Sound and fast footstep planning for humanoid robots

Nicolas Perrin, Olivier Stasse, Léo Baudouin, Florent Lamiraux, Eiichi Yoshida

Abstract— **In this paper we present some concepts for sound and fast footstep planning. The soundness is achieved with a two-stage trajectory generation process that uses a smoothing homotopy. The fastness is achieved with swept volume approximations precomputed offline.**

I. Introduction

This paper is roughly a condensed version of [16], where methods and results are presented in more details.

Footstep planning is the problem of generating walk motions for a humanoid robot that must go from an initial position to a goal in an environment with obstacles. In this paper we assume that the walking surface is flat and horizontal. Footstep planning is a key problem in the field of humanoid robotics, because one of the reasons for making humanoid robots is precisely that they should be able to move about in environments that are not accessible to wheeled robots. In the last decade, this problem has been studied extensively, and although no really satisfying solution exists so far, a lot of promising approaches have been experimentally validated (see [9], [4], [18], [5]).

Because of its complexity, footstep planning is usually not solved directly. Indeed, it combines high dimensionality with underactuation and complex balance constraints, three issues that make it a difficult motion planning problem. On top of that, in order to be compatible with human behaviours, the robot is often required to quickly generate walking motions to go to a given goal. To make this problem simpler, the solution usually adopted is to first plan only a sequence of footprints, and then, using a walking pattern generator, produce a trajectory in the configuration space so that the robot will follow the footprints and avoid the obstacles without falling. The good thing about this strategy is that the problem of planning footprints is much easier to solve than the full footstep planning problem, and what is more there exist efficient algorithms to generate walking motions from sequences of footprints. But the risk with this separation is to produce footprint sequences that do not lead to feasible trajectories, or on the contrary be overconservative and frequently miss solutions. When the first issue is avoided we say that the separation is sound, and when solutions are not missed we say it is complete. Completeness, or at least ”almost completeness”, is desirable, but soundness is crucial. In this paper we present an efficient solution that provides a sound separation and is not too overconservative. An example of overconservativeness we want to avoid is the one of the ”bounding box method” (see [20]): we want to robot to be able to use its stepping-over capabilities to overcome small obstacles on the floor. Let us recall the principles of the bounding box method: first, a box (usually a rectangular cuboid) is defined that contains the whole robot. Then, using classical path planning techniques, a continuous collision-free path is found for the box, from an initial location to a goal. Then, footsteps are generated such that the robot stays at all time inside the volume swept by the box during its motion. This approach is clearly sound: if the robot stays inside the volume swept by the box, and if the path found for the box is collision-free, then *a fortiori* the robot motion is collision-free. However this approach is not complete, and if there is for example a cable around the robot on the floor, then it is impossible to find a collision-free motion of the box that would escape from the region defined by the cable. So, even if the robot could in fact step over the cable with ease, the bounding box method will never find a solution. An other example of overconservativeness occurs when the variety of steps considered by the planner is not representative of the actual stepping capabilities of the robot. Many state-of-the-art approaches based on the use of the algorithm A* on a small set of steps are threatened by this issue (see for example [8], [2], [3], [4], [6]).

For the separation to be sound, a cautious analysis of the walking pattern generator is required. For example, with state-of-the-art walking pattern generators, the motion performed by the robot tend to depend not only on the current step, but also on the previous and next few steps (that is true in particular when preview control is used, see [7]), and that makes sound separations difficult to obtain. In Section II, we briefly present a two-stage walking pattern generator that uses a smoothing homotopy to enable the generation of state-of-the-art trajectories while keeping a sound separation. In Section III, we present experiments obtained on the robot HRP-2 with a footstep planning algorithm based on this walking pattern generator and on a finite but large set of steps for which swept volume approximations are precomputed offline.

Section IV is the conclusion.

II. A two-stage walking pattern generator based on a smoothing homotopy

In this section we briefly present a walking pattern generator that combines half-steps for dimensionality reduction, and a smoothing homotopy. The dimensionality reduction

Nicolas Perrin, Olivier Stasse, Léo Baudouin and Florent Lamiraux are with Université de Toulouse ; UPS, INSA, INP, ISAE ; CNRS ; LAAS ; F-31077 Toulouse, France (nperrin@laas.fr, ostasse@laas.fr, leo.baudouin@laas.fr, florent@laas.fr)

Eiichi Yoshida is with CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT AIST Central 2, Umezono 1-1-1, Tsukuba 305-8568 Japan (e.yoshida@aist.go.jp)

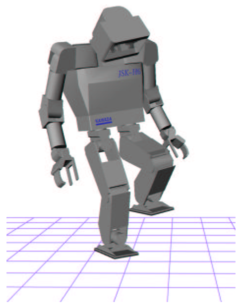

Fig. 1. The intermediate posture Q_{right} used in [8] for transitioning between left leg footstep placements.

brought by the use of half-steps is interesting for offline precomputations, as we will see in Section III, but in this section we focus on the smoothing homotopy which enables a sound separation between the (footprint) planning phase and the trajectory generation. For more details about this walking pattern generator, see [17], or [16].

The use of half-steps is a bit similar to the approach introduced in [8]: half-steps are obtained by introducing two statically stable, intermediate postures Q_{right} and Q_{left} that serve as via point configurations for all footstep transitions (see Fig. 1). Once these configurations are fixed, an isolated half-step (downward or upward) can be defined by three parameters only.

The goal of our walking pattern generator is of course to take in input a sequence of footprints, and return in output a (discretized) trajectory in the configuration space of the robot. This is done with two distinct phases. During the first phase, an initial configuration space trajectory is given: it corresponds to a simple concatenation of isolated half-steps. This concatenation is possible because each half-step starts and ends at zero speed in a balanced posture. With this simple concatenation it is easy to achieve soundness at the planning phase. Indeed, if the planner needs to verify the validity of random footsteps, it can simply use the trajectory corresponding to the concatenation of isolated half-steps: no matter what next and previous footsteps will eventually be decided, after the first phase the resulting trajectory will always correspond to this concatenation of half-steps. Thus, if the trajectory is verified and considered feasible by the planner, it will also be feasible inside the trajectory obtained after the first phase.

This trajectory could be used to move the robot from its initial configuration to the goal, and the method would arguably be not too conservative. However, this result would not be satisfying for several reasons. First, between each half-step the robot would come to a stop, and the walk motion would not be visually smooth. Furthermore, because of these stops the motion would be rather slow, and not energy-efficient. Recent walking pattern generators using preview control (see [7]) achieve much better results. Next, we present the principles of a smoothing homotopy that continuously modifies the initial trajectory in order to make it smoother and faster. Because the modifications are continuous, it is easy to keep the soundness of the whole process. Indeed, we know that the sequence of footprints found by the planner corresponds to a feasible initial trajectory, and then we can continuously smooth it and stop just before it becomes unfeasible.

The key principle of our smoothing homotopy is very similar to the one of the mixtures of motions introduced in ([14] and [15]), but in these papers the purpose was to create new steps, not to smooth them nor speed them up. To explain this principle, let us present first a few aspects of our trajectory generation process. We use a classical simplified model of the robot dynamics: the linear inverted pendulum model (see [7]). In this model the mass of the robot is assumed to be concentrated in its CoM (Center of Mass) which is supposed to be rigidly linked to the robot waist and directly above it at all time. Besides, the robot is supposed to have only coplanar point contacts with the horizontal walking surface. An analysis of the subsequent equations leads to a further approximation which enables the decoupling of the dynamic differential equations for the x-axis and y-axis. They can be written as follows:

$$p_x = Z(x) \tag{1}$$

$$p_y = Z(y) \tag{2}$$

$$\text{with } Z \triangleq Id - \frac{z_c}{g}\frac{d}{dt^2} \tag{3}$$

(x, y) are the (x-axis,y-axis) coordinates of the CoM of the robot, z_c is the height of the robot center of mass which is supposed constant during the steps, and (p_x, p_y) are the (x-axis,y-axis) coordinates of the virtual Zero Moment Point (ZMP). A classical dynamic balance criterion for biped walking is that the ZMP should stay at all time inside the polygon of support (see [19]). An important thing to notice in these equations is that Z is a linear operator. Using these equations, we use the following strategy to generate half-step trajectories: first, we define a trajectory for the swing foot. This trajectory depends on the 3 parameters defining the half-step. Then, we define a trajectory for the ZMP that also depends on the 3 parameters and that stays at all time inside the polygon of support. Finally, using the linear equations (1) and (1), we compute a corresponding horizontal trajectory for the CoM. All the half-step motions have the same duration T.

With concatenations of these half-step motions we obtain initial trajectories, and we will explain how to continuously modify a sequence of half-steps in order to make it faster and smoother along the same footprint sequence. We first show how to do so for a sequence of two half-steps, and start with the case of an upward half-step followed by a downward half-step.

1) Upward then downward: We consider an upward half-step followed by a downward half-step. The upward half-step is shown on Fig. 2 (we take as origin the center of the support foot). Together the two half-steps make a classical full step: double support phase, then quick ZMP shift and single support phase, and then second quick ZMP shift and double

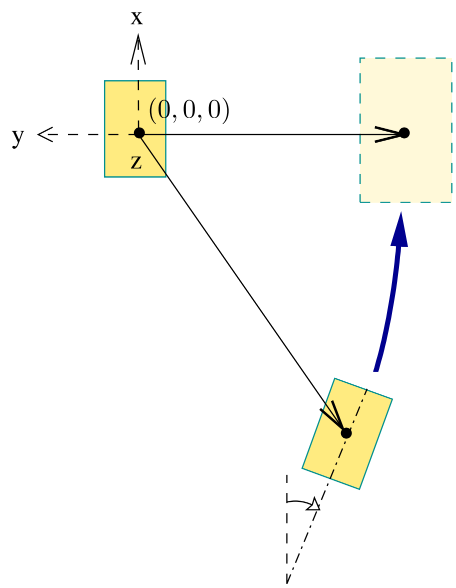

Fig. 2. Representation of an upward half-step from above.

support phase again. We denote by $\left((x_1(t), y_1(t))\right)_{t\in[0,T]}$ the horizontal CoM trajectory for the first half-step, and by $\left((x_2(t), y_2(t))\right)_{t\in[0,T]}$ the horizontal CoM trajectory for the second half-step. We explain only the smoothing process for the component along the y-axis, but it is exactly the same process for component along the x-axis. At the end of the upward half-step, the robot is in a balanced posture, with its CoM directly above the center of the support foot. Thus, we have: $y_1(T) = y_2(0) = 0$.

Let us now define two operators g^1_Δ and g^2_Δ such that:

$$g^1_\Delta(f)(t) = \begin{cases} f(t) \text{ for } t \in (0, T) \\ f(T) \text{ for } t \in (T, 2T - \Delta) \end{cases} \quad (4)$$

$$g^2_\Delta(f)(t) = \begin{cases} 0 \text{ for } t \in (0, T - \Delta) \\ f(t - T + \Delta) - f(0) \text{ for } t \in (T - \Delta, 2T - \Delta) \end{cases} \quad (5)$$

$g^1_0(y_1) + g^2_0(y_2)$ corresponds to the simple concatenation of y_1 and y_2 without overlap. Knowing that $p_{y_1} = Z(y_1)$, $p_{y_2} = Z(y_2)$, and $y_1(T) = y_2(0) = 0$, it is quite easy to verify that for any $0 \leq \Delta \leq T$, $g^1_\Delta(p_{y_1}) = Z(g^1_\Delta(y_1))$ and $g^2_\Delta(p_{y_2}) = Z(g^2_\Delta(y_2))$. And, since Z is a linear operator:

$$g^1_\Delta(p_{y_1}) + g^2_\Delta(p_{y_2}) = Z(g^1_\Delta(y_1) + g^2_\Delta(y_2)) \quad (6)$$

It follows that operators g^1_Δ and g^2_Δ enable us to obtain new combined CoM and ZMP trajectories that still verify the Linear Inverted Pendulum equations (eq. (1) and eq. (2)). Starting with $\Delta = 0$ and progressively increasing the value of Δ continuously modifies the CoM trajectory (starting from the initial trajectory $g^1_0(y_1) + g^2_0(y_2)$) to make the second ZMP shift (the one of p_{y_2}) happen earlier, creating an overlap of duration Δ between the two trajectories y_1 and y_2. Fig. 3 illustrates this effect. When we increase the value of Δ we can see for example that the position of the CoM does not need to reach the center of the support foot anymore.

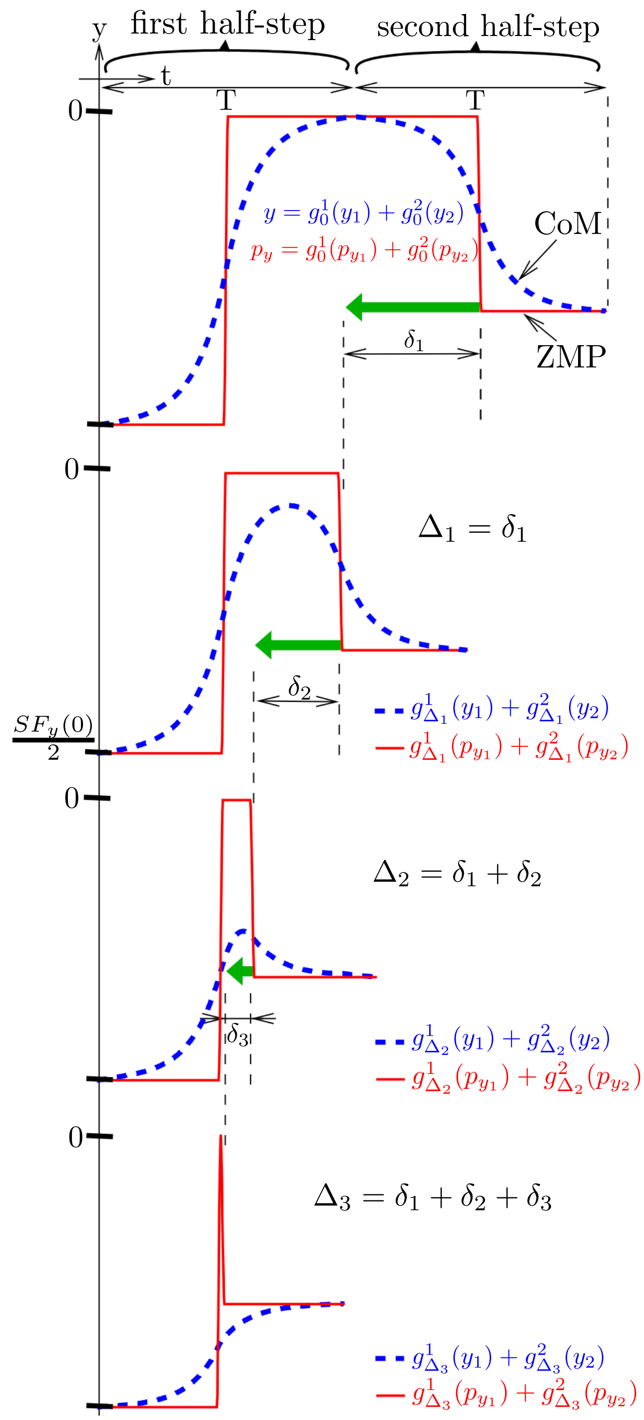

Fig. 3. Progressively increasing the overlap between two half-steps. We denote by $SF_y(0)$ the initial position of the swing foot along the y-axis. The plot on the top shows the trajectories $y(t)$ and $p_y(t)$ for a raw sequence of two half-steps (no overlap), the first half-step being the one of Fig. 2. Notice that the CoM reaches the ZMP between the half-steps. On the other plots, we show the effect of progressively increasing the overlap, using the operators g^1_Δ and g^2_Δ. We can see that the CoM trajectory becomes more natural: it does not need to reach the top of the ZMP curve between the two ZMP shifts anymore. Indeed, the overlap works a bit like a preview control: the first CoM trajectory is influenced by the second one during the overlap, so it is as if it already "knew" that there will be another ZMP shift, and adapted consequently.

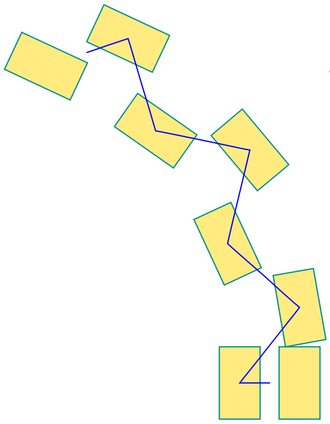

Fig. 4. We illustrate the "smoothing" of a raw sequence of half-steps. On the initial raw sequence (on the left), the support paths of the ZMP and CoM trajectories are superimposed. Then, after adjusting the overlaps, the ZMP support path stays the same but the CoM support path becomes smoother (on the right). We can smooth even more, but it reduces the duration of the single support phase that is directly linked with the swing foot speed. Therefore limitations on the swing foot speed constrain the smoothing process.

We use the same operators, g^1_Δ and g^2_Δ, to produce an overlap between the functions of time corresponding to the waist orientation and swing foot position and orientation. Since the inverse geometry for the legs is a continuous function as long as we stay inside the joint limits, these operators used on the bodies trajectories actually implement a simple homotopy that continuously deforms the initial C-space trajectory into a smoother, more dynamic trajectory.

In the case of an upward half-step followed by a downward half-step, increasing Δ reduces the duration of the single support phase, and therefore it increases the speed of the swing foot. To limit this effect we must bound Δ. Besides, if Δ is too large undesirable phenomena can occur, such as a negative swing foot height. To avoid these problems we set an upper bound such that the maximum overlap results in a moderately fast gait.

2) Downward then upward: We can apply the same technique to produce an overlap in the case of a downward half-step followed by an upward half-step. Since the last phase of the downward half-step and the first phase of the upward half-step are double support phases, the constraint on the swing foot motion disappears and the maximum bound on Δ becomes simply T.

For longer sequences of half-steps, we can simply repeat the procedure to smooth the whole trajectory. Fig. 4 shows the results obtained with an example of raw sequence. After the smoothing, the CoM trajectory is visually smoother and besides, the new trajectory is much faster (about 3 times faster).

Changing overlaps inside a sequence of half-steps modifies the whole configuration space trajectory: not only the CoM and ZMP, but also the swing foot trajectory. When the overlap is increased, the swing foot tends to move faster and closer to the ground. If one property must be preserved (for instance the absence of collision), it must be checked after every modification. Since the smoothing by overlap is a continuous operator, we can use dichotomies to quickly find large acceptable values of overlap. Let us consider an example for two consecutive half-steps. We predefine a maximum overlap Δ_{max} and, first, we simulate the part of the trajectory that is modified by the overlap $\Delta_{max}/2$, and check for collisions, self-collisions and joint limits violations. If these three constraints are verified, we try with $3\Delta_{max}/4$. Otherwise, we try $\Delta_{max}/4$, etc. We stop and go to the next overlap once we find a "good" overlap value close enough to a "bad" overlap value. Fig. 5 shows the effect of the smoothing process on the swing foot trajectory: with the dichotomy we can quickly find a large overlap that keeps the trajectory collision-free.

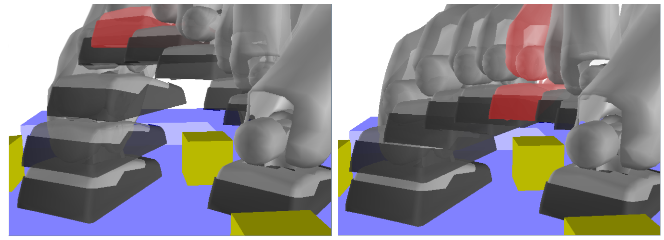

Fig. 5. *On the left*: a raw sequence of two half-steps avoiding a box on the ground. We can see that the swing foot reaches an unnecessarily high position. After smoothing (*on the right*), the trajectory has been modified so that the foot moves very close to the obstacle.

III. Precomputed Swept Volume Approximations for Fast Footstep Planning

The walking pattern generator briefly presented in the previous section is based on half-steps, and they have the good properties of being completely defined with only 3 parameters. Thus, we can obtain a quite dense grid in the parameter space with not too many points. Each point of the grid completely defines a half-step trajectory. Using extensive offline precomputations, we can build approximations of the volume swept by the legs of the robot during these half-steps. Then, during the footprint planning phase, we can use these approximations to speed up the collision checks. Indeed, the planner often requires the validation of footsteps. This is usually done by checking if collisions occur along the trajectory corresponding to a given footstep. If we only allow the planner to consider footsteps whose half-steps are parametrized by points of our predefined grid, then we can use our swept volume approximations: instead of checking for collisions along discretized trajectories (i.e. hundreds of collision checks per half-step), we simply perform one collision check per half-step: between the environment and the swept volume approximation. A great deal of computation time can be saved using this approach, but it is not trivial to obtain a good planning algorithm that can handle the large set of half-steps (276 in our case). As the complexity of A* quickly increases with the number of half-steps allowed, we opted for an adaptation of the algorithm RRT ([11]) to discrete footstep planning (see [16] for details).

We implemented and tested our approach in real-time replanning experiments with the robot HRP-2, in environments where the obstacles and the robot position are acquired by motion capture. This implementation and these experiments are also mentioned in [16], and precisely described in [1].

In these experiments, we used two distinct computers to plan and execute the robot motions: one computer for the planning, and one for the control, the latter being equipped with a real-time operating system. A CORBA server organizes and transfers communications between the 4 units of our architecture: the two computers just mentioned, the motion capture system responsible for obstacles and robot localization, and a viewer that shows in real-time the new paths found by the robot.

The control of the robot motion is made with a module called Stack of Tasks (see [12], [13]) which is a structure managing priorities between the active controllers. In an execution thread, the planned trajectory is progressively sent to the control part, while the localization information is read to check for potential collisions. If a collision is detected along the planned trajectory, a query is sent to the planner to generate a new path that tries to link the part of the current trajectory before the expected collisions to the part of the current trajectory after the expected collisions. When the environment is not changing, the planner uses its free time to try to improve parts of the current path, or to smooth the currently planned sequences of steps.

The two-phase approach of the trajectory generation is very convenient for online replanning because we control the independence between half-steps in the sense that the smoothing between two consecutive half-steps can easily be canceled. For instance it is easy to “unsmooth” a part of the currently planned sequence in order to delay a step, or to modify it without modifying the previous steps.

With 276 meshes representing our precomputed swept volume approximations, we use the PQP algorithm [10] for collision checks, and during the smoothing process we also use PQP to check for collisions between the environment and convex hulls of the robot bodies (using convex hulls is a bit conservative and slightly reduces the number of triangles).

Fig. 6 and Fig. 7 illustrate some of our results in experiments or simulations. The goal and 3D obstacles can be moved in real-time, and the robot tries to adapt its trajectory consequently. An interesting point of the results is that even when stepping over motions are required, the robot can often replan a trajectory very quickly. However, there is some latency and the robot can typically replan the trajectory only 1 or 2 steps after the current one.

IV. Conclusion

The framework for sound and fast footstep planning briefly presented in this paper and explained in more details in [16] includes a walking pattern generator based on half-steps, a simple homotopy for trajectory smoothing, swept volume approximations for fast collision checking, and an RRT variant for footstep planning. We used this framework on the robot HRP-2 to quickly plan dynamic sequences of

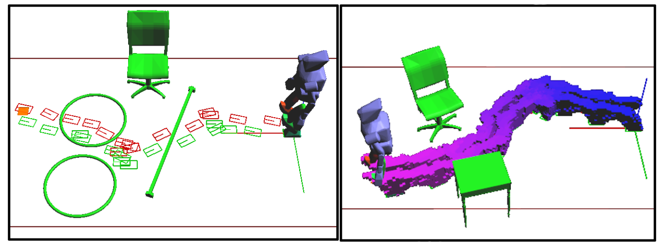

Fig. 6. On the left: a sequence of steps found in a complex environment. On the right, we show for one sequence of steps the concatenation of the swept volume approximation meshes. For the upper body simpler bounding boxes are used for the collision checks.

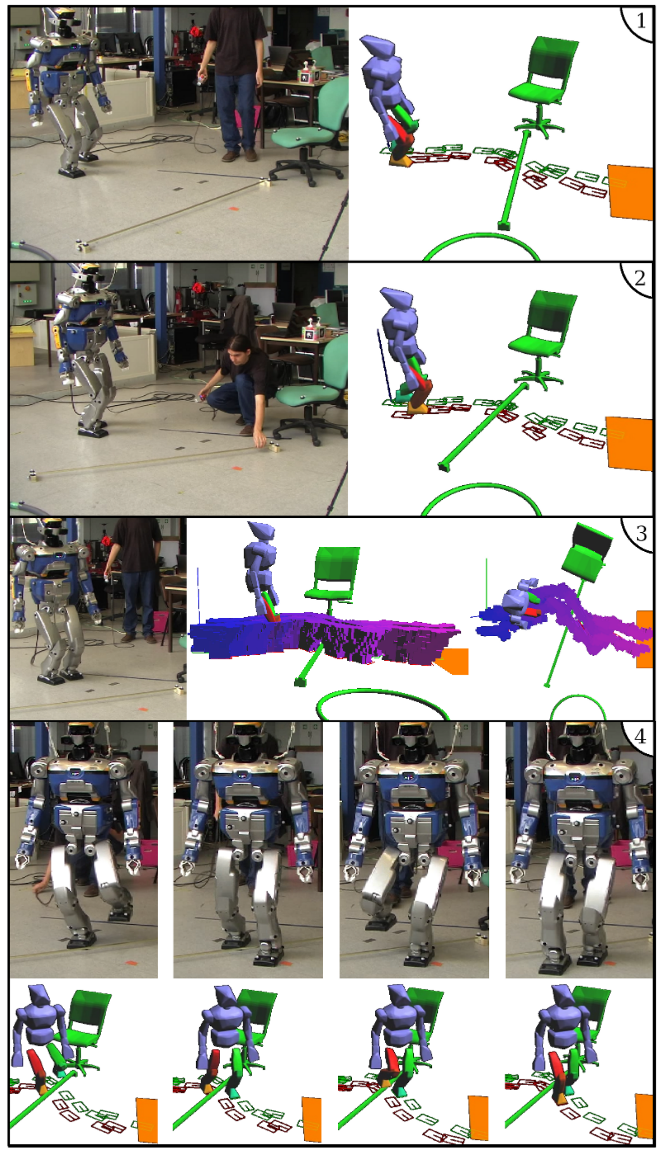

Fig. 7. In this experiment a bar placed 5cm above the ground is moved while the robot is executing its initial plan. (1): HRP-2 starts to execute the sequence initially found. (2): the bar is suddenly moved, and the current sequence of step would lead to collisions. (3): while walking, HRP-2 is able to compute a new sequence of steps towards the goal (we show the concatenation of the swept volumes which indeed avoid the bar). (4): the robot finally steps over the bar while at the same time it tries to optimize the rest of the path towards the goal. Remark: due to uncertainty on positions, we use a model of bar that is thicker than the actual one.

walk in environments cluttered with 3D obstacles. Although computed in a few seconds and with some guarantees given by the soundness of the approach, the executed trajectories seem very natural: no pauses, no exaggerated motions to avoid small obstacles, and a large diversity of foot placements.

V. Acknowledgements

This work was supported by a grant from the RBLINK Project, Contrat ANR-08-JCJC-0075-01.

References

[1] L. Baudouin, N. Perrin, T. Moulard, F. Lamiraux, O. Stasse, and E. Yoshida. Small-time controllability of a walking humanoid robot. Accepted to the 11th IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'11). Available at http://homepages.laas.fr/nperrin/accepted/humanoids11-lbaudouin.pdf, 2011.

[2] J.-M. Bourgeot, N. Cislo, and B. Espiau. Path-planning and tracking in a 3d complex environment for an anthropomorphic biped robot. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS'02)*, volume 3, pages 2509–2514, 2002.

[3] J. Chestnutt, J. J. Kuffner, K. Nishiwaki, and S. Kagami. Planning biped navigation strategies in complex environments. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'03)*, 2003.

[4] J. Chestnutt, M. Lau, G. Cheung, J. J. Kuffner, J. Hodgins, and T. Kanade. Footstep planning for the honda asimo humanoid. In *IEEE Int. Conf. on Robotics and Automation (ICRA'05)*, pages 631–636, 2005.

[5] J. Chestnutt, Y. Takaoka, K. Suga, K. Nishiwaki, J. J. Kuffner, and S. Kagami. Biped navigation in rough environments using on-board sensing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'09)*, 2009.

[6] J.-S. Gutmann, M. Fukuchi, and M. Fujita. Real-time path planning for humanoid robot navigation. In *Int. Joint Conf. on Artificial Intelligence (IJCAI05)*, pages 1232–1237, 2005.

[7] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, and K. Yokoi. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, pages 1620–1626, 2003.

[8] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Footstep planning among obstacles for biped robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'01)*, pages 500–505, 2001.

[9] J. J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue. Online footstep planning for humanoid robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA'2003)*, 2003.

[10] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Fast proximity queries with swept sphere volumes. In *IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, pages 3719–3726, 2000.

[11] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *4th Workshop on the Algorithmic Foundations of Robotics (WAFR'00)*, pages 293–308, 2000.

[12] N. Mansard and F. Chaumette. Task Sequencing for Sensor-Based Control. *IEEE Trans. on Robotics*, 23(1):60–72, February 2007.

[13] 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 *Int. Conf. on Advanced Robotics (ICAR'09).*, pages 1–6, 2009.

[14] K. Nishiwaki, K. Nagasaka, M. Inaba, and H. Inoue. Generation of reactive stepping motion for a humanoid by dynamically stable mixture of pre-designed motions. In *1999 IEEE International Conference on Systems, Man, and Cybernetics*, pages 902 – 907 vol.1, 1999.

[15] K. Nishiwaki, T. Sugihara, S. Kagami, M.y. Inaba, and H. Inoue. Online mixture and connection of basic motions for humanoid walking control by footprint specification. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 4110 – 4115 vol.4, 2001.

[16] N. Perrin, O. Stasse, Léo Baudouin, F. Lamiraux, and E. Yoshida. Fast humanoid robot collision-free footstep planning using swept volume approximations. 2011. Accepted to IEEE Trans. on Robotics. Available at http://homepages.laas.fr/nperrin/accepted/TRO_accepted2011.pdf.

[17] N. Perrin, O. Stasse, F. Lamiraux, and E. Yoshida. A biped walking pattern generator based on "half-steps" for dimensionality reduction. In *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, pages 1270–1275, 2011.

[18] M. Stilman, K. Nishiwaki, S. Kagami, and J. J. Kuffner. Planning and executing navigation among movable obstacles. In *IEEE/RSJ Int. Conf. On Intelligent Robots and Systems (IROS'06)*, pages 820–826, 2006.

[19] M. Vukobratovic and B. Borovac. Zero-moment point – thirty five years of its life. *Int. Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[20] E. Yoshida, C. Esteves, I. Belousov, J.-P. Laumond, T. Sakaguchi, and K. Yokoi. Planning 3D collision-free dynamic robotic motion through iterative reshaping. *IEEE Trans. on Robotics*, 24(5):1186–1198, 2008.