Experiments on whole-body manipulation and locomotion with footstep real-time optimization
Dang Duong, Jean-Paul Laumond, Florent Lamiraux

To cite this version:
Dang Duong, Jean-Paul Laumond, Florent Lamiraux. Experiments on whole-body manipulation and locomotion with footstep real-time optimization. Rapport LAAS n° 12484. 2012. <hal-00727600v1>

HAL Id: hal-00727600
https://hal.archives-ouvertes.fr/hal-00727600v1
Submitted on 4 Sep 2012 (v1), last revised 2 Nov 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Experiments on whole-body manipulation and locomotion with footstep real-time optimization

Duong Dang, Florent Lamiraux and Jean-Paul Laumond

Abstract—This paper focuses on the experiments on the HRP-2 humanoid robot using the framework of manipulation and locomotion with real-time footstep adaptation. Two classes of experiments are presented. On the one hand, a grasping task at various height level to illustrate a whole-body task in combination with locomotion. On the other, stepping over obstacle experiments illustrate the particularity of humanoid robots. In all presented examples, footsteps are considered as a part of the robot’s kinematic chain and are resolved as an optimization problem along with other degrees of freedom of the robot. The environment is perceived by the stereo vision system mounted on the robot which closes the loop with the control through a online footstep adaptation scheme.

Index Terms—locomotion, footsteps, adaptation, reactive, real-time, visual servoing

I. INTRODUCTION

A. Problem statement and related works

THE high degree of redundancy and legged locomotion are two of the particularities that make the humanoid robotics research field both challenging and exciting. Either with a small size, miniature like robot or with a large size humanoid such as HRP-2, the researcher is provided with a formidable platform that is complex, highly redundant and capable of performing a large set of manipulation tasks. With legs, a humanoid robot can access to an environment specifically built for human, interact with that environment in an interesting way, such as climbing up a stair case, jumping over obstacles, tasks that cannot be done by other type of robots, such as, say a wheeled mobile robot. If numerous works have been carried out in both manipulation and locomotion, the two aspects are usually treated as independent problems. Whole-body tasks are often considered completely separate from the footsteps.

Walking and running locomotion has been studied by a number of research group, [1]–[7], notably with the introduction of the Zero Moment Point (ZMP), the analysis of the cart model and the inverted pendulum.

On manipulation side, task-based methods have been developed since the eighties of the last century for industrial robot and robotic arms, [8], [9]. These methods have been extended to humanoids in recent years as more and more robots have been made available for research [10]–[12].

On-line generation of footsteps have been studied previously by several research groups [13]–[17]. These online footstep generation methods use search algorithms, which are good at finding a feasible solution without an emphasis on the optimality of the problem. This might results in unpredictable and “unnatural” footsteps and in practice leads to failure during experiment due to physical limitation of humanoid robots. In addition, whole-body manipulation is not integrated into the stepping decision.

Kanoun et al. [18] has the idea of considering footsteps as parts of the robot kinematics and is driven by “whole-body” tasks. The notion of “whole-body” in this case is expanded to the virtual degrees of freedom related to locomotion. Footstep placement is then resolved in an optimization problem, in harmony with the upper-body movement. This way of reasoning about locomotion has been put together into a framework presented in [19] which combines the manipulation, locomotion with a reactive footstep adjustment scheme in closed loop with perception. The goals of this framework are:

- Seamlessly integrate locomotion with whole body movement. Footsteps are considered as part of the robot and are dictated by the task applied to the augmented robot.
- Build a reactive scheme that helps the robot achieve the task even if the environment is changed during execution.
- Resolve the foot placement by optimization so that it preserves the optimality, hence, the high feasibility of the movement.
- Integrate with on-robot stereo vision to make the movement the most robust and portable possible.
Moreover, combined with a prior motion planning step, the method is less subject to local minima than classical numerical optimization approaches.

**B. Contribution**

This paper is the follow up of the primarily work introduced in [19]. The laid out framework is demonstrated in a number of new experimental situations (Figure 1). The implementation of stereo-vision on the HRP-2 robot is also improved to achieve tasks such as precise object grasping.

In addition, a new type of experiments is introduced (section III-B) illustrating the real-time footstep adjustment scheme in the typical context of humanoid robots, i.e. stepping over objects.

The representation of footsteps as the robot’s extra degrees of freedom can be used to calculate the initial footsteps as well as to adapt these footsteps on the fly during the experiment. The framework is flexible enough to take as input any initial footsteps sequence and adapt them in real-time.

**II. APPROACH AND TOOLS**

Figure 2 depicts the global architecture of the framework. The planner plays the role of a “visual servo” for footsteps. It optimizes the stepping sequence in real-time and in closed-loop with the vision system. The controller takes as input the information from the visual servo and resolves the prioritized hierarchy of the corresponding primary tasks to send command to the robot in real-time. The perception system includes an automatic calibration process which improves precision and allows the framework to perform precise tasks such as grasping.

**A. Perception**

The tracking method on the robot is the broadly used CAMShift [20] algorithm. The tracked 2D object is then projected into the PCL [21] point cloud. Once outliers have been filtered out, one obtains the 3D-points on the object, hence its estimated position (Figure 3 4.).

**Automatic extrinsic parameter calibration:** One major modification of the perception module is a better extrinsic parameter calibration. An automatic process has been developed which involved moving a chessboard fixed to the hand of the robot. During calibration, the robot hand was moved inside the vision field of the robot. (Figure 5) The recorded poses of the chessboard and corresponding joint angles are then recorded. The data is processed and fed to a calibrator using Tsai et. al. algorithm [22]. This automatic calibration process helped significantly improve the performance of the vision system and allowed the robot to achieve tasks with better precision. The calibration process is available as a ROS package on the paper website.

**B. Step deformation by localstepper**

In the same spirit the “elastic band” introduced by [23], [24] which connected path-planning and control for wheeled mobile robots, this framework uses optimisation to reactively build and adjust footsteps in real-time, hence provides the control corrections in a timely manner.

The core of localstepper is presented by Kanoun et al. [18]. The main idea here is to consider each footstep as a virtual link with three degrees of freedom (Figure 6). The augmented robot will be then resolved with the prioritized set of task, subjected to various constraints such as self-collision, obstacle avoidance, etc. With the introduction if inequality tasks, the constraints simply become tasks and are added to the prioritized set at the highest priorities. Let $J_i$ and $e_i$ are the Jacobians and errors corresponding to task $i$, in an hierarchy of $k$ successive tasks, the resolution of the robot state vector $q$ is summarized as follows

**Fig. 3. 2D tracking**

**Fig. 4. Projection into the point cloud**

**Fig. 5. Hand-eye calibration process**
Find $\dot{q}^* \in S_k$:

$$S_0 = \mathbb{R}^n$$

$$S_i = \text{arg} \left\{ \min_{\dot{q} \in S_{i-1}} \frac{1}{2} ||J_i \dot{q} - e_i||^2 \right\} \text{ for equality tasks}$$

$$S_i = \text{arg} \left\{ \min_{w, \dot{q} \in S_{i-1}} \frac{1}{2} ||w||^2 \quad \text{s.t.} \quad J_i \dot{q} - e_i \leq w \right\} \text{ for inequality tasks}$$

At the end of the process, (Figure 7) one obtains from the resulting $q$:

- The footprints (the first $3n$ terms in $q$ if the robot was to perform $n$ steps).
- The final posture (the rest of the terms).

C. Stack of Tasks

The controller in the framework is the StackOfTask, [25]–[27]. The role of this controller is to find out, given a prioritized stack of tasks and within the limit of the control cycle, the control law $\dot{q}_i$.

In the case of a single task, this control law is simply

$$\dot{q}_i = -\lambda J_i^+ e_i$$  \hspace{1cm} (1)

$J_i^+$ is the pseudo-inverse of the Jacobian $J_i$. $e_i$ is the difference between desired feature $s_i^*$ (i.e. a position in operational space, a posture, etc.) and its current value $s_i$.

$$e_i = s_i - s_i^*$$  \hspace{1cm} (2)

The control law on a prioritized set of tasks is written as [28]:

$$\dot{q}_i = \dot{q}_{i-1} + \lambda_i J_i^+ (\dot{e}_i - J_i \dot{q}_{i-1}), \quad \dot{q}_1 = \lambda_1 J_1^+ \dot{e}_1$$  \hspace{1cm} (3)

when $J_i$ is the projection of $J_i$ in the null space of the augmented Jacobian

$$J_i^A = [J_1, J_2, \ldots J_{i-1}]^T$$  \hspace{1cm} (4)

$$\tilde{J}_i = J_i P_i^A, \quad P_i^A = I - (J_i^A)^+ J_i^A$$  \hspace{1cm} (5)

$J_i^+$ is simply $J_i^A$. One recovers (1) if there is one task in the stack. This formulation guarantees that the task at $i$th stage does not disturb the previous tasks, i.e. with higher priority.

D. Pattern generator

The output of the planner, i.e. footprints is fed to a real-time pattern generator presented by Stasse et. al. [29] with the underlying algorithm proposed by Morisawa et. al. [5], [6], [30]. The role of the pattern generator is to generate trajectories of operational points (feet, center of mass), which can be used directly by the controller, as well as the ZMP’s trajectory which is fed to the stabilizer on the robot.

III. EXPERIMENTS

Figure 2 shows the information flow during the experiments. The “Upper body tasks “ arrow can be omitted in section III-A and III-B as these experiments involve only footstep placements.

A. Stepping towards a goal

This is the first application of the localstepper concept. Suppose that the robot needs to go to a goal position while avoiding $r$ holes on the ground (Figure 8(a)). To achieve this goal, an initial sequence of $n$ footsteps is added to the robot. In localstepper frameworks, this is translated into $3n$ additional degrees of freedom added to the robot kinematic chain. (Figure 8(b)). Since the upper body is not subject to a specific task, the optimization problem presented in II-B will simplify and only act on the vector $q$ representing the additional $3n$ degree of freedom.
arg min \limits_{q \in \mathbb{R}^{3n}} ||X_{final\_step} - X_{goal}|| \quad (6)
subject to \quad q_i^{min} \leq q^i \leq q_i^{max}, \forall i \in 1, 2 \ldots 3n \quad (7)
\quad D_{self\_collision}^{ij} > 0, \forall i \neq j \in 1, 2 \ldots n \quad (8)
\quad D_{obstacle\_step}^{ij} > 0, \forall i \in 1, 2 \ldots n \quad \text{and} \quad j \in 1, 2 \ldots r \quad (9)

Where \(X_{final\_step}\) and \(X_{goal}\) are 3 dimensional vector representing footsteps \((x, y, \theta)\).

The resulting footstep (Figure 8(c)) is then executed by the robot. When the goal moves, the footsteps are updated during experiment by resolving the same optimization problem. Since the initial guess of the new optimization problem is the current solution, provided the goal moves at reasonable pace, the optimization process is quick and takes typically tens of milliseconds to compute, more than enough for the control to change the footsteps reactively (the stepping period on the HRP-2 robot is 0.8s.)

B. Stepping over obstacles

In this experiment, the task assigned to the robot is to overcome a long cylindrical bar. The bar is long enough and its unknown characteristics make it impossible for the robot to step upon. This example illustrates a main specificity of legged locomotion.

To achieve the assigned task, the robot has to step over the obstacle whose position is estimated by the stereo vision system mounted on the robot. As any stereo system, the precision of the estimated position gets better when the robot gets closer to the tracked object (bar). Moreover, the bar is also intentionally moved by a human during the experiment. As a result, either to take into account the updated perceived position or a real displacement of the object, there is a need of reactive footstep adjustment.

1) Compute initial stepping sequence: As localstepper takes initial footprints and initial robot configuration as inputs, a stepping sequence computed by any method can be fed to localstepper. For instance, the 3D swept volume method as describe in [13] which allows stepping over obstacles up can be used as the initial sequence.

2) Online deformation: As the perceived position of the obstacle is continuously updated plus the fact that the obstacle might be moved during the experiment; the footsteps have to be recalculated as fast as possible.

Provided that form of the obstacle is unchanged (long cylindrical bar with known diameter), the robot only needs to make sure that the two subsequent steps that cross the bar stay unchanged with respect to the bar. We then recover the same situation as described in III-A: stepping towards a moving target.

The footstep adjustment scheme for the stepping over experiment can be written as algorithm 1, when \(x_0, y_0, x\) are three-dimensional vectors in the footprint coordinate \((x, y, \theta)\).

Algorithm 1 Footstep adjustment for stepping over experiment

Require: current plan.
Ensure: new plan

1: \(x_0\) ← initial target step
2: \(y_0\) ← initial bar position
3: loop
4: \(y\) ← current bar position
5: new target \(x\) ← \(x_0 + y - y_0\)
6: recompute footsteps
7: end loop

3) Experiment: In the experiment depicted in Figure 9, the bar is marked by a color orange band and detected by the module detailed in II-A.

Thanks to the online optimization scheme, the robot has no problem crossing both a fixed and a mobile bar. Figure 10 describes how the bar is tracked during the movement, Figure 11 depicts the estimated position of the bar.

C. Grasping objects

1) Stepping as a side effect of upper-body task: As presented in [18], [19], one important point to be noted here is the fact that stepping is simply the result of the upper-body task, in this case, the grasping task. By considering footsteps parts of the robot’s kinematic chain, the resolution of the
optimization problem presented in II-B automatically deploys the footsteps. In the same fashion, footstep deformation is also a consequence of the modification of the grasping task (target physically moves or the perceived position changes as the robot approaches the target).

2) Online footstep optimization: It is assumed in this experiment that the grasping target moves but the environment around it stay intact, i.e. the ball is always on the ground or on the table and no new obstacle appears during the experiment. With that assumption, one can “freeze” the posture of the standing robot. The modification of the grasping task will only affect the footsteps. The computation time (Table I) is well below the stepping period (0.8s for HRP-2 in this case) and allows a reactive walking scheme.

3) Walking-grasping transition: In Figure 2, the visual servo has two parts

- localstepper which regenerates posture and footsteps.
- a servo which feeds directly the target into the grasping task.

The perception module returns a 3D goal position. The planner only outputs the final posture and desired footsteps. An additional step is needed for the robot to use these pieces of information to generate a movement on the robot.

In order to achieve a feasible, fluid, movement, a posture task is added to the StackOfTasks at the lowest priority. This task does not affect the stability of the system and prepares the robot in the grasping posture even before the last step (Figure 12).

For the grasping task, a cubic spline is used to make sure that the hand passes by the appropriate way-point to successfully grasp the object. Experiments of grasping with objects at different height and position have been carried out on the robot (Figure 12 and 13) with the possibility to move the object while the robot is walking.

The combination of reactive localstepper, the StackOfTasks and the stereo vision system on the HRP-2 robot forms a powerful toolbox that can be used in a large set of applications.

Footstep deformation by localstepper provides a quick robust footstep planner to adapt arbitrary input stepping sequences to deal with changes in the environment.

One limitation of this footstep adjustment scheme is that it resolves a local problem. If the environment changes drastically, the planner can be stuck in local minima. To amend this limitation, the framework has to be combined with a global step planner.

### Table I

<table>
<thead>
<tr>
<th>Goal modification (m)</th>
<th>Number of tests</th>
<th>Max (ms)</th>
<th>Min (ms)</th>
<th>Average (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>180</td>
<td>16</td>
<td>8</td>
<td>10.7</td>
</tr>
<tr>
<td>0.02</td>
<td>180</td>
<td>38</td>
<td>7</td>
<td>11.4</td>
</tr>
<tr>
<td>0.05</td>
<td>180</td>
<td>12</td>
<td>9</td>
<td>11.0</td>
</tr>
<tr>
<td>0.1</td>
<td>180</td>
<td>48</td>
<td>9</td>
<td>15.5</td>
</tr>
<tr>
<td>0.2</td>
<td>180</td>
<td>23</td>
<td>11</td>
<td>20.0</td>
</tr>
<tr>
<td>0.5</td>
<td>180</td>
<td>117</td>
<td>15</td>
<td>33.6</td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS

FIG. 11. Perceived position of the bar (x and y components) which is moved while the robot is walking (around second 12)

FIG. 12. Walking and grasping an object at table level on HRP-2

FIG. 13. Walking and grasping an object at ground level on HRP-2

FIG. 12. Walking and grasping an object at table level on HRP-2

FIG. 13. Walking and grasping an object at ground level on HRP-2

PREPRINT SUBMITTED TO 2012 IEEE-RAS INTERNATIONAL CONFERENCE ON HUMANOID ROBOTS.

RECEIVED JULY 6, 2012.
REFERENCES


