2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Real-Time Pursuit-Evasion with Humanoid Robots

Marco Cognetti, Daniele De Simone, Federico Patota, Nicola Scianca, Leonardo Lanari, Giuseppe Oriolo

***Abstract*— We consider a pursuit-evasion problem between humanoids. 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 scheme is a maneuver planning module which makes use of closed-form expressions exclusively. This allows its use in a replanning framework, where each robot updates its motion plan upon completion of a step to account for the perceived motion of the other. Simulation and experimental results on NAO humanoids reveal an interesting asymptotic behavior which was predicted using unicycle as template models for trajectory generation.**

I. Introduction

Research on humanoid robots has flourished in the last decade and their use is currently envisaged in a number of applications. Since many of these involve the simultaneous presence of humans and humanoids in the same environment, it becomes imperative to investigate the safety-related issues arising from their coexistence. For example, the objective of the EU H2020 research project COMANOID[1] is to foster the deployment of humanoid robots in aeronautical shopfloors shared with human co-workers.

One of the most essential safety layers in a robot is arguably detection and avoidance of obstacles, static or dynamic. This is a long-standing problem in robotics since Khatib's pioneering work [1], and by now the literature includes a number of methods for fixed-base manipulators or mobile robots; e.g., see [2], [3] and the references therein. Recently, researchers have started looking at this issue in the context of safe human-robot coexistence and interaction [4], [5]. Specific criteria for robot navigation in the presence of humans have been investigated in [6].

While the fundamental issues are the same, the design of safety layers for humanoids must take into account the distinctive peculiarities of these robotic systems, namely the fact that their base can only be displaced through stepping gaits and that equilibrium must be maintained at all times during motion (see, e.g., [7]). One of the first works that showcased a humanoid avoiding moving obstacles was [8], where real-time vision and replanning were used for autonomous navigation with ASIMO; more recent results are presented in [9] and, using Model Predictive Control techniques, in [10], [11].

In a previous work [12], we have addressed a basic safety problem for humanoids: in particular, a situation was considered where the robot is threatened by a moving obstacle (e.g., a human, or another robot) that enters its safety area headed for collision. Under the assumption that the moving obstacle did not change its direction, we developed and implemented a method by which the humanoid could plan and execute in real time an evasion maneuver. In this paper, we shall remove that assumption, and consider a worst-case scenario where the obstacle is actively trying to reach and collide with the humanoid. This leads us to replace the moving obstacle with another humanoid, and to consider therefore a full-fledged *pursuit-evasion* problem with humanoids.

Pursuit-evasion is a well-known topic in robotics and has been studied from several perspectives, see [13] for a recent review. Our viewpoint 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.

At the core of the proposed control scheme is a maneuver planning module that implements a sequential procedure. From the desired direction of motion, a corresponding Cartesian trajectory is generated, around which footsteps are placed; from these, a stable trajectory for the humanoid CoM is computed. The whole procedure relies on closed-form expressions, thus making real-time implementation possible and ultimately allowing its use in a replanning framework, where each robot updates its motion plan upon completion of a step to account for the perceived motion of the other.

An interesting outcome of our study is that the pursuer and the evader converge to a circular limit cycle along which they travel at the same speed. This property will be first observed on unicycles, which will be used as template models for Cartesian trajectory generation, and then confirmed on NAO humanoids, both in simulations and experiments.

The paper is organized as follows. In Sect. II, we introduce feedback laws for pursuit-evasion with unicycle robots to be used as template models for real-time trajectory generation. Section III addresses the pursuit-evasion problem with humanoids and describes the control schemes that drive the two robots. In particular, we analyze the maneuver planning modules for both the pursuer and the evader, and discuss their use in a replanning framework. Simulations and experiments are presented in Sect. IV. Possible future work is mentioned in the concluding section.

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

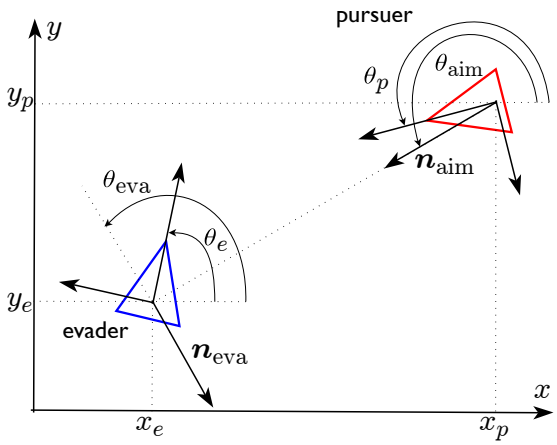

Fig. 1. Pursuit-evasion with unicycles: geometry of the problem

II. Pursuit-Evasion with Unicycles

Our method for pursuit-evasion with humanoids is based on the use of the unicycle[2] as a template model for real-time trajectory generation. We shall therefore discuss first pursuit-evasion with unicycles.

We have two unicycle robots, one of which acts as a pursuer and the other as an evader (see Fig. 1). 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.
- A3 There are no obstacles in the workspace.

In our framework, both the pursuer and the evader are controlled in pure feedback 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 $\boldsymbol{n}_{\text{aim}}$, and steers its course so as to align with $\boldsymbol{n}_{\text{aim}}$. The evader determines the line-of-sight to the pursuer, represented[3] by $-\boldsymbol{n}_{\text{aim}}$, computes from this an evasion direction $\boldsymbol{n}_{\text{eva}}$, and steers its course so as to align with $\boldsymbol{n}_{\text{eva}}$.

Below, we discuss the two robots separately.

A. Pursuer

The pursuer unicycle is represented by

$$\begin{aligned} \dot{x}_p &= v_p \cos\theta_p \\ \dot{y}_p &= v_p \sin\theta_p \\ \dot{\theta}_p &= \omega_p, \end{aligned}$$

where (x_p, y_p, θ_p) is the robot configuration and v_p, ω_p are its driving and steering velocity inputs.

The pursuer moves under the action of the following control law[4]

$$v_p = \bar{v} \quad (1)$$
$$\omega_p = k\,(\theta_{\text{aim}} - \theta_p) \quad (2)$$

where $\bar{v} > 0$, $k > 0$ and $\theta_{\text{aim}} = \angle \boldsymbol{n}_{\text{aim}}$ is the phase angle of $\boldsymbol{n}_{\text{aim}}$. The constant driving velocity $\bar{v}$ sustains a continued pursuing behavior, while the angular velocity forces the robot to align its sagittal axis with $\boldsymbol{n}_{\text{aim}}$, i.e., with the line-of-sight to the evader. Note that the pursuer directly measures the angular error $\theta_{\text{aim}} - \theta_p$, so that no absolute measurements are actually needed.

B. Evader

The equations of the evader unicycle are

$$\begin{aligned} \dot{x}_e &= v_e \cos\theta_e \\ \dot{y}_e &= v_e \sin\theta_e \\ \dot{\theta}_e &= \omega_e, \end{aligned}$$

where (x_e, y_e, θ_e) is the robot configuration and v_e, ω_e are the driving and steering velocity inputs.

The control law for the evader is structurally the same of the pursuer's:

$$v_e = -\bar{v} \quad (3)$$
$$\omega_e = k\,(\theta_{\text{eva}} - \theta_e). \quad (4)$$

where[5] $\theta_{\text{eva}} = \angle \boldsymbol{n}_{\text{eva}} - \pi$, and $\boldsymbol{n}_{\text{eva}}$ is the unit vector representing the chosen direction for evasion. Note the following important points.

- The driving velocity of the evader is chosen to be equal in magnitude to that of the pursuer to consider a fair 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 backward motion is the reason for the presence of a $-\pi$ offset in the definition of θ_{eva}.
- The choice of $\boldsymbol{n}_{\text{eva}}$ encodes the chosen evasion strategy. In [12] we have discussed two possibilities, i.e., *move back* and *move aside*. In this paper we will consider only the second, which is more effective in confined spaces. With this strategy, the evader moves backwards so as to align with a direction that is orthogonal to the line-of-sight to the pursuer. This corresponds to setting $\boldsymbol{n}_{\text{eva}} = \boldsymbol{n}_{\text{aim}}^{\perp}$, where $\boldsymbol{n}_{\text{aim}}^{\perp}$ is the normal unit vector to $\boldsymbol{n}_{\text{aim}}$ in the half-plane behind the robot; equivalently, we have
$$\theta_{\text{eva}} = \theta_{\text{aim}} - \pi/2. \quad (5)$$

[2]The main rationale for this choice is that biomechanical studies (see, e.g., [14]) have shown that the human sagittal axis is almost invariably tangent to the Cartesian path during fast locomotion. The consequence of this approach, which we already adopted successfully in [12], [15], is that the generated humanoid gaits will not include lateral steps.

[3]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.

[4]This control law is inspired to the Cartesian regulator described in [16, Sect. 11.6.2]. The main difference is that here we want to track a moving target (the evader) while the Cartesian regulator aimed at reaching a fixed point. For this reason, the driving velocity in eq. (1) is constant rather than modulated by the distance to the target.

[5]Do not confuse θ_e, the orientation of the evader, with θ_{eva}, the orientation associated to the evasion direction.

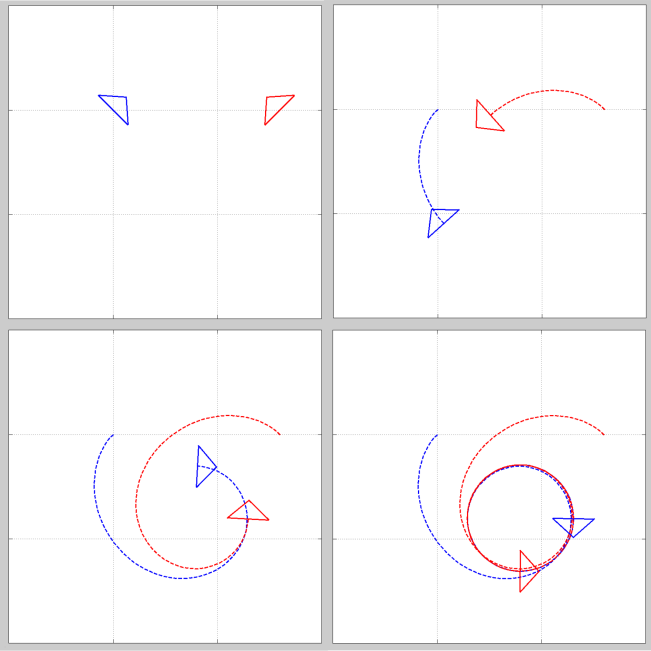

Fig. 2. Pursuit-evasion with unicycles: simulation under control (1–2) for the pursuer (red) and (3–4) for the evader (blue). Axis ticks are 0.5 m apart.

A possible modification of the orientation control law for the evader is

$$\omega_e = k\,\mathrm{sign}(\theta_{\mathrm{eva}} - \theta_e). \tag{6}$$

This would lead the evader to perform an evasion maneuver using the maximum possible curvature radius.

C. Simulations

The pursuit-evasion system with unicycles has been simulated in MATLAB. A typical result obtained using (1–2) for the pursuer (red) and (3–4) for the evader (blue) is shown in Fig. 2; the control parameters were chosen as $\bar{v} = 1$ m/s, k = 0.5. The first snapshot shows the initial configuration of the two robots. In the second, the pursuer moves towards the evader, while the latter starts the evasion maneuver. The last two snapshots show the robots approaching and settling on a circular limit cycle, along which they travel at the same speed; note that their relative orientation is $\pi/2$ (a fact that can easily be proven analytically).

We have also simulated the case in which the evader robot is controlled using the saturated control (6); plots are not shown for brevity. We have found that the robots tend again to a circular limit cycle, whose radius is $\bar{v}/k$. For the same values of $\bar{v}$ and k, this radius is always smaller than the radius of the limit cycle observed in the previous case. Also, the relative orientation between the two robots at steady-state is less than $\pi/2$.

An interesting generalization of the *move aside* evasion strategy is obtained by substituting $\pi/2$ in eq. (5) with a generic angle $\alpha \in [0, \pi/2]$. With this choice, simulations run using the proportional control (2) for the evader show that the relative orientation of the two robots at steady-state becomes exactly α. Moreover, the radius monotonically increases as α is decreased, and tends to infinity when α approaches zero. Note that $\alpha = 0$ corresponds to *move back*, which can then be seen as a limit case of this generalized evasion strategy.

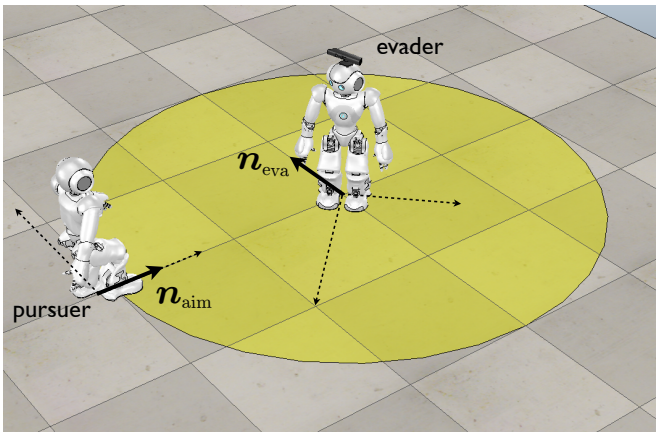

Fig. 3. Pursuit-Evasion with humanoids. The pursuer enters the safety area of the evader and heads towards it. The latter must plan and execute a fast evasive motion. Each robot must continuously replan its motion on the basis of the other's. Note the moving frame associated to each humanoid.

The above observations suggest that the pursuit-evasion system with unicycles possesses strong asymptotic properties which should be further investigated.

III. Pursuit-Evasion with Humanoids

We now proceed to the problem at the center of this paper, i.e., pursuit-evasion with humanoids. The situation of interest is shown in Fig. 3. There are two humanoid robots, one of which acts as a pursuer and the other as an evader. The pursuer is always aware of the presence of the evader, and steers its course trying to intercept it. The evader detects the pursuer when this enters its safety area, triggering the execution of an evasive maneuver.

We shall make the same assumptions A1-A3 of the unicycle case. Since evasion is now a reaction to intrusions in the safety area, A2 must be reinforced by assuming that the evader can also measure the distance to the pursuer.

There is, however, a more fundamental difference: a pure feedback scheme cannot be used, because in humanoids it is necessary to address the problem of gait generation. The proposed solution is to adopt a *replanning* approach: each robot computes a motion plan in real time based on its current perception of the other, and updates this plan at a fast rate to adapt to new perceptions. At its core, the real-time planning procedure still uses a feedback-controlled unicycle for Cartesian trajectory generation.

A. Control schemes

The control schemes for the two robots, outlined in Fig. 4, are structurally the same, although their objectives are obviously different. As with unicycles, 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 sensory system.

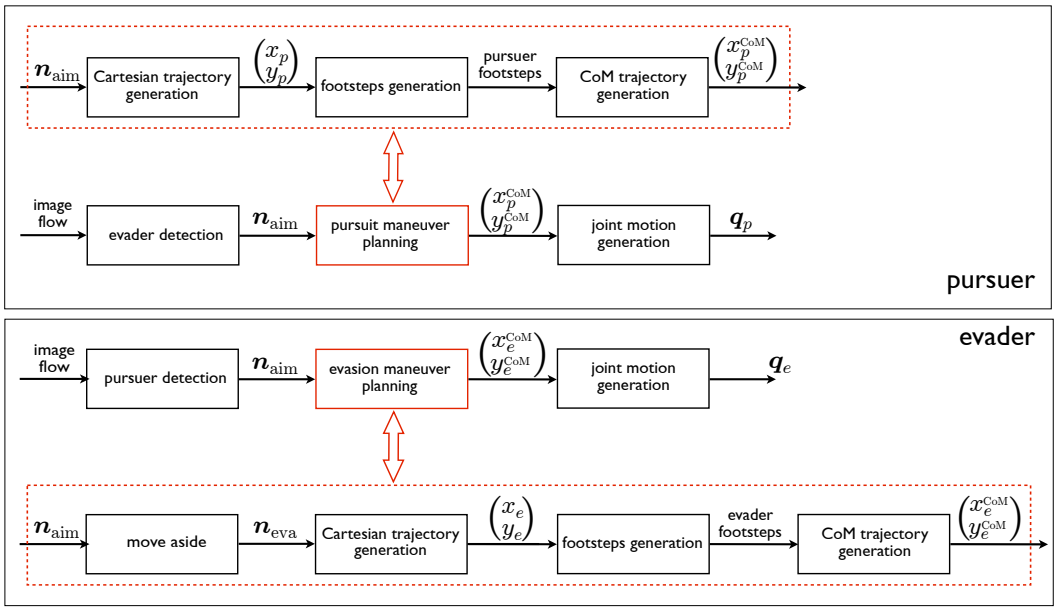

Fig. 4. The control schemes for the pursuer (above) and the evader (below), with the corresponding maneuver planning modules exploded (red box). The only difference between the two robots is that the pursuer tries to align with the line-of-sight to the evader, whose direction is $\boldsymbol{n}_{\text{aim}}$; whereas the evader tries to align with direction $\boldsymbol{n}_{\text{eva}}$, which is orthogonal to the line-of-sight to the pursuer (whose direction is $-\boldsymbol{n}_{\text{aim}}$).

Let us look at the pursuer first. The pursuer detects the evader and measures the corresponding line-of-sight (i.e., the current direction of the evader relative to itself) represented in the following by the unit vector $\boldsymbol{n}_{\text{aim}}$. Based on this information, the robot plans in real time a *pursuit maneuver*, expressed in terms of a reference motion for its own Center of Mass (CoM). This is obtained through a sequential procedure: (1) a Cartesian pursuit trajectory is generated using a unicycle robot as a template model and an orientation control law aimed at aligning the robot with $\boldsymbol{n}_{\text{aim}}$ (2) footsteps are placed around the pursuit trajectory (3) a stable trajectory for the CoM is generated accordingly. Once the CoM plan is ready, it is sent to a kinematic controller for computing appropriate joint motions.

The evader detects the pursuer and measures the corresponding line-of-sight, represented by $-\boldsymbol{n}_{\text{aim}}$. Based on this information, an evasion direction $\boldsymbol{n}_{\text{eva}}$ is computed and an *evasion maneuver* is planned in real time, expressed as a reference motion for the robot CoM. This is done by following the same sequential procedure outlined above, with the only difference that the evasion trajectory is generated by a unicycle robot subject to an orientation control law aimed at aligning the robot with $\boldsymbol{n}_{\text{eva}}$. From this point on, the planning procedure of the evader is an exact replica of that of the pursuer.

In the following, we will describe in some depth the structure of the *maneuver planning* module for both the evader and the pursuer[6]. In conclusion of this section, we discuss when and how replanning is performed.

[6]For brevity, we will not provide details on the structure of the *detection* and *joint motion generation* modules; in particular, the latter can use standard pseudoinverse-based kinematic control (see, e.g., [17])

B. Cartesian trajectory generation

For each humanoid, the *Cartesian trajectory generation* submodule computes a Cartesian reference trajectory using the controlled unicycle models of Sect. II as a template. To allow a unified treatment, we write the control laws (1–2) (pursuer) and (3–4) (evader) as follows

$$v = \pm\bar{v} \tag{7}$$
$$\omega = k\,(\theta^* - \theta). \tag{8}$$

The pursuer is obtained by taking the positive determination of v and $\theta^* = \theta_{\text{aim}}$; while for the evader one should take the negative determination of v and $\theta^* = \theta_{\text{eva}}$.

At the start of each maneuver planning phase, time is reset and the template unicycle is initialized at the origin of the current humanoid frame, with the same orientation; i.e., we let $t_{\text{ini}} = 0$, $x_{\text{ini}} = y_{\text{ini}} = 0$ and $\theta_{\text{ini}} = 0$. To generate a trajectory in real time, we assume that the angular velocity (8) is computed at $t_{\text{ini}} = 0$ (when the orientation error is $\theta^*_{\text{ini}} - \theta_{\text{ini}} = \theta^*_{\text{ini}}$) and then kept constant for the whole planning horizon. Under this premise[7], the unicycle equations can be easily integrated to obtain a closed-form expression for the Cartesian trajectory:

$$x(t) = \pm\bar{v}\,\frac{\sin k\,\theta^*_{\text{ini}}\,t}{k\,\theta^*_{\text{ini}}} \tag{9}$$
$$y(t) = \pm\bar{v}\,\frac{1-\cos k\,\theta^*_{\text{ini}}\,t}{k\,\theta^*_{\text{ini}}} \tag{10}$$
$$\theta(t) = k\,\theta^*_{\text{ini}}\,t \tag{11}$$

[7]Since we are operating in a fast replanning framework, this is an acceptable assumption.

for $t \leq t_s$ and

$$x(t) = x(t_s) \pm \bar{v}(t - t_s)\cos\theta^*_{\rm ini} \quad (12)$$
$$y(t) = y(t_s) \pm \bar{v}(t - t_s)\sin\theta^*_{\rm ini} \quad (13)$$
$$\theta(t) = \theta^*_{\rm ini}, \quad (14)$$

for $t > t_s$, with $t_s = 1/k$. The Cartesian part of this trajectory consists of an arc of circle of radius $\bar{v}/k\,|\theta^*_{\rm ini}|$ (until t_s, where the tangent to the arc has exactly the desired orientation $\theta^*_{\rm ini}$), followed by a straight line.

C. Footstep generation

The *footstep generation* submodule generates a sequence of footsteps around the Cartesian trajectory. The idea is simply to use a constant stepsize Δ along the trajectory itself. This is realized by sampling (9–10) using a constant time interval $\Delta t = \Delta/\bar{v}$ and displacing the samples alternatively to the right and to the left of the trajectory. The orientation of each of these footsteps is that of the tangent to the Cartesian trajectory at the sample point, and is given by the corresponding sample of (11).

D. CoM trajectory generation

The *CoM trajectory generation* submodule computes a stable CoM trajectory for the humanoid robot. It receives in input the footstep sequence, from which a reference trajectory for the ZMP is generated by polynomial interpolation. Computation of a stable CoM trajectory associated to the ZMP reference is performed using the same LIP-based method described in [12]; see also [18] for further details.

Below, we quickly recall the computation of the sagittal coordinate $x_{\rm CoM}$ of the CoM; equivalent formulas for the coronal motion can be easily obtained. Let $\eta = \sqrt{g/z_{\rm CoM}}$, with $z_{\rm CoM}$ the height of the CoM (assumed to be constant in the LIP model), and denote by $x^*_{\rm ZMP}$ the sagittal coordinate of the ZMP reference. We have

$$x_{\rm CoM}(t) = e^{-\eta t}x_{\rm CoM}(0) + \frac{x_{\rm s}(t) - e^{-\eta t}x_{\rm u}(0) + x_{\rm u}(t)}{2},$$

where

$$x_{\rm u}(t) = \eta\int_0^\infty e^{-\eta\tau}x^*_{\rm ZMP}(t+\tau)d\tau$$
$$x_{\rm s}(t) = \eta\int_0^t e^{-\eta(t-\tau)}x^*_{\rm ZMP}(\tau)d\tau.$$

The integrals in $x_{\rm u}(t)$ and $x_{\rm s}(t)$ can be easily computed for polynomial ZMP profiles, ultimately leading to a closed-form computation of $x_{\rm CoM}(t)$.

E. Replanning

Once a robot has planned a (pursuit or evasion) maneuver based on its current perception of the other as explained above, it executes a short portion of it and then recomputes a new plan to adapt to new sensor information.

The maneuver planning procedure makes use of closed-form expressions exclusively, and is therefore suitable for real-time implementation. In principle, then, we could perform replanning at the same rate at which visual data is updated. However, we have chosen to allow the robot to compute a new plan only upon completion of a step, and more precisely at the end of each double support phase. The rationale for this is to guarantee that the reference profile for the CoM is updated only when the robot has both feet on the ground, so as to avoid any destabilizing effect.

As soon as the new plan is available, it replaces the remaining part of the previous plan. In practice, this implies that the straight line part (12–13) of the Cartesian trajectory is never actually traveled by a robot, at least as long as the line-of-sight to the other robot keeps changing.

IV. Simulations and Experiments

The proposed approach was validated using two NAO humanoids. One of them, acting as evader, has a depth camera (ASUS Xtion) mounted on its head. This camera provides also the distance to the closest obstacle, making it possible to detect intrusions into the safety area. The pursuer does not need a measurement of the distance to the evader, and therefore only uses the built-in monocular camera.

For simulations, we have used the V-REP environment. The evader safety area is assumed to have a radius of 0.8 m. Cartesian trajectories for the pursuer and the evader are generated as explained in Sect. III-B, with $\bar{v} = 0.1$ m/s and $k = 0.2$. Footsteps are distributed around these trajectories using $\Delta = 0.04$ m, a value consistent with the NAO gait capabilities. A ZMP trajectory is computed from the footsteps using single and double support durations of 0.122 s and 0.425 s, respectively. The value of $z_{\rm CoM}$ used for CoM trajectory generation is 0.268 m.

A typical simulation is summarized in Fig. 5 (see the accompanying video for a clip), where each frame contains a side view and a top view for a given time instant. In spite of the adaptations needed for the humanoid case, the results fully confirm the pursuit-evasion behavior observed for the unicycle case: the two robots converge to a circular limit cycle, along which they travel at the same speed with a relative orientation of $\pi/2$.

For the experiments, the various control parameters have been set to exactly the same values of the simulations. In spite of the rather limited processing capabilities (each NAO is equipped with an Intel Atom running at 1.6 GHz) we were able to perform all computations on-board; in particular, each call to the maneuver planning module takes less than 10 ms.

Figure 6 shows snapshots taken during an experiment (see the accompanying video for a clip). The expected limit cycle behavior is observed again, although its radius is slightly reduced with respect to the simulation. This is mainly due to the fact that the actual robot speed is less than 0.1 m/s due to significant feet slippage on the smooth floor.

V. Conclusions

We have considered a pursuit-evasion problem between humanoids. This is an evolution of the setting considered in our previous work [12], where a humanoid had to avoid an incoming intruder which was initially headed for collision but did not alter its course to pursue the evader.

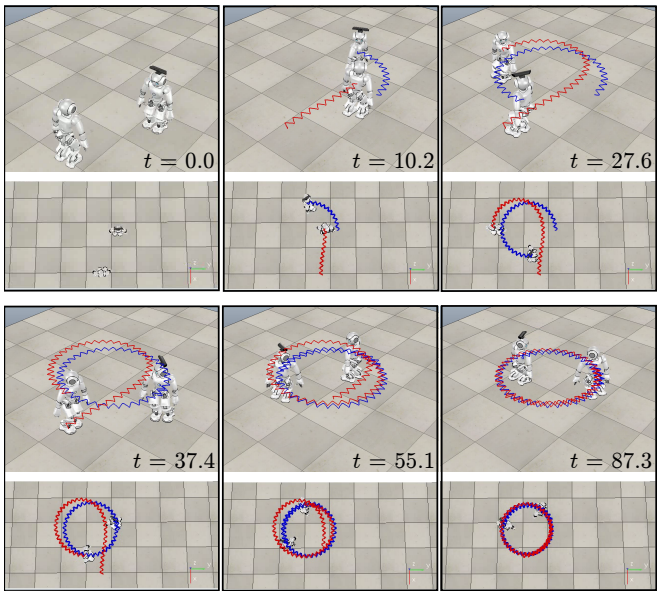

Fig. 5. Pursuit-evasion with humanoids: snapshots from a simulation. The trajectories of the CoMs are shown in red (pursuer) and blue (evader).

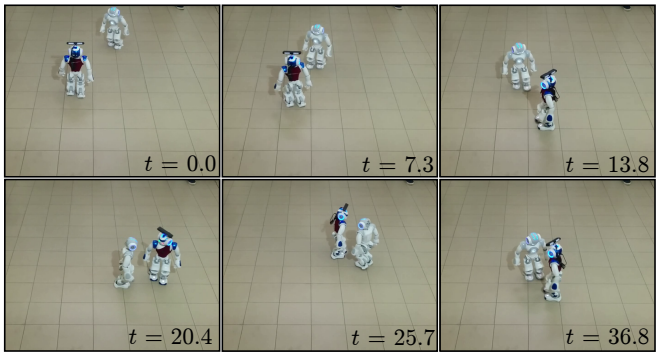

Fig. 6. Pursuit-evasion with humanoids: snapshots from an experiment.

We have designed control schemes 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 scheme is a maneuver planning module that implements a sequential procedure. From the desired direction of motion, a corresponding Cartesian trajectory is generated, around which footsteps are placed; from these, a stable trajectory for the humanoid CoM is computed. The whole planning procedure makes use of closed-form expressions, thus making real-time implementation possible, and is repeated upon completion of each step to account for the motion of the other robot.

An interesting outcome of our study is that the pursuer and the evader converge to a circular limit cycle along which they travel at the same speed. This property has been first observed on unicycles, which have been used as template models for Cartesian trajectory generation, and then fully confirmed on NAO humanoids, both in simulations and experiments.

Future work will address several points, among which:

- asymptotic properties of pursuit-evasion with unicycles;
- design and validation of more sophisticated evasion strategies;
- how to perform evasion maneuvers in the presence of obstacles in the workspace;
- the use of MPC to compute robust, stable trajectories for the robot CoMs [19].

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.

[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] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726 – 1743, 2013.

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[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] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion – an inverse optimal control approach," *Autonomous Robots*, vol. 28, no. 3, pp. 369–383, 2010.

[15] A. Faragasso, G. Oriolo, A. Paolillo, and M. Vendittelli, "Vision-based corridor navigation for humanoid robots," in *2013 IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 3190–3195.

[16] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*. Springer, 2009.

[17] M. Cognetti, P. Mohammadi, and G. Oriolo, "Humanoid whole-body planning based on CoM movement primitives," in *2015 IEEE-RAS Int. Conf. on Humanoid Robots*, 2015, pp. 1090–1095.

[18] L. Lanari, S. Hutchinson, and L. Marchionni, "Boundedness issues in planning of locomotion trajectories for biped robots," in *2014 IEEE-RAS Int. Conf. on Humanoid Robots*, 2014, pp. 951–958.

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