS-MPC approach on the ideal biped, so as to generate as output the ZMP velocity. After that exploiting the prediction done on the ZMP position we perform a disturbance check, considering a compute_feet block. This will end up into a flag variable called "CHANGED". If that variable is set to TRUE, it means that a disturbance is occurred and so a second QP is performed so as to compute a new footplan for the quadruped. Otherwise nothing will change.

3 Quadruped parametrization

The quadruped has been described by mean of five parameters (Figure 2):

1. Displacement B (*disp_B*) is the distance along the y-direction between the center of a foot and the middle of the body.
2. Displacement C (*disp_C*) is the distance between two feet from the same side when the robot is at initial rest.
3. Displacement A (*disp_A*) is the desired distance imposed by the operator for a footstep.
4. Angle ϕ is the desired gait orientation.
5. Height h_c is the z-coordinate of the CoM.

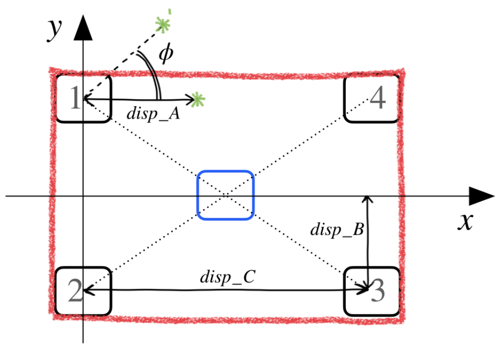


Figure 2: Quadruped parameters in the xy-plane

It has been considered an initial state (expressed in meters):

$$[x(0), y(0), z(0)]^T = [disp\_C/2, 0, h_c]^T$$

while, for the initial velocity the robot is at rest. Each footstep is a square polygon where each side has length of 0.02 m. Looking again at Figure 2 it is possible to see in red the contour of the support polygon and in blue a square centered at the intersection of the two diagonals that pass respectively in the center-foot 1-3 and center-foot 2-4. This square can be considered as an uncertainty area where the ZMP lies when the robot is at rest conditions with no tilt angles for the body. For each foot we consider kinematic limitation as a rectangular region described by four parameters : F for the forward range, B as backward, I to address the limitation for the inside direction and O for outside admissible range (Figure 3).

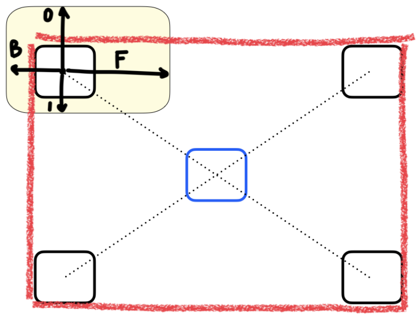


Figure 3: Foot kinematic limit

Since "*disp_A*" could exceed the kinematic limitation imposed by the hardware of the quadruped, a further analysis is taken in consideration.
The actual (x,y) step displacementare described by:

$$x\_passo = disp\_A * cos(\phi) \tag{1}$$
$$y\_passo = disp\_A * sin(\phi) \tag{2}$$
$$x\_passo\_dummy = disp\_A * cos(\phi)/2 \tag{3}$$
$$y\_passo\_dummy = disp\_A * sin(\phi)/2 \tag{4}$$

If these component exceed the limitation range shown in Figure 3 the desired footstep is recomputed in order to preserve the angle ϕ and at same time to be kinematically feasible.
The idea is the following: at the beginning, when we assign the "*disp_A*" parameter and the gait angle "ϕ" we compute the displacements in terms of x and y coordinate of each consecutive step as (1),(2),(3),(4). The fist step ("dummy" step) is actually performed as half of the regular step size. In this way during the gait the quadruped never returns in a rest configuration, where by *rest* we mean a configuration with a rectangular support polygon.

Once we have done this, we check if the robot is able to perform such (x,y) step displacement, by looking if the step is falling inside the kinematic admissible area (the yellow one in Figure 3). If this is not the case, we "truncate" in some sense the (x,y) step displacement at the robot's kinematic limit, preserving the gait orientation given by parameter "ϕ".
At this point we are sure that the initial "*foot_plan*", is actually feasible by the quadruped.
More than that, considering a fictitious square to describe the region where the ZMP should be constrained, we have centered that region at the diagonals cross. In this way we are able to easily translate the quadruped's kinematic limitations to some corresponding limitations (Figure 4) of an ideal humanoid whose step locations are exactly the blue fictitios square depicted during the gait (Figure 6). In this case the forward and backward ranges remain the same, while for the lateral limitation L the new parameter is connected to the feet parameter as $L = (I + O)/2$.

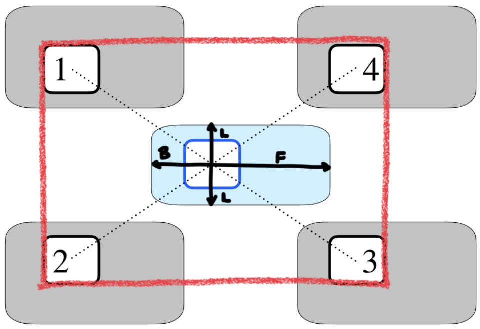


Figure 4: Admissible kinematic region for ZMP fictitious foot.

4 Trotting

Trotting is a particular animal gait, characterize by moving at the same time opposite legs (e.g. front right with back left and front left with back right). In the table below we have encoded the method to generate the desired footstep considering just three consecutive steps starting from an initial step configuration, because the procedure is recursive. The first row represents the initial position of the quadruped feet; then in the next rows the necessary operation to do in order to obtain the desired steps are encoded. As can be seen the first feet that are moving are the number 1 (e.g. Back left) and 3 (e.g. Front Right) recalling the numeration in Figure 4. In the table the terms "$+y\_pd$" and "$+x\_pd$" stand for the initial dummy step displacement, while "$+y\_p$" and "$+x\_p$" address the normal step displacement. Finally the operator "$\bullet$" indicates to hold the previous value of the cell above without performing any calculation like the addition of "$+x\_p$","$+y\_p$" at the actual footstep coordinations.

Back Left		Back Right		Front Right		Front Left	
x	y	x	y	x	y	x	y
0	disp_B	0	-disp_B	disp_C	-disp_B	disp_C	disp_B
+x_pd	+y_pd	•	•	+x_pd	+y_pd	•	•
•	•	+x_p	+y_p	•	•	+x_p	+y_p
+x_p	+y_p	•	•	+x_p	+y_p	•	•

The main idea of the presented work is given looking at Figure 5 where a square of the same size of the footsteps is built centered in the point where the two diagonals of a four feet support polygon cross each other. This square is a region where we constrained the ZMP to lie. Thinking as a biped this is the equivalent of a single support phase. While swinging the quadruped goes to the next four-feet support polygon passing through a two feet support polygon. Translating the idea in [1] to our case, we used the concept of moving constraints to shift the position of the ZMP during the double support phase. Since the desired ZMP position in quad-support is always in the intersection of the two diagonal linking the opposite feet, during the double support phase the ZMP square roto-translates along the diagonal to reach the next desired position.

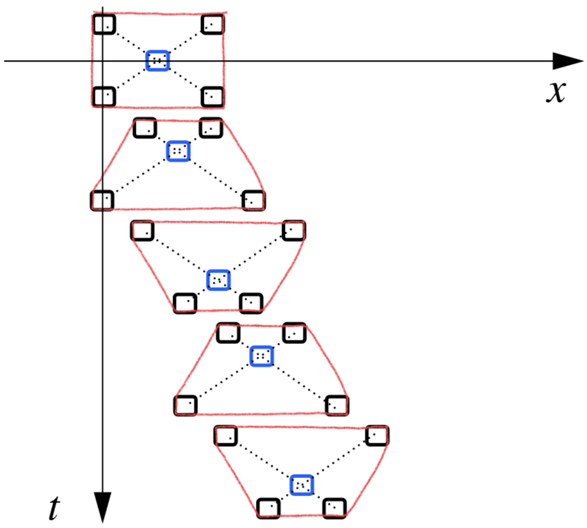


Figure 5: Gait phases for trotting maneuver.

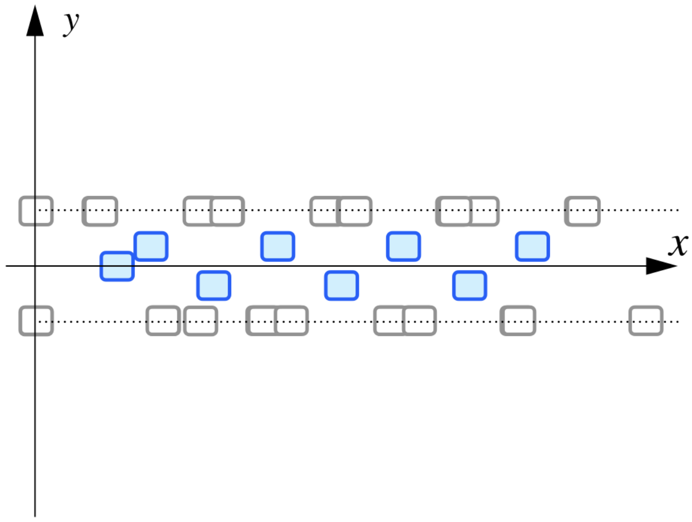


Figure 6: Developed gait during trotting maneuver: in grey the original footsetps and in blue the ZMP fictitious feet.

In Figure 5 we can see how the position of the footsteps along the x axis are changing in time, while looking at Figure 6 it' s possible to see the ZMP square together with the actual footsteps. It is clear from the figure that the blu squares could be interpreted as the footstep locations of a walking biped.

5 Walking

A walking maneuver is instead characterizes by the fact that only one leg is moved during a swing phase. As it is possible to see from Figure 7, recalling the feet number of Figure 4 the sequence of moving feet proposed is $1-3-4-2$. For the walking gait, the same reasoning used for the trotting is not directly applied; during the swing phase from a four feet support polygon to the next there exist phases of three feet support polygon that have to be considered.
For these "phases" we have decided to constraint the ZMP inside the fictitious square considered at the previous quad-support. This choice is made because of the fact that otherwise we will have an unsmooth reference trajectory, that will end up into a similar situation for the CoM's generated trajectory.
So the walking gait is decomposed as shown in Figure 7.

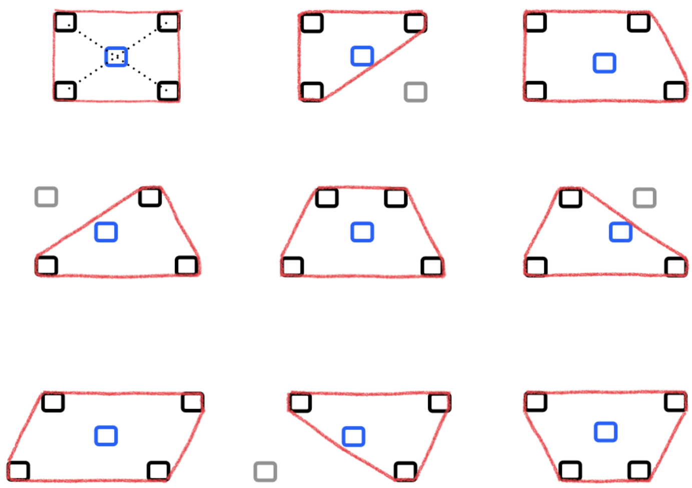


Figure 7: Gait phases for walking maneuver

The fictitious ZMP foot could in principle be placed in any configuration inside the red perimeter of Figure 7 but, in order to obtain trajectory for the ZMP as smooth as possible it has been decide to let the fictitous ZMP foot lies centered at the centroid point when the quadruped is at a four support polygon and remain there also during a three feet support polygon.

6 IS-MPC for quadruped

So following the results presented by N. Scianca [1], we set up an Intrisically Stable Model Predictive Control approach (Eq.11) using as a prediction model the LIP whose dynamics is expressed by Eq.5. This choice is driven by the assumption of describing the CoM position for a biped walking on a flat horizontal surface at constant height.

$$\ddot{x}_c = \eta^2(x_c - x_z) \tag{5}$$

Where $\eta = \sqrt{g/h_c}$, x_c and x_z represent the coordinate of the CoM and ZMP, while h_c is the height of the CoM and g is the gravity term.

$$\begin{cases} \min_{\substack{\dot{X}_z^k,\dot{Y}_z^k \\ X_f^k, Y_f^k}} \|\dot{X}_z^k\|^2 + \|\dot{Y}_z^k\|^2 + \beta(\|X_f - \hat{X}_f\|^2 + \|Y_f - \hat{Y}_f\|^2) \\ \text{subject to} \\ \text{ZMP constraint} \\ \text{kinematic constraint} \\ \text{stability constraint for x and y} \end{cases} \tag{6}$$

All the decision variables are collected in vectors below:

$$
\begin{aligned}
\dot{X}_z^k &= (\dot{x}_z^k, ..., \dot{x}_z^{k+C-1})^T \\
\dot{Y}_z^k &= (\dot{y}_z^k, ..., \dot{y}_z^{k+C-1})^T \\
X_f^k &= (x_f^1, ..., x_f^{F'})^T \\
Y_f^k &= (y_f^1, ..., y_f^{F'})^T
\end{aligned}
$$

The results of this QP (6) that are generated are:

1. The control variables, (i.e. the ZMP velocities) $\dot{x}_z^{k+i}, \dot{y}_z^{k+i}$ over the control horizon C for $i = 0, ..., C-1$.
2. The fictitious ZMP footstep position x_f^j, y_f^j for $j = 1, ..., F'$,where F' denotes the number of footsteps that fall inside the control horizon.
3. The output history $x_c(t), y_c(t)$ for the position of the CoM.

6.1 ZMP constraint

This constraint addresses the fact that the ZMP has to lie inside the current support polygon at all time instants within the control horizon. In this case a more conservative apprach is consider since the fictitious ZMP footstep is actually smaller than the actual support polygon. When the quadruped is in four feet support, the fictitious ZMP footstep represent the actual admissible region as a biped in single support on one foot, which can be approximated as a rectangle of dimension $d_{z,x}, d_{z,y}$ centered at the actual jth-footstep position (x_f^j, y_f^j). Moreover since the ZMP is piecewise linear this constraint can be expressed as Eq.7

$$
R_j^T \begin{pmatrix} \delta \sum_{l=0}^{i} \dot{x}_z^{k+l} - x_f^j \\ \delta \sum_{l=0}^{i} \dot{y}_z^{k+l} - y_f^j \end{pmatrix} \leq \frac{1}{2} \begin{pmatrix} d_{z,x} \\ d_{z,y} \end{pmatrix} - R_j^T \begin{pmatrix} x_z^k \\ y_z^k \end{pmatrix} \tag{7}
$$

Where R_j^T is the planar rotation matrix w.r.t the foot orientation θ_j. While the quadruped is swinging instead the support polygon would be a convex hull of the two fictitious ZMP footsteps and in order to preserve linearity, it is adopted an approach based on moving constraint. The high level concept is that the support polygon has exactly the same size when is in single support and it roto-translate from one footstep to the next.

6.2 Kinematic constraint

Considering at the jth and the ZMP fictitious step centered in (x_f^{j-1}, y_f^{j-1}) and oriented as θ^{j-1}, the admissible region to place the footstep (x_f^j, y_f^j) is a rectangle oriented as θ^{j-1} recalling the dimension and the placement looking at Fig.4. The constraint can be written as Eq.(8)

$$
R_{j-1}^T \begin{pmatrix} x_f^j - x_f^{j-1} \\ y_f^j - y_f^{j-1} \end{pmatrix} \leq \pm \begin{pmatrix} 0 \\ disp\_L/2 \end{pmatrix} + \begin{pmatrix} disp\_F \\ disp\_L/2 \end{pmatrix} \tag{8}
$$

Recall that both equations (7) and (8) should be completed by a left hand side and imposed for all footsteps in the control horizon ($j = 1, ..., F'$).

Another problem to take into account is the foot slippage which can occur depending on many factors. The presence of uneven terrain, the acceleration of the CoM, the acceleration of the feet during the swing phase and the friction model are all aspects that need to be taken into consideration. In works like [2] the Coulomb friction model has been taken into account, and in particular it is shown how the friction cone imposes constraints on the region available for the reaction forces. Since our kinematic constraints are quite conservative, the plane is horizontal and the chosen footstep timing allows a slow motion of the robot, we can neglect the constraints related to the friction cone.

6.3 Stability constraint

The last condition represents a constraint that guarantees the CoM trajectory to be bounded w.r.t. the ZMP. The results shown in [1] for the stability constrained can be translated to this study case obtain, (following the proof in the cited paper above as):

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+1} = -\sum_{i=C}^{\infty} e^{-i\eta\delta} \dot{\tilde{x}}_z^{k+i} + \frac{\eta}{1-e^{-\eta\delta}}(x_u^k - x_z^k) \tag{9}$$

where x_u is the Divergent Component of Motion, $\dot{\tilde{x}}_z^{k+i}$ for $i = k+C, k+C+1, ...$ are the ZMP velocities after the control horizon or *tail* and are obviously unknowns. To tackle this issue we have used anticipative tails: the candidate fictitious ZMP foot plan has been used to generate a trajectory for the ZMP, allowing to numerically differentiate it to recover the missing velocities in the time interval $[T_c,\ T_p]$. In the interval $[T_p,\ \infty]$, instead, the ZMP velocities are truncated (imposed equal to zero) or periodic, since one can assume that the movement will be replicated in the future. Setting $\dot{\tilde{x}}_z^{k+i} = \dot{x}_{z,ant}^{k+i}$ for $i = C, ..., P-1$ and $\dot{\tilde{x}}_z^{k+i}$ for $i = P, P+1, ...$ the stability constraint becomes:

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+1} = -\sum_{i=C}^{P-1} e^{-i\eta\delta} \dot{x}_{z,ant}^{k+i} - \sum_{i=P}^{\infty} e^{-in\delta} \dot{\tilde{x}}_{z,ant}^{k+i} + \frac{\eta}{1-e^{-\eta\delta}}(x_u^k - x_z^k) \tag{10}$$

From now on it is used an anticipative tail such that the ZMP trajectory in $[T_c, T_p]$ is always at the center of the ZMP admissible region and the residual part of the tail is truncated. Eq.10 now gives a closed form expression that can be used for real time implementation.

7 Quadratic programming for foot replacement

When a disturbance is acting on the quadruped, in order to let the ZMP trajectory remain in the center of the support polygon, a function (QP approach), is used such that the final position of the moving foot is recomputed in the right way.

We check if a disturbance is occurred just looking at the intersection between the diagonals, that are generated by the given feet, and see if that intersection mismatches the predicted ZMP location. If there is no mismatch, then a flag variable called "changed" is set to FALSE in order to do not perform the QP relative to the foot replacement. Otherwise "changed" is set to TRUE and the QP for the foot replacement is performed.

7.1 Quadratic programming for trotting foot replacement

For the trotting gait at each step two feet are moving (foot n.1 with n.3 or foot n.2 with n.4) and so they are the decision variables for the cited QP, that is structured as:

$$\begin{cases} \min\limits_{\substack{x_{fl}, y_{fl} \\ x_{fr}, y_{fr}}} \ (X_{fl} - x_{fl})^2 + (Y_{fl} - y_{fl})^2 + (X_{fr} - x_{fr})^2 + (Y_{fr} - y_{fr})^2 \\ \text{Subject to} \\ \text{Kinematic foot constraints (Figure 3)} \end{cases} \tag{11}$$

where x_{fl}, y_{fl} are the coordinate of the moving left foot and x_{fr}, y_{fr} for the right foot, with the capital letter is address instead the desired foot position. Our main idea, in order to generalize as much as possible this function, is trying to "absorb" the disturbance effect, maintaining the gait orientation w.r.t. the *x-axis* (e.g. ϕ angle) and moving the feet consequently just along the gait direction.
First of all we compute the line equation passing through the "fixed" feet. Then we consider the line equation with opposite angular value and passing through the ZMP location at the same time. At this point we have considered the location given by the intersection of these two equations; if no disturbances are present, this intersection has to coincide exactly with the predicted ZMP location. This means that the two variables *dist_x*, *dist_y* are equal to zero. Otherwise, a disturbance has occurred. So the new step location is computed as:

$$\begin{aligned} eq1 &= \tan(\phi)(x - x_{1free}) + y_{1free} \\ eq2 &= \tan(\phi)(x - x_{2free}) + y_{2free} \end{aligned} \tag{12}$$

Then referring as *sol1* and *sol2*, to the *x-solution* of the intersection between Equation 12 and the equation with opposite angular value and passing through the ZMP (denoted as *equation_2*), we obtain for the new feet coordinates:

$$\begin{aligned} x_{1free} &= sol1, & y_{1free} &= subs(equation\_2, sol1) \\ x_{2free} &= sol2, & y_{2free} &= subs(equation\_2, sol2) \end{aligned}$$

Obviously, this approach has a singularity when $\phi = \frac{\pi}{2}$. We have handled this case manually with an "if" condition.

7.2 Quadratic programming for walking foot replacement

Since in walking we have three fixed feet attached to the ground we have used a different quadratic programming (13) in order to reconfigure the foot in case of disturbances.

$$\begin{cases} \min\limits_{x_f, y_f} (X_f - x_f)^2 + (Y_f - y_f)^2 \\ \text{subject to} \\ \text{feet constraints} \end{cases} \tag{13}$$

A function that recompute the desired moving foot position is used. Given the fixed feet positions, the expected position for the free one and the predicted position of the ZMP, we check if a disturbance is occurred just looking at the intersection between the diagonals, that are generated by the given feet, and see if that intersection mismatchs the predicted ZMP location. If there is no mismatch, then a flag variable called "changed" is set to FALSE in order to do not perform the QP relative to the foot replacement. In the other case, the function "compute_one_feet_walk" has provided an intuitive foot location based on the same reasoning done for the trotting. Then the second QP is activated in order to try to perform that changing in the new foot location considering the feet constraints.

8 MATLAB Simulations

To validate the approach simulations were performed in Matlab [3] using the dynamic parameters of the ANYmal robot. The below table reports the parameters used in the following simulations.

Parameter	Symbol	Value	Unit
Robot Mass	m	30.5	kg
Gravity Acceleration	g	9.81	ms^{-2}
Foot size	f_s	0.02	m
Height	h	0.56	m
Length	$disp_C$	0.9303	m
Half body width	$disp_B$	0.259394	m
Lenght of footstep	$disp_A$	0.15	m
Sampling Time	δ	0.01	s
Trotting Control horizon	$C\delta$	160δ	s
Trotting Prediction horizon	$P\delta$	320δ	s
Walking Control horizon	$C\delta$	100δ	s
Walking Prediction horizon	$P\delta$	200δ	s
Footstep in the control horizon	F'	3	-

Table 1: Parameters used for the following simulations.

We have investigated six simulation scenarios: three trotting gait with different orientation ϕ and three walking one to see if the approach can be generalized also for this type of gait.

8.1 Trotting with $\phi = 0$

The first simulation is the "basic" one, since we have considered a trotting gait with $\phi = 0$. No disturbances are taken into account and the terrain is considered as an even one. Moreover the gait is performed with a stepping time of 0.8 seconds and with a step distance of 0.15 meters. In Figure 8 it is highlighted both the steps sequence of the quadruped as well the ZMP and CoM trajectories. We can see that both trajectories are quite a smooth ones and really similar with the biped CoM and ZMP behaviour in [1]. The first step is a "dummy_step" in fact the fictitious square (blue square) is travelling an half distance w.r.t. to all the next steps.

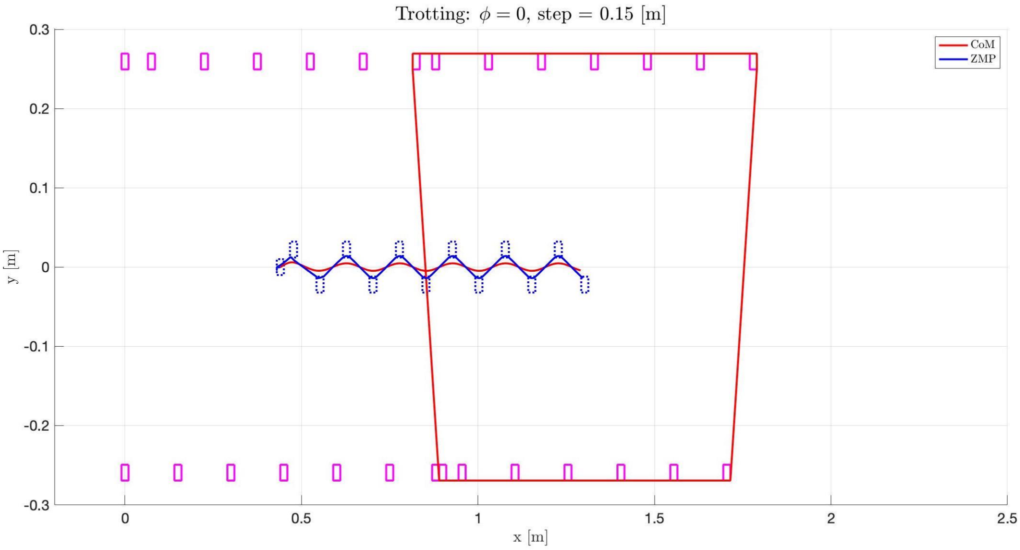


Figure 8: Trotting $\phi = 0$, step distance of 0.15 [m].

8.2 Trotting with $\phi = \frac{\pi}{2}$

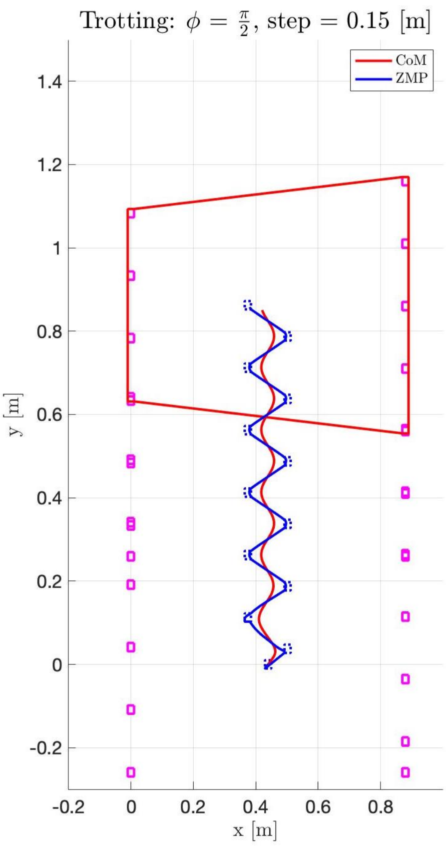


Figure 9: Trotting $\phi = \frac{\pi}{2}$, step distance of 0.15 [m].

The same considerations done for simulation with $\phi = 0$ are still valid. In this case the orientation of the trotting is at $\phi = \frac{\pi}{2}$, in fact the gait is along a *constant-x-direction*. The gait is performed in the exact settings of the previous simulation, so considering a step timing of 0.8 seconds and a step distance of 0.15 meters. We have done this to see if the robot was able to achieve a similar results considering different ϕ values, and so it is.

8.3 Trotting with $\phi = \frac{\pi}{4}$

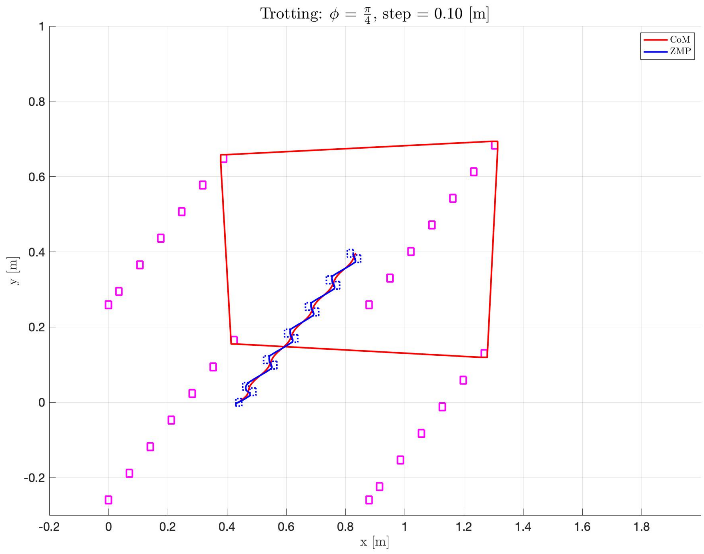
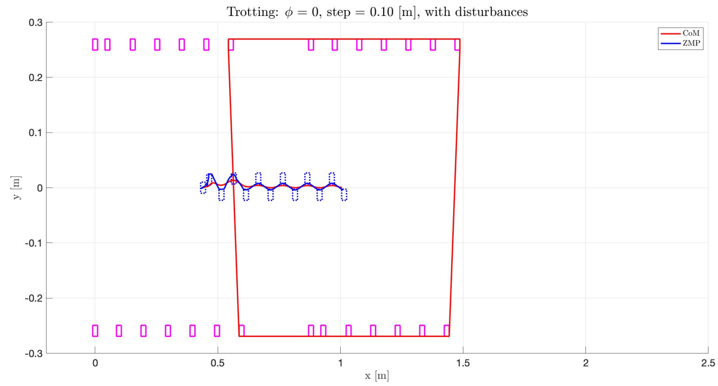


Figure 10: Trotting $\phi = \frac{\pi}{4}$, step distance of 0.10 [m].

The simulation in Figure 10 is the last experiment done considering a trotting gait. In this scenario the quadruped is moving with a step size of 0.10 meters along a direction line of orientation $\phi = \frac{\pi}{4}$ w.r.t. the *x-axis* and with a step timing of 0.8 seconds.
We can see that basically the feet are placed in the right displacement one w.r.t. each other and this is obtained by the second QP that actually is not changing the initial *"foot_plan"* since no disturbances occur. We check if a disturbance is present by looking at the difference of the predicted ZMP location and the actual one. If there is no difference then a flag, named "changed" in the MATLAB code, is set to FALSE and so there is no utility in performing also the second QP.

8.4 Trotting with disturbance

For the case in which ϕ is equal to zero we have introduced a small disturbance along the y axis in order to see if the robot was able to recompute the footsteps. The scenarios considered are two: the difference between them stands in the magnitude of the disturbance, which is given at acceleration level. In the first scenario, during the fifth step of the gait is introduced a disturbance along y of $0.4\ \frac{m}{s^2}$.



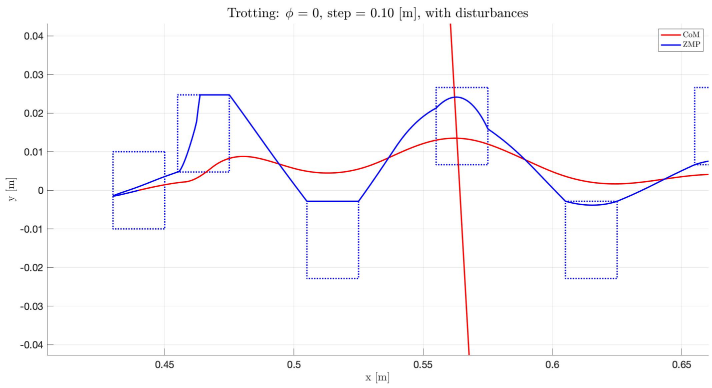
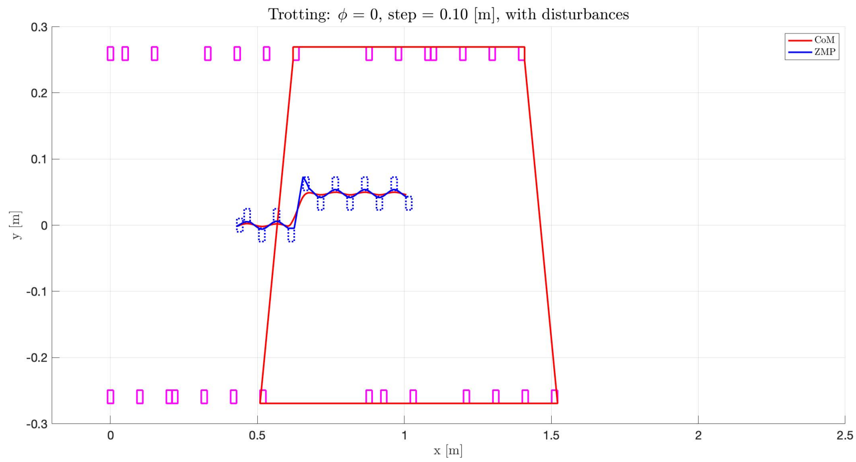


Figure 11: Trotting $\phi = 0$ with disturbance of $0.4\ \frac{m}{s^2}$ along y

We can clearly see that the magnitude of the disturbance is not enough to cause a recomputation of the footsteps. In fact from the zoomed figure we notice that even though the ZMP is disturbed, it never exits the blue centroid, so the gait can proceed as planned in advance.
In the second scenario the disturbance magnitude is $0.5\ \frac{m}{s^2}$ along y axis and the result is shown in the next figure.



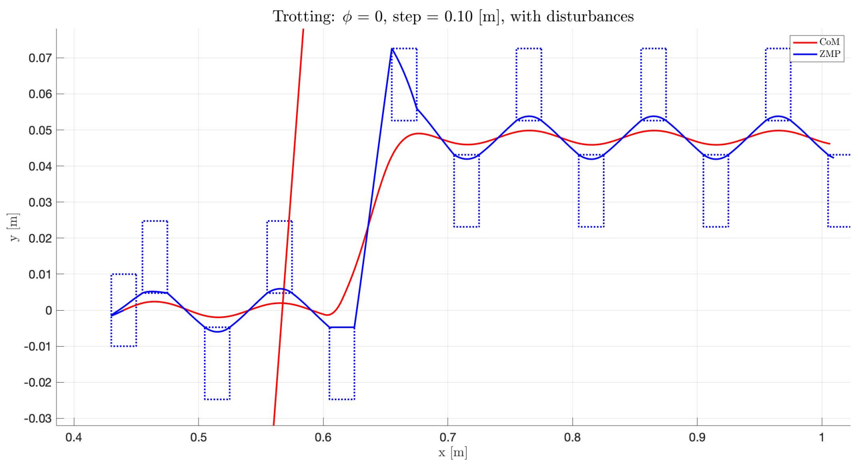


Figure 12: Trotting $\phi = 0$ with disturbance of 0.5 $\frac{m}{s^2}$ along y

We notice that in this second case the disturbance is shifting the position of the center of mass, and as a consequence the ZMP cannot solve the QP problem while remaining in the centroid, so the steps are recomputed in such a way that the blue centroid is on the diagonal of the next double support,

8.5 Walking with $\phi = 0$

In this simulation, we have tried to check if the reasonings applied for a trotting gait are also compatible in a walking scenario. So we have considered a walking gait with a step size of 0.10 meters and a step duration of 0.5 seconds, on a flat and even terrain with no disturbances.

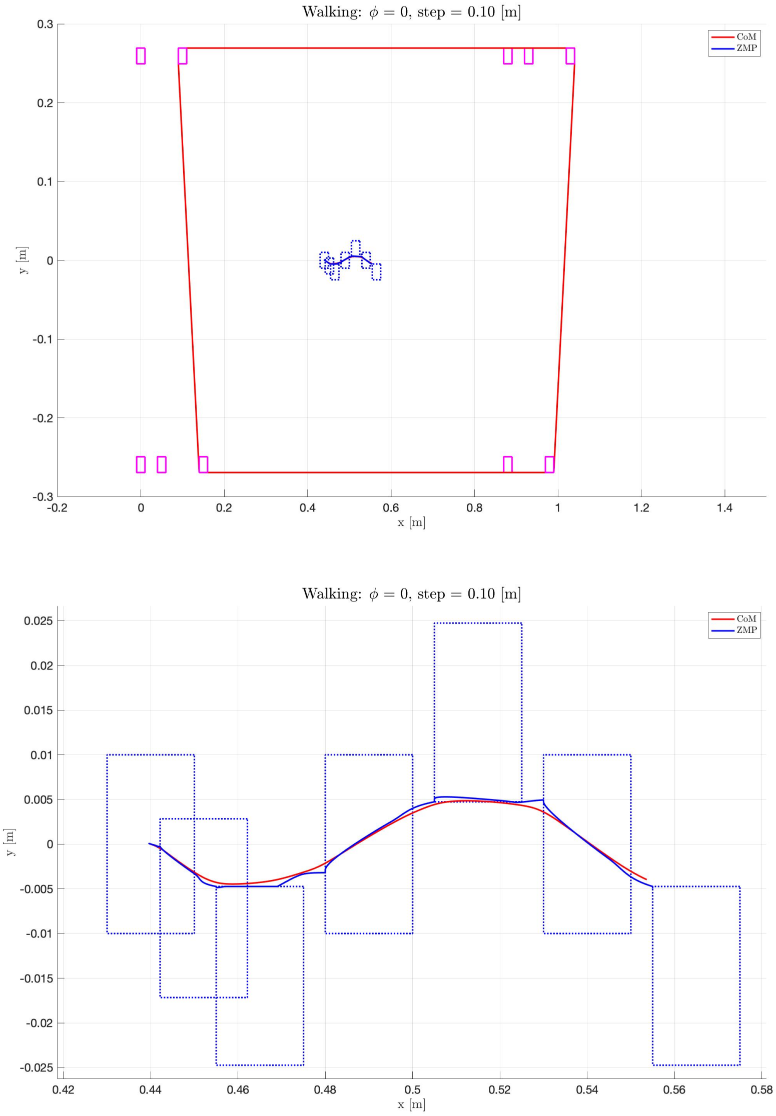


Figure 13: Walking $\phi = 0$, step distance of 0.10 [m].

First of all we want to underline that the simulation time for this walking case is exactly the same considered for all the previous trotting cases. As we expected since we are looking to a walking scenario, the CoM movement between consecutive steps is really small. Consequently the gait is almost a static one, and so the distance covered by the quadruped considering the same amount of time is less w.r.t. the trotting case, even if we have considered a stepping time of just 0.5 seconds.

8.6 Walking with $\phi = \frac{\pi}{2}$

Proceding along this "direction" we were able to generalize this approach considering a generic ϕ: $\phi = \frac{\pi}{2}$ in Figure 14 and a case with $\phi = \frac{\pi}{4}$ in Figure 15.

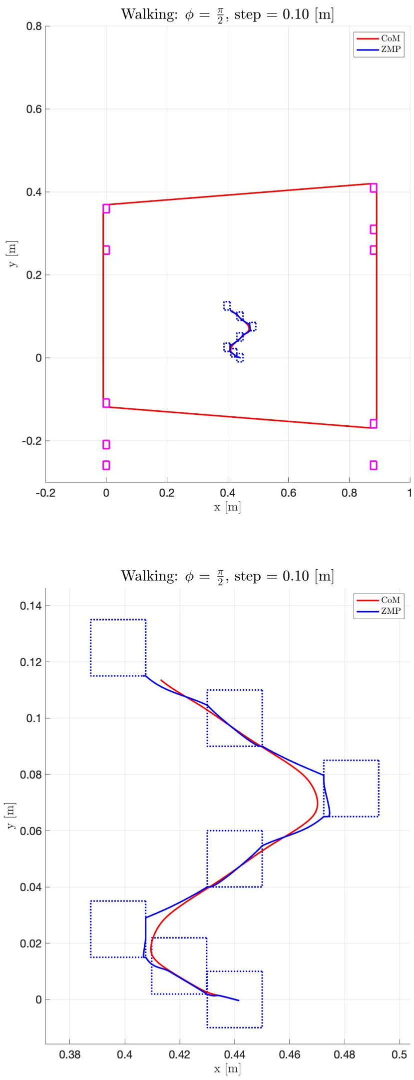
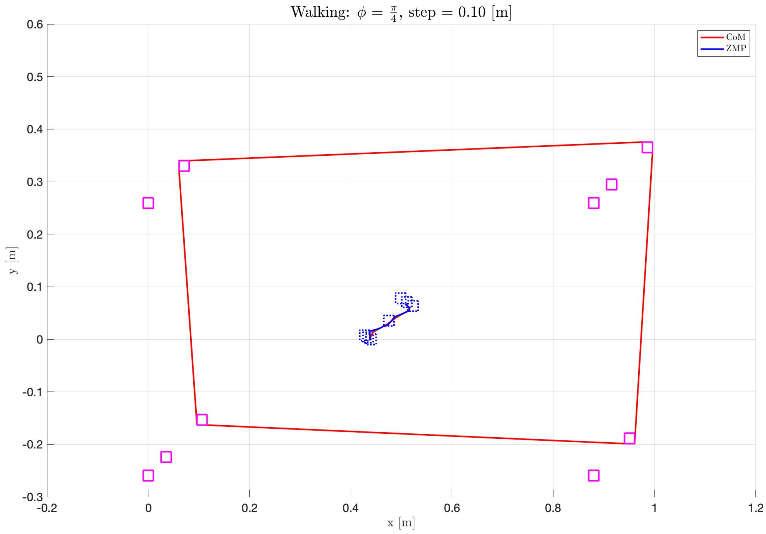


Figure 14: Walking $\phi = \frac{\pi}{2}$, step distance of 0.10 [m].

8.7 Walking with $\phi = \frac{\pi}{4}$



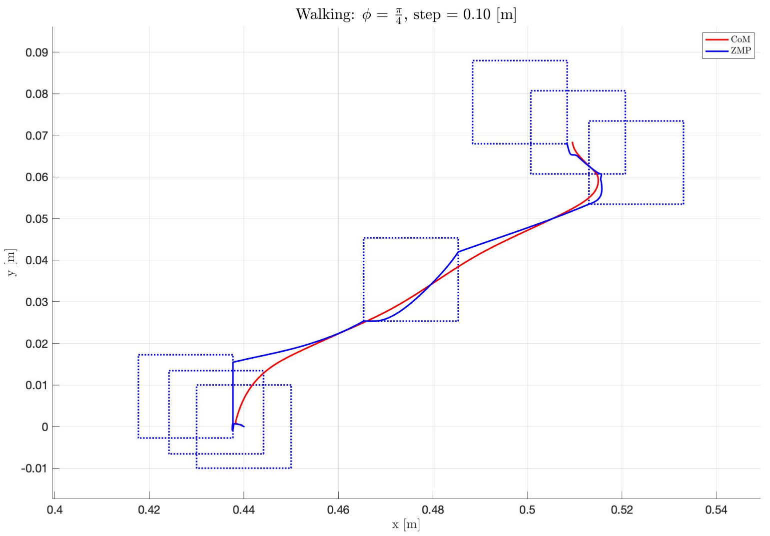


Figure 15: Walking $\phi = \frac{\pi}{4}$, step distance of 0.10 [m].

From these two last simulations it seems that everything is working well. We will see next from Dart's simulations thet something goes wrong when we perform a walking gait with $\phi \neq 0$.

9 DART Simulations

We now present dynamic simulations conducted on the ANYmal robot in the DART environment. In order to perform the gait in DART, we have first of all extracted the CoM's position and velocity trajectory as well the swing trajectory of each foot. For the latters we have considered a quadratic trajectory for the step's height, with step maximum height of 0.02 [m] and swing time equal to the stepping time considered in the corresponding MATLAB simulations.
The basic idea was to give all these trajectories as references, and perform an inverse kinematic to generate the corresponding control inputs in order to obtain reference tracking.
So the MPC problem is handled by MATLAB code and we have just checked if in a dynamic scenario all our reasonings are realizable.

Figure 16: ANYmal quadruped.

9.1 Trotting

Proceding with the same simulated scenarios seen in Section VIII, we have considered for all DART simulations: a simulation time of 20 seconds, a sampling time of $\delta = 0.01$ [s], a flat ground, without disturbances and with a robot's body that does not perform any type of rotation (e.g. roll/pitch/yaw angles equal to zero).
For the robot's parameters, we have considered the ones in Table 1.
The three simulations are performed considering respectively a ϕ angle equal to "0" Figure 17, equal to "$\frac{\pi}{2}$" Figure 18 and equal to "$\frac{\pi}{4}$" Figure 19. In these pictures we have tried to highlight as much as possible the gait direction, considering six snapshots that have to be looked as first and second "row-images" from left side to the right side.

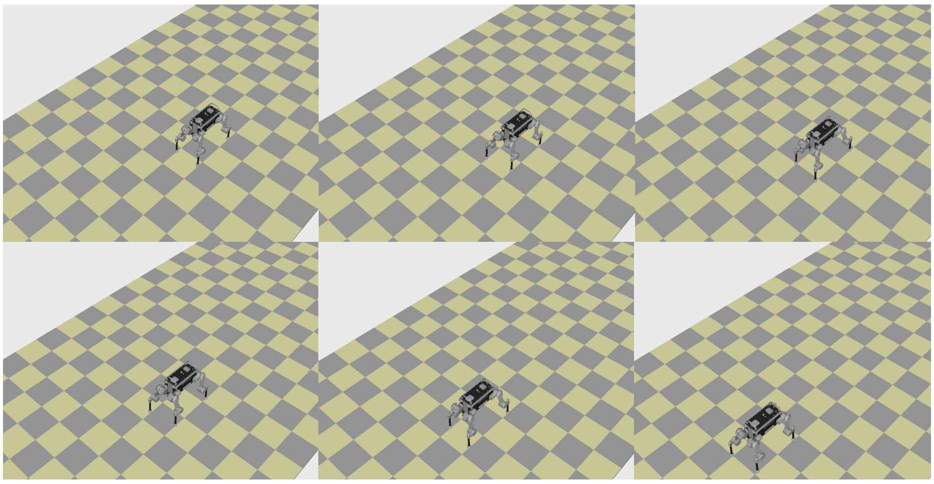


Figure 17: Trotting $\phi = 0$, step distance of 0.15 [m].

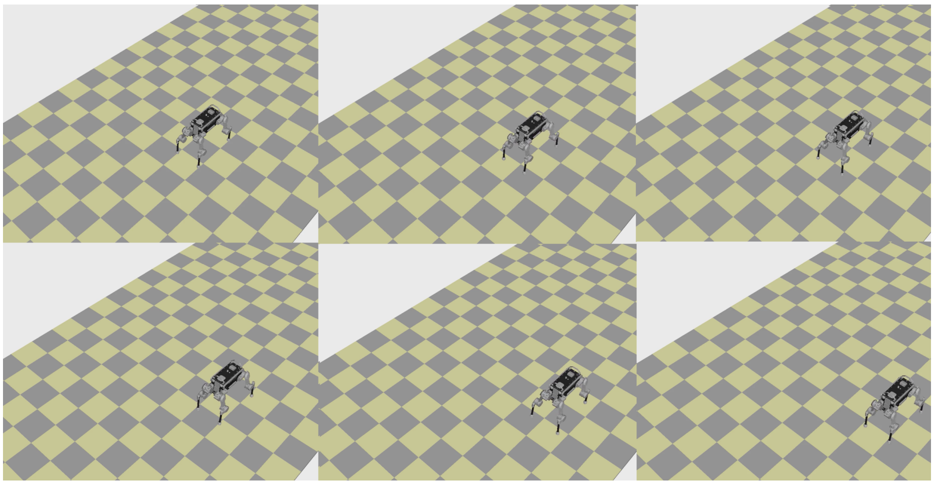


Figure 18: Trotting $\phi = \frac{\pi}{2}$, step distance of 0.15 [m].

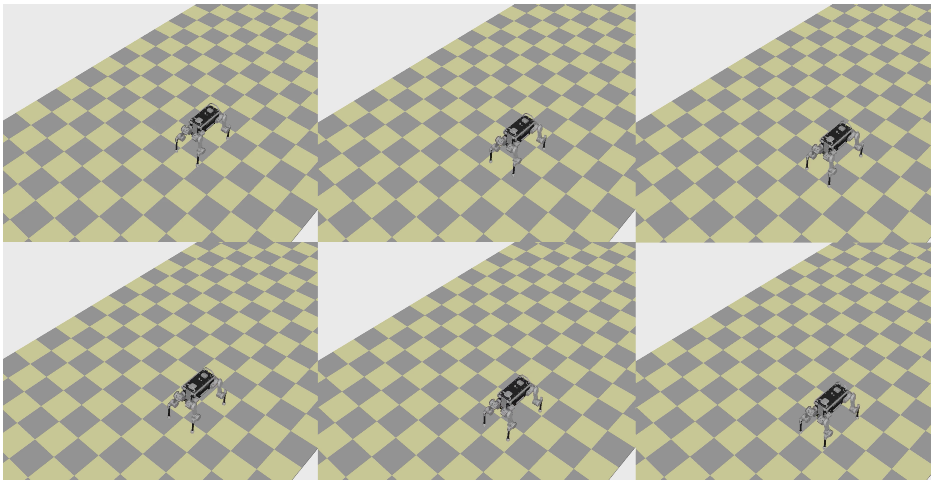


Figure 19: Trotting $\phi = \frac{\pi}{4}$, step distance of 0.10 [m].

We can observe from the previous three pictures that the approach is dynamically realizable for a trotting gait scenario. The quadruped seems to trot along the right direction axis defined by the angle ϕ without falling down, and at the same time performing the right step distance comparing it with the squares depicted on the ground. In fact, in Figure 17 the robot is moving longitudinally along its own initial front sequence of squares depicted on the ground. Instead for the second simulation Figure 18 we can see that the gait is purely lateral, while in Figure 19 the quadruped is performing a diagonal gait.

9.2 Walking

The DART simulation we have performed for the walking gait is the case in which $\phi = 0$.

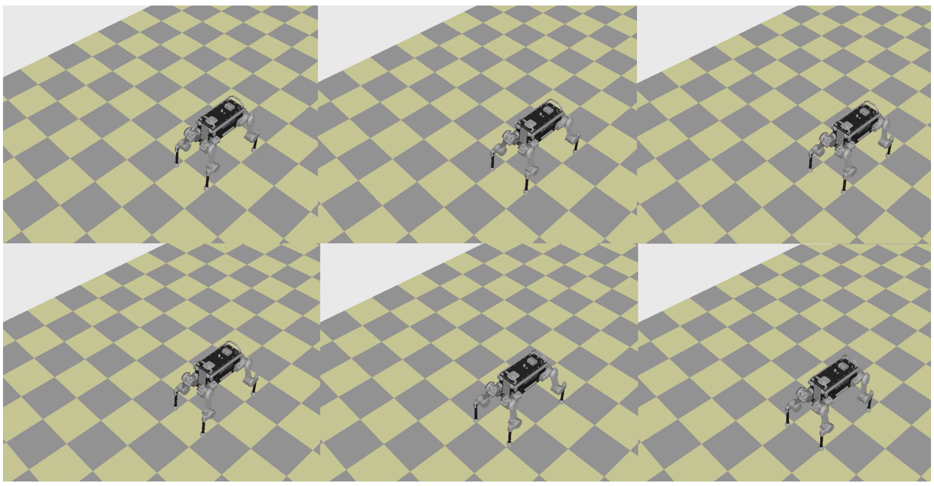


Figure 20: Walking $\phi = 0$, step distance of 0.10 [m].

In Figure 20 we have considered a zoomed view w.r.t. the DART simulations for the trotting gait. This is mainly due to the fact that as explained during MATLAB simulations the walking scenario seems to be a static gait so, even if the view is zoomed, we are sure that the quadruped will be always inside the view during the entire simulation time of 20 seconds.
From this simulation we have just checked that also for the walking case, the approach is dynamically realizable, and so the robot cal perform a gait without falling down.

10 Conclusion

With this project we have investigated a different approach, w.r.t. the state of the art, to generate a quadruped gait based on an IS-MPC framework.
This approach has some pros and cons. The main reason that lead us to investigate this approach was the preservation of linearity of the problem and the fact that it is quite computationally inexpensive. This is more highlighted when no disturbances are present, because of the fact that the second QP for the foot replacements is not "called". With all our simulations we have checked that this method is quite generalizable for a trotting and a walking scenarios. As a con, we were very conservative considering the allowable support polygon for the ZMP to lie and also quite conservative considering the approach for the foot replacement. Moreover the reasonings done for the ZMP constraints can be performed in a better way, considering the gait timing and so the the instants on which the robot is in quad support or not. A possible further approach could be consider a different foot replacement policy without any priority to the gait orientation ϕ, and maybe consider an approach that leads the CoM to be centered at the center of the quadruped body so as to do not unbalance the robot shape.

11 References

[1] N. Scianca et al. "MPC for Humanoid Gait Generation". In: *IEEE Transactions on Robotics* 36.4 (2020), pp. 1171–1188.

[2] Mahdi Khorram and S. Ali A. Moosavian. "Stable walking of a quadruped robot on uneven terrains with friction constraints". In: *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*. 2014, pp. 364–369.

[3] MATLAB. *version 9.9.0.1 (R2021a)*. Natick, Massachusetts: The MathWorks Inc., 2021.

[4] Yapeng Shi, Pengfei Wang, Mantian Li, Xin Wang, Zhenyu Jiang, and Zhibin Li. "Model Predictive Control for Motion Planning of Quadrupedal Locomotion". In: *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*. 2019, pp. 87–92.

[5] Alexander W. Winkler, Carlos Mastalli, Ioannis Havoutis, Michele Focchi, Darwin G. Caldwell, and Claudio Semini. "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 5148–5154.

[6] Guiyang Xin, Songyan Xin, Oguzhan Cebe, Mathew Jose Pollayil, Franco Angelini, Manolo Garabini, Sethu Vijayakumar, and Michael Mistry. "Robust Footstep Planning and LQR Control for Dynamic Quadrupedal Locomotion". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4488–4495.