Robust 3D Walking Using Inverse Dynamics And Footstep Planning With Model Predictive Control

Salman Faraji, Soha Pouya and Auke Jan Ijspeert
Biorobotics Laboratory, EPFL, Lausanne, Switzerland
salman.faraji@epfl.ch, soha.pouya@epfl.ch, auke.ijspeert@epfl.ch

1 Motivation

Controlling legged robots is challenging due to the fact that they should keep balance and perform locomotion. Bipeds are even more prone to failure since the support region for their Center of Mass (CoM) is limited. Although many methods exist in literature that keep the CoM inside the support region during locomotion, the resulting statically-stable walking is not fast or natural like humans. Dynamically-stable walking however produces faster and more natural motions as it lets the CoM fall outside the support region, while the robot takes proper steps to maintain balance dynamically and walk. Challenging aspects of this approach are proper prediction of the future states, avoiding tilting or rolling of feet, fast motions, using heels and toes for faster walking, keeping knees straight in order to avoid consuming unnecessary energy, human-like CoM height profile and finally being compliant to softly handle impacts produced at each touchdown. These aspects are in addition to considering physical constraints like torque limits, frictions and self collisions. In this work we want to propose a generic method that needs minimal tuning, covers a wide range of speeds and does not rely on off-line optimizations. We aim at reducing the problem size from joint to Cartesian space in order to avoid dealing with high dimensional joint trajectories and their complicated geometrical consistency. The proposed method should be computationally affordable and robust against various sources of perturbations internally or externally.

2 Our Approach

Here we break down the controller into three layers where we simplify the algorithm step by step.

First Layer: In this layer we generate actuator torques given the desired motion in the Cartesian (task) space. Various methods in the literature use kinematic models of the robot to convert Cartesian velocities [4, 3] or forces [1] to actuator variables. For our torque controlled robot Coman [12], considering the dynamics-model of the robot however will help us calculating the required torques when targeting agile and versatile motions. Although large feedback gains will increase the tracking accuracy, they make the robot less compliant. Adding feed-forward torques will help to achieve the same preciseness with smaller feedback gains, still being able to reject perturbations and behaving more compliantly. Inspired from [9] we use task-space inverse dynamics to generate the joint torques given Cartesian accelerations. However similar to [10] we use the whole body optimization approach instead of pseudo-inversion as it enables us to take into account the torque limits, the CoP region and friction polyhedrals as inequality constraints. In this method, one optimizes contact forces, joint accelerations and torques at the same time under aforementioned constraints, together with the equation of motion and joint/Cartesian space acceleration relations as equality constraints. Our problem is efficiently formulated in a fast quadratic optimization library called CVXGEN [7] which solves it per time-step in less than $1.2ms$. Introduction of soft constraints makes our method slightly different from [11] and [10] in the sense that we can accept infeasible input accelerations and still produce a feasible motion.

Second layer: Our whole-body optimization in the first layer helps us planning trajectories for the CoM and feet in the task-space with reduced dimensions. Thus in the second layer of our controller, we define arc trajectories for swing foot and track them by PD controllers which produce Cartesian accelerations. Various methods like [6], [2] and [8] optimize these trajectories off-line which makes their method harder to generalize for other environment conditions. However we use soft transitions and SLERP (spherical linear interpolation) functions to smoothen these trajectories. The input to this layer is the next footstep location that is determined by the third layer of the controller.

Third layer: One needs to simplify the robot's model to predict the future motion with less computational cost during dynamic walking. A common approach is to use simple inverted pendulums or with springs, ankles or inertia mass. Spring Loaded Inverted Pendulum (SLIP) model is usually used to predict running. Although Mordatch [8] for example has derived a closed form solution for evolution of the CoM state with polynomial approximations, such nonlinear prediction is computationally expensive. For the purpose of walking however, Linear Inverted Pendulum Models (LIPM) appear more frequently in the literature and they enable simpler or even linear predictions of motion if no inertia is assumed for the mass. Some methods have control over the CoP trajectory during the swing phase to follow the LIPM model [6, 4]. However they rely on ankle torques to modulate the motion which is less robust in case of arbitrary foot shapes, unstructured terrains or contacts made of soft materials. Similar concept is used in [5] for 1-step prediction. The swing foot touches down in a location where the energy of the robot is captured after being pushed and thus going to rest condition.

In [3], the top level planner uses LIPM and controls both the CoP trajectory and multiple footstep locations which is more elaborated and robust, but still being linear and in closed form. This method solves a Model Predictive Control (MPC) problem which optimizes a sequence of footsteps and CoP trajectories that induce a desired given reference speed. However relying on ankle torques has previously-mentioned drawbacks, in addition to the fact that the low level control method in [3] is kinematics-based and less compliant.

Inspired by this work, we have formulated a similar MPC problem which merely optimizes footstep locations assuming no ankle for the LIPM unlike [3]. Given reference forward and steering speeds via a joystick, the third layer forms a reference footstep plan. It then optimizes future states of the robot and footsteps that act as discrete inputs to this system, regarding the reference footstep plan and the LIPM model. The first footstep location is then used to calculate swing arc trajectories in the second layer. With such novel linear and closed form simple formulation which is again implemented in CVXGEN, we are able to adjust future footsteps in every time-step. Together with the low-level compliant controller, our method is responsive to pushes or other large perturbations by taking proper footsteps. It only uses ankle torques for smaller perturbations which is done compliantly in the low-level controller, not in the planning level.

3 Results and Discussions

The proposed controller is able to produce a wide range of forward speeds from $-0.2m/s$ to $0.4m/s$ on our simulated kid-sized robot Coman which has only around $1m$ height. This performance is comparable with methods like [2] which produce $1.14m/s$ on Atlas robot which has a height of around $190cm$. However we do not have heel touchdown and toe lift-off phases unlike [2] which restrict our motion geometrically regarding ankle joint limits. Our robot is also able to rotate about the yaw axis with maximum speed of $\pm 1rad/s$ which is notably more than similar methods like [3]. However walking forward and steering at the same time can not be done at these maximum speeds due to large swing motions of feet. Our method can dynamically recover from strong pushes of $10N$ lasting $0.3s$ on the kid-sized robot Coman. It can also perform blind walking robustly over slopes of -10^o to 15^o and an uneven terrain with $\pm 5cm$ height variations. We have also verified its robustness against internal perturbations such as sensor noises up to 3^o of standard deviation, communication loops with $10ms$ delays and various types of model errors like shifting the CoM $1cm$ or making different parts of the body $1kg$ heavier.

In summary, we have transferred the joint control problem to task space in the first layer with our whole-body inverse dynamics formulation. We have also proposed an on-line simple trajectory planning method to replace the use of off-line optimizations. Finally in the third layer which is the novelty of this work, we have formulated a discrete-time MPC problem that optimizes future footsteps of the robot in closed form and linearly. Our controller is compliant and precise while considering various physical constraints in its first layer. It covers a wide range of speeds and performs robustly when subject to various internal or external perturbations. The method is generic and we have applied it on another humanoid robot (Atlas) in simulations, achieving similar performance [13]. All movies of this work could be found on http://biorob.epfl.ch/page-99800-en.html.

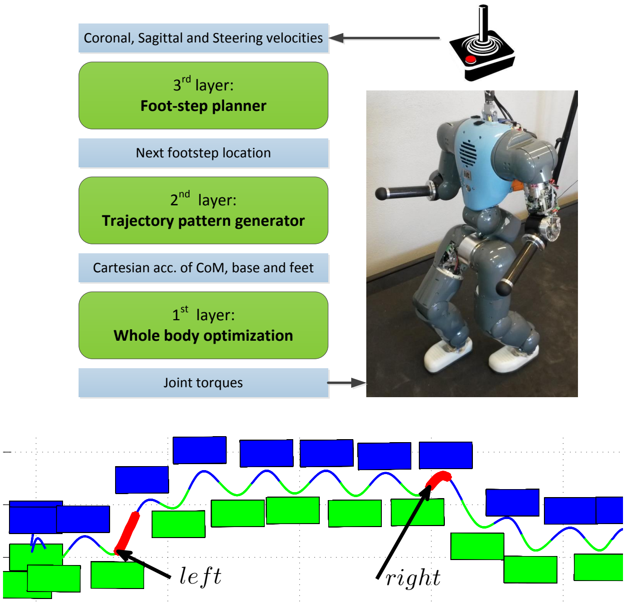

Figure 1: Top: The Coman robot and our hierarchical control architecture. Bottom: The walking sequence at $0.2m/s$ where the robot is pushed twice by a $10N$ force lasting $0.3s$. It recovers by taking proper correcting footsteps.

4 Acknowledgments

This project is funded under the European Community's 7th Framework Programme: FP7-ICT 611832 (WALK-MAN). Cognitive Systems and Robotics: FP7-ICT-2013-10.

References

[1] J. Pratt et. al., (2001), I. J. of Robotics Research.
[2] S. Feng et. al., (2013), Humanoids Conf.
[3] A. Herdt et. al., (2010), IROS Conf.
[4] J. Englsberger et. al., (2011), IROS Conf.
[5] T. Koolen et. al. (2012), I. J. of Robotics Research.
[6] S. Kuindersma et. al., (2013), In arXiv.
[7] J. Mattingley et. al., (2012), vol. 13, Springer.
[8] I. Mordatch et. al., (2010), ACM T. on Graphics.
[9] L. Sentis et. al., (2010), IEEE T. On Robotics.
[10] A. Herzog et. al., (2013), In arXiv.
[11] B. Stephens et. al., (2010), IROS Conf.
[12] E. S. Papastavridis, (2013), IEEE I. C. Mechatronics.
[13] S. Faraji et. al., (2014), ICRA Conf.