Trajectory Following for Legged Robots

Thomas Moulard, Florent Lamiraux, Olivier Stasse

Trajectory Following for Legged Robots

Thomas Moulard, Florent Lamiraux, Olivier Stasse

LAAS-CNRS, Université de Toulouse

7, avenue du Colonel Roche

31077 Toulouse cedex 4, France

`thomas.moulard@laas.fr, olivier.stasse@laas.fr`

Abstract—While robust trajectory following is a well-studied problem on mobile robots, the question of how to track accurately a trajectory on a humanoid robot remains open.

This paper suggests a closed-loop trajectory tracking strategy aimed at humanoid robots. Compared to approaches from mobile robotics, this control scheme takes into account footsteps alteration, equilibrium constraints and singularities avoidance for humanoids. It provides a robust way to execute long and/or precise motion with the ability of correcting on-line preplanned trajectories in a very reactive manner. Results have been validated on the HRP-2 humanoid platform.

Index Terms—Humanoid robots, Motion planning, Robot control

I. Introduction

Following a planned trajectory on a robot while compensating execution errors has been extensively studied in the 90's for mobile robots [1], [2]. Surprisingly, this issue has not been explicitly addressed in the literature concerning navigation for legged robots, although these machines are also prone to execution errors while moving. Previous experiments such as [3] illustrate how imprecise trajectory following on a humanoid robot can be. After executing a five meters long trajectory, the difference between the planned and real position can reach 0.4m. Such an error cannot be ignored anymore and invalidate the whole planning stage. Therefore, solving this issue is crucial and will allow the achievement of complex movements where a high precision is needed. For instance, obstacle crossing is only feasible at the beginning of the trajectory where the drift is not too important. This paper objective is to provide a generic framework for robust trajectory following on a humanoid robot. One way of indirectly tackling the problem consists of regularly replanning the motion of the robot from its current configuration to the goal after localizing obstacles with respect to the robot. This strategy enables the robot to be reactive to environment changes as well as to execution errors [4]–[6]. On the other hand, it requires short planning time and induces heavy CPU load. It might even not be always be possible. Indeed most fast replanning schemes rely on a simplified model [7] of the robot neglecting momenta generated by the leg motions. These assumptions are not met for small robots like Nao with a large ratio of mass distributed in the legs and with a small CPU. Moreover, to produce a really feasible movement, additional constraints must be satisfied: no auto-collision should occur during the movement for instance. For all theses reasons, validating a complex movement remains a computationally

Fig. 1. HRP-2 robot walking in a constrained environment. Compensating for execution errors is crucial in this scenario to avoid collisions.

expensive operation. Therefore, an alternative solution to on-line replanning and regeneration of the walking trajectory such as [4], [8]–[10] is the continuous deformation of walking trajectories. This combination of dynamic trajectories and high probability for the robot to enter in auto-collision makes naive correction algorithm fail which is why it is important to define a sound framework for trajectory following. This paper presents a "blink of an eye" reshaping of the trajectory associated with a generic method to follow trajectories on a humanoid robot. These two features together provide a way to follow a trajectory while compensating for errors during the movement execution. This opens many possible applications such as moving in extremely constrained environments in a reliable manner, going to specific places of the environment precisely, etc. Most of the state of the art demonstration of reactive pattern generators are, in fact, open loop trajectories

with no sensors feedback. This work has been fully integrated into the LAAS/JRL planning and control frameworks and a motion capture system has been used to close the loop and evaluate the execution errors. This allowed HRP-2 humanoid robot to perform precise and/or long locomotion tasks where usual open loop approaches would have drifted so much that the task would have failed.

II. MOTIVATION

This work has been motivated by the previous experimental setup described in [3]. In this paper, fast online replanning is used to handle environment changes. Using replanning to cancel the drift has been considered at first but suffers from several drawbacks. The initial idea was to accelerate replanning and consider that there is no need to take into account the execution errors as it can be handled by changing the robot starting position to the position given by the localization system and regenerate the part of the trajectory which is yet to be executed. This is difficult in practice for several reasons. First, using the localization system as an input of the planning component is dangerous. If the localization is imprecise, so will be the plan. Therefore, it is required to filter the robot position explicitly to ensure a high quality localization at every point of time. On the opposite, as our control scheme only modifies the next step and is bounded by a correction limit, a low-pass filter is implicitly applied and protects the control scheme from temporary erroneous localization. Second, if the localization is imprecise and is near an obstacle, the estimated position may be in collision with the obstacle. It is possible to project the robot position outside the obstacle but additional efforts are required during the motion planning step. To finish, even if fast replanning is possible, randomized planning such as RRT-based methods cannot be used safely in a real-time context as they cannot guarantee to compute a solution within a determined time frame. In [3], when a replanning is required, the three next steps cannot be modified to avoid discontinuities in the robot trajectory. It means that the correction cannot be as reactive as it may be necessary: the execution error may be taken into account too late. On the opposite, the next section will demonstrate that real-time correction is possible.

III. PROBLEM STATEMENT

A. Notation and definitions

A *robot* is a kinematic chain the configuration of which is denoted by $q \in \mathcal{C}$, where $\mathcal{C}$ is the robot configuration space. The robot position is represented by the x, y components and yaw rotation r_z of a reference body in the 3D space and is denoted $\mathbf{x} \in \mathrm{SE}(2)$, $\mathrm{SE}(2)$ the rigid motions in the 2D space. The height z, roll r_x and pitch r_y of the reference body are stored with the kinematic chain angles. This reference body is often the body attached to the root of the kinematic chain. In this paper, the position of the robot waist defines the robot position. Therefore the robot configuration is:

$$\begin{aligned} \mathbf{x} &= [x, y, r_z] \\ \mathbf{q} &= (\mathbf{x}, [z, r_x, r_y, \mathbf{q}_{\mathrm{int}}]) \in \mathcal{C} = \mathrm{SE}(2) \times \mathcal{C}_{\mathrm{int}} \end{aligned} \quad (1)$$

Therefore a *trajectory* is a continuous function γ associating to each point of time of the interval $[t_{\min}, t_{\max}]$ a particular robot configuration $\mathbf{q}(t)$. Walking is a sequence of one or many *footstep(s)*. At the beginning, both feet are on the floor. This phase is called *double support phase*. Then, a foot moves until it reaches a desired position. This interval of time, until the foot lies on the floor again is called a *single support phase*. The moving foot is the *swing foot*, the static foot is the *support foot*. A footstep position is a 2D position on the plane. Steps will be denoted by $S \in \mathrm{SE}(2)$. A walking movement can be described as a sequence of footsteps S_i, $0 \le i \le n^{\mathrm{step}}$. Step duration is constant and equal to T_{step}.

B. Trajectory following: from mobile robots to humanoids

As trajectory following has been extensively studied, a direct use of previously studied ideas such as illustrated by Fig. 2 would seem natural. However, this section will demonstrate that this naive approach is not sufficient. Fig. 2 depicts what would be a mobile-robot closed loop tracking system applied to a humanoid robot. The reference trajectory γ would be modified by a feedback provided by some external localization system integrated to the robot and providing an estimation of the robot position and orientation denoted by $\hat{\mathbf{x}}$. From this estimation, an error is computed and added to the original trajectory component by component using a proportional gain, i.e. k_x, k_y, k_{r_z} on the figure.
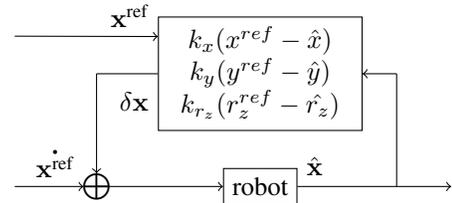
Fig. 2. Naive correction system for a humanoid robot. $\mathbf{x}^{\mathrm{ref}} = (x^{\mathrm{ref}}, y^{\mathrm{ref}}, r_z^{\mathrm{ref}})$, $\dot{\mathbf{x}}^{\mathrm{ref}} = (\dot{x}^{\mathrm{ref}}, \dot{y}^{\mathrm{ref}}, \dot{r_z}^{\mathrm{ref}})$ and $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{r_z})$ are respectively the current planned robot position, velocity and position estimated by an external localization system. The planned control $\dot{\mathbf{x}}$ is rectified by summing $\delta\mathbf{x}$ a correction directly computed by the position error of the waist between the plan and the perception.

This system provides an updated trajectory of the waist taking into account execution error through a feedback loop. As long as the humanoid robot has at least one foot on the floor, the waist becomes locally fully actuated and this correction can be applied as long as $\mathbf{q}_{\mathrm{int}}$, the joint values are recomputed accordingly.

a) Corrected trajectory stability: Unlike mobile robots, humanoid robots do not have to take into account the non-holonomic constraint which simplifies the control scheme. However, humanoids robots must preserve equilibrium during motion. This constraint is equivalent to the center of pressure $\mathbf{z}$ remaining in the convex hull of the contact points of the feet on the ground:

$$\mathbf{z} = \mathbf{x} + \frac{1}{m(\ddot{z}_c + g)} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \dot{\mathbf{L}} - \frac{z_c}{\ddot{z}_c + g} \ddot{\mathbf{x}} \quad (2)$$

where z_c is the height of the center of mass with respect to the ground, m is the mass of the robot, g is the gravity constant $\mathbf{x} = (x_c, y_c)$ is the projection of the center mass of the robot on the ground and $\mathbf{L}$ is the angular momentum of the robot about the center of mass. Naively applying a correction of the robot waist trajectory as suggested by Fig. 2 induces a perturbation of the center of mass and thus of the center of pressure trajectories that may violate the equilibrium constraint. By constraining the center of mass to remain at constant height, and neglecting variations of the angular momentum, the above equation simplifies into the following linear relation:

$$\mathbf{z} = \mathbf{x} - \frac{z_c}{g}\ddot{\mathbf{x}} \qquad (3)$$

Linearity implies that perturbing the center of mass trajectory by a function of time $\delta\mathbf{x}$ perturbates the center of pressure trajectory according to the same relation:

$$\begin{aligned} \mathbf{z}' &= (\mathbf{x} + \delta\mathbf{x}) - \frac{z_c}{g}\cdot\frac{d^2(\mathbf{x} + \delta\mathbf{x})}{dt^2} \\ &= \mathbf{x} - \frac{z_c}{g}\ddot{\mathbf{x}} + \underbrace{\delta\mathbf{x} - \frac{z_c}{g}\ddot{\delta\mathbf{x}}}_{\text{induced ZMP perturbation}} \end{aligned} \qquad (4)$$

Using the simplified linear model, computing a trajectory correction is thus the same problem as computing a dynamically balanced trajectory. However, if the initial trajectory has been computed using the exact multi-body model Eq. (2), computing a trajectory correction using the linear model implies that approximation is performed only on the correction and may result in trajectories of better quality from a dynamic balance point of view.

b) Corrected trajectory singularities: As the waist is only locally fully actuated, it is important to compute a correction which does not introduce singularities during motion. The presence of singularities is directly related to the relative position of waist and contact points. However, applying directly the correction does not trigger any modification of the contact points, i.e. the footsteps. In practice, it means that as errors happen the gap between the waist and the feet will increase and finally there is a high risk to be unable to recompute the joints values due to the robot mechanical limits. From this discussion, it appears clearly that these two drawbacks make the naive solution unsatisfactory. Therefore, correcting a humanoid robot trajectory cannot be solved by considering it as a mobile robot: a new strategy is required. To solve these issues, a better control scheme allowing larger corrections and more suited to humanoid robots will be introduced in the next section.

IV. CLOSED-LOOP TRAJECTORY FOLLOWING FOR HUMANOID ROBOTS

This section proposes a control system for closed-loop trajectory tracking while taking into consideration humanoid robots specificities. Closed-loop trajectory tracking consists in following a precomputed trajectory while compensating for execution errors. Systems are often composed of four components:

1) a trajectory generator component,
2) a localization component providing an estimation of the robot position,
3) an error estimation component computing the error between the planned position and the localization of the robot,
4) and a component reshaping the planned trajectory to compensate for the above error.

The trajectory generator provides two reference data: the footstep sequence, a set of footsteps S_i such as $0 \leq i \leq n^{\text{step}}$ and a whole-body trajectory $\gamma(t \in [t_{\min}, t_{\max}]) \in \mathcal{C}$. One advantage of the proposed control scheme is to alter the future footstep positions to avoid singularities. Given a known perturbation of the footstep sequence, it is then possible to deduce the correction that should be applied to the feet and center of mass trajectories. Once those trajectories are computed, inverse geometry can be used to regenerate the joints trajectories. One iteration of the control loop is described by Algorithm 1 and can be summarized as:

1) estimate the robot position,
2) compute the position error $\delta\mathbf{x}$,
3) filter the error to avoid perturbing too much the initial trajectory and to absorb localization noise,
4) recompute the next steps positions to compensate execution errors and make sure the feet will land on the planned position,
5) check if the recomputed next step is feasible,
6) regenerate smooth trajectories for the feet, center of mass and ZMP.
7) regenerate the joints trajectories. This step is denoted by $\gamma \bigoplus \delta\gamma$ in the algorithm. The $\gamma \bigoplus \delta\gamma$ operation returns γ altered by the rigid transformation $\delta\gamma$. $\delta\gamma$ is the perturbation applied to the whole-body trajectory and is not directly computed as the inverse geometry is directly applied on the updated body positions.

Algorithm 1 Control loop at time t_{current} achieving a closed-loop following of trajectory γ (next correction will be applied at $t_{\text{next\_correction}}$).

Input: γ, t_{current}, $t_{\text{next\_correction}}$
Output: γ, t_{current}, $t_{\text{next\_correction}}$
  if $\gamma(t_{\text{current}})$ is double support **and** $t_{\text{current}} \geq t_{\text{next\_correction}}$ **then**
    estimate robot position $\hat{\mathbf{x}}$
    compute robot position error $\delta\mathbf{x}$
    compute offset $\delta\gamma$ absorbing the execution error $\delta\mathbf{x}$
    if the perturbation $\delta\gamma$ can be applied **then**
      $\forall t \in [t_{\text{current}}, t_{\max}], \gamma(t) \leftarrow \gamma(t) \bigoplus \delta\gamma(t)$
      $t_{\text{next\_correction}} \leftarrow t_{\text{current}} + 2T_{\text{step}}$
    end if
  end if
  $\mathbf{q} \leftarrow \gamma(t_{\text{current}})$
  $t_{\text{current}} \leftarrow t_{\text{current}} + \Delta t$

First, the robot position $\hat{\mathbf{x}}$ is perceived. The localization system will not be detailed in this paper, see [11], [12] for instance for more details. Although common limitations of these systems are taken into account. The precision of the robot estimation does not decrease over time, but can vary during the execution. This produces a noise which may perturb the control scheme. The localization system can also fail to provide an estimation or even sometimes provide aberrant values. Secondly, an error $\delta\mathbf{x}$ is computed by comparing the planned and estimated position of the tracked reference body. A threshold is applied to this value to bound the applied corrections. In practice, it also filters out outliers and the noise that the localization system may introduce in the system. Then, the relative position
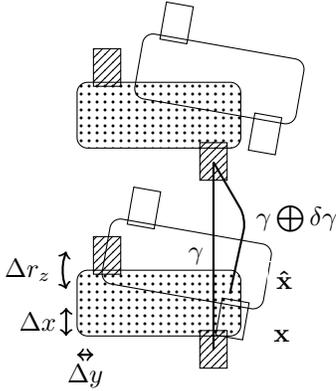

Fig. 3. Correction of the next step due to a position error. Dotted rectangles are the planned positions $\mathbf{x}$ of the robot waist and feet before and after the next step. Non-dotted rectangles corresponds to the robot localization $\hat{\mathbf{x}}$. Error w.r.t to axis X, Y and yaw rotation is $(\Delta x, \Delta y, \Delta r_z) = \delta\mathbf{x}$, γ the reference trajectory and $\delta\gamma$ the corrected trajectory reaching the planned step.

of the next footstep w.r.t. to the current one is changed to compensate the perceived error. Fig. 3 illustrates this process. From this point, smooth trajectories can be regenerated for feet and center of mass. To ensure smoothness, the trajectory correction is progressively applied during the next two steps. To finish, joint values are recomputed using these new reference trajectories. Additionally, a test is added to check that a correction can be computed for the current time t_{current}. A correction can be applied if no correction is being applied, i.e. $t_{\text{current}} \geq t_{\text{next\_correction}}$ and if the robot is in the double support phase. Indeed starting a correction in the middle of a step would be dangerous and starting a correction while another correction is being applied would lead to erroneous results. Previous works such as [13], [14] aim at allowing sudden changes in the robot trajectory. The proposed control scheme is different and aim at following as closely as possible a preplanned trajectory.

c) Estimation of the position error: We make the assumption that an external system provides $\hat{\mathbf{x}} \in \mathrm{SE}(2)$, an estimation of the current robot position. If $\mathbf{x} \in \mathrm{SE}(2)$ is the robot planned position and $\hat{\mathbf{x}} \in \mathrm{SE}(2)$ the robot localization, the position error is defined by: $\delta\mathbf{x} = \mathbf{x}.\hat{\mathbf{x}}^{-1}$ In the previous equation, $\delta\mathbf{x}(t)$ can be interpreted as the planned robot posi-

tion with respect to the current robot position at time t. By consequence, at the beginning of the trajectory t_{min}, the error is always equal to zero: $\delta\mathbf{x}(t_{\text{min}}) = 0$

d) Footstep sequence modification: Given a position error of the waist, it is possible to alter the remaining steps in the footstep sequence to absorb this offset. The purpose of this step is to take into consideration that the previous step has not been executed correctly, leading to a different relative position than the one which has been initially planned. To cancel the error, the next footsteps positions will be modified so that the robot will step in the planned locations. The footstep positions are $S \in \mathrm{SE}(2)^{n^{\text{step}}}$. Let consider $\delta\mathbf{x}$, the current position error, $S^{\text{future}} \subset S$ the steps which have not been played yet. The footstep positions will be changed according to the following computation: $\forall s \in S^{\text{future}}, s \leftarrow \delta\mathbf{x}.s$

e) Whole-body trajectories modification: A new placement of the next feet has been computed. It is now necessary to modify the two feet and the center of mass trajectories synchronously to reach the corrected foot prints. The correction is computed by considering the simplified model introduced in section III. Hence, no hypothesis is done on the strategy used to plan the reference trajectories. One interest of this approach is to totally dissociate the planning and correction algorithms. To compute a small perturbation the simplified model is sufficient. It allows extremely reactive correction without compromising the overall trajectory quality. The linearized inverse pendulum model allows the computation of the center of mass trajectory $\mathbf{c}(t)$ given a ZMP trajectory $\mathbf{z}(t)$ by solving Eq. (2). Considering $\mathbf{r}$ a polynomial depending only of $\mathbf{z}$, (V_x, V_y, W_x, W_y) free parameters used to constrain the initial position and velocity of the center of mass, a general following form of a polynomial center of mass trajectory is:

$$\mathbf{c}(t) = \cosh(\sqrt{\frac{g}{z_c}}.t).\mathbf{V} + \sinh(\sqrt{\frac{g}{z_c}}.t).\mathbf{W} + \mathbf{r}(t) \quad (5)$$

Given the formulation in Eq. (5), it is possible to continuously modify the center of mass trajectory to make it follow $\bar{\mathbf{z}}(t)$ the corrected trajectory. This new trajectory can be expressed as the sum of two polynomials:

$$\bar{\mathbf{c}}(t) = \cosh(\sqrt{\frac{g}{z_c}}.t).\mathbf{V} + \sinh(\sqrt{\frac{g}{z_c}}.t).\mathbf{W} + \mathbf{r}(t) + \boldsymbol{\Delta}(t) \quad (6)$$

To apply smoothly the correction from t_1 to t_2, several constraints expressed in Eq. 7 have to be respected. $\delta\mathbf{x}_x$ and $\delta\mathbf{x}_y$ are respectively the x and y components of the 2d rigid transformation $\delta\mathbf{x}$.

$$\boldsymbol{\Delta}(t_1) = \frac{\partial\boldsymbol{\Delta}}{\partial t}(t_1) = \frac{\partial\boldsymbol{\Delta}}{\partial t}(t_2) = 0$$
$$\boldsymbol{\Delta}(t_2) = \begin{pmatrix} \delta\mathbf{x}_x \\ \delta\mathbf{x}_y \end{pmatrix} \quad (7)$$

These four constraints determine the polynomial four parameters leading to the curve illustrated by Fig. 4. This reshapes the center of mass trajectory by taking advantage of the linear formulation of the simplified model. Additionally, the feet trajectories are modified to reach the corrected positions at
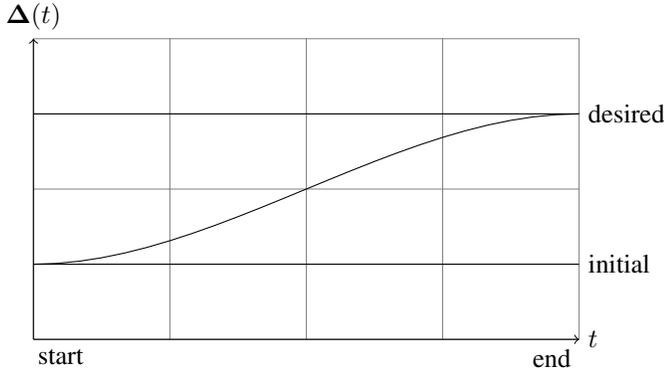
Fig. 4. Polynomial curve $\mathbf{\Delta}(t)$ providing a smooth transition between feet and center of mass trajectories.
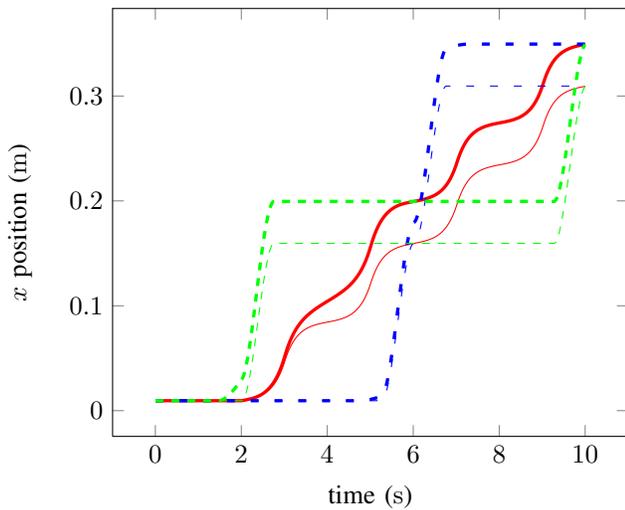

Fig. 5. Evolution, during two steps, of the x position of the center of mass (——), left foot (– –) and right foot (– – –). Bold curve depicts a step a 0.3m forward. Dashed curve depicts the center of mass (——), left foot (– –) and right foot (– – –) trajectories with a correction of 0.05m in the forward direction.

the end of the step. The smooth correction is also obtained by using a third degree polynomial with similar constraints: initial position remains as before, the goal position must fit the corrected position and the velocity of the correction is equal to zero at the beginning and the end of the transition. Fig. 5 compares resulting trajectories before and after the correction. These three corrections must be executed in the correct order: as stated before a correction is computed during a double support phase and applied during the next two steps. We will make the assumption, without any loss of generality, that the next flying foot is the left one. In that case the correction of the left foot is progressively applied during the single support phase. During the ZMP shift of this step, the center of mass correction is also applied. Then, during the next step, the correction of the right foot is applied. The timeline of the correction is illustrated by Fig. 5.

f) Error thresholding and new step feasibility: Correction does not compromise the robot stability. However, any correction is not feasible due to the robot mechanical limits and auto-collision. Therefore, it is important to bound corrections in order to avoid producing infeasible steps. It is also important to validate new steps w.r.t to auto-collision. The base hypothesis is that the reference trajectory which has been computed offline is safe i.e. stable and without any auto-collision. By consequence, the goal is to determine how much this initial trajectory can be modified without compromising its safety. First, a maximum error has been determined empirically. The maximum lateral perturbation is ± 0.04m , the maximum forward perturbation ± 0.05m and the maximum angular perturbation is ± 0.1rad every two steps. Secondly, the new step
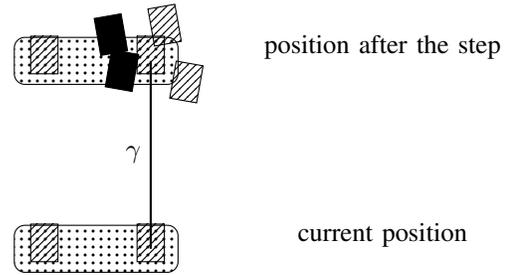

Fig. 6. Validation of the recomputed next step. Waist position is symbolized by dotted rectangle, valid steps by hashed rectangles and invalid steps by black rectangles.

is validated. The decision is based on the relative movement of the first corrected foot. Let consider that, for instance, the next moving foot is the left one. The original left foot position is $s \in \mathrm{SE}(2)$, the new corrected left foot position is $s' \in \mathrm{SE}(2)$. The relative position of the original and new foot is computed. If the y component of the result is positive, the new step is accepted. In practice, it means that the step is "pushed" away and will not induce an auto-collision. On the opposite, with the right foot, the step is accepted if the y component is negative.
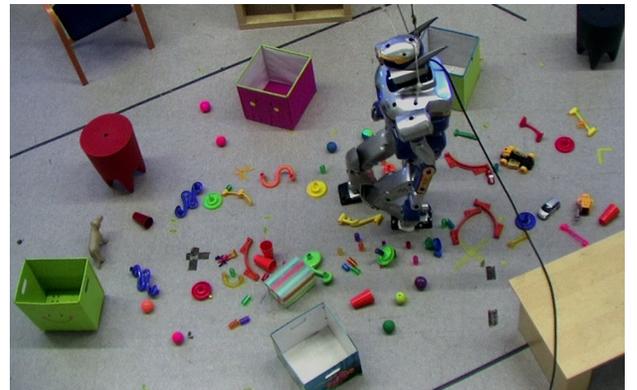

Fig. 7. HRP-2 walking in a cluttered environment while avoiding obstacles. The final position is reached with a precision of ± 3cm. The final error is the consequence of both noise in the robot position estimation and drifts in the two last steps which happen too late to be not compensated.

If a correction is invalidated, another one is computed during the next double support phase. In practice, the estimated error during the next step will be close to the current estimation, in particular the sign (i.e. direction) of the error will remain the same. As the flying foot will change the delayed correction has a high probability of being accepted. This naive system has been empirically validated. However, it prevents some valid corrected steps. In the future, this will be replaced by a fast on-line steps validation algorithm described by [15].

V. EXPERIMENTAL RESULTS

The proposed method has been validated on HRP-2 using the following scenarii: the robot walks in a narrow and constrained environment while stepping over obstacles. The trajectory length is 2.5m and is executed in approximately 40s. This setup demonstrates that the execution using this control scheme is precise and reliable. The robot reaches its goal position with an error of a few centimeters whereas open loop control schemes is off by more than a meter. Moreover, the control scheme is reliable: during the movement, the localization precision can change, the robot position tracking can fail. . . The control scheme is designed to be robust toward these situations. This experiment uses the pattern generator described by [15] and the stack of tasks formalism [16] to implement the control scheme. The pattern generator provides reference trajectories for the center of mass and the feet. Trajectory following tasks for these particular bodies are then inserted into the control framework. Computing the control such as the tasks are satisfied is then realized by the solver. This experiment video is available on the web[1]. Fig. 7 provides an overview of the scenario: the robot starts on the right side and walks toward the left side. In this context, the planned final position is reached with a precision of ± 2cm. This trajectory has been executed five times consecutively with any obstacle collision. Using a different footsteps sequence with smaller steps, the planned position can be reached with an error lower than 1cm.

VI. FUTURE WORK AND CONCLUSION

This paper has discussed the difference between closed-loop trajectory tracking on a mobile robot and on a humanoid robot. Section III demonstrates that even if this problem may seem simple, a direct use of the mobile robots control schemes is not realistic. Section IV suggests a new strategy which takes into consideration the humanoid robot specificities, in particular the need to alter contact points. Section V provides an illustration of an experiment where the HRP-2 robot follows a trajectory and achieve a result which would be impossible using an open loop scheme, validating the proposed approach. However, additional work is needed to enhance this strategy: the heuristic step validation should be replaced by a better criteria and maintaining a precise localization during a long movement is still a problem. Additionally, more complex movements should be planned and executed to validate the generality of this approach.

[1]http://youtu.be/cUZ0nNiPs70

REFERENCES

[1] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, apr 1991, pp. 1136–1141.

[2] A. de Luca, G. Oriolo, and C. Samson, *Robot motion planning and control*, Jean-Paul Laumond ed., ser. Lectures Notes in Control and Information Sciences 229. Springer, 1998, no. ISBN 3-540-76219-1, ch. Feedback Control of a Nonholonomic Car-like Robot.

[3] L. Baudouin, P. Perrin, T. Moulard, O. Stasse, and Y. E., "Real-time replanning using 3d environment for humanoid robot," in *11th IEEE/RAS International Conference on Humanoid Robots (Humanoids), to appear*, 2011. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00601300/en/

[4] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, "Vision-guided humanoid footstep planning for dynamic environments," in *5th IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, dec. 2005, pp. 13–18.

[5] P. Michel, J. Chestnut, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, "Online environment reconstruction for biped navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 3089–3094.

[6] J. Chestnut, *Motion planning for humanoid robots*, K. Harada and E. Yoshida and K. Yokoi ed. Springer, 2010, ch. Navigation and Gait Planning.

[7] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 239–246.

[8] D. Dimitrov, A. Paolillo, and P.-B. Wieber, "Walking motion generation with online foot position adaptation based on l1- and l∞-norm penalty formulations," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011. [Online]. Available: http://hal.inria.fr/inria-00567671/en

[9] 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. [Online]. Available: http://hal.inria.fr/inria-00391408/en

[10] K. Nishiwaki and S. Kagami, "High frequency walking pattern generation based on preview control of zmp," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2006, pp. 2667–2672.

[11] O. Stasse, B. Verrelst, A. Davison, N. Mansard, F. Saidi, B. Vanderborght, C. Esteves, and K. Yokoi, "Integrating walking and vision to increase humanoid autonomy," *International Journal of Humanoid Robotics,special issue on Cognitive Humanoid Robots*, vol. 5, no. 2, p. accepted, 2008.

[12] S. Thompson, S. Kagami, and K. Nishiwaki, "Localisation for autonomous humanoid navigation," in *Proc. of the 6th IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, dec. 2006, pp. 13 –19.

[13] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," in *4th IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, vol. 2, nov. 2004, pp. 640–655.

[14] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *IEEE International Conference on Robotics and Automation (ICRA)*, april 2007, pp. 3989 –3994.

[15] N. Perrin, O. Stasse, F. Lamiraux, and E. Yoshida, "Approximation of feasibility tests for reactive walk on hrp-2," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2010, pp. 4243–4248.

[16] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *Proc. of the International Conference on Advanced Robotics (ICAR)*, june 2009, pp. 1–6.