German Journal on Artificial Intelligence (KI) 29(4): 401-405, Springer, 2015.

Noname manuscript No.
(will be inserted by the editor)

Online Learning of Bipedal Walking Stabilization

Marcell Missura · Sven Behnke

Received: 2015-02-25 / Accepted: [date]

Abstract Bipedal walking is a complex whole-body motion with inherently unstable dynamics that makes the design of a robust controller particularly challenging. While a walk controller could potentially be learned with the hardware in the loop, the destructive nature of exploratory motions and the impracticality of a high number of required repetitions render most of the existing machine learning methods unsuitable for an online learning setting with real hardware. In a project in the DFG Priority Programme Autonomous Learning, we are investigating ways of bootstrapping the learning process with basic walking skills and enabling a humanoid robot to autonomously learn how to control its balance during walking.

Keywords Online Learning · Bipedal Walking · Push Recovery

1 Introduction

Using machine learning is thought to be a promising approach to produce a capable bipedal walk controller. In simulation, artificially evolved muscle reflex control [1] can produce a convincingly natural looking bipedal walk. However, as the motions require hours to days on a large number of cores to optimize, the applicability to real robots is not foreseeable. When a real robot is in the loop, the feasible number of trials and the risk of damaging the hardware become limiting factors, and dictate that the learning process must reach a reliable walking performance after only a low number of experiences. Successful learning projects on real hardware to date typically start with an already stable walk and optimize the walking speed or stability during execution [8, 2], or learn the parameters of a motion skeleton such that some form of stable walking is achieved within a feasible amount of iterations [7]. Balancing has mostly been ignored in the context of machine learning so far.

We investigate a learning method that incorporates a Central Pattern Generator (CPG) to produce coordinated stepping motions and an analytic balance controller that initializes the learning process with a robot that already has a concept of balance. The learning algorithm is executed online and improves the walking capabilities of a biped using the feedback it gains from every step the robot makes. A simple physical model is exploited to gain an approximate gradient that boosts the learning performance to real hardware feasibility.

2 Gait Control Framework

Our online learning concept is embedded into the bipedal gait control framework shown in Figure 2. The focus of the ongoing research is the Learning Control component. The other components are already in place. The robot itself (bottom right) is part of the loop. It receives motor targets from the control software and provides sensor data about its internal state. A low-level CPG [4] is used as a Motion Generator (top right) of open-loop stepping motions. The CPG hides the complexity of the full-body walking motion and exhibits parameters to control the size and the timing of the steps. The State Estimation module (bottom left) estimates the angle and the angular velocity of the trunk and re-

This work has been supported by grant BE 2556/6-1 of the German Research Foundation (DFG)

Marcell Missura · Sven Behnke
Rheinische Friedrich-Wilhelms-Universität
Friedrich-Ebert-Allee 144, 53113 Bonn
E-mail: missura@ais.uni-bonn.de

constructs a tilted full-body pose of the robot. From the pose reconstruction the step size can be measured by computing the distance between the feet at the end of the step, and low-dimensional features are extracted that serve as inputs for the analytic and the learning balance controllers.

In the Analytic Control module (top middle), closed-form mathematical expressions of the Linear Inverted Pendulum Model (LIPM) are used to compute the timing and the location of the next footstep in order to keep the robot balanced while obeying a step size commanded by a higher control instance. In its core, the Analytic Control drives the Center of Mass (CoM) towards a limit cycle by means of Zero Moment Point (ZMP), step timing, and foot placement control strategies. Figure 1 and a video[1] demonstrates the capabilities of the analytic footstep controller. More information about its implementation is given in [5].

The Learning Control (top left) improves the performance of the robot in terms of balance and reference tracking by learning offsets to the step size and step timing outputs of the analytic footstep controller. The offsets are learned based on measurable errors the robot makes during walking. Errors can stem from imprecise actuation, latency, and the imperfection of the low-dimensional analytic balance model.

3 The Online Learning Process

Our concept of learning a bipedal walk controller [6] hinges on a strong reduction of the input and output dimensions of the learning task and the initialization of walking skills. The use of the CPG to generate stepping motions reduces the high-dimensional task of learning whole-body control to a low-dimensional task of learning only Cartesian footstep coordinates and the step timing. Furthermore, we decompose the learning task by learning the sagittal footstep coordinate, the lateral footstep coordinate, and the step timing as independent instances of even lower dimensionality. The dimensional decomposition approach has already been a key concept for the implementation of the analytic controller [3]. We investigate the learning of a walk control function, which is formally defined as

$$(\Delta\mathbf{S}, \Delta\mathsf{T}) = \mathcal{W}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \check{\mathbf{S}}). \tag{1}$$

The walk control function $\mathcal{W}$ receives the trunk angle and angular velocity $(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ in pitch and roll directions, and the commanded step size $\check{\mathbf{S}}$ as inputs, and outputs a step size offset $\Delta\mathbf{S}$ and a step timing offset $\Delta\mathsf{T}$ that are added to the output of the analytic controller. We represent the walk control function $\mathcal{W}$ with a function approximator and train it during the control process, as illustrated in Figure 3. The walk control function is initialized with a value of zero for all outputs. At the end of each step, the trunk angle $\boldsymbol{\theta}_\mathsf{E}$ is measured as an indicator of balance, and the step size $\mathbf{S}_\mathsf{E}$ as an indicator of the reference tracking error. We compute a gradient function $\mathcal{G}(\boldsymbol{\theta}_\mathsf{E}, \mathbf{S}_\mathsf{E})$ based on the pendulum-cart model that resembles the angular dynamics of a biped and suggests a change of the step size. Then we update the function approximator with the update rule

$$\mathcal{W}(\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i, \check{\mathbf{S}}_i)_{k+1} = \mathcal{W}(\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i, \check{\mathbf{S}}_i)_k + \eta\, \mathcal{G}(\boldsymbol{\theta}_\mathsf{E}, \mathbf{S}_\mathsf{E}), \forall i \in \mathrm{I}, \tag{2}$$

[1] http://youtu.be/PoTBWV1mOlY

Fig. 1 Push recovery with bipedal robot Dynaped.

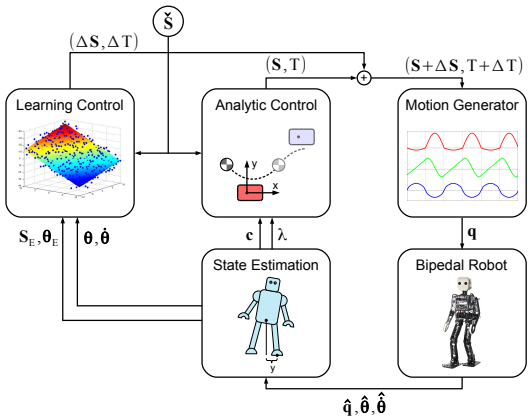

Fig. 2 Overview of our gait control framework. The input into the gait control framework is the desired step size $\check{\mathbf{S}}$ commanded by a higher layer. The Analytic Control (top middle) and the Learning Control (top left) both have the task to obey the commanded step size while maintaining the balance of the biped. The Analytic Control computes the time T and the location $\mathbf{S}$ of the next footstep with the Linear Inverted Pendulum Model. The Learning Control component observes the errors the robot makes during walking and learns the corrective offsets $\Delta\mathbf{S}$ and $\Delta\mathsf{T}$ to the outputs of the Analytic Control. Both controllers use low-dimensional features extracted from a whole-body pose reconstruction by the State Estimation component (bottom left). The Motion Generator (top right) generates joint position targets $\mathbf{q}$ for a timed stepping motion towards the desired footstep coordinates.

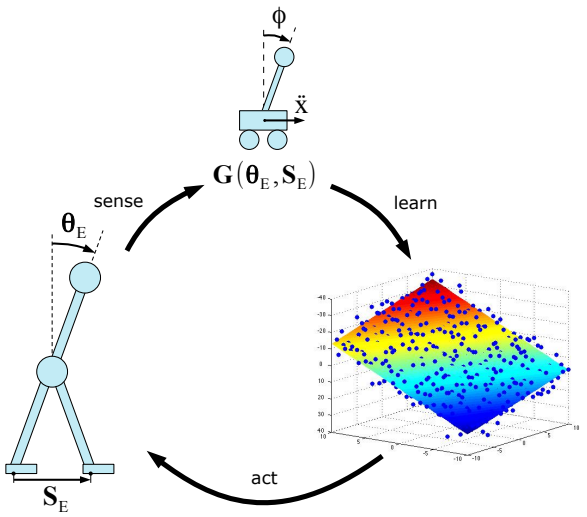

Fig. 3 The learning process uses the trunk attitude θ_E and the step size S_E at the end of the step to infer a gradient $\mathcal{G}(\theta_E, S_E)$ that suggests a change of the step size. The gradient is used to update a function approximator that takes charge of controlling a balanced walk.

where η is a learning rate, and $\{\theta_i, \dot{\theta}_i\}, i \in I$, is the set of trunk angles and angular velocities that were measured during the step. In words, we query the function approximator at the locations that were seen during the step, add the gradient to the resulting values, and present the results as the new desired outputs to the walk control function approximator.

The main control loop queries the function approximator with a high frequency—typically 100 Hz—to drive the walking motion. The function approximator has to deliver a time-critical response, even when it is being updated with new data. Neither the response time nor the memory consumption of the function approximator should degrade with the ever increasing amount of seen data, otherwise the learning process will eventually have to terminate. Gaussian processes, regression trees, and random forests, all degrade when used in an incremental learning setting. The Locally Weighted Projection Regression (LWPR) algorithm represents a function with a bounded number of locally linear kernels, such that old training data can be discarded. Thus, the memory consumption, update times, and recall times are bounded. We use an open-source implementation[2] that fully satisfies our requirements.

4 Experiments

We performed experiments with learning the sagittal step size on a simulated biped and evaluated different aspects of the learning process. We evaluated whether the learning component is able to improve the overall walking stability. We applied 400 randomly timed push impulses directed in the forward direction to the back of a robot walking in place. The magnitudes of the impulses were sampled from a range that included strong enough pushes that forced the robot to make forward steps in order to avoid falling. By dividing the number of falls by the number of pushes for several ranges of impulses, we estimated the probability to fall depending on the magnitude of the disturbance. The results are shown in Figure 4. In addition to the *analytic* and the *learned* balance augmentation, we also included an open-loop controller that walks in place with a fixed frequency and does not react to the pushes. The analytic controller significantly increases the push resistance compared to what the robot can absorb passively. Our online learning technique increases the stability even further.

We evaluated the ability of the robot to return to a reference step size after a disturbance. We commanded the robot to walk forward with a fixed step size. We pushed the robot repeatedly forward with an impulse of

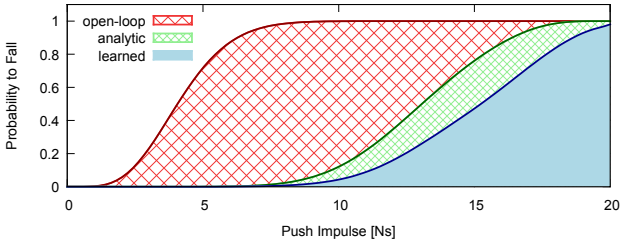

Fig. 4 Probability to fall of an open-loop, analytic, and a learned controller with respect to varying push impulses from the back.

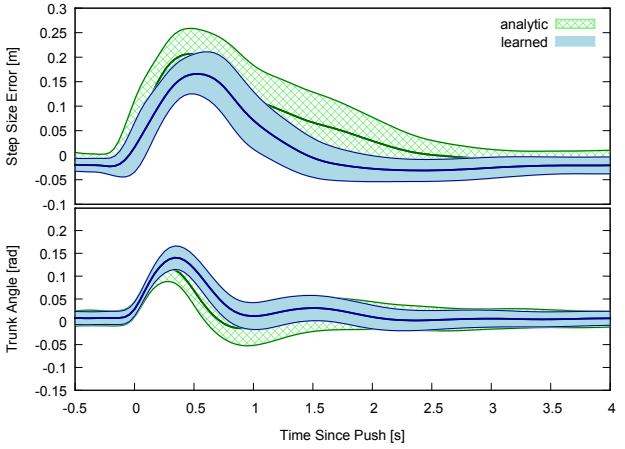

Fig. 5 The robot learned to react to a push with a smaller step size error than with the analytic controller by allowing a larger, but tolerable tilt. The robot is able to return faster to the reference step size with the learning component enabled.

[2] http://wcms.inf.ed.ac.uk/ipab/slmc/research/software-lwpr

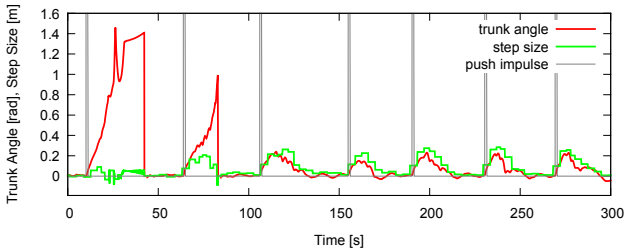

Fig. 6 Even if not initialized with the analytic controller, the learning framework manages to learn how to stabilize the robot after only three pushes. The pushes are indicated by the vertical lines.

a constant magnitude that is strong enough to force the robot to adapt its foot placement, but not too strong for the *analytic* and the *learned* balance controllers to handle. The pushes were triggered at random times in order to hit the robot in different phases of the walking motion. Synchronized at the moment of the push, Figure 5 shows the mean and the variance of the step size error and the trunk angle after the push. Both of these quantities return to their reference values. The learned controller reacts to the pushes with a smaller step size error than the analytic controller by utilizing a larger, but tolerable inclination of the body. The learned controller also returns faster to the reference step size.

Finally, we evaluated the potential of our learning approach with an experiment that is focused on the speed and robustness of learning. In this experiment we did not use the analytic controller for initialization. The robot starts with stepping in place and no prior knowledge of step size control. We disturb the robot with push impulses from the back and observe how quickly the robot learns to absorb the push without falling. The result of the experiment is shown in Figure 6. The first two pushes made the robot fall, but the controller learned from this experience and managed to stabilize the robot already on the third push. After the third push, the learning process has mostly settled and the controller has learned how to balance after the push.

5 Conclusion

We identified the initialization of walking skills and the reduction of the complexity of the learning problem as key ingredients for successfully learning a bipedal walk controller under real hardware conditions. By investing a simple model assumption, we limited the competence of the learning algorithm to inverted pendulum-like balancing tasks, but we gained a competitive learning performance that is able to balance a humanoid robot after a strong push based only on the experience of a few failed recovery steps.

In future work, we intend to complete all learning components for the footstep coordinates and the timing of the steps, and to investigate how much learning and walking performance can be achieved with isolated learners. By enabling the robot to disturb itself, we plan to give life to a learning instance that autonomously explores and learns its own balance. In the successor project, we intend to investigate learning of walking in rough terrain with restricted footholds.

References

1. Geijtenbeek T, van de Panne M, van der Stappen AF (2013) Flexible Muscle-Based Locomotion for Bipedal Creatures. ACM Transactions on Graphics
2. Kohl N, Stone P (2004) Policy gradient reinforcement learning for fast quadrupedal locomotion. In: IEEE Int. Conf. on Robotics and Automation
3. Missura M, Behnke S (2013) Omnidirectional Capture Steps for Bipedal Walking. In: IEEE-RAS Int. Conf. on Humanoid Robots
4. Missura M, Behnke S (2013) Self-Stable Omnidirectional Walking with Compliant Joints. In: Workshop on Humanoid Soccer Robots, Atlanta, USA
5. Missura M, Behnke S (2014) Balanced walking with capture steps. In: RoboCup 2014: Robot Soccer World Cup XVIII (to appear), Springer
6. Missura M, Behnke S (2014) Online Learning of Balanced Foot Placement for Bipedal Walking. In: IEEE-RAS Int. Conf. on Humanoid Robots
7. Morimoto J, Atkeson CG (2007) Learning biped locomotion. In: IEEE Robotics and Automation Magazine, IEEE
8. Tedrake R, Zhang TW, Seung HS (2005) Learning to walk in 20 minutes. In: 14th Yale Workshop on Adaptive and Learning Systems

Marcell Missura is a PhD candidate at the Autonomous Intelligent Systems group from the University of Bonn. He is a fivefold world champion in the Humanoid League of RoboCup and twofold winner of the Louis Vuitton Best Humanoid Award. His research interests include bipedal walking, push recovery, and footstep planning.

Sven Behnke obtained his PhD in Computer Science from Freie Universität Berlin in 2002. Since April 2008, he is a professor for Autonomous Intelligent Systems at the University of Bonn and director of the Institute of Computer Science VI. His research interests include cognitive robotics, computer vision, and machine learning.