Improving Biped Walk Stability Using Real-Time Corrective Human Feedback

Çetin Meriçli[1,2] and Manuela Veloso[1]

[1] Computer Science Department,
Carnegie Mellon University
Pittsburgh, PA 15213, United States
[2] Computer Engineering Department,
Boğaziçi University
Bebek, Istanbul, Turkey
{cetin,veloso}@cmu.edu

Abstract. Robust walking is one of the key requirements for soccer playing humanoid robots. Developing such a biped walk algorithm is non-trivial due to the complex dynamics of the walk process. In this paper, we first present a method for learning a corrective closed-loop policy to improve the walk stability for the Aldebaran Nao robot using real-time human feedback combined with an open-loop walk cycle. The open-loop walk cycle is obtained from the recorded joint commands while the robot is walking using an existing walk algorithm as a black-box unit. We capture the corrective feedback signals delivered by a human using a wireless feedback mechanism in the form of corrections to the particular joints and we present experimental results showing that a policy learned from a walk algorithm can be used to improve the stability of another walk algorithm. We then follow up with improving the open-loop walk cycle using advice operators before performing real-time human demonstration. During the demonstration, we then capture the sensory readings and the corrections in the form of displacements of the foot positions while the robot is executing improved open-loop walk cycle. We then translate the feet displacement values into individual correction signals for the leg joints using a simplified inverse kinematics calculation. We use a locally weighted linear regression method to learn a mapping from the recorded sensor values to the correction values. Finally, we use a simple anomaly detection method by modeling the changes in the sensory readings throughout the walk cycle during a stable walk as normal distributions and executing the correction policy only if a sensory reading goes beyond the modeled values. Experimental results demonstrate an improvement in the walk stability.

Keywords: complex motor skill acquisition, learning from demonstration, motion and sensor model learning, human-robot interfaces.

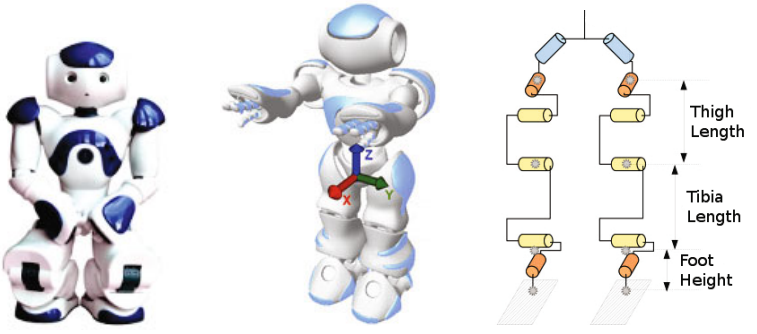
1 Introduction

Biped walk learning is a challenging problem in humanoid robotics due to the complex dynamics of walking. Developing efficient biped walking methods on commercial humanoid platforms with limited computational power is even more challenging since the

J. Ruiz-del-Solar, E. Chown, and P.G. Ploeger (Eds.): RoboCup 2010, LNAI 6556, pp. 194–205, 2011.

developed algorithm should be computationally inexpensive, and it is not possible to alter the hardware.

The Nao (Fig.1), is a 4.5 kilograms, 58 cm tall robot with 21 degrees of freedom (www.aldebaran-robotics.com). It does not have separate hip yaw joints for the legs. Instead, both legs have mechanically connected hip yaw-pitch joints perpendicular to each other along the Y-Z plane (Fig.1(c)) and these two joints are driven by a single motor. The Nao is equipped with a variety of sensors including a 3-axis accelerometer, a 2-axis (X-Y) gyroscope, and an inertial measurement unit for computing the absolute torso (upper body of the robot) orientation using accelerometer and gyroscope data. The inertial measurement unit, the accelerometer, and the gyroscope sensors use a right-hand frame of reference (Fig.1(b)). We use the term "Yaw" for rotation along the Z axis, "Roll" for rotation along the X axis, and "Pitch" for rotation along the Y axis throughout the text.



(a) (b) (c)

Fig. 1. a) The Nao robot b) The frame of reference for sensors. c) Kinematic configuration of the legs.

Numerous research studies have been published on the biped walk algorithms for the Nao robot since the introduction of the Nao to the RoboCup SPL (www.robocup.org). Graf *et al.* present an omni-directional walking algorithm using parameterized gaits and sensor feedback [1,2]. Liu and Veloso develop an efficient Zero Moment Point (ZMP) search method [3,4]. Gökçe and Akın utilize Evolutionary Strategy (ES) to tune the parameters of a Central Pattern Generator (CPG) based walk [5]. Strom *et al.* present a ZMP based omnidirectional walking algorithm [6]. Czarnetzki et al. propose a preview control based method for keeping the ZMP on the desired path [7].

There have also been approaches utilizing the learning from demonstration paradigm for task and skill learning. Nakanishi *et al.* present a method for biped walk learning from human demonstrated joint trajectories using dynamical movement primitives [8]. Grollman and Jenkins propose a learning from demonstration framework called "Dogged Learning" [9] and successfully applied it to learning quadruped walking and a set of skills related to playing soccer on a Sony Aibo robotic dog [10]. Argall *et al.* present a learning from demonstration and corrective feedback method for low level motion planning of a

Segway RMP robot [11,12]. Chernova and Veloso present a sliding autonomy framework for teaching tasks to single robots [13] and multi-robot systems [14].

In this paper, We first describe a method for obtaining a single walk cycle using an existing walk algorithm and how the obtained walk cycle can then played back to presented an overview of the method, along with initial experimentation on two different walk algorithms. We present experimental results demonstrating how a correction policy learned using an existing walk algorithm is able to improve the walk stability on both the initial algorithm and a second algorithm, showing that the learned correction policy using the proposed method does not depend on the underlying walk algorithm.

We then contribute a biped walk stability improvement algorithm using human feedback consisting of three phases, where the first phase being the walk cycle capture and playback presented in the first part. In the second phase, we present an offline improvement method for the open-loop walk using advice operators. Finally, we introduce a closed-loop feedback policy learning method which uses the corrective human demonstration given in real-time in the form of foot position displacements to learn a mapping from the sensory readings to a corresponding correction value for the positions of the feet. We present experimental results for the performance evaluation of the learned policy against the open-loop playback algorithm, and the open-loop playback algorithm improved using advice operators. The results show improvement at second and third phases over the performance of the initial phase.

2 Proposed Approach

Walking is a periodic phenomenon and consists of consecutive walk cycles which starts with a certain configuration of the joints and ends when the same configuration is reached again. A walk cycle wc is a motion segment of duration T timesteps, where $wc_j(t), t \in [0, T)$ be 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 12 of them named $Joints$: arm roll, hip roll, hip pitch, knee pitch, ankle pitch, and ankle roll joints for the left and the right arms and the legs.

2.1 Obtaining an Open-Loop Walk

We use an existing walk algorithm as a black-box and we collect a number of walk sequences where the robot is walking forwards for a fixed distance at a fixed speed using the black-box algorithm. We save the sequences in which the robot was able to travel the determined distance without losing its balance. A set of many example walk sequences where the robot walks without falling provide

Many examples of the robot walking without falling provide data D for each $t, t \in [0, T)$, in the form of the commands received for each joint $\overrightarrow{D_j}(t)$ and the sensory readings $\boldsymbol{S}(t)$ for the set of sensors $Sensors$. 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 addition, we fit a normal distribution on the readings of each sensor at each t $\overrightarrow{\mu(t)}, \overrightarrow{\sigma(t)}$ where $\mu_s(t)$ is the mean, and $\sigma_s(t)$ is the standard deviation for the readings of the sensor $s \in Sensors$ at time t in the walk cycle (Fig.2).

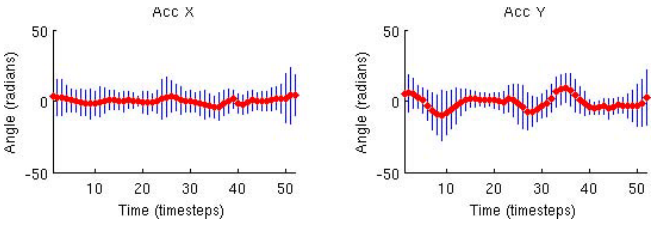


Fig. 2. Distribution of the sensor values over the complete walk cycle for a stable walk sequence. The middle line denotes the mean and the vertical lines denote +/- 3σ variance. The X axis is timesteps, and the Y axis is the sensor value.

The actual movement of the robot differs from the desired movement as a result of the various sources of uncertainty associated with the sensing and actuation (Fig.3) and therefore it is not possible to have an open-loop walk behavior that can walk indefinitely without falling.

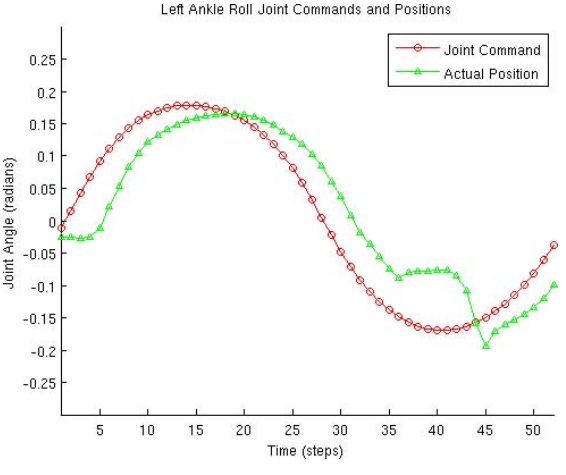


Fig. 3. An example to the actuation error from the ankle roll joint of the left leg. The plot with circles shows the joint commands, and the plot with triangles shows the actual trajectory the joint has followed. The error towards the end is caused by the weight of the robot on the left ankle of the robot while it is taking a right step and is standing on its left leg.

The changes in sensory readings when the robot is about to lose its balance can be used to derive a correction policy by mapping these changes to appropriate modifications on the walk cycle (Fig.4). The next subsection describes a method for obtaining a closed-loop walk using sensory readings and human demonstration.

2.2 Correction Using Sensor-Joint Couplings

In our previous research, we presented a method where we introduce the idea of using human demonstration to learn a closed-loop correction policy [15]. We used the hip

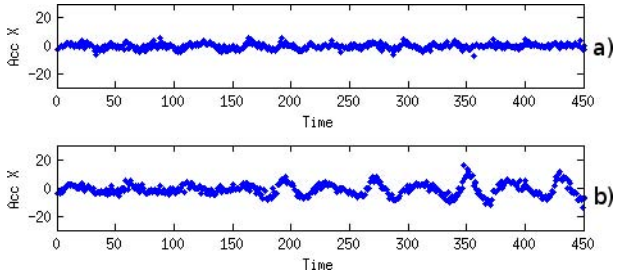


Fig. 4. Sample accelerometer readings: a) a stable walk sequence, and b) a walk sequence in which robot starts losing its balance after walking for some amount of time

roll and the hip pitch joints to apply the correction signals. We defined the correction function 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 noisy nature of sensors causes fluctuations in the sensory readings which may result in jerky motions and therefore loss of balance when used directly to generate a correction signal. We smooth the sensory readings using running mean smoothers. We use human demonstration to learn the mapping function from sensor readings to the correction signals. The human demonstrator provide the correction signals in the form of angle offsets to the joint commands using a wireless game controller interface. We model the received demonstration data as a function of accelerometer data by fitting normal distributions.

We used the walking algorithm proposed by Liu and Veloso which uses online ZMP sampling [3,4] for learning the correction policy. The efficiency of learned feedback policy is then evaluated using the default walk algorithm provided by Aldebaran Robotics with default parameters and 30 timesteps per walking step (W1), and Liu and Veloso's walking algorithm [4,3] based on online ZMP sampling (W2).

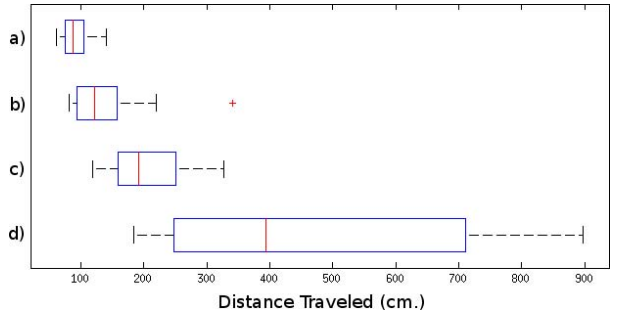


Fig. 5. Performance evaluation results: a) W1, open-loop, b) W1, learned policy using accelerometer readings, c) W2, open-loop, d) W2, learned policy using accelerometer readings

For each algorithm, 10 runs with original open-loop method and 10 runs using the learned policy for correction were conducted and the distance traveled before falling was recorded. The results are given in Fig.5. The plus sign is an outlier, the maximum distance that the learned policy on original Aldebaran walk traveled, 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. The learned policy has improved the performance of both algorithms signicantly despite that the policy was derived only using Liu and Veloso's algorithm.

Following up this experimentation, we extend the correction framework to include offline advice operators and multi-joint corrections with locally weighted regression as the function approximator.

2.3 Advice Operators

Advice Operators Policy Improvement (A-OPI) is a method for improving the execution performance of the robot in a human-robot LfD setup [12]. Advice operators provide a *language* between the human teacher and the robot student, allowing the teacher to give *advice* as a mathematical function to be applied on the observations and/or actions. The resulting data is then used to re-derive the execution policy. Advice operators are especially useful in domains with continuous state/action spaces where the correction must be provided in continuous values.

We use A-OPI for correcting the obtained walk cycle in its open-loop form based on human observations of the executed walk behavior. We define three advice operators that are applied on the walk cycle:

- **ScaleSwing(f):** Scales the joint commands of hip roll joints (along X axis) in the walk cycle by a factor f where $f \in [0, 1]$. Hip roll joints generate the lateral swinging motion while walking.
- **ChangeFeetDistance(d):** Applies an offset of d millimeters to the distance between the feet along Y axis.
- **SetArms(**$angle$**):** Raises or lowers the arms by $angle$ radians along the Y-Z plane with respect to their baseline.

After a set of iterations consisting of execution of the walk behavior, receiving advice from the teacher, and revising the walk cycle accordingly, an improvement has been achieved. The initial and improved versions of hip roll joint values to generate lateral swinging motion are shown in Fig.6 as an example.

2.4 Using Human Demonstration for Learning Correction Policy

We introduced a wireless feedback delivery method without touching the robot in [15]. The proposed feedback method utilizes the Nintendo Wiimote commercial game controller [16] to provide corrective demonstration to the robot (Fig.7). 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.

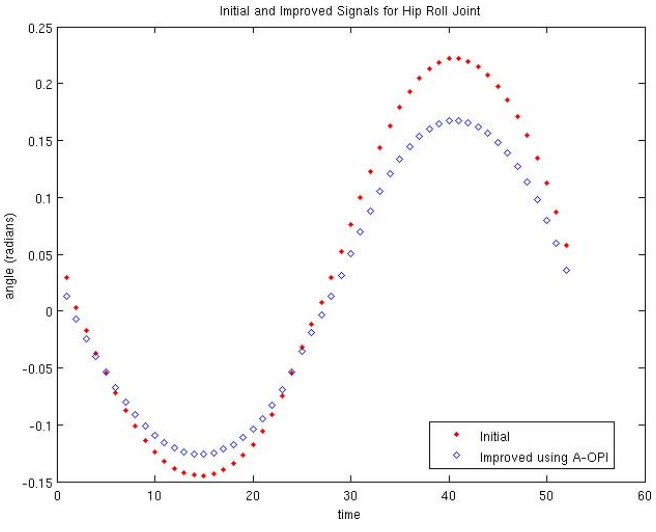


Fig. 6. Initial and improved joint commands for hip roll joints generating swinging motion while walking

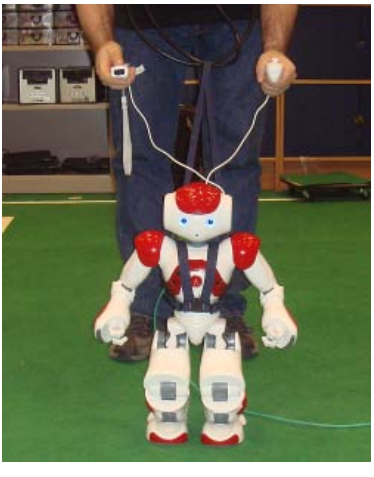


Fig. 7. A snapshot from a demonstration session. A loose baby harness is used to prevent possible hardware damage in case of a fall. The harness neither affects the motions of the robot nor holds it as long as the robot is in an upright position.

A scaling factor γ is applied on the Wiimote readings before they are sent to the robot. We used $\gamma = 50$ in our implementation. The changes in the orientations of the Wiimote handles (in radians) are mapped to the foot position displacements on the robot (in millimeters). Four measurable axes of the controller handles, namely Nunchuk yaw, Nunchuk pitch, Wiimote yaw, and Wiimote pitch, are used to control the displacement

of the left foot along the X axis, the left foot along the Y axis, the right foot along the X axis, and the right foot along the Y axis, respectively (Fig.8).

2.5 Correction Using Sensor-Feet Position Couplings

At each timestep of playback, the vector of joint command angles for that timestep is used to calculate relative positions of the feet in 3D space with respect to the torso using forward kinematics.The calculated corrections (in the autonomous mode), or the received corrections (during the demonstration) are applied on the feet positions in 3D space and the resulting feet positions are converted back into a vector of joint command angles using inverse kinematics.

Due to the physically connected hip-yaw joints of the Nao, inverse kinematics for feet positions cannot be calculated independently for the feet. Graf *et al.* propose an analytical solution to inverse kinematics of the Nao presenting a practical workaround for the connected hip-yaw pitch joints constraint [1]. We use a simplified version of this approach by assuming the hip-yaw joints to be fixed at 0 degrees for the straight walk.

The demonstrator uses the wireless interface to modify the robot's motion in real time while the robot is walking using the refined open-loop walk cycle. The correction values received during the demonstration are recorded synchronously with the sensory readings, tagged with the current position in the walk cycle. Each point in the resulting demonstration dataset is a tuple $< t, \overrightarrow{S}, \overrightarrow{C} >$ where t is the position in the walk cycle at the time when this correction is received, $\overrightarrow{S}$ is the vector of sensory readings, and $\overrightarrow{C}$ is the vector of received correction signals with $\overrightarrow{S} = \{Acc_X, Acc_Y\}$ being the accelerometer readings, and $\overrightarrow{C} = \{C_X^{left}, C_Y^{left}, C_X^{right}, C_Y^{right}\}$ being the received correction values for the left foot along the X axis, the left foot along the Y axis, the right foot along the X axis, and the right foot along the Y axis, respectively.

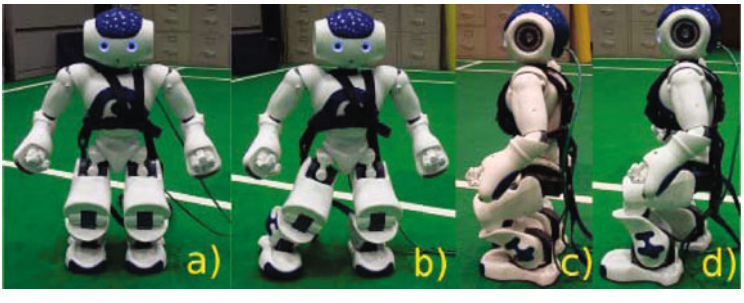


Fig. 8. Example corrections using the Wiimote. Rolling the Wiimote to the right takes the right leg of the robot from its neutral posture (a) to a modified posture along the Y axis (b). Similarly, tilting the Wiimote forward brings the right leg of the robot from its neutral posture (c) to a modified posture along the X axis (d).

We utilize locally weighted regression with a Gaussian kernel [17] for generalizing a policy using the recorded correction and sensor values. For each received sensor reading vector $\overrightarrow{S}$, we calculate a correction vector $\overrightarrow{C}$ as follows:

$$d_i = e^{-\sqrt{(\overrightarrow{S} - \overrightarrow{S_i}(t))^T \Sigma^{-1} (\overrightarrow{S} - \overrightarrow{S_i}(t))}}$$

$$\overrightarrow{C} = \frac{\sum_i d_i \overrightarrow{C_i}(t)}{\sum_i d_i}$$

where Σ is the covariance matrix of the sensory readings in the demonstration set, $\overrightarrow{C_i}(t)$ is the i^{th} received correction signal for the walk cycle position t, $\overrightarrow{S_i}(t)$ is the i^{th} sensory reading for the walk cycle position i, $\overrightarrow{S}(t)$ is the current sensory reading, $\overrightarrow{C}$ is the calculated correction value to be applied, and t is the current position in the walk cycle.

The calculated correction values are applied only if any of the sensor values are not in the range $\mu_t \pm K\sigma_t$ (i.e., an abnormal value is read from that sensor) where K is a coefficient, and t is the current position in the walk cycle. In our implementation, we chose $K = 3$ so the correction values are applied only if the current sensory readings are outside the $\mu_s(t) \mp 3\sigma_s(t)$, corresponding to the $\%99$ of the variance of the initial sensory model. Algorithm 1 uses sensor-foot position couplings to perform a closed-loop walk.

Algorithm 1 Closed-loop walking using sensor-foot position couplings. Pos_{left} and Pos_{right} are the positions of the feet in 3D space.

$t \leftarrow 0$
loop
  $\overrightarrow{S(t)} \leftarrow readSensors()$
  $\overrightarrow{S(t)} \leftarrow smooth(\overrightarrow{S(t)})$
  $Pos_{left}, Pos_{right} \leftarrow forwardKine(wc(t))$
  if $(\mu_s(t) - K\sigma_s(t) \leq S_s(t) \leq \mu_s(t) + K\sigma_s(t))$ **then**
    $C_{left}, C_{right} \leftarrow 0$
  else
    $C_{left}, C_{right} \leftarrow correction(\overrightarrow{S(t)})$
  end if
  $Pos_{left} \leftarrow Pos_{left} + C_{left}$
  $Pos_{right} \leftarrow Pos_{right} + C_{right}$
  $NextAction \leftarrow inverseKine(Pos_{left}, Pos_{right})$
  $t \leftarrow t + 1 \pmod T$
end loop

3 Experimental Results

We evaluated the performance of the proposed method on a flat surface covered with regular RoboCup SPL field carpet. We used the walking algorithm proposed by Liu and Veloso as the black-box open-loop algorithm. The duration of the extracted walk cycle is 52 individual timesteps, approximately corresponding to one second. During two

demonstration sessions of about 18 minutes, a total of 53014 data points are recorded. 19428 data points corresponding to about 373 walk cycles are selected as good examples of corrective demonstration by visually inspecting the demonstration data based on the changes in the sensory readings towards the recovery of the balance.

We evaluated the following algorithms:

- Initial open-loop playback walk.
- Open-loop playback walk after offline correction using advice operators.
- Closed-loop playback walk using the learned policy from real-time corrective demonstration.

For each algorithm, we made 10 runs and we measured the distance traveled before falling. The results are given in Fig.9. Although an improvement has been achieved by the sole application of the advice operators, the learned policy was able to improve the stability furthermore. Both algorithms were able to reach 1130 centimeters, which was the maximum distance available in the experimental setup. While the open-loop playback walk with advice operators was able to reach the limit only once, the learned policy was able to reach the limit several times.

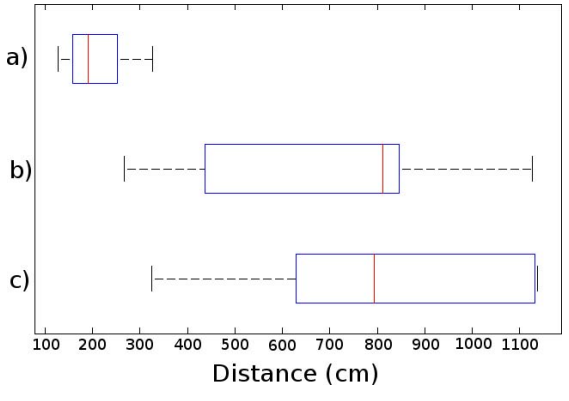


Fig. 9. Performance evaluation results: a) initial open-loop playback walk, b) open-loop playback walk improved using advice operators, c) closed-loop playback walk using learned correction policy

4 Conclusions

In this paper, we presented a method for learning a corrective policy for improving the walk stability on the Nao humanoid robot. Our method plays a single walk cycle obtained using an existing walk algorithm back to obtain an open-loop walk behavior and uses real-time corrective human demonstration in the form of single joint angle corrections delivered using Wiimote wireless game controllers to learn a correction policy for the open-loop playback walk. In the first part, we investigated the possibility of using a learned policy on a different walk algorithm and presented experimental results

showing that the learned policies do not depend on the underlying walk algorithm. In the second part, we proposed an extension over the initial version where the corrective feedback signals are in the form of foot position displacements and the sensory readings recorded at the time of correction signal are then used to learn a mapping from the sensory readings to a corresponding correction value. We also introduced an offline improvement using advice operators to improve the stability of the open-loop walk cycle. We presented experimental results demonstrating the learned policy outperforms the initial open-loop and improved open-loop using advice operators.

Addressing the delay between the perception and the actuation of the demonstrator, generalizing the proposed three-phase approach to a multi-phase learning framework applicable to other skill learning problems, investigating better policy derivation methods, improving the demonstration interface usability, relaxing the flat surface assumption to cope with uneven terrain, and extending the balance maintenance capability to endure against moderate pushes in adversarial domains (i.e., the robot soccer) are among the issues we aim to address in the future.

Acknowledgments

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

1. Graf, C., Härtl, A., Röfer, T., Laue, T.: A robust closed-loop gait for the standard platform league humanoid. In: Zhou, C., Pagello, E., Menegatti, E., Behnke, S., Röfer, T. (eds.) Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2009 IEEE-RAS International Conference on Humanoid Robots, Paris, France, pp. 30–37 (2009)
2. 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., Worch, J.-H.: B-Human 2009 Team Report. Technical report, DFKI Lab and University of Bremen, Bremen, Germany (2009), `http://www.b-human.de/download.php?file=coderelease09_doc`
3. Liu, J., Chen, X., Veloso, M.: Simplified Walking: A New Way to Generate Flexible Biped Patterns. In: Tosun, O., Akin, H.L., Tokhi, M.O., Virk, G.S. (eds.) Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2009), Istanbul, Turkey, September 9-11. World Scientific, Singapore (2009)
4. Liu, J., Veloso, M.: Online ZMP Sampling Search for Biped Walking Planning. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), Nice, France (September 2008)
5. Gokce, B., Akin, H.L.: Parameter optimization of a signal-based biped locomotion approach using evolutionary strategies. In: Tosun, O., Akin, H.L., Tokhi, M.O., Virk, G.S. (eds.) Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2009), Istanbul, Turkey, September 9-11. World Scientific, Singapore (2009)

6. Strom, J., Slavov, G., Chown, E.: Omnidirectional walking using ZMP and preview control for the NAO humanoid robot. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS, vol. 5949, pp. 378–389. Springer, Heidelberg (2010)
7. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems 57, 839–845 (2009)
8. Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., Kawato, M.: Learning from demonstration and adaptation of biped locomotion. Robotics and Autonomous Systems 47 (2-3), 79–91 (2004); Robot Learning from Demonstration
9. Grollman, D.H., Jenkins, O.C.: Dogged learning for robots. In: International Conference on Robotics and Automation (ICRA 2007), Rome, Italy, pp. 2483–2488 (April 2007)
10. Grollman, D.H., Jenkins, O.C.: Learning elements of robot soccer from demonstration. In: International Conference on Development and Learning (ICDL 2007), London, England, pp. 276–281 (July 2007)
11. Argall, B., Browning, B., Veloso, M.: Learning from demonstration with the critique of a human teacher. In: Second Annual Conference on Human-Robot Interactions (HRI 2007) (2007)
12. Argall, B., Browning, B., Veloso, M.: Learning robot motion control with demonstration and advice-operators. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008) (2008)
13. Chernova, S., Veloso, M.: Interactive policy learning through confidence-based autonomy. Journal of Artificial Intelligence Research 34 (2009)
14. Chernova, S., Veloso, M.: Teaching collaborative multirobot tasks through demonstration. In: Proceedings of AAMAS 2008, the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems, Estoril, Portugal (May 2008)
15. Meriçli, Ç., Veloso, M.: Biped walk learning through playback and corrective demonstration. In: AAAI 2010: Twenty-Fourth Conference on Artificial Intelligence (2010)
16. Nintendo. Nintendo - Wii Game Controllers (2007), http://www.nintendo.com/wii/what/controllers
17. Atkeson, C., Moore, A., Schaal, S.: Locally weighted learning. AI Review 11, 11–73 (1997)