Adaptive step duration in biped walking: a robust approach to nonlinear constraints

Néstor Bohórquez and Pierre-Brice Wieber

***Abstract*— When a biped robot is walking in a crowd, being able to adapt the duration of steps is a key element to avoid collisions. Model Predictive Control (MPC) schemes for biped walking usually assume a fixed step duration since adapting it leads to a nonlinear problem, in general. Nonlinear solvers do not guarantee the satisfaction of nonlinear constraints at every iterate and this can be problematic for the real-time operation of robots. We propose a method to make sure that all iterates satisfy the nonlinear constraints by borrowing concepts from robust control: we make the problem robust to nonlinearities within some bounds. These bounds are linear with respect to the variables of the problem and can be adapted online.**

I. Introduction

When walking in a crowd, a biped robot should naturally try to avoid collisions with people. One strategy to do this is to adapt the position of steps in response to the motion of the crowd. This can be formulated as a linear problem if one considers the duration of steps to be fixed, as usually done [1], [2], [3]. Nevertheless, the combination of maximal step length and fixed step duration limits the maximal speed the robot can achieve and, consequently, its ability to avoid collisions [4]. We need, therefore, to adapt the duration of the steps of the robot.

However, by doing this, the problem becomes either nonlinear or combinatorial, depending on whether we choose to vary the duration of the steps in a continuous or discrete manner. On the continuous side, nonlinear MPC schemes have been proposed using Sequential Quadratic Programming (SQP) [5] or Interior Point (IP) methods [6]. On the discrete side, these MPC schemes have been formulated as Mixed Integer Quadratic Programs [7]. Alternatively, others have shown that the problem of adapting the duration of the steps can be made completely linear if we either: 1) limit the number of steps in the future to one [8] or 2) consider the relation between the Capture Point and the positions of the future footsteps [9]. However, it is not clear whether these two last approaches are suitable for making a robot walk in a crowd: the first one proposes a preview horizon that might be too short and the second does not account explicitly for the motion of the Center of Mass (CoM).

Standard solvers for nonlinear problems (SQP and IP methods) implement a form of Newton method in which constraints are iteratively linearized. However, the satisfaction of these linearized constraints does not imply the satisfaction of the original nonlinear constraints. There is no guarantee that the nonlinear constraints are satisfied at every Newton iterate. This is a problem for real-time operation where we may need to interrupt the solver before convergence and still need a solution that can be applied safely to the robot. This issue has been reported in [10] but for a different humanoid locomotion problem.

We propose here a method to guarantee safe Newton iterates. It is similar to what was done in [11], [12] for the problem of adapting the vertical motion of the CoM of the robot: we design constraints that are robust to changes in the variable that introduces the nonlinearity. To do so, we make sure the constraints are feasible over a given interval of this variable. We can then make safe linear approximations of the objective function inside of this interval and always generate feasible iterates.

With this method it suffices to solve a single Quadratic Program (which takes 0.29[ms] on average) to obtain a feasible solution. Simulations show that, by adapting the step duration in this way, we can reduce the collision rate of the robot walking in a crowd from 75% to less than 6%.

We proceed as follows: in Section II we present the specific nonlinearity that we face when we adapt the duration of the steps and how we develop robustness with respect to it. We then apply this technique for robustness to all dynamic, kinematic and collision avoidance constraints of the robot in Section III. We introduce in Section IV the Optimal Control Problem (OCP) we need to solve. Finally, we present some numerical results that showcase the capabilities of our controller in Section V.

II. Problem definition

The motion of the CoM of a biped walking robot is commonly modeled as a triple integrator [13]

$$x^+ = \begin{bmatrix} 1 & \tau & \tau^2/2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} \tau^3/6 \\ \tau^2/2 \\ \tau \end{bmatrix} \dddot{c}, \tag{1}$$

where c is the position of the CoM, the jerk $\dddot{c}$ is the input of the system, $x = (c, \dot{c}, \ddot{c})$ and $x^+ = (c^+, \dot{c}^+, \ddot{c}^+)$ are two consecutive states. Here, the jerk is considered to be constant over intervals of time of length τ that we define to be $1/8^{th}$ of the duration of one step of the robot. The successor state x^+ is linear with respect to $\dddot{c}$ but nonlinear with respect to τ. In particular, x^+ is a cubic polynomial of τ. The value of τ is commonly set to a constant in order to obtain a completely linear formulation.

We base our approach for adapting the duration of the steps on varying τ, what gives a nonlinear problem to solve.

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP*, LJK, 38000 Grenoble, France nestor-alonso.bohorquez-dorante, pierre-brice.wieber@inria.fr.

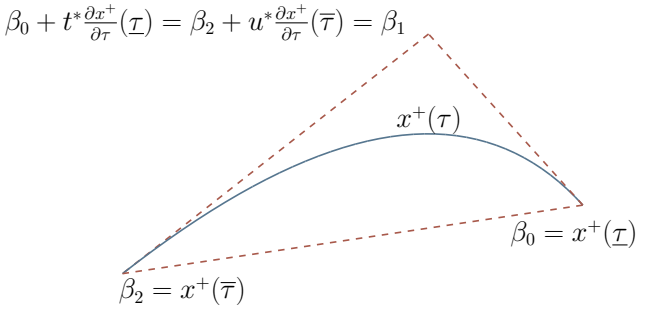


Fig. 1: Boundaries of $x^+(\tau)$ defined as the convex hull of its Bernstein coordinates $(\beta_0, \beta_1, \beta_2)$. The control points β_0 and β_2 are located at $x^+(\underline{\tau})$ and $x^+(\overline{\tau})$, respectively. The control point β_1 is located at the intersection of the lines tangent to $x^+(\underline{\tau})$ and $x^+(\overline{\tau})$.

We propose to restrict τ to an interval $[\underline{\tau}, \overline{\tau}]$ and build bounds on $x^+(\tau)$ that depend on $\underline{\tau}, \overline{\tau}$ but not on τ.

The complexity of the bounds that we define on $x^+(\tau)$ depend on the degree of the polynomial involved. In order to simplify this issue, we reduce the degree of $x^+(\tau)$ by rewriting the last equation of (1) as

$$\dddot{c} = \frac{\ddot{c}^+ - \ddot{c}}{\tau}, \tag{2}$$

and substituting it back in the triple integrator to obtain:

$$x^+ = \begin{bmatrix} 1 & \tau & \tau^2/3 \\ 0 & 1 & \tau/2 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} \tau^2/6 \\ \tau/2 \\ 1 \end{bmatrix} \ddot{c}^+. \tag{3}$$

In this way, the successor state $x^+(\tau)$ is a quadratic polynomial of τ that preserves the C^2 continuity of the motion of the CoM.

We define the bounds on $x^+(\tau)$ by changing its representation from the monomial basis $\{1, \tau, \tau^2\}$ to the Bernstein basis [14]:

$$\{b_0(\tau),\, b_1(\tau),\, b_2(\tau)\}, \tag{4}$$

(traditionally used in Bézier curves) where

$$b_0(\tau) = (1-\tau')^2,\ b_1(\tau) = 2\tau'(1-\tau'),\ b_2(\tau) = \tau'^2, \tag{5}$$

and

$$\tau' = \frac{\tau - \underline{\tau}}{\overline{\tau} - \underline{\tau}}, \tag{6}$$

for a given choice of $\underline{\tau}, \overline{\tau}$. These polynomials have the desirable property that for all $\tau \in [\underline{\tau}, \overline{\tau}]$, $b_i(\tau) \in [0, 1]$, $\sum b_i(\tau) = 1$ and therefore

$$x^+(\tau) = \sum_{i=0}^{2} \beta_i b_i(\tau) \in \text{conv}\{\beta_i\}. \tag{7}$$

So, the polynomial $x^+(\tau)$ is always contained in the convex hull of the points $\beta_0, \beta_1, \beta_2$ whenever $\tau \in [\underline{\tau}, \overline{\tau}]$, as can be seen in Figure 1. These points can be interpreted geometrically as control points of $x^+(\tau)$. Details on the transformation from monomial to Bernstein coordinates can be found in the Appendix.

III. Constraints robust to the step duration

A. Constraint on the steps

We introduce a constraint on the sequence of steps made by the robot $(s_1, \ldots, s_J)$ to prevent it from crossing legs while walking:

$$s_{j+1} - s_j \in \mathcal{S}_j. \tag{8}$$

The set $\mathcal{S}_j$ alternates between left and right according to s_j.

B. Constraint on the CoP

When the robot is walking with its CoM at a constant height over a horizontal flat ground, the position of the Center of Pressure (CoP) can be written as:

$$p^+(\tau) = \begin{bmatrix} 1 & 0 & -h/g \end{bmatrix} x^+(\tau), \tag{9}$$

where h is the height of the CoM and g is the norm of the gravity vector. Due to the unilaterality of the contact forces with the ground, the CoP can only reside inside the support polygon of the robot, which is a convex set $\mathcal{P}$ translated to the j^{th} footstep position s_j:

$$p^+(\tau) - s_j \in \mathcal{P}. \tag{10}$$

This involves a quadratic polynomial of τ that can be expressed in Bernstein basis as outlined earlier:

$$p^+(\tau) - s_j = \sum \lambda_i b_i(\tau) \in \text{conv}\{\lambda_i\}, \tag{11}$$

where $\lambda_i = \begin{bmatrix} 1 & 0 & -h/g \end{bmatrix} \beta_i - s_j$.

We can enforce this nonlinear constraint for all $\tau \in [\underline{\tau}, \overline{\tau}]$ by imposing instead that

$$\forall i:\ \lambda_i \in \mathcal{P}, \tag{12}$$

since by definition of the Bernstein basis:

$$\forall i:\ \lambda_i \in \mathcal{P} \implies p^+(\tau) - s_j \in \mathcal{P}. \tag{13}$$

As mentioned before, each λ_i is linear with respect to the initial state x and the control input $\ddot{c}^+$. By satisfying the linear constraints on $\{\lambda_i\}$ we satisfy the nonlinear constraints on $p^+(\tau) - s_j$.

C. Constraint on the CoM

Since the length of the legs is limited, we do not allow the CoM to be far from the position of the footsteps. We define a convex set $\mathcal{C}_j$ that represents the possible positions of $c^+(\tau)$ with respect to s_j:

$$c^+(\tau) - s_j \in \mathcal{C}_j. \tag{14}$$

The set $\mathcal{C}_j$ alternates between left and right accordingly.

Since $c^+(\tau) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x^+(\tau)$, this involves the quadratic polynomial

$$c^+(\tau) - s_j = \sum \pi_i b_i(\tau) \in \text{conv}\{\pi_i\}, \tag{15}$$

where $\pi_i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \beta_i - s_j$. Then, by enforcing the linear constraint

$$\forall i:\ \pi_i \in \mathcal{C}_j, \tag{16}$$

we make sure that the nonlinear constraint (14) is satisfied for all $\tau \in [\underline{\tau}, \overline{\tau}]$ since

$$\forall i:\ \pi_i \in \mathcal{C}_j \implies c^+(\tau) - s_j \in \mathcal{C}_j. \tag{17}$$

D. Constraint on collision avoidance

In order to avoid collisions, we want the CoM of the robot to be at least a certain distance d from each person m_k in the environment:

$$\forall k: \ \|c^+(\tau) - m_k(\tau)\|_2 \geq d. \tag{18}$$

If we define $m_k(\tau) = \sum \mu_i b_i(\tau)$ then this constraint involves, again, the quadratic polynomial

$$c^+(\tau) - m_k(\tau) = \sum \sigma_i b_i(\tau) \in \text{conv}\{\sigma_i\}, \tag{19}$$

where $\sigma_i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \beta_i - \mu_i$.

We define $\overline{\sigma}$ as the centroid of the control points $\{\sigma_i\}$ and $\mathcal{B}_i$ as a ball centered at the origin with radius $d + \|\overline{\sigma} - \sigma_i\|_2$. We enforce the linear constraint

$$\forall i: \ \overline{\sigma} \notin \mathcal{B}_i, \tag{20}$$

which is a sufficient condition for the satisfaction of the nonlinear constraint (18) for all $\tau \in [\underline{\tau}, \overline{\tau}]$, that is

$$\forall i: \ \overline{\sigma} \notin \mathcal{B}_i \implies \forall k: \ \|c^+(\tau) - m_k(\tau)\|_2 \geq d. \tag{21}$$

E. Constraint on capturability

In order to impose 0-step capturability at the end of the horizon, we conduct the last position of the Capture Point ξ_N to the interior of the support polygon of the robot

$$\xi_N(\tau) - s_J \in \mathcal{P}. \tag{22}$$

Given that $\xi^+(\tau) = \begin{bmatrix} 1 & \sqrt{h/g} & 0 \end{bmatrix} x^+(\tau)$ we can write, once again, the quadratic polynomial

$$\xi_N(\tau) - s_J = \sum \psi_i b_i(\tau) \in \text{conv}\{\psi_i\}, \tag{23}$$

where $\psi_i = \begin{bmatrix} 1 & \sqrt{h/g} & 0 \end{bmatrix} \beta_i - s_J$, and check the linear constraint

$$\forall i: \ \psi_i \in \mathcal{P}, \tag{24}$$

which is sufficient to ensure the 0-step capturability of the robot for all $\tau \in [\underline{\tau}, \overline{\tau}]$

$$\forall i: \ \psi_i \in \mathcal{P} \implies \xi_N(\tau) - s_J \in \mathcal{P}. \tag{25}$$

F. Adapting the positions of the steps as a function of their duration

Constraints (10), (14) and (22) are not always possible to satisfy with a single choice of s_j for all values of τ. In particular, constraint (10) is the most restrictive as can be easily seen in Figure 2: different trajectories of the CoP for different τ do not fit all inside a single sequence of steps.

We propose to vary the positions of the steps of the robot as a linear function of τ:

$$(s_1(\tau), \ldots, s_J(\tau)). \tag{26}$$

To do so, we choose to relate the positions of the steps to the positions of the CoP by defining $s_j(\tau)$ as the centroid of $P_j(\tau)$:

$$s_j(\tau) = \frac{1}{N_j} \sum P_j(\tau) + s'_j, \tag{27}$$

where $P_j(\tau)$ is the set of all future positions of the CoP belonging to the j^{th} future step and N_j is the cardinality of this set. The free parameter s'_j adds flexibility in the choice of the position of the footstep.

By doing this, constraint (8) is now a polynomial of τ (and, therefore, of the step duration) and the polynomials (11), (15), (23) change to

$$\begin{aligned} s_{j+1}(\tau) - s_j(\tau) &= \sum \gamma_i b_i(\tau), \\ p^+(\tau) - s_j(\tau) &= \sum \lambda'_i b_i(\tau), \\ c^+(\tau) - s_j(\tau) &= \sum \pi'_i b_i(\tau), \\ \xi_N(\tau) - s_J(\tau) &= \sum \psi'_i b_i(\tau). \end{aligned} \tag{28}$$

Consequently, their respective sufficient conditions for feasibility become

$$\forall i: \ \gamma_i \in \mathcal{S}_j, \quad \lambda'_i \in \mathcal{P}, \quad \pi'_i \in \mathcal{C}_j, \quad \psi'_i \in \mathcal{P}. \tag{29}$$

IV. Model Predictive Control for walking

Having defined robust constraints with respect to an interval of τ, we can safely choose any value $\tau^* \in [\underline{\tau}, \overline{\tau}]$ that minimizes a given cost function. The desired behaviour of our robot, reflected in this cost function, is to: 1) keep each person in the crowd at a comfortable distance, *i.e.* no less than d_{ref} from the CoM, 2) move the CoM at a reference velocity $\dot{c}_{ref}$, 3) keep the CoP at the center of the foot for improved balance, 4) minimize the jerk of the CoM for smoothness of its motion and regularization, 5) penalize step durations different from a given reference value $8\tau_{ref}$ where energy expenditure is optimal and 6) stop the robot at the end of the horizon to ensure capturability. In summary, we consider the following objectives:

$$\begin{aligned} o_1 &: \|c^+(\tau) - m_k(\tau)\|_2 \geq d_{ref} > d \quad \forall k, \\ o_2 &: \dot{c}^+(\tau) &&= \dot{c}_{ref}, \\ o_3 &: p^+(\tau) - s_j(\tau) &&= 0, \\ o_4 &: \ddot{c}(\tau) &&= 0, \\ o_5 &: \tau &&= \tau_{ref}, \\ o_6 &: p_N(\tau) &&= \xi_N(\tau), \end{aligned} \tag{30}$$

altogether as $f(\ddot{c}^+, \tau, s'_j) \leq q$ in the following OCP over a horizon of length N:

$$\begin{aligned} \underset{\ddot{c}^+, \tau, s'_j, v}{\text{minimize}} \ & \|v\|_2^2 \\ & f(\ddot{c}^+, \tau, s'_j) + v \leq q \\ \text{subject to} \ & (20), \ (29) \\ & \tau \in [\underline{\tau}, \overline{\tau}]. \end{aligned} \tag{31}$$

The function f is nonlinear with respect to τ so we apply a Newton method and consider successive linearizations. After each iteration we can adapt, for the next iteration, the interval $[\underline{\tau}, \overline{\tau}]$ based on the previous optimal τ^*:

$$\underline{\tau} = \tau^* - \frac{\Delta\tau}{2}, \quad \overline{\tau} = \tau^* + \frac{\Delta\tau}{2}. \tag{32}$$

We propose to keep $\Delta\tau$ constant.

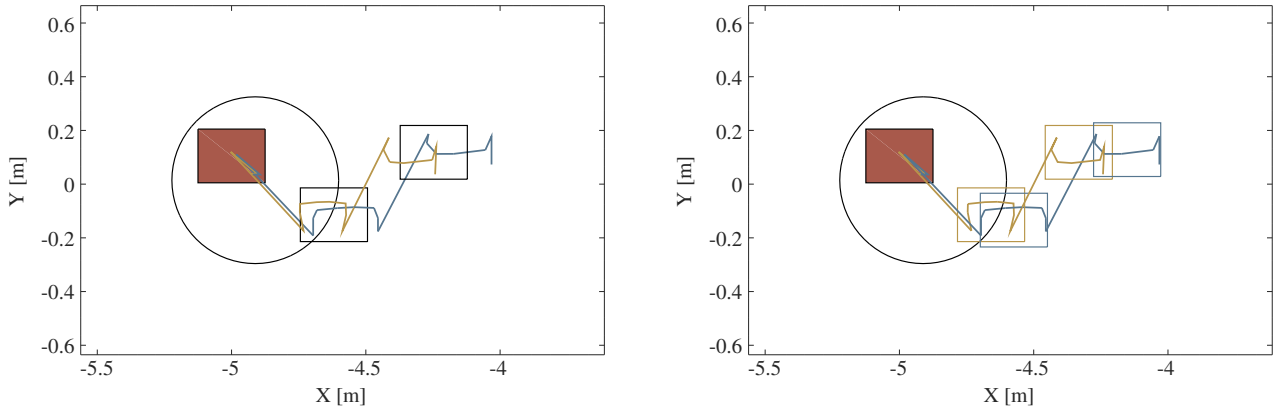


Fig. 2: Future trajectories of the CoP for $\tau = \underline{\tau}$ (in yellow) and $\tau = \overline{\tau}$ (in blue). Figures show future steps that do not vary with τ (on the left) and that vary with τ (on the right). The circle represents the body of the robot, the filled rectangle is the current step, the unfilled rectangles are the previewed future steps.

TABLE I: Parameters of biped robot

Parameter	Symbol	Value	Unit
CoM height (9)	h	0.80	m
Feet dim. (10)	$\mathcal{P}$	0.25×0.125	m $\times$ m
Leg stride (14)	$\mathcal{C}_j$	0.30×0.30	m $\times$ m
Feet separation (8)	$\mathcal{S}_j$	0.21	m
Safety dist. (18)	d	1	m
Radius of FoV	R	6	m
Horizon length	N	18	samples
Double supp. dur.	τ	-	s
Single supp. dur.	7τ	-	s
Ref. step dur. (30)	$8\tau_{ref}$	0.8	s
Comfort dist. (30)	d_{ref}	3	m
Ref. vel. (30)	$\dot{c}_{ref}$	$(0.5,\ 0)$	$(\text{m.s}^{-1}, \text{m.s}^{-1})$

TABLE II: Parameters of scenario A

Parameter	Symbol	Value	Unit
Size of crowd	M	16	persons
Vel. of the crowd	$\dot{m}_k^x$	-1.5	m.s^{-1}
	$\dot{m}_k^y$	$\mathcal{N}(0, 0.4)$	m.s^{-1}
Length of interval of step duration	$\Delta\tau$	5	ms

V. Simulation Results

We evaluate the performance of our controller by simulating a robot and a crowd walking in opposite directions. As in [4], we assume that: people walk at constant velocities, they do not try to avoid the robot and collisions among people are disregarded. The parameters of the robot are specified in Table I.

A. Typical behaviour

Figure 3 shows the results obtained from one illustrative simulation. The parameters of the controller and the crowd are specified in Table II. During the first 2[s] there are no obstacles present in the Field of View (FoV). The robot walks freely and tries to reach its reference velocity $\dot{c}_{ref}$ making steps of nominal duration (0.8[s]).

TABLE III: Parameters of scenario B

Parameter	Symbol	Value	Unit
Size of crowd	M	16	persons
Vel. of the crowd	$\dot{m}_k^x$	{-1, -1.25, -1.5}	m.s^{-1}
	$\dot{m}_k^y$	$\mathcal{N}(0, 0.2)$	m.s^{-1}
Length of interval step duration	$\Delta\tau$	$\{0, 0.5, 1, 2, 4, 8\}$	ms

Between 2[s] and 16[s] the robot encounters people walking in opposite direction and moves backwards to avoid them. At first (between 2[s] and 3[s]) the robot only makes larger steps while maintaining its nominal step duration. However, its maximal speed under such conditions is only about -0.75m.s^{-1} whereas people in the crowd walk at -1.5m.s^{-1}. Consequently, it is forced (between 3[s] and 11[s]) to also make faster steps to be able to avoid collisions.

At 11[s] the robot resumes its forward walking motion once it has overcome most people in its way. It has to stop occasionally a couple of times to walk around the last members of the crowd. From 16[s] until 20[s] the robot continues walking uninterrupted at its objective velocity.

The adaptation of $[\underline{\tau}, \overline{\tau}]$ allows the robot to avoid infeasibilities due to difficult collision-avoidance situations. The key here is objective o_1: it makes the cost function take into account potential collisions in the future so that the optimal solution $(\ddot{c}^*, \tau^*, s_j^*)$ moves the system away from infeasibilities.

B. Effect of the length of the interval of τ

We investigate the effect of $\Delta\tau$ on the ability of the robot to avoid collisions with the crowd. We make 900 simulations

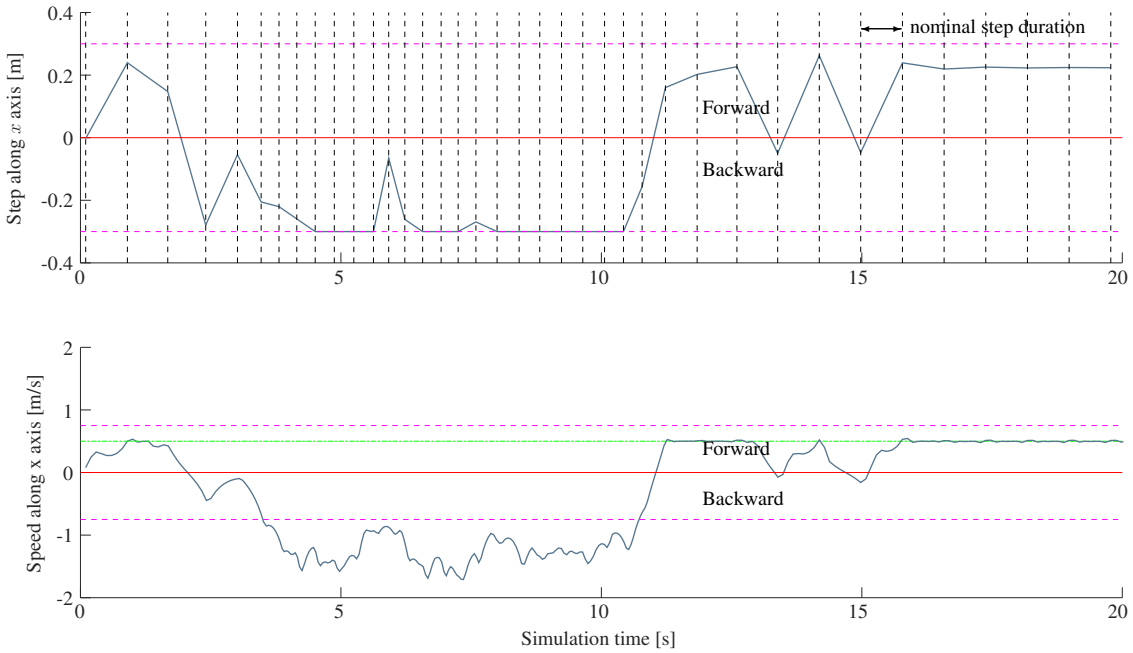


Fig. 3: Adapting the step duration while walking in a crowd. The top subfigure shows in solid blue the x component of the steps $(s_j^x - c^x)$ made by the robot during the simulation. Values above and below the red line indicate forward and backward motion, respectively. The vertical dashed lines mark the instants at which steps were made. The magenta dashed lines indicate the maximal step the robot can make. The bottom subfigure shows the x component of the velocity of the robot $(\dot{c}^x)$ during the simulation. The magenta dashed lines indicate the maximal speed the robot can reach at a nominal step duration of 0.8[s]. The green dash-dotted line indicates the objective speed.

of the robot walking in randomly generated crowds according to the parameters shown in Table III. For each specified crowd speed $\dot{m}_k^x$ we generate and store 50 different crowds that only differ in the initial positions $\{m_k\}$ and speeds $\{\dot{m}_k^y\}$ of each person. Their initial positions vary uniformly over an area of $10 \times 8[\text{m}^2]$ while their speeds $\{\dot{m}_k^y\}$ follow a normal distribution $\mathcal{N}(0, 0.2)$. During simulations, we test the performance of the controller for each $\Delta\tau$ against each of the 50 crowds generated for $\dot{m}_k^x$. Simulations last 20[s] or until a collision occurs.

A longer interval leads to a more constrained OCP that is more likely to become infeasible, which is what is shown in the left plot of Figure 4. For $\Delta\tau \in \{0.5, 1\}$[ms] this issue occurs less than 2% of the time and for $\Delta\tau = 2$[ms] this rises to 9%. When the OCP becomes infeasible we can continue executing the last computed trajectory until the OCP becomes feasible again. If this does not happen, the last computed trajectory ensures that the robot will come to a stop before any collision happens, as in [4].

The plot on the right of Figure 4 shows the proportion of such outcomes. No collisions occur when the speed of the crowd is less than $1.5[\text{m.s}^{-1}]$ for any of the intervals we proposed. When the crowd reaches this speed collisions do occur: 1) with $\Delta\tau \in \{0.5, 1\}$[ms] because people move faster than what the robot can adapt its step duration; 2) with $\Delta\tau = 8$[ms], due to the high number of consecutive infeasible OCPs that make the robot stop in the middle of the walking crowd.

A choice of $\Delta\tau = 1$[ms] appears, therefore, to be the best option. The improvement in collision avoidance we obtain over a controller with fixed $\tau = 0.1[s]$, shown in Figure 4 as $\Delta\tau = 0$, is clear for the speeds of the crowd we considered.

C. Computational efficiency

The OCP (31) contains 41 decision variables. The number of constraints is proportional to the number of obstacles in the FoV (less or equal to M). This number is multiplied by three due of the robustness we introduce using the Bernstein basis. In our simulations the total number of constraints oscillates between 500 and 900. A single iteration of this problem is solved on average in 0.29[ms] on a laptop with a 3[GHz] Intel Core i7 CPU.

VI. Conclusions

We presented a method to adapt the duration of the steps of the robot to improve its ability to avoid collisions when walking in a crowd. It is based on varying τ in a continuous manner while guaranteeing the satisfaction of nonlinear constraints at every Newton iteration. To do so, we identified bounds on the state of system for a given interval of τ and we made sure that the bounds satisfied all constraints. Solving a single carefully constructed Quadratic Program was sufficient to obtain a feasible solution. We reduced significantly the collision rate of the robot from 75% to less than 6%.

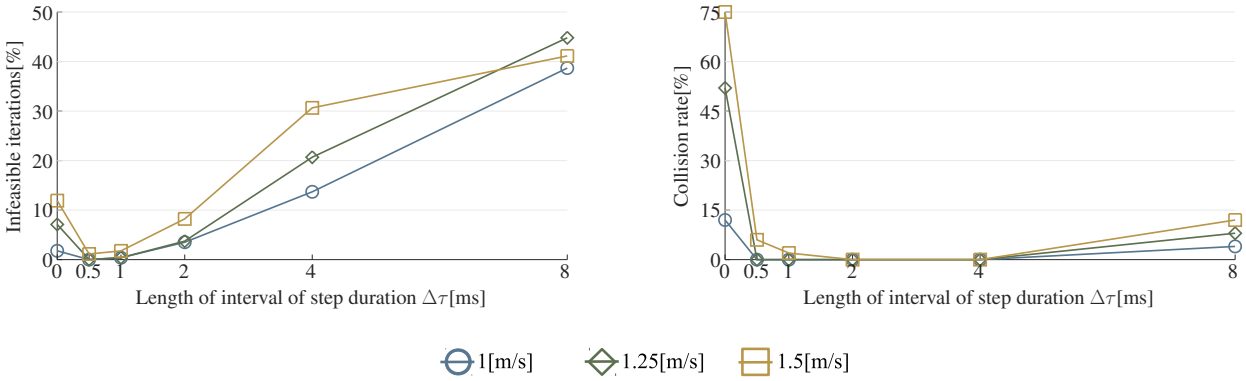


Fig. 4: Infeasible iterations and collision rate for different lengths of interval of τ.

In the future, we would like to apply this method to locomotion problems with multiple contacts where timing is a critical issue.

APPENDIX

Given a quadratic polynomial $x(\tau)$ for $\tau \in [\underline{\tau}, \overline{\tau}]$ with coordinates $\alpha = (\alpha_0, \alpha_1, \alpha_2)$ in the monomial basis $\{1, \tau, \tau^2\}$, we first find the equivalent coordinates in the normalized monomial basis $\{1, \tau', \tau'^2\}$, for $\tau' \in [0, 1]$, by defining $\tau = \underline{\tau} + (\overline{\tau} - \underline{\tau})\tau'$ and solving for all τ'

$$\alpha_0 + \alpha_1\tau + \alpha_2\tau^2 = \alpha'_0 + \alpha'_1\tau' + \alpha'_2\tau'^2,$$

to obtain

$$\begin{bmatrix}\alpha'_0\\ \alpha'_1\\ \alpha'_2\end{bmatrix} = \begin{bmatrix}1 & \underline{\tau} & \underline{\tau}^2\\ 0 & \overline{\tau}-\underline{\tau} & 2(\overline{\tau}-\underline{\tau})\underline{\tau}\\ 0 & 0 & (\overline{\tau}-\underline{\tau})^2\end{bmatrix}\begin{bmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\end{bmatrix}.$$

Then we find its coordinates $\beta = (\beta_0, \beta_1, \beta_2)$ in the Bernstein basis $\{(1-\tau')^2,\ 2\tau'(1-\tau'),\ \tau'^2\}$ by solving, once again, for all τ'

$$\alpha'_0 + \alpha'_1\tau' + \alpha'\tau'^2 = \beta_0(1-\tau')^2 + \beta_1 2\tau'(1-\tau') + \beta_2\tau'^2,$$

to obtain:

$$\begin{bmatrix}\beta_0\\ \beta_1\\ \beta_2\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 1 & 1/2 & 0\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}\alpha'_0\\ \alpha'_1\\ \alpha'_2\end{bmatrix}.$$

ACKNOWLEDGMENTS

This work has been funded by the PSPC Romeo 2 project and EU H2020 Comanoid Research and Innovation Action (RIA).

REFERENCES

[1] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[2] H. Diedam, D. Dimitrov, P. B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 1121–1126.

[3] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-dimensional bipedal walking control based on divergent component of motion," *Robotics, IEEE Transactions on*, vol. 31, no. 2, pp. 355–368, April 2015.

[4] N. Bohórquez, A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Safe navigation strategies for a biped robot walking in a crowd," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. Cancun, Mexico: IEEE, Nov. 2016, pp. 379–386.

[5] Z. Aftab, T. Robert, and P.-B. Wieber, "Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, nov 2012.

[6] P. Kryczka, P. Kormushev, N. G. Tsagarakis, and D. G. Caldwell, "Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2015.

[7] A. Ibanez, P. Bidaud, and V. Padois, "Emergence of humanoid walking behaviors from mixed-integer model predictive control," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014.

[8] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Step timing adjustment: A step toward generating robust gaits," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, nov 2016.

[9] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. E. Pratt, "Walking stabilization using step timing and location adjustment on the humanoid robot, atlas," *CoRR*, vol. abs/1703.00477, 2017.

[10] S. Caron and A. Kheddar, "Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-base Inverted Pendulum," Mar. 2017, working paper or preprint.

[11] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.-B. Wieber, "A robust linear MPC approach to online generation of 3d biped walking motion," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. Institute of Electrical & Electronics Engineers (IEEE), nov 2015.

[12] D. Serra, C. Brasseur, A. Sherikov, D. Dimitrov, and P.-B. Wieber, "A Newton method with always feasible iterates for Nonlinear Model Predictive Control of walking in a multi-contact situation," in *IEEE-RAS 2016 - International Conference on Humanoid Robots (Humanoids)*. Cancun, Mexico: IEEE, Nov. 2016, pp. 932–937.

[13] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, Sept 2003, pp. 1620–1626 vol.2.

[14] G. Lorentz, *Bernstein polynomials*, ser. Mathematical expositions. University of Toronto Press, 1953.