2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada
pp. 5245-5250

MPC-based Humanoid Pursuit-Evasion in the presence of Obstacles

Daniele De Simone, Nicola Scianca, Paolo Ferrari, Leonardo Lanari, Giuseppe Oriolo

***Abstract*— We consider a pursuit-evasion problem between humanoids in the presence of obstacles. In our scenario, the pursuer enters the safety area of the evader headed for collision, while the latter executes a fast evasive motion. Control schemes are designed for both the pursuer and the evader. They are structurally identical, although the objectives are different: the pursuer tries to align its direction of motion with the line-of-sight to the evader, whereas the evader tries to move in a direction orthogonal to the line-of-sight to the pursuer. At the core of the control architecture is a Model Predictive Control scheme for generating a stable gait. This allows for the inclusion of workspace obstacles, which we take into account at two levels: during the determination of the footsteps orientation and as an explicit MPC constraint. We illustrate the results with simulations on NAO humanoids.**

I. Introduction

Research activity on humanoid robots has grown remarkably in the last decade, together with their envisaged use in a number of applications. Many of these require that human and robotic co-workers share the same environment; for example, the objective of the EU H2020 research project COMANOID[1] is to foster the deployment of humanoid robots in aeronautical shopfloors. It becomes then imperative to investigate the safety-related issues arising from human-robot coexistence.

A fundamental safety layer in a robot is certainly detection and avoidance of obstacles. This is a classical problem in robotics since Khatib's pioneering work [1], and a plethora of methods for fixed-base manipulators or mobile robots can be found in the literature; e.g., see [2], [3] and the references therein. Recently, researchers interested in safe human-robot coexistence and interaction have also started looking at this issue [4], [5].

While the basic issues are the same, the design of safety layers for humanoids must account for the peculiar characteristics of these robots: that is, the fact that their base can only be displaced through steps and that balance must be maintained at all times during motion (see, e.g., [6]). Among the early works showing a humanoid avoiding dynamic obstacles was [7], where real-time vision and replanning were used for autonomous navigation with ASIMO; more recent results include [8] and, based on Model Predictive Control (MPC) techniques, [9], [10].

In a previous paper [11], we have studied a basic safety problem for humanoids. In particular, a situation was considered where the robot was threatened by a moving obstacle (e.g., a human, or another robot) entering its safety area headed for collision. Under the assumption that the moving obstacle did not change its direction, we developed a method by which the humanoid could plan and execute in real time an evasion maneuver. In [12] we removed that assumption, and considered a worst-case scenario where the obstacle is actively chasing the humanoid. This led us to replace the moving obstacle with another humanoid, and to consider therefore a *pursuit-evasion* problem with humanoids.

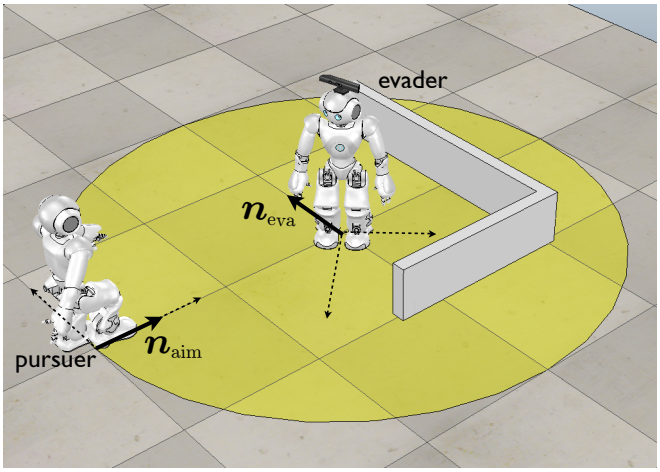

Fig. 1. Pursuit-evasion with humanoids. The pursuer enters the safety area of the evader and heads towards it. The latter must plan and execute a fast evasive motion. Each robot must continuously replan its motion on the basis of the other's and, at the same time, avoid workspace obstacles. Note the moving frame associated to each humanoid.

Pursuit-evasion is a long-standing topic in mobile robotics and has been investigated from different viewpoints, see [13] for a recent review. Our perspective is to consider a coupled dynamic system consisting of two identical humanoids with equivalent control schemes but different objectives: the pursuer tries to align with the line-of-sight to the evader, whereas the latter attempts to move away from the line-of-sight to the pursuer, e.g., in a direction orthogonal to it.

In this work, we extend the work in [12] by considering the presence of obstacles in the environment where pursuit-evasion takes places. This obviously complicates an already difficult problem, because each robot must perform its own main task and, at the same time, avoid nearby obstacles. The simple inclusion of obstacle reaction behaviors, such as those generated by an artificial potential field, is not an effective solution because local minima are very easily generated by the combined action of the main motion intention and the

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy. E-mail: *lastname*@diag.uniroma1.it. This work is supported by the EU H2020 project COMANOID.

[1] www.comanoid.eu

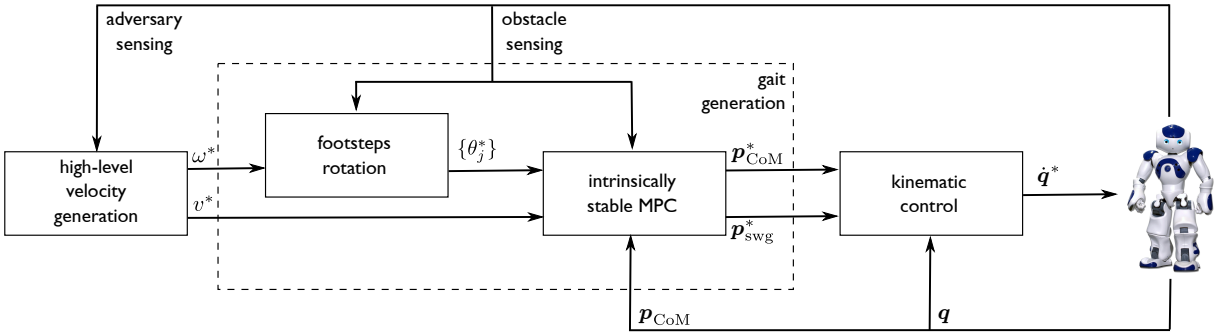

Fig. 2. A block scheme of the control architecture for humanoid pursuit-evasion in the presence of obstacles, valid for both the pursuer and the evader.

obstacle reaction itself. Therefore, we chose a completely different approach.

The proposed control scheme consists of two stages, which decouple the pursuit-evasion task from the obstacle avoidance behavior. The first uses unicycle template models to produce high-level reference velocities for pursuit-evasion. The second stage generates a humanoid gait that tracks these reference velocities and, at the same time, takes care of obstacle avoidance. To this end, it uses an adaptation of our intrinsically stable MPC framework presented [14]. In addition to realizing a fast replanning scheme with strong robustness properties, this allows to take into account workspace obstacles at two levels: during the determination of the footsteps orientation and as an explicit MPC constraint.

The paper is organized as follows. Section II formulates the considered problem and outlines our solution approach. Section III presents the module in charge of generating high-level velocity commands, while Sect. IV describes MPC-based gait generation in the presence of obstacles scheme. Simulation results on NAO robots are presented in Sect. V. Some future work is mentioned in the concluding section.

II. Formulation and Approach

The situation of interest is shown in Fig. 1. There are two humanoid robots moving in an environment containing fixed obstacles. One of the robots acts as a pursuer and the other as an evader. The pursuer is always aware of the presence of the evader, and tries to intercept it. The evader detects the pursuer when this enters its safety area, triggering the execution of an evasive maneuver.

Each robot performs computations in its own moving frame consisting of the sagittal and the coronal axes, and only uses local information made available by its own sensory system. We make the following assumptions:

- A1 The evader is not performing any particular task, or it is ready to abort it immediately.
- A2 Each robot can determine (and measure the orientation of) the line-of-sight to the other. The evader can also measure the distance to the pursuer in order to detect intrusions in its safety area.
- A3 Each robot can detect nearby obstacles. In particular, each robot can measure the position of the closest obstacle point relative to itself (expressed, e.g., as range and bearing in the robot frame), provided that this point falls within a certain field of view that depends on the specific sensing equipment.

In our framework, both the pursuer and the evader are controlled in a purely reactive mode; that is, there is no anticipative action based on an estimate of the other robot's intention of motion. At any instant, the pursuer determines the line-of-sight to the evader, represented by the unit vector $\boldsymbol{n}_{\mathrm{aim}}$, and steers its course so as to align with $\boldsymbol{n}_{\mathrm{aim}}$. The evader determines the line-of-sight to the pursuer, represented[2] by $-\boldsymbol{n}_{\mathrm{aim}}$, computes from this an evasion direction $\boldsymbol{n}_{\mathrm{eva}}$, and steers its course so as to align with $\boldsymbol{n}_{\mathrm{eva}}$. Both robots must take into account the presence of nearby obstacles and deform their trajectories accordingly.

However, a fundamental issue related to the nature of humanoid motion must be taken into account: a pure feedback scheme cannot be used, because the problem of gait generation must be addressed. The proposed solution is to use feedback control to generate high-level motion commands implementing the pursuit-evasion behavior, and Model Predictive Control (MPC) for generating a stable gait that tracks the high-level commands and, at the same time, avoids workspace obstacles.

In particular, consider the control architecture schematized in Fig. 2, which applies to both the pursuer and the evader robots. The whole process is driven by exteroceptive sensory information, which is conceptually twofold: *adversary sensing* and *obstacle sensing*. The first is information about the other robot (the evader for the pursuer, and vice versa), in the form made precise in Assumption A2. The second concerns nearby obstacles, and is made available as described in Assumption A3.

Feedback control based on adversary sensing is used to compute high-level velocity commands for the robot: since we use unicycles as template models, these commands are encoded in a driving velocity v^* and a steering velocity ω^*. The commands are then sent to a gait generation module

[2]Although the direction of the two lines-of-sight is the same, each robot will obviously obtain and express the corresponding measurement in its own moving frame.

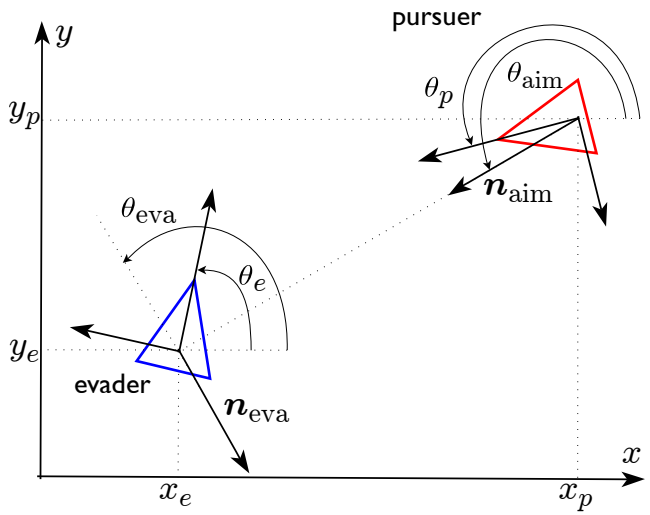

Fig. 3. Pursuit-evasion with unicycles: geometry and relevant symbols

that performs two basic actions. First, it computes footsteps orientations that are consistent with v^*, ω^* and, at the same time, with the avoidance of workspace obstacles, if any. Then, once these orientations have been determined, it uses an MPC framework for generating an associated stable gait for the humanoid, expressed as reference trajectories $\boldsymbol{p}^*_{\text{CoM}}$, $\boldsymbol{p}^*_{\text{swg}}$ for the CoM and swinging foot, respectively. Finally, standard pseudoinverse-based kinematic control is used to compute reference joint velocities $\dot{\boldsymbol{q}}^*$ which will be sent to the robot actuators. Note that proprioceptive feedback is used for both MPC and kinematic control.

The above architecture runs in real-time: the velocity commands are updated at the exteroceptive sensors' rate, and MPC-based gait generation works with a very short planning horizon thanks to the inherent stability guaranteed by our MPC method [14].

In the next sections we describe separately the modules for high-level velocity generation and gait generation. For the sake of compactness, in the following we will omit all the asterisk superscripts in the reference signals.

III. High-level velocity generation

High-level velocity generation is based on the use of unicycles as template models. We summarize here the main points of this procedure and refer the reader to [12] for further details on pursuit-evasion with unicycles.

Figure 3 shows two unicycle robots, one of which acts as a pursuer and the other as an evader. Refer to the figure for the definition of the relevant symbols. Both the pursuer and the evader obey the same model equations:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega, \end{aligned}$$

where all variables take either the p (pursuer) or e (evader) subscript. Both unicycles are controlled by a feedback law aimed at aligning their motion with a desired orientation θ_{des}:

$$v = \pm\bar{v} \quad (1)$$

$$\omega = k\,(\theta_{\text{des}} - \theta), \quad (2)$$

where $\bar{v} > 0$ and k is a positive gain. In particular, the pursuer takes the positive determination of v and $\theta_{\text{des}} = \theta_{\text{aim}}$; while the evader takes the negative determination of v and $\theta_{\text{des}} = \theta_{\text{eva}} = \theta_{\text{aim}} - \pi/2$. Note the following facts.

- The driving velocity of the evader is chosen to be equal in magnitude to that of the pursuer to consider a situation where neither robot has an advantage. However, its sign is opposite because the evader moves backwards in order to keep the pursuer in its field of view.
- The above definition of θ_{eva} encodes a *move aside* evasion strategy [11] which is expected to be effective in confined space. Note that while $\theta_{\text{aim}} = \angle\boldsymbol{n}_{\text{aim}}$, we have $\theta_{\text{eva}} = \angle\boldsymbol{n}_{\text{eva}} - \pi$ in view of the fact that the evader is moving backwards.

Simulations in [12] have shown that in the absence of obstacles the pursuit-evasion unicycle system exhibits an interesting asymptotic behavior: the two robots converge to the same circular limit cycle along which they travel at the same speed, with a relative orientation of $\pi/2$.

IV. Gait generation

According to the block scheme of Fig. 2, the gait generation module takes as input the high-level velocity commands (1–2) produced by the previous module and computes a suitable humanoid gait using an MPC framework. To this end, two sequential steps are performed. In the first, footstep orientations are computed over a time horizon by solving a quadratic optimization problem. In the second, an adaptation of the intrinsically stable linear MPC algorithm proposed in [14] is used to compute reference trajectories for the CoM and the swinging foot over the same horizon.

Workspace obstacles in the vicinity of the robot are taken into account in both steps of gait generation. During footsteps rotation, the cost criterion used for setting up the optimization problem includes a term aimed at steering the robot away from the closest obstacle. In the MPC module, explicit constraints are introduced that prevent the humanoid from placing footsteps within the obstacle. We will show that this dual strategy provides an adequate level of safety.

A. Footsteps rotation

Choosing the orientation of the footsteps before computing the solution of the MPC problem is aimed at maintaining the linearity of the latter.We denote by T_{h} the prediction horizon, by T_s the constant duration of the steps, and by $M = \text{ceil}(T_h/T_s)$ the number of footsteps to be planned within T_{h}. The goal is then to compute the orientations $\theta_1, \ldots, \theta_M$ of these footsteps w.r.t. the robot sagittal axis.

Assume that the robot detects a workspace obstacle in the vicinity (see Fig. 4, top). According to Assumption 3, both θ_{obs} and d will be measured by the robot sensors. The first is used to compute $\theta_{\text{avo}} = \theta_{\text{obs}} \pm \pi/2$, the orientations of the tangent half-lines originating at the closest obstacle point.

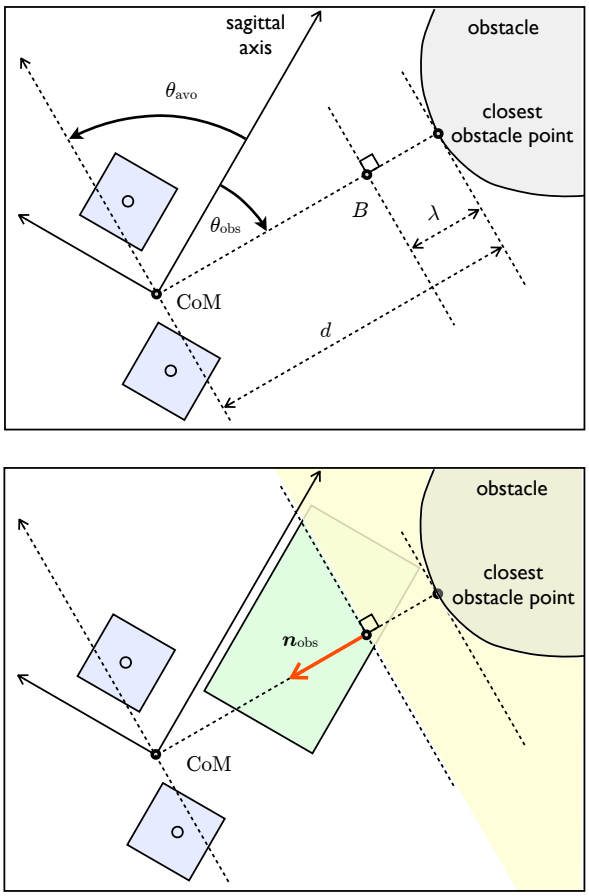

Fig. 4. Taking into account the closest obstacle during gait generation. The current robot placement is defined by its CoM: current footstep locations are also shown (light blue). Top: definition of the relevant quantities. Bottom: kinematically feasible zone (green) and forbidden zone (yellow) for the next footstep as defined in the MPC.

The footstep orientations are obtained by minimizing the quadratic cost function

$$\sum_{j=1}^{M} \left(\left(\frac{\theta_j - \theta_{j-1}}{T_s} - \omega \right)^2 + k_{\text{obs}} \frac{w(\theta_{\text{obs}})}{d^2} (\theta_j - \theta_{\text{avo}})^2 \right) \tag{3}$$

under the linear constraint $|\theta_j - \theta_{j-1}| \leq \theta_{\max}$, where $\theta_{\max}$ is the bound on the relative orientation between two consecutive footsteps.

Function (3) combines two different objectives. The first term simply rewards the reproduction over T_h of the angular velocity ω. The second term forces the robot to align its footsteps with the tangent to the closest obstacle point; in particular, the sign of the $\pi/2$ offset in the definition of θ_{avo} is chosen according to the sign of θ_{obs}. Note that the second term is modulated through a scaling factor k_{obs} by a weight function $w(\theta_{\text{obs}})$ and the inverse of the squared distance.

Figure 5 shows the different definitions of function w for the pursuer and the evader. The idea here is that the pursuer moves forward, and therefore only obstacles lying in its front half-plane should be considered; and vice versa for the evader.

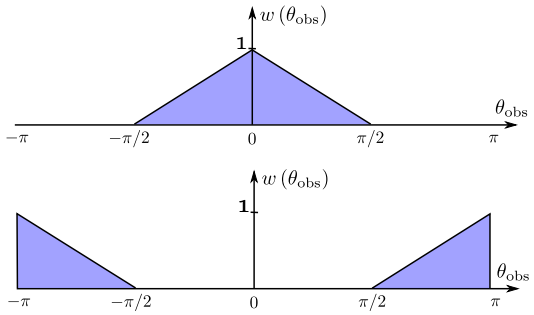

Fig. 5. The weight functions $w(\theta_{\text{obs}})$ for the pursuer (above) and the evader (below).

Note that the quantities ω, $w(\theta_{\text{obs}})$ and d are assumed to be constant at their initial value throughout the time horizon. This is acceptable in view of the high rate at which the MPC scheme runs.

B. Intrinsically stable MPC

The second and main stage of gait generation is based on an adaptation of the intrinsically stable MPC scheme proposed in [14], which uses as a motion model the Linear Inverted Pendulum with a dynamic extension. As a consequence, the dynamics along the sagittal (x) and coronal (y) axes are decoupled and identical.

Let us consider the x coordinate for illustration. The motion model is

$$\begin{pmatrix} \dot{x}_{\text{c}} \\ \ddot{x}_{\text{c}} \\ \dot{x}_{\text{z}} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \eta^2 & 0 & -\eta^2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{\text{c}} \\ \dot{x}_{\text{c}} \\ x_{\text{z}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{x}_{\text{z}} \tag{4}$$

where x_{c} and x_{z} are the x coordinates respectively of the CoM and the ZMP, $\eta = \sqrt{g/h}$, and h is the constant height of the CoM. We assume piecewise-constant control $\dot{x}_{\text{z}}$ over time intervals of duration δ, with the prediction horizon $T_h = N\delta$. Below, $\varphi^i = \varphi(t_i)$ denotes the value of a generic function φ sampled at $t_i = i\delta$.

In our MPC scheme, the decision variables are the ZMP velocities $(\dot{x}^i_{\text{z}}, \dot{y}^i_{\text{z}})$, $i = 1, \ldots, N$, and the footstep locations $(x^j_{\text{f}}, y^j_{\text{f}})$, $j = 1, \ldots, M$. The support foot at time t_k is located at $(x^0_{\text{f}}, y^0_{\text{f}})$.

The cost function to be minimized along the prediction horizon is

$$\sum_{i=1}^{N} \Big(\big(\dot{x}^{k+i}_{\text{z}}\big)^2 + k^x_{\text{vel}} \big(\dot{x}^{k+i}_{\text{c}} - v\cos(i\omega\delta)\big)^2 + \big(\dot{y}^{k+i}_{\text{z}}\big)^2 + k^y_{\text{vel}} \big(\dot{y}^{k+i}_{\text{c}} - v\sin(i\omega\delta)\big)^2 \Big) \tag{5}$$

and penalizes inaccurate tracking of the high-level reference speed v. Note that this cost function is independent of the footstep locations, thus making the problem non-strictly convex. However, these influence the QP problem through the constraints introduced to guarantee balance, stability, kinematic feasibility and obstacle avoidance. Below, we briefly illustrate each type of constraint (only one side in case of inequality constraints).

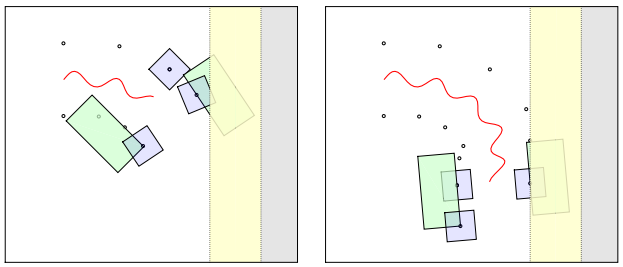

Fig. 6. Avoidance of a wall (shown in gray) with the proposed gait generation method. The CoM trajectory is shown in red.

The bounding box which defines the ZMP constraint is a rectangle with width $x_{\mathrm{z}}^{\max}$ and length $y_{\mathrm{z}}^{\max}$, and therefore the balance constraints are of the form

$$R_j^T \begin{pmatrix} \delta \sum_{l=k+1}^{k+i} \dot{x}_{\mathrm{z}}^l - x_{\mathrm{f}}^j \\ \delta \sum_{l=k+1}^{k+i} \dot{y}_{\mathrm{z}}^l - y_{\mathrm{f}}^j \end{pmatrix} \leq \frac{1}{2} \begin{pmatrix} x_{\mathrm{z}}^{\max} \\ y_{\mathrm{z}}^{\max} \end{pmatrix} - R_j^T \begin{pmatrix} x_{\mathrm{z}}^k \\ y_{\mathrm{z}}^k \end{pmatrix} \tag{6}$$

where R_j is the rotation matrix associated to angle θ_j. The ZMP constraints are only active during single support in order to maintain their linearity.

The stability constraint first introduced in [14] takes the form

$$\frac{1}{\eta} \frac{1 - e^{\delta\eta}}{1 - e^{N\delta\eta}} \sum_{i=1}^{N} e^{i\delta\eta} \dot{x}_{\mathrm{z}}^{k+i} = x_{\mathrm{c}}^k + \frac{\dot{x}_{\mathrm{c}}^k}{\eta} - x_{\mathrm{z}}^k \tag{7}$$

and will guarantee that the computed CoM trajectory is bounded regardless of the choice of T_{h}.

Kinematic feasibility constraints on the footstep locations are expressed as

$$R_{j-1}^T \begin{pmatrix} x_{\mathrm{f}}^j - x_{\mathrm{f}}^{j-1} \\ y_{\mathrm{f}}^j - y_{\mathrm{f}}^{j-1} \end{pmatrix} \leq \pm \begin{pmatrix} 0 \\ L \end{pmatrix} + \frac{1}{2} \begin{pmatrix} x_{\mathrm{f}}^{\max} \\ y_{\mathrm{f}}^{\max} \end{pmatrix} \tag{8}$$

in which $x_{\mathrm{f}}^{\max}$ and $y_{\mathrm{f}}^{\max}$ are the size of the feasibility zone (in green in Fig. 4, bottom), L is a reference distance between two consecutive footsteps, and the sign alternates for the two feet. There are a total of $2M$ constraints, being (8) only one side of the inequality and $j = 1, \ldots, M$.

The last MPC constraint is related to obstacle avoidance. With reference to Fig. 4, top, consider a point $B = (x_B, y_B)$ located along the line connecting the CoM with the closest obstacle point at a safety distance λ from the latter, and draw the normal to the same line through B. The half-plane beyond this line (in yellow in Fig. 4, bottom) is a forbidden zone for the footstep locations. This constraint is easily written as

$$\boldsymbol{n}_{\mathrm{obs}}^T \left\{ \begin{pmatrix} x_{\mathrm{f}}^j \\ y_{\mathrm{f}}^j \end{pmatrix} - \begin{pmatrix} x_B \\ y_B \end{pmatrix} \right\} \geq 0 \tag{9}$$

with $\boldsymbol{n}_{\mathrm{obs}}$ the unit vector defined in Fig. 4.

Solving the Quadratic Programming (QP) problem defined by cost function (5) and constraints (6–9) yields the reference CoM trajectory during the prediction horizon T_h, together with the associated footstep locations. Swing foot trajectories are then generated accordingly. As customary in MPC, only the first sample of the reference trajectories is sent to the robot kinematic controller, and the QP computation is repeated for the next interval.

We conclude this section by giving in Fig. 6 an illustration of how the robot trajectory is deformed by a nearby obstacle. Effective obstacle avoidance is achieved by the combined effect of the second term in (3) on footstep orientations and of constraint (9) on footstep locations.

V. Simulations

The proposed approach was validated in V-REP simulations using two NAO humanoids, one acting as the pursuer and the other as the evader. Both are equipped with standard cameras, which are used for line of-sight determination. The evader can also measure range via an additional RGB-D camera. This allows to detect intrusions into the robot safety area and, in principle, to identify the closest obstacle point in the field of view. For simplicity, however, no obstacle sensing is actually performed, assuming instead that each robot computes d and θ_{obs} from an environment map.

High-level reference velocities are produced by (1–2) with $\bar{v} = 0.1$ m/s and $k = 0.2$. Footsteps rotation is performed using $k_{\mathrm{obs}} = 0.05$ and $\theta_{\max} = \pi/16$. For MPC, we set $k_{\mathrm{vel}}^x = k_{\mathrm{vel}}^x = 10$ in the cost function (5) and used $\delta = 10$ ms over a prediction horizon $T_h = 0.6$ s. The duration of the single and double support phase is fixed at respectively 0.2 s and 0.1 s. For the ZMP bounding box we used $x_{\mathrm{z}}^{\max} = y_{\mathrm{z}}^{\max} = 0.02$ m, and for the feasibility constraints $L = 0.125$ m, $x_{\mathrm{f}}^{\max} = 0.05$ m, $y_{\mathrm{f}}^{\max} = 0.025$ m. Overall, the control scheme runs at 100 Hz and can be implemented on the NAO hardware.

Figure 7 shows pursuit-evasion in an environment containing a single cylindrical obstacle. In this case, the humanoids converge to a limit cycle around the obstacle. The effect of the latter can be appreciated in Fig. 8, where the motion of the robots is shown both in the absence and the presence of the obstacle. Note how the trajectory deformation is smooth thanks to the modulation mechanism with d used in the footsteps rotation module.

In the second simulation, shown in Fig. 9, the environment obstacle is a long wall. At the beginning, the humanoids move as in the previous simulation and tend to align with the wall. In this case, however, the combination of the pursuit-evasion and the obstacle avoidance actions ultimately drives the robots away from the obstacle.

Pursuit-evasion in an environment with several obstacles is simulated in Fig. 10. In spite of the more complex geometry, the humanoids always manage to avoid the obstacles, with the pursuer aggressively chasing the evader but the latter always escaping.

We encourage the reader to watch the simulation clips in the accompanying video to better appreciate the naturality and effectiveness of the generated motions.

VI. Conclusions

We have presented a method for humanoid pursuit-evasion which extends our previous work [12] on the subject by

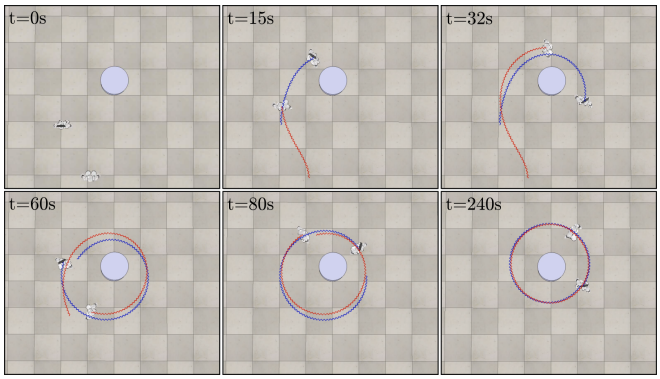

Fig. 7. Simulation 1: Pursuit-evasion in the presence of a cylindrical obstacle. COM trajectories are shown in red (pursuer) and blue (evader).

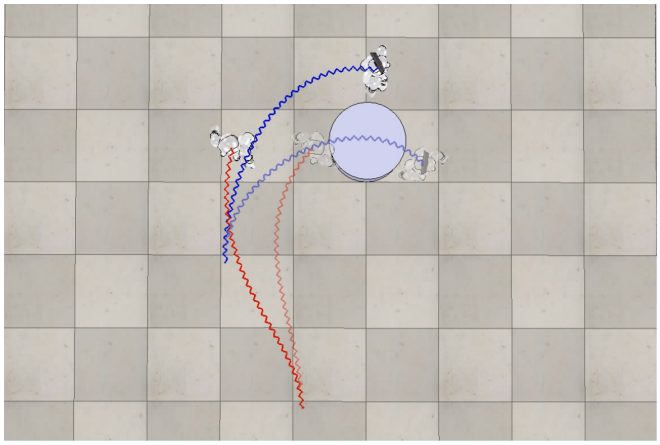

Fig. 8. Simulation 1: Comparison of motions with and without the obstacle.

allowing the presence of obstacles in the environment.

The proposed control scheme consists of two stages: the first uses unicycle template models to produce high-level reference velocities, which are then tracked using an intrinsically stable MPC [14] for gait generation. Workspace obstacles in the vicinity of the robot are taken into account in both steps of gait generation. NAO simulations show that the adoption of an MPC framework results in a fast replanning scheme with good safety and robustness properties.

Future work will address several points, such as implementing the proposed method (including sensing) on actual NAOs, considering more general scenarios (e.g. multiple pursuers), and improving the MPC architecture to combine footstep orientations and position, along with ZMP in a single optimization problem.

References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *1985 IEEE Int. Conf. on Robotics and Automation*, 1985, pp. 500–505.

[2] E. Yoshida and F. Kanehiro, "Reactive robot motion using path replanning and deformation," in *2011 IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 5456–5462.

[3] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, 2012.

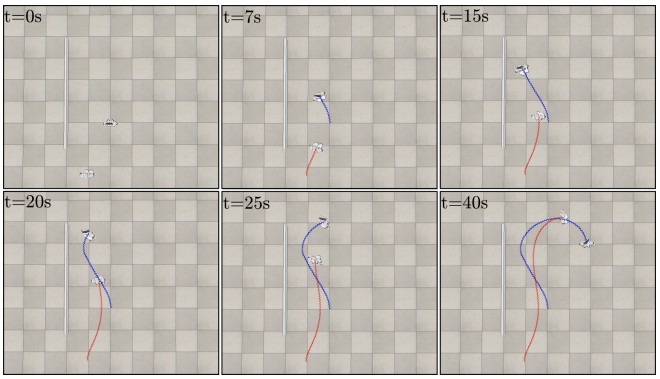

Fig. 9. Simulation 2: Pursuit-evasion in the presence of a wall obstacle. CoM trajectories are shown in red (pursuer) and blue (evader).

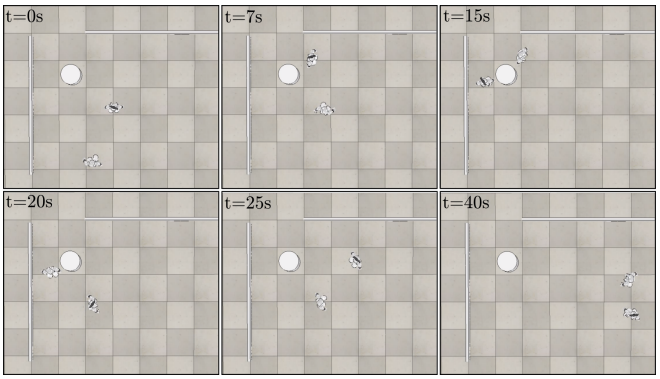

Fig. 10. Simulation 3: Pursuit-evasion in a complex environment.

[4] A. De Luca and F. Flacco, "Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration," in *2012 4th IEEE RAS EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, 2012, pp. 288–295.

[5] B. Lacevic, P. Rocco, and A. Zanchettin, "Safety assessment and control of robotic manipulators using danger field," *IEEE Trans. on Robotics*, vol. 29, no. 5, pp. 1257–1270, 2013.

[6] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*. Springer, 2014.

[7] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, "Vision-guided humanoid footstep planning for dynamic environments," in *2005 IEEE-RAS Int. Conf. on Humanoid Robots*, 2005, pp. 13–18.

[8] J. Chestnutt, "Navigation and gait planning," in *Motion Planning for Humanoid Robots*, K. Harada, E. Yoshida, and K. Yokoi, Eds. Springer, 2010, pp. 1–28.

[9] N. Bohorquez, A. Sherikov, D. Dimitrov, and P. B. Wieber, "Safe navigation strategies for a biped robot walking in a crowd," in *16th IEEE-RAS Int. Conf. on Humanoid Robots*, 2016, pp. 379–386.

[10] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2017.

[11] M. Cognetti, D. De Simone, L. Lanari, and G. Oriolo, "Real-time planning and execution of evasive motions for a humanoid robot," in *2016 IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 4200–4206.

[12] M. Cognetti, D. De Simone, F. Patota, N. Scianca, L. Lanari, and G. Oriolo, "Real-time pursuit evasion with humanoid robots," in *2017 IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 4090–4095.

[13] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, 2011.

[14] N. Scianca, M. Cognetti, D. De Simone, L. Lanari, and G. Oriolo, "Intrinsically stable MPC for humanoid gait generation," in *16th IEEE-RAS Int. Conf. on Humanoid Robots*, 2016, pp. 101–108.