On the Problem of Online Footsteps Correction for Humanoid Robots

Nicolas PERRIN (CNRS-LAAS), Olivier STASSE, Florent LAMIRAUX (CNRS-LAAS), Paul EVRARD (CNRS-LIRMM), Abderrahmane KHEDDAR (CNRS-LIRMM)

CNRS-AIST JRL, UMI3218/CRT, Tsukuba, Japan

Abstract— In humanoid robotics, many applications involving walking require unfeasible footsteps to be corrected very quickly. In this paper, we introduce and discuss that issue, and expose our preliminary work and results on the topic.

Key Words: Humanoid robots, Footsteps planning, Gait

1. Introduction and Background

Several biped robots that are capable of walking have been designed in the past 2 decades. To make these robots actually walk, the classical approach is to combine a gait generator and a path planner.

The gait generator's purpose is to achieve the fundamental task of generating a trajectory from a list of footsteps. This trajectory uses the legs, waist, arms of the robot to create a dynamically stable motion which realizes the desired sequence of steps. Here are some of the most recent techniques for gait generation: [5], [1], [2], [3]. Of course, the robot has limitations, and thus the gait generator also: from an initial configuration, only a limited set of footprints is accessible for the next step. If a footprint out of this set is given as an input to the gait generator, it will still attempt to produce a trajectory. Even if nice models for humanoids (e.g. the 3D Linear Inverted Pendulum –see [4]– or the Cart-Table model –see [5]–) have been developped in order to simplify the calculations of stable trajectories based on the stability criterion known as ZMP (Zero-Moment Point, see [6]), the generation of a trajectory remains rather time-consuming: typically a few hundreds of milliseconds are needed for one step. Moreover, the problems that can make a trajectory fail on the real robot are manifold: self-collisions, violation of joint limits, torque limits, bad behavior of the ZMP due to differences between the robot's inertia and the one of the model used for calculations...

To sum up, there is to this day no method to guess whether a footstep or sequence of footsteps *will* lead to a generated trajectory actually feasible on the real robot in few hundreds of microseconds. The main consequence of this issue takes place in the other component of walking: the path-planner. From a high-level task and a sensed environment, the path-planner generates sequences of footsteps that should be realized by the robot. Indeed, the *sine qua non* condition is that the footsteps must lead to an actually feasible trajectory after the use of the gait generator. The current methods for path planning are based on rapidly growing trees of footprints sequences whose aim is to reach a goal while avoiding obstacles (see [7]). Therefore, we cannot include actual simulation with the gait generator in the process: with more than 0.1s. per footstep, the full process of planning would take dozens of minutes at least, which is of course not acceptable if we hope for a *reactive walk*. The work of Kuffner and Chestnutt et al. ([8], [9]) shows how this problem is tackled currently. They force the path-planner to only consider a small finite set of steps on which the robot has been tested. The steps are selected so that they enable all the basic motions: go forward, backward, turn right, left, etc. This leads to efficient planning and it is enough for many applications. But here, the role of the gait generator is not crucial anymore: all the trajectories can be learned offline. On top of that a problem where a bit of flexibility oin the footsteps is required won't be solved. Obviously it doesn't push the robot to its limits. In our work we consider another point of view where the problem becomes harder and the current approach not satisfying. We consider that, in the same way you can actually guide a blind person to some location without knowing exactly how she walks, the path planner should not necessarily be built with much knowledge on the gait generator which will be used. Let's give a specific example (there are many): consider the problem of walk imitation, where a robot wants to imitate the walk of a human. The trivial way to generate the sequence of footsteps is to simply copy the steps of the human. Since the human and the robot don't share the same characteristics, even after homothetic transformation some steps won't be feasible, and here arises the problem of footstep correction: given a footstep that might not be feasible for the robot, we want to produce very quickly a feasible footstep, as close as possible to the first one, where here then sense of "close to" should respect the achievment of similar "types" of high-level motions: turn right, left,

. . . If we simply project the footstep on a small finite set of known steps, the results won't be convicing, so we have to come up with a new solution. We will suppose no specific knowledge on the gait generator, so we can assume that information (on whether a step is feasible or not) can be obtained only through testing.
Three approaches can be considered:

- The discrete approach: use the same principles as the current approach, but instead of selecting a small set of steps, automatically construct a similar but large set of steps with much more expressiveness.

- The continuous approach: use machine learning techniques which can generalize on unseen footsteps, and build a structure on the space of steps which enables to efficiently project a footstep on the accepted area. One of the problems of this approach is that considering the complexity of the gait generators, we most probably won't be ensured that the generalization is safe, and walking safely is a critical requirement. Yet, some emergency behavior in case of wrong guess might cope with that issue.
- The mixed approach: combine the discrete and the continuous approaches.

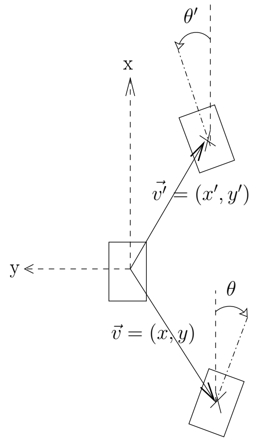
2. A remark on dynamic walk and unicity of trajectories

Up to now, we made the implicit assumption the the trajectory generated for one given step is unique. This is not obvious at all: for example, recent gait generators produce fully dynamic walks, and thus take into account the initial speed of the robot's center of mass: the same step inserted in two different sequences of other steps lead to different motions. But we consider that restrictions (e.g on the initial speed of the center of mass, on the height of the waist during the step, . . .) have been made in order to obtain the unicity of trajectories.

3. The space of steps

With the assumptions made in the previous section, a step is fully parametrized by its geometry. This corresponds to a 6-dimensional space: a step is parametrized by $(x, y, \theta, x', y', \theta')$, where $\vec{v} = (x, y)$ is the vector from the center of the stable foot to the initial position of the swing foot, θ the relative orientation of the swing foot in its initial configuration, $\vec{v} = (x', y')$ is the vector from stable foot to the final position of the swing foot, and θ' the relative orientation of the final configuration of the swing foot. Here is an illustration of that space:

When we correct a step, we only correct the final footprint, but the initial position cannot be changed.



Remark: normally we should precise which is the swing foot (left or right), but in our work we suppose that the left foot is always to the left of the right foot: either both y and y' are positive, and the swing foot is the left one, or they are both negative and the swing foot is the right one.

Fig.1 Description of the space of steps

Hence the function of footstep correction should take in input a 6-uple $(x, y, \theta, x', y', \theta')$, and return a 6-uple of the form $(x, y, \theta, x'', y'', \theta'')$.

4. The discrete approach

The set of steps can be represented as a graph where, in our formalism, there is an arrow from $(x_1, y_1, \theta_1, x'_1, y'_1, \theta'_1)$ to $(x_2, y_2, \theta_2, x'_2, y'_2, \theta'_2)$ if and only if $x'_1 = -x_2$, $y'_1 = -y_2$, and $\theta'_1 = -\theta'_2$. The challenge is to create a functional graph (i.e. its expressiveness enables the robot to perform all "kinds" of steps that one might think of), and which is balanced: there should be only a small standard deviation on the degree of the vertices (the number of outgoing arrows). Besides that, the graph not only has to be globally functional, but also locally functional: at each vertex the outgoing edges must express nice possibilities of movements. The obvious drawback of the discrete approach is its lack of flexibility, which can be seen in two dual issues:

- First, lack of flexibility on the arrival footprint: for example, if the robot has to obtain a precise orientation (for example in order to face a slope it wants to climb). In order to do it with a discrete set of steps, the robot might have to do complicated sequences of steps before obtaining a satisfying stance (if it can find one). Let's say for example that the robot can spin around itself by an increment of 10° to the right, and by an increment of 7° to the left. Then, to spin by 5° to the left, it has to spin three times to the right, and 5 times to the left. In [10], that problem arises. Many other cases where a bit of flexibility for the arrival footprint would be of great interest can be thought of (e.g. make the length of the steps correspond to the length of some stairs –see [11]–, etc.).
- The dual problem, lack of flexibility for the initial footprint, is easy to understand: if for some reason the robot ends up in a position slightly

different than the one planned (e.g. because it has been pushed, has touched some unseen part of the environment, etc.), then its position might not correspond to any initial position of any step of the graph: no safe movement can be quickly found at this point.

Because of these two drawbacks, we chose to first focus on the other approach: the continuous approach.

5. The continuous approach

Here, the challenge is to use machine learning techniques in order to build an approximation of the indicator function of the set of acceptable steps in the 6-dimensional space. Through extensive sampling, we first have been able to define a (probabilistically) safe region for the steps of the robot HRP2. We have been able to successfully use this region for online footprints correction in an experiment where HRP2 is guided by a human holding its hands, and where the footprint to be corrected corresponds to a reference position of the left or right foot relatively to the current position of the (resp. left or right) hand, which the robot endlessly tries to go back to. The figure 2 shows the results of this experiment, and the figure 3 describes the safe region we defined. Such a safe region can be seen in the chapter 5 of [11], but it is explained neither how it was obtained nor what are its limitations.

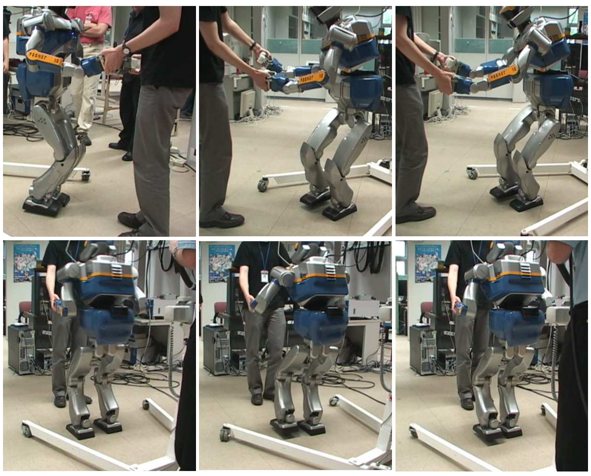


Fig.2 Experimental results

This first result was encouraging, but the ranges for orientation or position shifts are limited, and we know that HRP2 can perform much more efficient steps. Therefore we tried to design a new approximation algorithm in order to tackle the hard problem of approximation of an unknown function over a 6-dimensional space. A naive approach would probably require billions of samples before obtaining good results, which would take years to generate. Therefore, the reduction of the required number of samples was our primary concern.

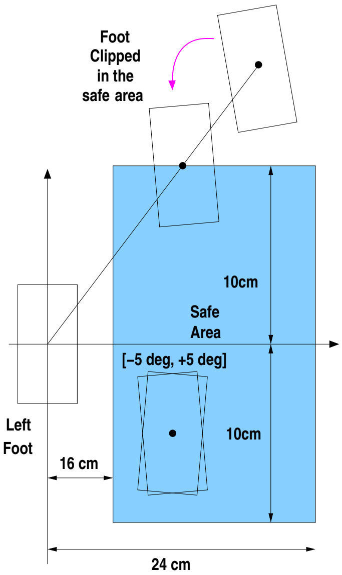


Fig.3 Safe region for a transition between two positions of the right foot

Our method combines decision trees and an original algorithm of adaptive sampling, as well as the resolution of QP problems. More precisely, a Decision Tree mechanism is used to recursively split the input space into disjoint cells. The importance of each cell is evaluated, and we sample points on them accordingly. When a small set of points has been sampled on a cell, we instantiate a QP problem, where the objective is to find a function in a vector space (in the work presented here, the vector space of affine functions) which minimizes the distance to the set of samples, subject to sign constraints: a positive sample must lead to a positive evaluation, a negative sample must lead to a negative evaluation. The results of these QP problems influence the further division of the cells, and therefore the further sampling. Hence, the QP problems and the Decision Tree structure combine nicely to permit an adaptive sampling which focuses on critical regions (where the cells will consequently be split faster).

Figures 4 and 5 depict a preliminary result, and show how is calculated the safe arrival area for a given initial position, with the additional assumption that there is no change of the orientation. Figure 4 shows the mechanism of the decision tree which recursively divides the input space into a disjoint union of rectangular cells. It also shows the negative samples (unfeasible steps); we can see some negative samples in what one would picture as the feasible area: this is probably due to some unstability in our simulation process. These mistakes induce a waste of time in the computation since the algorithm “thinks” that a new frontier has been found. Several techniques used in

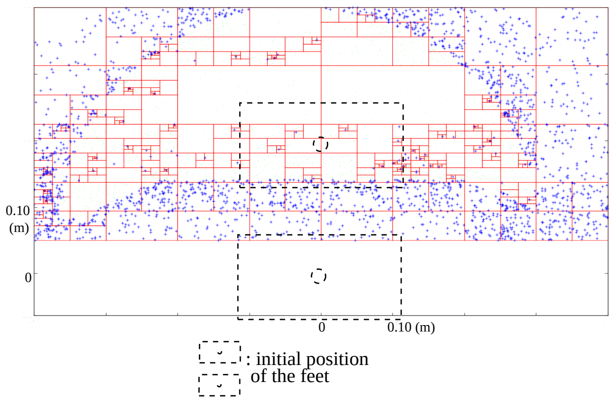


Fig.4 The construction of the approximation

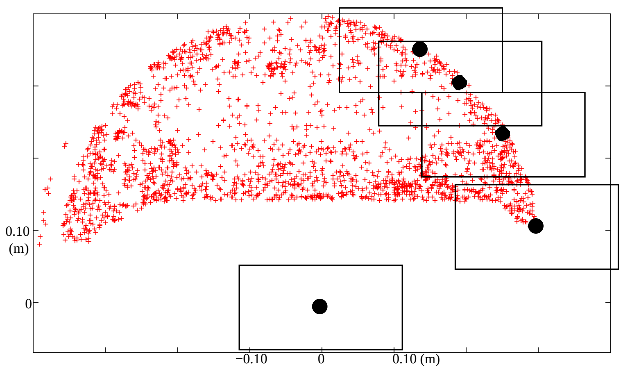


Fig.5 The positive samples generated and some feasible steps

machine learning for error tolerance might be helpful if we cannot make our simulation process more reliable. Figure 5 shows the positive samples. We can see that they are concentrated near the frontier between feasible and unfeasible arrival footprints, which is indeed the region on which a learning algorithm should focus. Nevertheless, even with adaptive sampling, our further attempts showed that approximating the full 6-dimensional space might be an impossible task regarding the amount of computation time we dispose of. This remark motivates our search for a new approach.

6. Conclusion and Further Work: the mixed approach

Considering the main problems of the two possible approaches previously presented, our conclusion is pretty straightforward and oriented towards a natural compromise: the solution to our problem might be found with a good combination of both discrete and continuous approaches: a mixed approach which would not have the two fundamental drawbacks of the discrete approach, and would avoid the insurmountable computation time needed for the continuous approach. Such a mixed approach, combining graphs construction and optimized machine learning, is what we are currently trying to design.

7. ACKNOWLEDGMENTS

This work was supported by grants from the R-BLINK Project, Contract ANR-08-JCJC-0075-01, and the ROBOT@CWE EU CEC project, Contract No. 34002 under the 6th Research program.

REFERENCES

[1] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa: "An Analytical Method on Real-Time Gait Planning for a Humanoid Robot," Proc. of IEEE-RAS/RSJ Int. Conf. on Humanoid Robotics, 2004.

[2] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich and F. Pfeiffer: "A Collocation Method for Real-Time Walking Pattern Generation," Proc IEEE-RAS Int Conference on Humanoid Robotics (Humanoids), 2007

[3] 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," Proc. of 2007 IEEE Int. Conf. on Robotics and Automation, 2007.

[4] 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," Proc. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Volume 1, 29 Oct.-3 Nov. 2001 pp. 239 - 246 vol.1

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa: "Biped walking pattern generation by using preview control of zero-moment point," Proc. of IEEE International Conference on Robotics and Automation (ICRA2003). pp. 1620-1626.

[6] M. Vukobratovic, B. Borovac: "Zero-Moment Point-Thirty Five Years of its Life," Int. Jour. of Humanoid Robotics, Vol. 1, No.1, March 2004, pp. 157-173

[7] S.M. LaValle and J.J Kuffner: "Rapidly-exploring random trees: Progress and prospects," In Robotics: The Algorithmic Perspective. 4th Int'l Workshop on the Algorithmic Foundations of Robotics (WAFR)., March 2000. A. K. Peters

[8] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba and H. Inoue: "Motion Planning for Humanoid Robots," Proc. 11th Intl Symp. of Robotics Research (ISRR 2003)

[9] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami: "Planning Biped Navigation Strategies in Complex Environments," Proc. 2003 IEEE Intl Conf. on Humanoid Robotics (Humanoids 2003)

[10] J.-M. Bourgeot, N. Cislo and B. Espiau: "Path-planning and tracking in a 3D complex environment for an anthropomorphic biped robot" Proc. of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems

[11] J. Chestnutt: "Navigation Planning for Legged Robots," doctoral dissertation, tech. report CMU-RI-TR-07-44, Robotics Institute, Carnegie Mellon University, December, 2007