Biped Walk Learning Through Playback and Corrective Demonstration

Çetin Meriçli[1,2] and **Manuela Veloso**[1]
{cetin,veloso}@cmu.edu

[1] School of Computer Science, Carnegie Mellon University, Pittsburgh, United States

[2] Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey

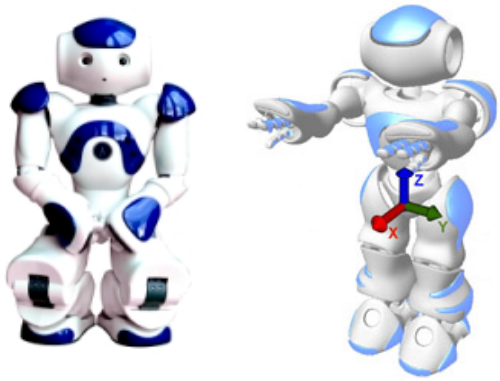
Abstract

Developing a robust, flexible, closed-loop walking algorithm for a humanoid robot is a challenging task due to the complex dynamics of the general biped walk. Common analytical approaches to biped walk use simplified models of the physical reality. Such approaches are partially successful as they lead to failures of the robot walk in terms of unavoidable falls. Instead of further refining the analytical models, in this work we investigate the use of human corrective demonstrations, as we realize that a human can visually detect when the robot may be falling. We contribute a two-phase biped walk learning approach, which we experiment on the Aldebaran NAO humanoid robot. In the first phase, the robot walks following an analytical simplified walk algorithm, which is used as a black box, and we identify and save a walk cycle as joint motion commands. We then show how the robot can repeatedly and successfully play back the recorded motion cycle, even if in open-loop. In the second phase, we create a closed-loop walk by modifying the recorded walk cycle to respond to sensory data. The algorithm learns joint movement corrections to the open-loop walk based on the corrective feedback provided by a human, and on the sensory data, while walking autonomously. In our experimental results, we show that the learned closed-loop walking policy outperforms a hand-tuned closed-loop policy and the open-loop playback walk, in terms of the distance traveled by the robot without falling.

Introduction

Biped walk learning is one of the most challenging problems in humanoid robotics research due to the complex dynamics of the walking process. Developing efficient biped walking methods on commercial humanoid platforms with limited computational power is particularly difficult due to the high computational complexity of sophisticated analytical walking approaches.

In this work, we use the Aldebaran Nao robot, (Figure 1(a)), which is a 4.5 kilograms, 58 cm tall humanoid robot with 21 degrees of freedom (www.aldebaran-robotics.com). The Nao has an on-board 500 MHz processor, to be shared between the low level control system and the autonomous perception, cognition, and motion algorithms. The Nao is equipped with a variety of sensors including two color cameras, two ultrasound distance sensors, a 3-axis accelerometer, a 2-axis, gyroscope (X-Y), an inertial measurement unit for computing the absolute orientation of the torso, 4 pressure sensors on the sole of each foot, and a bump sensor at the tiptoe of each foot. The inertial measurement unit, the accelerometer, and the gyroscope sensors use a right-hand frame of reference (Figure 1(b)). We use the common terms "roll," pitch," and "yaw" for the rotation along the X, Y, and Z axis, respectively.



(a) (b)

Figure 1: a) The Aldebaran Nao robot. b) Frame of reference for the sensors.

As opposed to many other humanoid robot designs, the Nao does not have separate hip yaw joints (along the Z axis), making it difficult to develop efficient walking motions. Nao's relatively heavy body also makes it rather challenging to keep the robot in balance when walking.

Since 2008, the Nao has been used as the robot for the RoboCup Standard Platform League (SPL) (www.robocup.org), in which teams of autonomous humanoid Nao robots play robot soccer (www.tzi.de/spl).

Since the introduction of the Nao as the common robot platform of the RoboCup SPL, competing teams have been drawn to investigate efficient biped walking methods, in particular suitable for the Nao hardware. The proposed approaches include an effective omni-directional walking algorithm using parameterized gaits and sensor feedback (Röfer et al. 2009), an efficient Zero Moment Point (ZMP) search method (Liu and Veloso 2008), an evolutionary strategy to tune the parameters of a Central Pattern Generator based walk (Gokce and Akin 2009), a ZMP based omnidirectional walking algorithm (Strom, Slavov, and Chown 2009), a preview control based method for keeping the ZMP on the desired path (Czarnetzki, Kerner, and Urbann 2009), and the default open-loop uni-directional walk algorithm of the Nao, provided by Aldebaran Robotics. In this paper, we present our work to explore biped walk learning from demonstration (Argall et al. 2009).

Multiple approaches to robot motion learning utilize learning from demonstration, in particular biped walk learning from human demonstrated joint trajectories using dynamical movement primitives (Nakanishi et al. 2004), and quadruped walk learning for a SONY AIBO robot using a regression-based approach (Grollman and Jenkins 2007). Learning from demonstration was introduced in the form of advice operators as functional transformations for low level robot motion, and demonstrated for a Segway RMP robot (Argall, Browning, and Veloso 2007; 2008). Furthermore the "confidence based autonomy" approach enables robot learning from demonstration of general behavior policies for single robots (Chernova and Veloso 2008a) and multi-robot systems (Chernova and Veloso 2008b).

We contribute a novel approach for learning biped walking on the Aldebaran Nao humanoid robot. We break the learning process into two phases. In the first phase, we take advantage of an existing simplified biped analytical walking algorithm, recording good walking sequences to summarize in a single walk cycle. We then play back the obtained walk cycle, resulting in an open-loop walking behavior. In the second phase, we derive a corrective policy from a set of human demonstrations in real time while the robot is performing the open-loop walk behavior. We utilize a commercial wireless game controller for providing feedback signals without physically touching the robot. We then use the learned corrective policy to modify the joint commands from the open-loop walk cycle in a way to avoid that the robot falls at some point while walking. We present experimental results showing the improvement in the distance traveled by the robot without falling using the learned correction policy in comparison to the open-loop walking and a hand-tuned closed-loop policy.

Two-Phase Biped Walk Learning

The biped walking process consists of sending commands to the joints of the robot over time. Walking is a periodic phenomenon and consists of consecutive walk cycles which start with a certain configuration of the joints and end when the same configuration is reached again. A walk cycle wc is a motion segment of duration T timesteps, where $wc_j(t), t \in [0, T)$ is the *command* to the joint j provided at timestep t.

Although the Nao robot has a total of 21 joints, for our approach, we use a subset of 10 of them named $Joints$: hip roll, hip pitch, knee pitch, ankle pitch, and ankle roll joints for the left and the right legs.

Phase One: Open-Loop Walking

We begin by collecting a number of trials of the robot walking forwards for a fixed distance at a fixed speed using an existing walk algorithm. We then save the trials in which the robot was able to complete the walking without losing its balance. Many examples of the robot walking taken from the saved trials provide data D for each $t, t \in [0, T)$, in the form of the commands received for each joint $\overrightarrow{D_j}(t)$. We acquire a single walk cycle wc using D as $wc_j(t) = \mu(\overrightarrow{D_j}), j \in Joints, t \in [0, T)$.

In principle, if we have a walk cycle, it would be possible to make the robot walk indefinitely by executing this cycle in an infinite loop. However, in reality, various sources of uncertainty are associated with the sensing, planning, and the actuation for biped walking.

- In ***sensing***, the main source of uncertainty is the noise in the sensor readings due to the lack of precision/accuracy, or the environmental effects (e.g., electromagnetic fields affect compasses negatively).
- In ***planning***, the simplifications and assumptions that have been made while building the mathematical model of the system prevent the developed model from capturing all physical aspects of the real world.
- In ***actuation***, several factors such as the friction inside the gearboxes, unmodeled payload effects, and the noise in the position sensor readings which affect the joint angles constitute the main sources of the uncertainty.

As a result, the actual movement of the robot differs significantly from the desired movement (Figure 2). The plot with circles shows the joint commands, i.e., the desired trajectory, and the plot with triangles shows the actual trajectory the joint has followed. The heavily disturbed part towards the end corresponds to the phase where the robot is taking a right step and is standing on its left leg, hence, the weight of the whole body prevents the left ankle joint from following the desired trajectory.

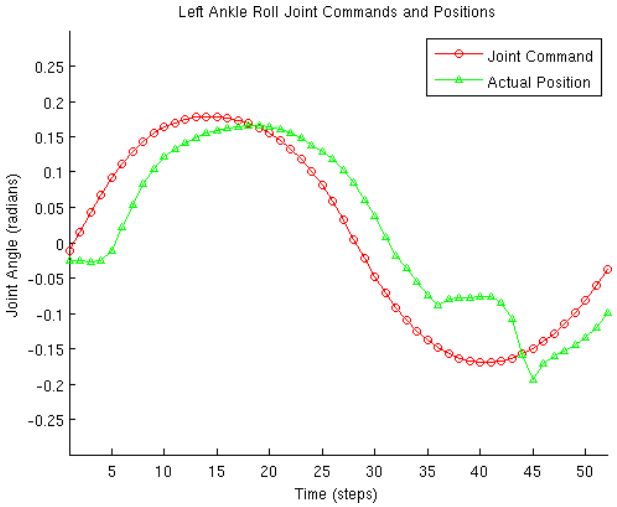


Figure 2: An example to the actuation error from the ankle roll joint of the left leg.

Phase Two: Closed-loop Corrections

It is very difficult to obtain a walking behavior that can walk forever using an open-loop playback mechanism without any corrections. The changes in sensor readings when the robot is about to lose its balance (Figure 3) can be used to derive a correction policy by mapping these changes to corrective feedback signals.

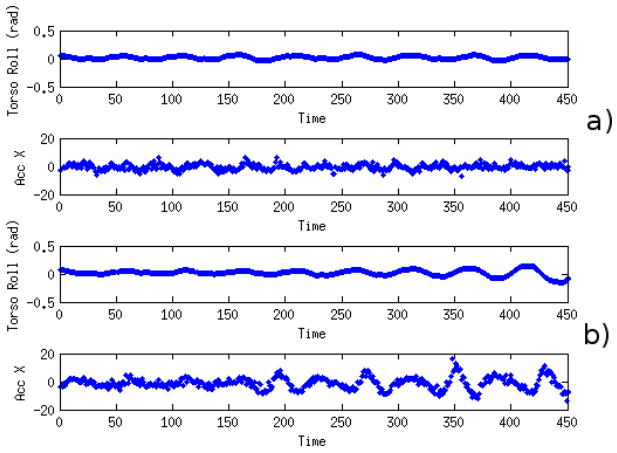


Figure 3: Sample torso orientation and accelerometer readings: a) a stable walk sequence, and b) a walk sequence where the robot starts losing its balance.

The noisy nature of the sensors causes fluctuations in the sensor readings which may result in jerky motions and therefore loss of balance when the correction values calculated as a function of the sensor readings are applied to the joints directly. Running mean and median smoothers are widely known methods for filtering noisy data. In running smoothers, the data point in the middle of a running window of size N is replaced with the mean or the median of the data points lying in that window. Filtered signal gets smoother as the window size increases. The delicate trade-off in filtering lies in the selection of an appropriate window size for smoothing the data enough to filter out the noise without rendering the patterns in the data hard to detect.

We evaluated the running mean and running median smoothers with window sizes 5 and 10 (Figure 4) and we decided to use a running mean filter with window size 5 since it filters out the noise reasonably well and it is computationally cheaper than the running median filter. Also, considering our sensor sampling rate of 50 Hz, we can still detect a significant change in the sensor readings in at most $1/10^{th}$ of a second.

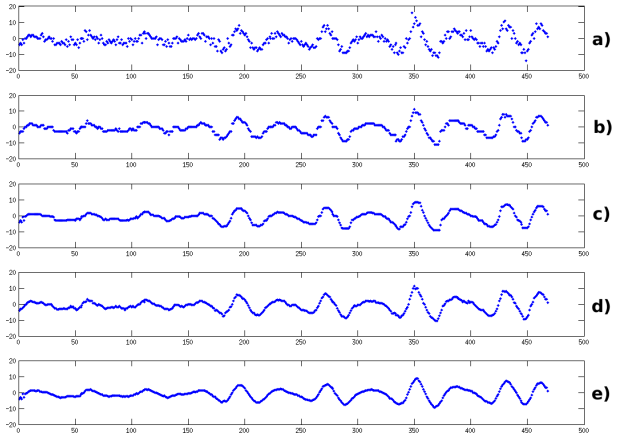


Figure 4: Various smoothers on an example accelerometer data: a) the raw data, b) median smoother with window size 5, c) median smoother with window size 10, d) mean smoother with window size 5, e) mean smoother with window size 10.

Correction Using Single Sensor-Joint Couplings

We use the hip roll and the hip pitch joints to apply the correction signals. We define the correction value for each joint as a transformation function applied on a single sensor reading. At each timestep, we compute the correction values for all joints $j \in Joints$ using the recent sensor readings and the defined correction functions. We then add the calculated values to the joint command values in the walk cycle for that timestep before sending the joint commands to the robot. The correction is applied to the system at each m^{th} timestep where $1 <= m <= T$ where T is the length of the walk cycle in timesteps. The pseudo-code of that process is given in Algorithm 1.

The definition of the correction function is a difficult and non-trivial problem. We first started with a hand-tuned simple linear function to be used as a benchmark method in comparisons. We used the roll and the pitch angles of the torso, calculated by the inertial measurement unit as the sensor readings and we coupled them with the hip roll and the

Algorithm 1 Closed-loop walking using sensor-joint couplings.

$t \leftarrow 0$
loop
  $\vec{S} \leftarrow readSensors()$
  $\vec{S} \leftarrow smooth(\vec{S})$
  for all $j \in Joints$ **do**
    if $timestep$ MOD $correctioninterval = 0$ **then**
      $C_j = Correction(\vec{S}, j)$
    else
      $C_j = 0$
    end if
    $NextAction_j \leftarrow wc_j(t) + C_j$
  end for
  $t \leftarrow t + 1 \pmod T$
end loop

hip pitch joints. The inertial measurement unit returns the roll and pitch orientations of the torso in radians with respect to the ground. The used linear coupling functions are of the form $C = AX + B$ where A is a gain value, B is an offset value, X is the sensor reading, and C is the calculated correction value. For the four hip joints to be corrected, we have four functions with the individually set A and B values. We hand-tuned the parameters of these four functions using expert knowledge and visual observation of the robot walking. The resulting hand-tuned policy provided an improvement over the open-loop walk. Details of the results are given in the Experimental Results section.

Correction Policy Learning From Demonstration

Given that hand-tuning is inevitably a tedious and hard to generalize method, we now present an automated method to correct the walk cycle. We consider a human demonstrator providing corrective feedback signals in real time while the robot is performing the playback walk.

One of the major engineering problems in using real time human feedback is that the feedback must be provided without interfering with the robot dynamics. In other words, the feedback should be delivered without touching the robot. We developed a wireless control interface using the Nintendo Wiimote commercial game controller (Nintendo 2007) to provide corrective demonstration to the robot. The Wiimote controller and its Nunchuk extension are equipped with accelerometers which not only measure the acceleration of the controllers, but also allow their absolute roll and pitch orientations to be computed. The computed roll and the pitch angles are in radians and they use the right-hand frame of reference.

The user gives the correction to the robot by changing the orientations of the Wiimote and the Nunchuk controllers. The Nunchuk extension and the Wiimote control the left and the right side corrections, respectively. Tilting a handle forward or backwards makes the corresponding half of the robot bend forward or backwards. Similarly, rolling a handle to left or right causes the corresponding half of the robot to bend left or right (Figure 5). This setup allows the demonstrator to provide corrective feedback signals without physically contacting the robot (Figure 6).

To provide a finer control ability to the demonstrator, a scaling factor γ is applied on the Wiimote readings before they are sent to the robot. We used $\gamma = 0.1$ in our implementation. The received roll corrections are applied on the hip roll joints and the received pitch corrections are applied on the hip pitch joints. To keep the feet parallel to the ground, the following correction values are applied on the ankle roll and the ankle pitch joints:

$$C_{AnkleRoll} = -C_{HipRoll}$$
$$C_{AnklePitch} = -C_{HipPitch}$$

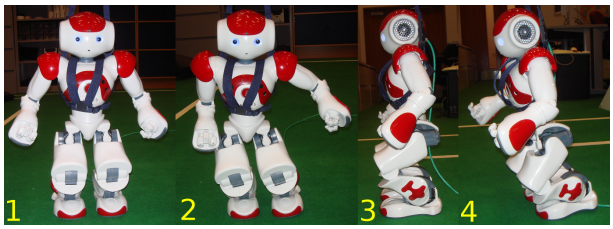


Figure 5: Example corrections using the Wiimote. Rolling the Wiimote to the right takes the robot from its neutral posture (1) to a posture bent along the Y axis (2). Similarly, tilting the Wiimote forward brings the robot from its neutral posture (3) to a posture bent along the X axis (4).

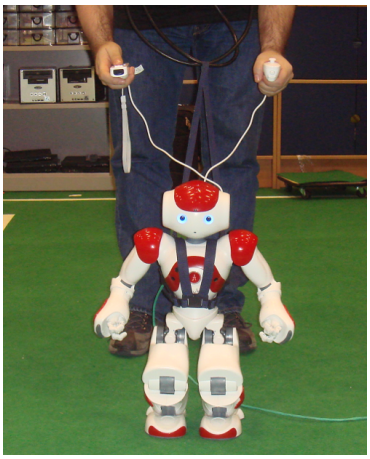
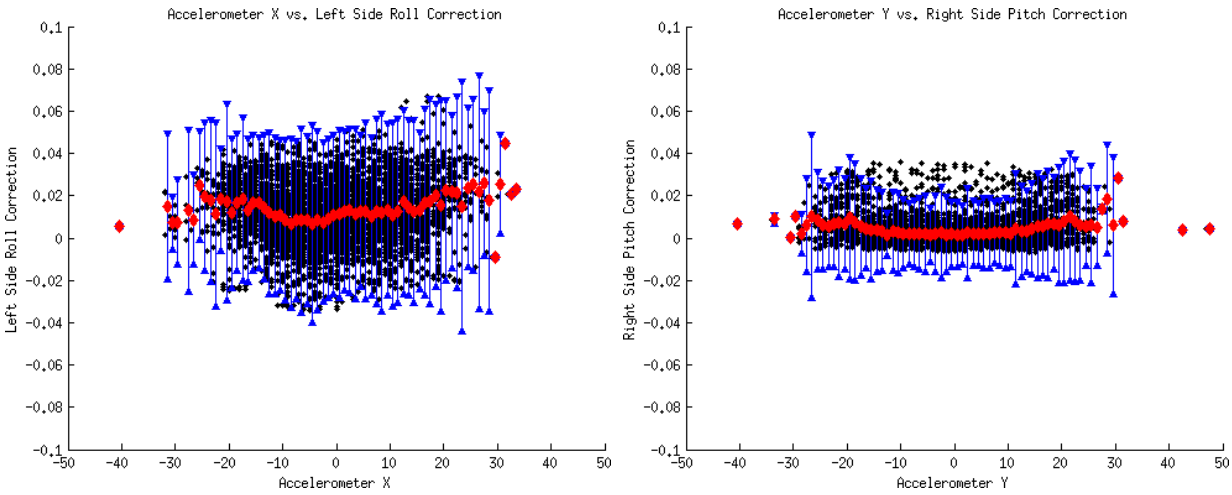


Figure 6: A snapshot from a demonstration session. A loose baby harness is used to prevent possible hardware damage in case of a fall. The harness neither affects the motions of the robot nor holds it as long as the robot is in an upright position.

The demonstrator presents a set of demonstrations to the robot using the wireless interface to modify the motion in



(a) Acc. X vs. Left Side Roll (b) Acc Y vs. Right Side Pitch

Figure 7: Accelerometer values versus received correction data. Bold points in the middle denote the mean, and vertical lines denote the variance of the normal distribution fit on that sensor value interval.

real time while the robot is walking using the open-loop walk cycle. We record the received correction signals during the demonstrations synchronously with the rest of the sensor readings at 50Hz.

We then map the recorded correction data to the sensor readings at that timestep. We use the accelerometer readings as the sensory input. The accelerometers on the Nao can measure accelerations in the range $[-2g, 2g]$ where g is the standard gravity and their readings are integer values in the interval $[-128, 127]$. To model the noise associated with the demonstration data, we fit a normal distribution on the correction data points received for all 256 possible values of the accelerometer (Figure 7). During the autonomous execution, a correction value should be populated for the value of the sensor reading at that time. Any discontinuity or a sudden change in the correction signal causes a jerky movement of the robot and further contributes to the loss of balance. To deal with it, the correction is a mapping from the sensory reading to the mean of each joint command to be corrected, namely, the left hip roll, the left hip pitch, the right pitch, and the right hip roll. During the autonomous execution, given the perceived sensory data, the corresponding mean is added to the walk cycle commands.

Experimental Results

We performed our experimental study with the Nao robot walking on a flat carpeted surface. The Nao walk cycle is of approximately one second duration, corresponding to 52 joint command and sensing points. For the corrective human demonstration, we ran a Nao in open-loop for approximately 10 minutes, equivalent to 600 walk cycles, and about 30,000 command and sensing points. A corrective action by the demonstrator using the Wiimote took in the order of half a second, leading to about 26 affected points in the walk cycle being executed. During the demonstration session, we collected a total of about 6,000 corrected data points, which were used to generate the learned policy.

We evaluated the performance of the learned correction policy against the open-loop walking and the hand-tuned closed-loop policy. We conducted 20 runs per method and we measured the distance traveled before falling. Figure 8 shows the results in. The learned policy demonstrated considerable improvement over the open-loop walking and the hand-tuned closed-loop policy, reaching a maximum traveled distance of 956 centimeters. [1] The maximum distances for the open-loop walking and the hand-tuned closed-loop policy were 327 and 689 centimeters, respectively.

Conclusion and Future Work

In this paper, we contributed a two-phase biped walk learning algorithm that combines playback of a walk cycle and corrective human demonstration. We analyzed the Nao robot in terms of the variations of joint commands and sensor readings. Our method utilizes the data collected from the human demonstration and the smoothed sensory readings of the robot to learn the appropriate correction values to the joint commands in the open-loop walk. The feedback given to the robot using the Wiimote controller is captured in terms of hip roll and hip pitch joint commands which are in turn translated into ankle roll and ankle pitch corrections.

[1] The open-loop walk performance was comparable to the performance of the original ZMP-based walk, which was not available to be accounted for in this empirical comparison.

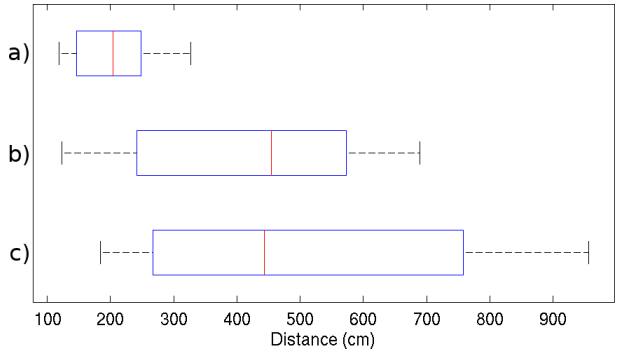


Figure 8: Distance traveled using different biped walk methods: a) open-loop, b) hand-tuned closed-loop, c) learned closed-loop policy from corrective demonstration. The lines within the boxes mark the median, the marks at both ends of boxes indicate minimum and maximum distances, and the left and right edges of boxes mark 25^{th} and 75^{th} percentiles, respectively.

We presented empirical results that show that the learned policy outperformed the open-loop walk and a hand-tuned closed-loop walk.

Following up on this work, we have investigated additional functional correction mechanisms based on Advice Operators (Argall, Browning, and Veloso 2008) and human demonstration as robot feet displacement, which we translate into corrections for multiple joint commands. The results show further improvements in walk distance travelled by the robot (Çetin Meriçli and Veloso 2010).

Investigating better policy derivation methods, extending the proposed approach to sideways walking and turning at variable speeds, and relaxing the flat surface assumption to deal with uneven terrains are among the issues we aim to address in the future.

Acknowledgements

The authors would like to thank Stephanie Rosenthal, Tekin Meriçli, and Brian Coltin for their valuable feedback on the paper. We further thank the Cerberus team for their debugging system, and the CMWrEagle team for their ZMP-based robot walk. The first author is supported by The Scientific and Technological Research Council of Turkey Programme 2214.

References

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Automation Systems* 57(5):469–483.

Argall, B.; Browning, B.; and Veloso, M. 2007. Learning from demonstration with the critique of a human teacher. In *Second Annual Conference on Human-Robot Interactions (HRI'07)*.

Argall, B.; Browning, B.; and Veloso, M. 2008. Learning robot motion control with demonstration and advice-operators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*.

Çetin Meriçli, and Veloso, M. 2010. Improving biped walk stability using real-time corrective human feedback. In *RoboCup 2010: Robot Soccer World Cup XIV*.

Chernova, S., and Veloso, M. 2008a. Confidence-based demonstration selection for interactive robot learning. In *In Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI'08), Amsterdam, The Netherlands, March 2008*.

Chernova, S., and Veloso, M. 2008b. Teaching collaborative multirobot tasks through demonstration. In *In Proceedings of AAMAS'08, the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems, Estoril, Portugal, May 2008*.

Czarnetzki, S.; Kerner, S.; and Urbann, O. 2009. Observer-based dynamic walking control for biped robots. *Robotics and Autonomous Systems* 57:839–845.

Gokce, B., and Akin, H. L. 2009. Parameter optimization of a signal-based biped locomotion approach using evolutionary strategies. In *Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2009), Istanbul, Turkey, 9-11 September 2009, O. Tosun, H. L. Akin, M. O. Tokhi and G S Virk (Ed.), World Scientific, 2009*.

Grollman, D., and Jenkins, O. 2007. Dogged learning for robots. In *International Conference on Robotics and Automation (ICRA 2007)*, 2483–2488.

Liu, J., and Veloso, M. 2008. Online ZMP Sampling Search for Biped Walking Planning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08), Nice, France, September 2008*.

Nakanishi, J.; Morimoto, J.; Endo, G.; Cheng, G.; Schaal, S.; and Kawato, M. 2004. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems* 47(2-3):79 – 91. Robot Learning from Demonstration.

Nintendo. 2007. Nintendo - Wii Game Controllers. http://www.nintendo.com/wii/what/controllers.

Röfer, T.; Laue, T.; Müller, J.; Bösche, O.; Burchardt, A.; Damrose, E.; Gillmann, K.; Graf, C.; de Haas, T. J.; Härtl, A.; Rieskamp, A.; Schreck, A.; Sieverdingbeck, I.; and Worch, J.-H. 2009. B-Human 2009 Team Report. Technical report, DFKI Lab and University of Bremen, Bremen, Germany. http://www.b-human.de/download.php?file=coderelease09_doc.

Strom, J.; Slavov, G.; and Chown, E. 2009. Omnidirectional walking using zmp and preview control for the nao humanoid robot. In *RoboCup 2009: Robot Soccer World Cup XIII*.