In Proceedings of 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, November 2014.

Online Learning of Foot Placement for Balanced Bipedal Walking

Marcell Missura and Sven Behnke

***Abstract*— Due to the high complexity of the humanoid body, and its inherently unstable inverted pendulum-like dynamics, the development of a robust and versatile walking controller proves to be a difficult task. Using machine learning algorithms with hardware in the loop is a promising way of achieving balanced and dynamic gaits. In this work, we propose an online learning technique that learns how to step onto a reference footstep location while maintaining the balance of a bipedal walker in the presence of disturbances. The ability to step with the help of a parametrized motion generator simplifies the learning problem to the low-dimensional space of footstep coordinates. To quickly adapt the produced step sizes from learned experience, we update an online-capable function approximator with a pendulum-cart motivated gradient function that incorporates the trade-off between maintaining balance and stepping onto a desired location. While our method is able to robustly learn suitable footstep locations without prior knowledge, we gain advantage from initializing the learning with an analytic controller and show experimentally that the learning process can further improve the capabilities of the robot.**

I. Introduction

Bipedal walking for humanoid robots is one of the most interesting challenges in robotics to date. The widespread state of the art covers basic walking skills on a flat surface without disturbances. Push recovery, walking on rough terrain, and agile step control are active research topics.

The main reason why the conception of a bipedal walk is difficult, is that walking is a balance-critical full-body motion with inherently unstable inverted pendulum-like dynamics. Even small disturbances can destabilize a robot and lead to a fall if no corrective action is taken. The complexity of the humanoid body requires the computation of a high-dimensional full-body motion under time-critical balance constraints, which is not a trivial task. So far, analytically engineered controllers have produced the best results. The dominant strategy is to abstract from the complex body and represent it as a point mass model with inverted pendulum dynamics. Then, the mathematically tractable linear inverted pendulum model is used to deduct controllers that steer and balance the underlying point mass model. The full-body walking motion arises by mimicking the pendulum motion with the pelvis and connecting the pendulum base locations with smooth swing foot trajectories. This approach works to some extent, but has not yet achieved human-like performance in terms of versatility and robustness.

Using machine learning as a universal tool to generate a walking controller is thought to be a promising approach to improve upon the current state of the art. However, the naïve approach of learning a mapping from the high-dimensional robot configuration to another high-dimensional action space of joint torques does not even produce a stable walk in simulation. The exploitation of low-dimensional manifolds in the actuation space, that result from coordinated joint motion, and the point mass abstraction of balance can simplify the learning problem to a tractable level of difficulty. Furthermore, the biological example of flight animals, who learn how to balance their gait shortly after birth based on a genetic disposition to step roughly in the right direction, suggests that some form of initialization is beneficial to successfully learn how to walk. We follow these paradigms in this contribution and present an online learning method that makes use of a central pattern generator to hide the complexity of creating a walking motion on the level of single joints. The pattern generator exhibits an interface to control the step size of rhythmic stepping motions, and thus allows the learning task to be reduced to the learning of Cartesian coordinates of footstep locations. Additionally, we initialize with an analytically engineered footstep controller to a point where a simulated biped is already able to track a reference step size and to maintain its stability after a disturbance. We augment its step size output with a corrective offset that is learned online during walking with a gradient-based update capable of learning both reference tracking and preservation of a stable upright pose. We provide experimental evidence to highlight isolated features of our approach, such as the increase of the overall stability and reference tracking precision of the analytic controller, the robustness to learn a stable controller even without initialization, and the competitive performance of learning to absorb strong pushes after the experience of only a few impacts.

II. Related Work

Zero moment point (ZMP) tracking with preview control [1] is the most popular approach to bipedal walking to date. A number of pre-planned footsteps are used to define a future ZMP reference. A continuous center of mass (CoM) trajectory that minimizes the ZMP tracking error is then generated in a Model Predictive Control [2] setting. Using ZMP preview control, high performance hardware [3], [4], [5] can walk reliably on flat ground as long as disturbances are small. Next generation walking controllers from the ZMP preview family [6], [7], [8] also consider foot placement in addition to ZMP control either by including the footstep locations in the optimization process, or by using a simplified model to compute a footstep plan online. These approaches have not yet matured to real hardware capability, however.

All authors are with the Autonomous Intelligent Systems Group, University of Bonn, Germany. Email: missura@ais.uni-bonn.de. This work was partially supported by grant BE 2556/6-1 of the German Research Foundation (DFG).

Urata et al. [9] presented an impressive foot placement-based controller on a real robot that is capable of recovering from strong pushes. Instead of optimizing the CoM trajectory for a single ZMP reference, a fast iterative method is used to sample a whole set of ZMP/CoM trajectory pairs for three steps into the future. Triggered by a disturbance, the algorithm selects the best available footstep plan according to given optimization criteria. Resampling during execution of the footstep plan is not possible. The robot has to be able to track a fixed motion trajectory for the duration of the recovery. Specialized hardware was used to meet these stringent precision requirements.

Englsberger et al. [10] proposed to use a capture point trajectory as a reference input for gait generation instead of the ZMP. The capture point approach is much simpler and faster to compute than ZMP preview. It was demonstrated on Toro [11] to produce a walk of the same quality. However, adaptive foot placement has not been considered so far.

The Capture Step Framework [12], [13] has recently been developed by the authors, and has been demonstrated to generate a stable, omnidirectional walk with strong disturbance rejection capabilities on a real robot [14]. It computes foot-placement, step-timing, and ZMP control strategies in closed form, and thus significantly reduces the computational costs compared to the aforementioned approaches. We chose this method to initialize our machine learning approach.

As an alternative to engineered controllers, online learning strategies deployed on real systems can potentially learn to control their own dynamics. Rebula et al. [15], for example, improved the reactive step of a simulated biped from a standing position by learning to step onto an offset from the capture point. Focusing on the walking speed, bipedal and quadrupedal gaits were successfully optimized using policy gradient reinforcement learning methods in high-dimensional state spaces [16], [17], [18]. With the same learning method, adjusted to a neuronal gait controller, the sagittal-only robot Runbot learned to walk fast, and to cope with irregular terrain [19]. All of these experiments started from an already stable, hand-designed gait.

The most autonomously learning bipedal system was presented by Tedrake et al. [20]. Using an online stochastic policy gradient estimation, the robot Toddler learns to walk on different surfaces in less than 20 minutes. The robot was designed in such way that it can passively walk down a slope without actuation. The success of this experiment can mostly be attributed to a strong simplification of the learning task that actuates only the ankles in order to imitate a passive dynamic gait without the need for a slope.

Yi et. al. [21] have investigated online learning on real hardware using a reinforcement learning method. The approach is built on top of an open-loop gait trajectory generator and learns to optimize the input parameters of three biologically inspired disturbance rejection strategies. To make online learning on real hardware feasible, the reinforcement learning was simplified by a discretization of the input space and the assumption that the control parameters are restricted to lie on parametric functions. The achieved balance is convincing on a stationary robot.

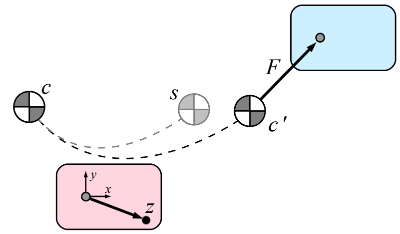

Fig. 1. The balance controller of the Capture Step Framework. The influence of the zero moment point Z steers the center of mass c towards the desired target location s. The actual footstep F is computed with respect to a predicted location c'.

Morimoto et. al used Gaussian processes [22], [23] and receptive field weighted regression [24] to learn a Poincaré map that approximates the periodic dynamics of a biped. Using this map, a policy gradient-based reinforcement learning method was used to train bipedal gaits in simulation and on real robots. Upright walking with an unspecified walking velocity in the absence of disturbances was successfully achieved.

III. Capture Step Control

Before elaborating on our machine learning concept, we briefly introduce the analytic controller that is used for initialization. The Capture Step bipedal gait controller [13] separates the walking task into an open-loop pattern generator for step motions [25] and a balance control module that commands the motion generator when and where to step in order to obey a commanded walking velocity while maintaining balance. In contrast to the ZMP preview family of approaches, the Capture Step control framework plans the center of mass trajectory first. The footstep locations arise as the output of the algorithm in such a way that they instantly adapt to disturbances. The details of the balance controller are illustrated in Figure 1. The state of the biped is reduced to a point mass c. Its position and velocity are estimated by tracking the ground projection of the center of mass with a full-body kinematic model during walking. A target end-of-step state s encodes the desired walking velocity. A ZMP location Z is first computed in order to steer the center of mass towards the desired state s, but since the ZMP is constrained to remain inside the support foot polygon, it can only have limited effect. When a disturbance is large, the desired state cannot be reached and the footstep location needs to be adjusted accordingly in order to prevent a fall. The footstep location F is computed with respect to the achievable end-of-step state c', which is predicted using a linear inverted pendulum model that takes the effect of the ZMP into account. The time T of the step is determined as the time when the center of mass is expected to reach the lateral coordinate of the desired state s. The computed parameters (F, Z, T) are then used by the step motion generator to realize a physical step to the desired coordinates at the right time.

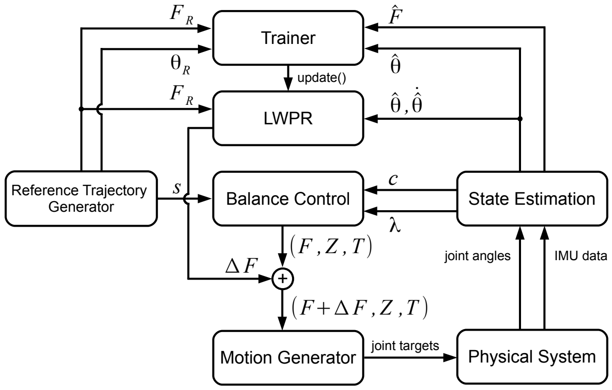

Fig. 2. The architecture of our machine learning concept.

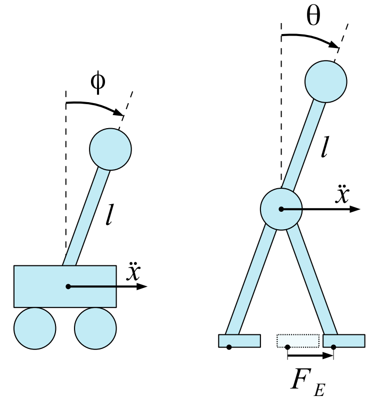

Fig. 3. The pendulum-cart model (left) resembles the angular dynamics of a biped (right). When the cart accelerates in the positive direction, the pendulum angle is accelerated in the negative direction. The biped accelerates its center of mass by increasing or decreasing its step size and can counteract undesired angular momentum at the cost of a possible error F_E with respect to a reference footstep location.

IV. Machine Learning Framework

We interface a machine learning component with the Capture Step controller by changing the footstep output of the balance control module before it is passed on to the motion generator. The components and data paths of our control architecture are illustrated in Figure 2. The machine learning component learns an offset ΔF that is added to the step size output F in order to maintain an upright posture and to follow the reference input more precisely— even in the presence of disturbances. The step size output of the balance control module can be turned off, in which case the learned controller is responsible for producing the right step sizes on its own. The learning layer consists of two modules. A trainer module measures the reference tracking error and the balance of the robot in the instant of the touch-down of the swing foot, computes a gradient that is expected to improve the physical step size under the conditions that were encountered during the step, and updates a function approximator using this gradient. The function approximator is constantly queried during walking with a high frequency and provides the offset ΔF. We restrict our considerations to the sagittal direction and leave the zero moment point Z and step timing T outputs unchanged. Since the function approximator has a strong influence on balance, a low-latency response is crucial. We achieved good results with the online capable LWPR [26] algorithm which was found to have convincing update and response times.

A. Reference Tracking

We express the reference input as a desired step size F_R that we want the biped to produce. This choice is motivated by the fact that footstep planning [27], [28] is a gradually improving, versatile method to command a robot where to walk. A footstep plan can be used to encode simple, constant velocity walking on a flat surface and scales up to careful stepping onto constrained locations in cluttered environments, stepping over obstacles, and elevated footholds in rough terrain. By learning to step more accurately onto a commanded footstep location while maintaining balance, we improve the ability of a gait controller to follow a footstep plan. We assume the desired step size F_R and a matching end-of-step CoM location s is the input into our system, given by a higher level reference trajectory generator. The state estimation component of the Capture Step Framework is based on a kinematic model that we can use to obtain step size measurements $\hat{F}$ by computing the distance between the feet in the moment of the support exchange. The step size error F_E is then trivially given by

$$F_E = \hat{F} - F_R. \tag{1}$$

B. Balance

To quantify balance, we introduce the concept of the Trunk Deviation Angle, denoted with the shorthand TDA. Typically, a motion controller has a notion of a desired trunk inclination θ_R with respect to the world frame. Usually the trunk is upright with a desired tilt of zero degrees, if it is not being used for balance or manipulation purposes. We assume that the reference trajectory generator provides the desired tilt input θ_R as well. The trunk attitude $\hat{\theta}$ can be estimated from an IMU. The TDA θ_E is then given by

$$\theta_E = \hat{\theta} - \theta_R. \tag{2}$$

Ideally θ_E should be zero, meaning that the robot has exactly the desired pose. Pushes, collisions, and other disturbances can rotate the entire body around an edge of the support foot and result in a non-zero TDA that can only be tolerated within a certain margin. When a large disturbance induces critical angular momentum, the robot has to step adequately to counteract the rotation.

To model a step controller that attempts to maintain a zero TDA, we derive a simple control law from the pendulum-cart model illustrated in Figure 3. The equations of motion of the pendulum-cart model are given by

$$\ddot{\phi} = \frac{g \sin\phi - \ddot{x}\cos\phi}{l}, \tag{3}$$

where ϕ is the pendulum angle with respect to the world vertical, x is the position of the cart, g is the gravitational constant and l is the length of the pendulum. When $\phi \approx 0$, a proportional controller $\ddot{x} = k\phi$ with gain k can be used as a simple controller that manages to balance the pendulum on the cart. A biped walker is not a cart, but it can accelerate its center of mass by increasing or decreasing its step size. This translates to a rough approximation of a balancing step controller

$$\Delta F \approx k\theta_E. \tag{4}$$

An example is shown in Figure 3. If at the end of a step the TDA is positive, i.e. the robot is rotated 'forward', the robot needs to take a larger step next time in the same situation in order to accelerate the TDA towards a more upright position. The direct application of this model to control the step size provides only a small amount of balance, and it cannot possibly track a reference input on its own. As our goal is to learn the right step size, we only borrow this concept to construct a learning gradient.

C. Learning

As the reference footstep location and the right place to step to counteract a disturbance do not necessarily coincide, following a footstep plan under balance constraints is an ill-posed problem. A trade-off must be found between stepping into a desired location and avoiding a fall. We combine the control law (4) we derived from the pendulum-cart model and the footstep error F_E into the parameterized gradient function

$$\mathcal{G}(\theta_E, F_E) = \theta_E - p_\theta \tanh(p_F F_E). \tag{5}$$

The characteristic saturation of the hyperbolic tangent function bounds the influence of the step size error F_E to a configurable limit p_θ for two specific purposes. Making no assumption about the commanding layer, the reference step size —and thus the step size error F_E— can change abruptly to arbitrarily large values. The parameterized saturation makes sure that the robot learns to track the reference step size carefully and avoids instability during the learning process due to abrupt changes in the step size. Furthermore, critical tilts of $\theta \gg p_\theta$ dominate the gradient and balance takes priority over reference tracking when a fall is imminent. p_F is a weight to fine-tune the influence of the step size error within the permitted bounds. Throughout our experiments we used $p_\theta = 0.15$ and $p_F = 30$. Please note that the gain k has been absorbed by the learning rate that we multiply the gradient with when we apply the update rule to the function approximator in the following equation.

Updates are made at the end of each step, where we measure the trunk angle $\hat{\theta}$ and the step size $\hat{F}$ and compute the step size error F_E and the TDA θ_E using (1) and (2), respectively. We also estimate the TDA rate $\dot{\theta}_E = \dot{\hat{\theta}}$. We train a function approximator over the input space $\theta_E \times \dot{\theta}_E \times F_R$ with the update rule

$$\mathcal{F}(\theta_{E_i}, \dot{\theta}_{E_i}, F_R) = \mathcal{F}(\theta_{E_i}, \dot{\theta}_{E_i}, F_R) + \eta\,\mathcal{G}(\theta_E, F_E), \forall i \in I, \tag{6}$$

where I is an index set and $\{\theta_{E_i}\}, i \in I$ is the set of TDAs 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 result as the new desired output to the function approximator.

V. Experimental Results

To evaluate and demonstrate isolated features of our learning framework, we performed a series of experiments in a Bullet Physics simulation with a humanoid model of 2 m height and a human-like distributed body-weight of 13.5 kg in total. In all of the following experiments, the algorithm learned online during the experiment. It was not pre-trained and it was operational during the entire evaluation time with a constant learning rate $\eta = 1$. Although the following experiments are focused on the sagittal direction, the motion of the robot was not restricted in any way and the lateral Capture Step components were fully operational.

A. Evaluation of Reference Tracking

In the first experiment, we evaluate the ability to track a reference step size. We compare the Capture Step controller on its own, which we refer to as the *model*, with the addition of a *learned* controller that was trained during the experiment. We sample the reference step size from a range of $[-10, 20]$ cm and keep it constant for 4 to 8 seconds. We observe how quickly both controller versions can adapt the step size to the correct value. To preserve comparability, the same random step size sequence was presented to both controllers. Figure 4 shows statistical data averaged over 1000 steps. The moments of the reference step size changes are synchronized at zero seconds. We observe the mean and standard deviation of the step size error. Since the reference step size was uniformly sampled, the mean is near zero. With learning, the standard deviation of the step size error decreased at all points in time. This means that the learned controller not only follows the reference faster than the capture step controller alone, but it also learned to step onto the right location altogether more precisely. In the right hand plot in Figure 4 we show a time series extract of the reference and the measured step sizes.

B. Evaluation of Disturbance Rejection

In the second experiment, we demonstrate the ability to return to a reference walking velocity after a disturbance. We command a fixed step size of 20 cm. While the robot is walking, we push it forward with constant impulse magnitudes of 8 Ns. The pushes are triggered at random times in order to avoid hitting the robot repeatedly in the same motion phase. Synchronized at the moment of the push, Figure 5 shows how the step size (left) and the TDA (right) return to their reference values. Again, the learning component reduces the time it takes for the robot to return to the commanded step size. The fact that the TDA appears to have increased after learning might be a surprise at first, but seeing as we set the tolerated TDA parameter p_θ to 0.15 rad, the learned controller utilized this margin to sacrifice the allowed amount of balance in order to better obey the step reference.

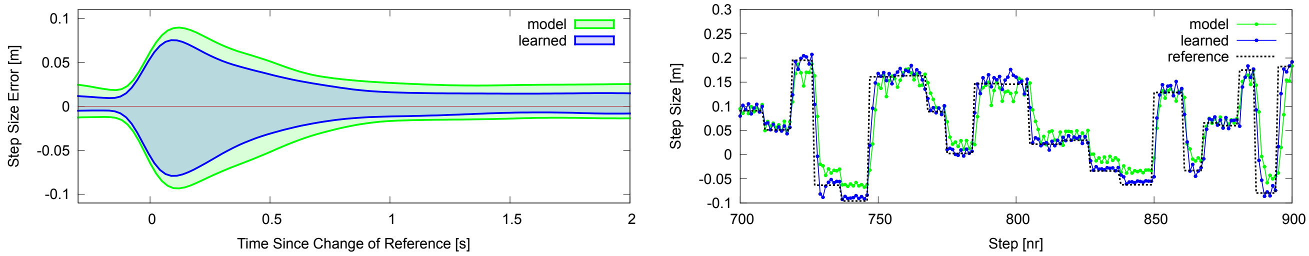

Fig. 4. In an experiment with random changes of the reference step size, the standard deviation of the step size error decreases faster with learning, than without (left). A time series of the abruptly changing reference step size and the step sizes produced by the evaluated controllers is shown on the right.

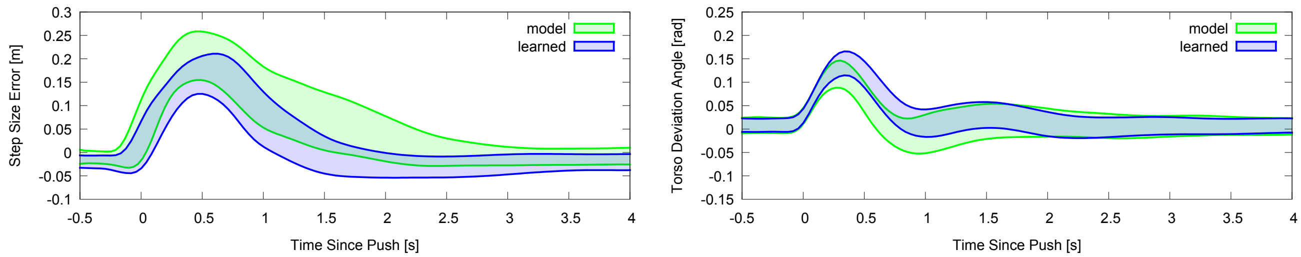

Fig. 5. After the robot is pushed, the step size error is reduced and the robot is able to return faster to the reference step size with the learning component enabled (left). The learned controller allows a larger Torso Deviation Angle (right) within the allowed margin of 0.15 radians in order to better obey the step reference.

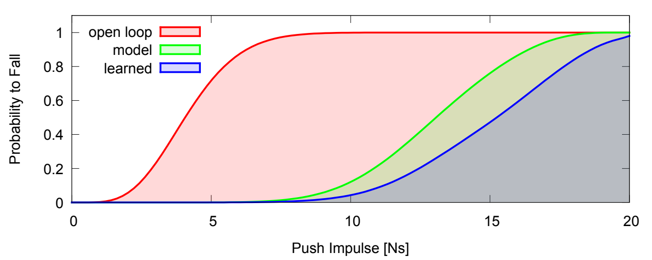

Fig. 6. Probability to fall of an open-loop, analytic, and a trained controller with respect to varying push impulses from the back.

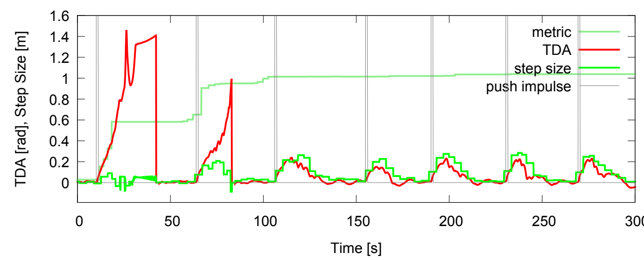

Fig. 7. Even if not initialized with the analytic controller, the machine learning layer manages to learn a step controller that stabilizes the robot after only three pushes. The pushes are indicated by the vertical lines of the push impulses.

C. Evaluation of Stability

In the third experiment, we aim to evaluate whether the learning component is able to improve the overall walking stability. We apply 400 randomly timed push impacts with magnitudes sampled from a range of $[0, 20]$ Ns to a robot walking in place. The pushes are directed in the forward direction and force the robot to make forward steps in order to avoid falling. By counting falls and pushes, we estimate the probability of a fall depending on the magnitude of the disturbance. In addition to the Capture Step *model* and the *learned* augmentation, in this experiment we also included an open-loop controller that walks in place with a fixed frequency and does not react to the pushes. The results are shown in Figure 6. The capture step controller significantly increases the push resistance compared to what the robot can absorb passively. Our online learning technique, however, increases the stability even further.

D. Evaluation of Robustness

Finally, we demonstrate the potential of our learning approach with an experiment that is focused on speed and robustness of learning. This time we do not use the capture step controller for initialization. The robot starts with stepping in place with no prior knowledge of step size control. We disturb the robot from the back with 8 Ns push impulses and observe how quickly the robot learns to absorb the push without falling. The result of the experiment is shown in Figure 7. 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. To observe a learning curve, we created a metric of the function approximator by computing the Euclidean norm over the output values of evenly distributed samples in the input space. Initially the metric is zero. Plotting the metric over time shows that in the beginning of the experiment large learning steps are taken. After the third push, the learning process has mostly settled and the controller appears to have learned how to balance in this specific case. The accompanying video shows a recording of this experiment.

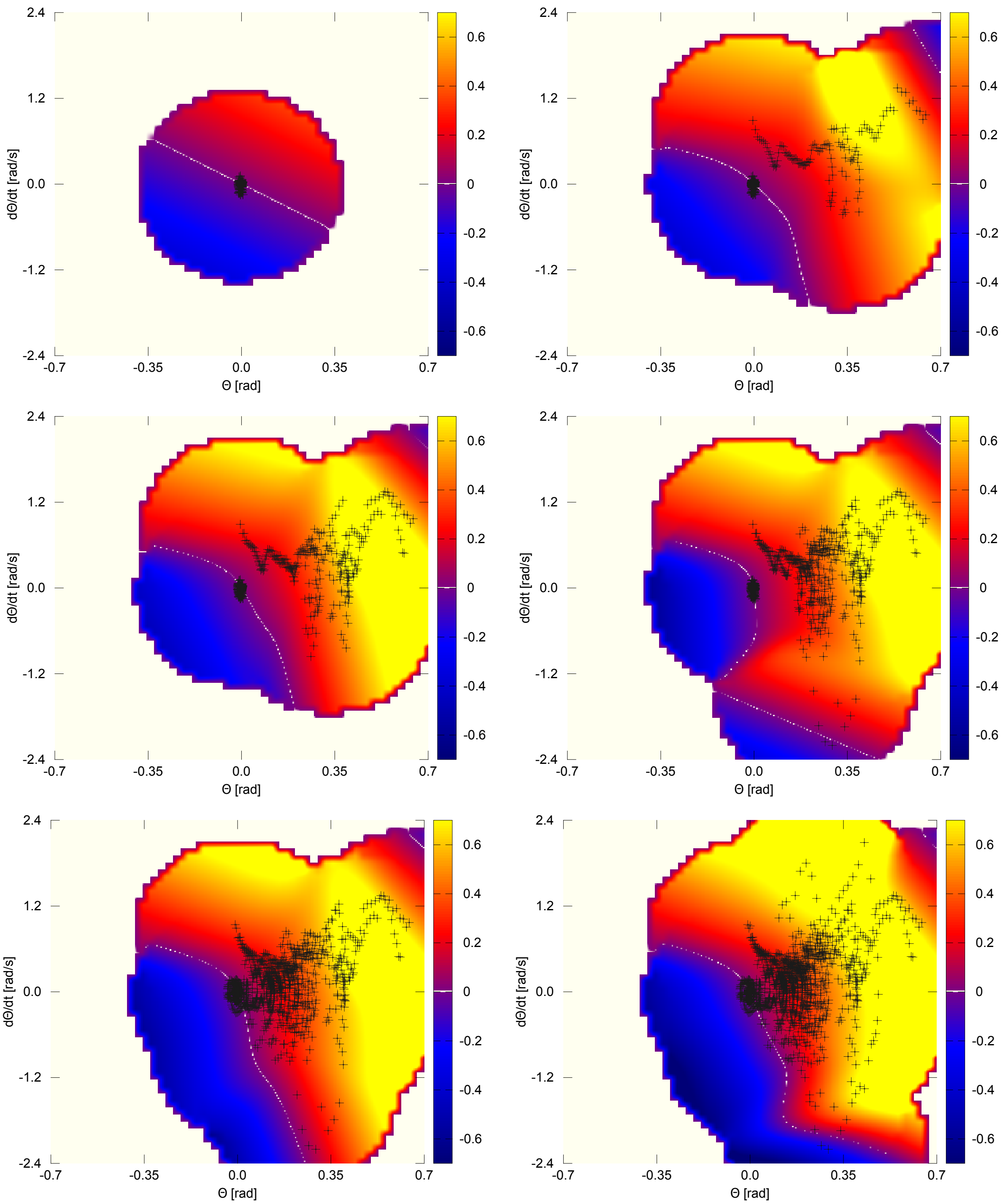

Fig. 8. Maps of the learned step size offsets after each of six pushes. The order of the plots is from left to right and from top to bottom. In this experiment, the step size controller was not initialized with the analytic controller and thus the entire step sizes have to be learned. The learning settles after three pushes and the simulated model is able to absorb an 8 Ns push without falling.

The evolution of the LWPR function approximator is shown in Figure 8. Each plot was generated in the moment of a push, i.e. the effect of a push is shown in the next plot. Data points are marked with black crosses. The first plot on the top left shows the state of the function approximator before the first disturbance. The relatively large size of the linear kernel we use within the LWPR can be clearly seen. Data points are located only in the center of phase space. It is remarkable that the LWPR algorithm managed to align the first kernel in a way that it extrapolates sensible values only from a small cloud of data points that were collected during stable walking in place. The first push (top right) forces the robot to fall and the collected data leaves a broad trace. The second push results again in a fall, and the function approximator is updated in the relevant part of the phase space. The third push is successfully absorbed, and subsequent pushes change the function approximator only slightly. The last plot (bottom right) includes a trunk angle trajectory that returned to a stable position after reaching an inclination of nearly 0.5 radians.

Note that this experiment does not render the initialization with the analytic controller obsolete. The initialization not only speeds up the learning process, because the learner can start from an already good result, but it also covers the entire input space. Without any initialization, an exploration of a large number of reference step size and disturbance magnitude combinations would be necessary in order to train a reliable controller. Furthermore, the initialization reduces the probability to fall during learning and thus reduces the risk of sustaining hardware damage when training with a real robot.

VI. Conclusions

We presented a simple and robust online learning concept that learns both, balancing a humanoid robot after strong disturbances and following a commanded step size. We used the Capture Step framework to initialize the controller with a reasonable output and used the learning layer to improve its capabilities.

A clear distinction between our method and other approaches investigated so far is the use of a gradient-based update— rather than reinforcement learning. To compute the gradient, we made the assumption that the angular dynamics of a biped behaves like the pendulum-cart model, and derived a monotonic coupling between the step size variation and the change of the angular momentum. While less generic than model-free reinforcement learning, the assumption appears to be beneficial for learning from a low number of experiences.

In future work, we intend to extend the learning controller to the lateral direction and to investigate whether two uncoupled learners are able to learn a stable omnidirectional walk. Furthermore, we plan to experiment with self disturbance-driven exploration strategies to develop an autonomously learning algorithm.

References

[1] 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)*, 2003.

[2] P.-B. Wieber. Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2006.

[3] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The Development of Honda Humanoid Robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1998.

[4] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi. Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[5] I.-W. Park, J.-Y. Kim, J. Lee, and J.-H. Oh. Mechanical Design of Humanoid Robot Platform KHR-3 (KAIST Humanoid Robot 3: HUBO). In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2005.

[6] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl. Online Walking Gait Generation with Adaptive Foot Positioning Through Linear Model Predictive Control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.

[7] Mitsuharu Morisawa, Fumio Kanehiro, Kenji Kaneko, Nicolas Mansard, Joan Sola, Eiichi Yoshida, Kazuhito Yokoi, and Jean-Paul Laumond. Combining suppression of the disturbance and reactive stepping for recovering balance. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3150–3156, 2010.

[8] B. J. Stephens and C. G. Atkeson. Push Recovery by Stepping for Humanoid Robots with Force Controlled Joints. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2010.

[9] J. Urata, K. Nishiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba. Online Decision of Foot Placement Using Singular LQ Preview Regulation. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2011.

[10] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger. Bipedal Walking Control Based on Capture Point Dynamics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[11] Christian Ott, Christoph Baumgrtner, Johannes Mayr, Matthias Fuchs, Robert Burger, Dongheui Lee, Oliver Eiberger, Alin Albu-Schffer, Markus Grebenstein, and Gerd Hirzinger. Development of a biped robot with torque controlled joints. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 167–173, 2010.

[12] M. Missura and S. Behnke. Omnidirectional Capture Steps for Bipedal Walking. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2013.

[13] M. Missura and S. Behnke. Balanced walking with capture steps. In *RoboCup 2014: Robot Soccer World Cup XVIII (to appear)*. Springer, 2014.

[14] M. Missura and S. Behnke. Walking with Capture Steps. `http://www.ais.uni-bonn.de/movies/WalkingWithCaptureSteps.wmv`.

[15] John Rebula, Fabian Canas, Jerry Pratt, and Ambarish Goswami. Learning Capture Points for Humanoid Push Recovery. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.

[16] T. Hemker, M. Stelzer, O. von Stryk, and H. Sakamoto. Efficient walking speed optimization of a humanoid robot. *International Journal of Robotics Research*, 28(2):303–314, 2009.

[17] Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[18] Felix Faber and Sven Behnke. Stochastic optimization of bipedal walking using gyro feedback and phase resetting. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.

[19] Tao Geng, Bernd Porr, and Florentin Wörgötter. Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *International Journal of Robotics Research*, 2006.

[20] Russ Tedrake, Teresa Weirui Zhang, and H. Sebastian Seung. Learning to walk in 20 minutes. In *14th Yale Workshop on Adaptive and Learning Systems*, 2005.

[21] Seung-Joon Yi, Byoung-Tak Zhang, Dennis Hong, and Daniel D. Lee. Online Learning of a Full Body Push Recovery Controller for Omnidirectional Walking. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2011.

[22] Jun Morimoto and Christopher G. Atkeson. Nonparametric representation of an approximated poincaré map for learning biped locomotion. In *Auton Robot*. Springer, 2009.

[23] Jun Morimoto, Christopher G. Atkeson, Gen Endo, and Gordon Cheng. Improving humanoid locomotive performance with learnt approximated dynamics via guassian process for regression. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2007.

[24] Jun Morimoto and Christopher G. Atkeson. Learning biped locomotion. In *IEEE Robotics and Automation Magazine*. IEEE, 2007.

[25] M. Missura and S. Behnke. Self-Stable Omnidirectional Walking with Compliant Joints. In *Workshop on Humanoid Soccer Robots*, Atlanta, USA, 2013.

[26] Sethu Vijayakumar, Aaron D'souza, and Stefan Schaal. Incremental online learning in high dimensions. *Neural Comput.*, 17(12):2602–2634, December 2005.

[27] K.M. Cheung J. Kuffner J.K. Hodgins J. Chestnutt, M. Lau and T. Kanade. Footstep planning for the honda asimo humanoid. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005.

[28] Armin Hornung, Andrew Dornbush, Maxim Likhachev, and Maren Bennewitz. Anytime search-based footstep planning with suboptimality bounds. In *Proc. of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, Osaka, Japan, November 2012.