The 10th Workshop on Humanoid Soccer Robots
15th IEEE-RAS International Conference on Humanoid Robots, Seoul, Korea, November 2015

Reinforcement Learning Inspired Disturbance Rejection and Nao Bipedal Locomotion

Bernhard Hengst
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
Email: bernhardh@cse.unsw.edu.au

***Abstract*—Competitive bipedal soccer playing robots need to move fast and react quickly to changes in direction while staying upright. This paper describes the application of reinforcement learning to stabilise a flat-footed humanoid robot. An optimal control policy is learned using a physics simulator. The learned policy is supported theoretically and interpreted on a real robot as a linearised continuous control function. The paper also describes other components, including foot-step coordination, of bipedal locomotion integrated to achieve reactive omni-directional locomotion for Nao robots used in the RoboCup Standard Platform League.**

I. Introduction

Soccer playing bipedal robots require competitive omnidirectional speed and the agility to change their dynamic pose quickly without falling over. For example, a robot may need to accelerate to a fast forward pace, then reverse direction quickly, and possibly walk backwards at the same fast pace. In addition to the forces generated by these changes in direction the robot has to stay balanced given imperfections in the field and while being jostled by other robots.

The challenge is greater using stubby and inexpensive robots such as the Naos. Inverted pendulum models are frequently used in modelling bipedal locomotion. We know that a short pendulum falls faster than a tall one. This makes the use of foot-step placement to balance a stubby robot more difficult. Keeping cost contained to make these robots widely affordable has meant that sensors are generally more noisy and less reliable than on more expensive models. In addition, the manufacturing process and robot wear-and-tear introduces variations in the electro-mechanical properties. All these factors make it difficult to achieve stable bipedal locomotion.

The objective is to overcome these challenges and to develop competitive omnidirectional bipedal locomotion for the Nao robot for use in the RoboCup Standard Platform League competitions. Our approach to bipedal locomotion is to develop closed-loop motion independently in the sagittal and coronal planes. These motions are synchronised to produce an omni-directional gait. This paper will largely focus on the stabilisation in the sagittal plane to keep the robot body upright and arrest any forward or backward sway. The approach is to first learn a control policy using reinforcement learning techniques, which is then adapted for use on the real-robot.

As real robots wear rapidly and require expensive repairs when they break, an accurate physics simulator is used for the repeated trials necessary to learn a policy using reinforcement learning. Figure 1 shows the simulated rendition of the robot. Despite the accurate state information available from the simulator, it was not possible to learn to balance a Nao with point feet in the sagittal plane with a few discrete foot-step actions. Point feet are easily simulated by collapsing the feet to a thin blade at their centre-of-mass. Our reinforcement learner was not able to learn to arrest the fall by changing the step-size alone at 4 Hz.

On the real robot noisy inertial and foot pressure measurements make even continuous control via stride length adjustment difficult. The stride length adjustment was calculated algebraically using the inverted pendulum equations as in [1], but this alone did not achieve a smooth stable walk.

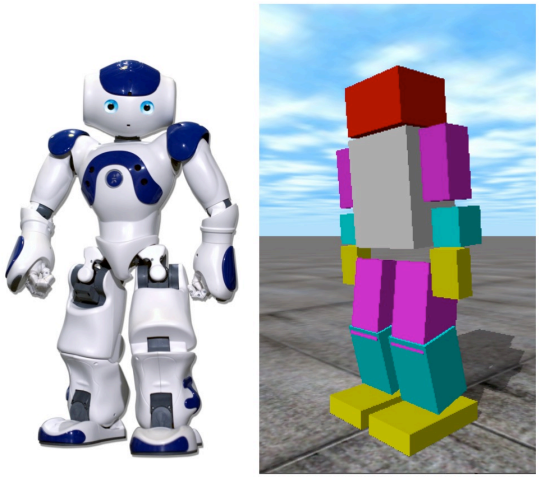


Fig. 1. Nao Robot and Simulated Version. While the box-like rendition of the simulated Nao may not look realistic, the ODE simulator has been programmed with the precise dimensions, masses, and joint-locations from the manufacturer's specification.

Fortunately flat feet provide another method of balance control. By adjusting the ankle-tilt the centre-of-pressure (CoP) on the ground can be shifted to lie between the heel and the toe of the support foot. As the motor controller runs at 100 Hz on the Nao, the ankle-tilt can assert higher fidelity control, in which the fast fall of a stubby robot can be made to work to our advantage to rapidly accelerate and decelerate the robot.

The rest of this paper first provides a short review of related work on bipedal locomotion. After an introduction to reinforcement learning, its application to sagittal balance control is described. The learned policy is interpreted and justified theoretically using a linear inverted model and implemented on the Nao with a simple continuous control function. The paper also discusses foot-step control and other components that integrate to make bipedal locomotion for the Nao complete.

II. Related Work

Accounts of the fascination with humanoid locomotion date back to King Mu of Zhou's time (976-922 BC) [2] and now, more than ever, it is still an active research area.

Passive-dynamic walkers based on a "rimless wheel" model were originally developed by McGeer [3], improved by others, and powered to walk on level ground [4]. These machines have no explicit controllers yet exhibit human-like motions. They are limited in their behaviour repertoire.

Much research has been devoted to planar bipedal locomotion. An extensive exposition is given by Westervelt et al [5]. Planar research has been extended to 3D [6]. Grizzle reminded researchers that even the simplest bipedal locomotion is challenging, let alone aperiodic walks, non-flat ground, etc [7]. These approaches follow the control system methodology where system identification is manual and specified using differential equations. Tracking and control of the Zero Moment Point (ZMP) using modern control theory is employed for the HRP series of robots [8] [9]. The approach uses preview-control, a feedforward mechanism that plans ahead using the anticipated target ZMP.

Reinforcement learning (RL) is a machine learning technique that can learn optimal control actions given a goal specified in terms of future rewards. RL can be effective when the system dynamics are unknown, are highly non-linear or complex. The literature on bipedal walking is extensive with several approaches using RL, for example: neural network function approximation to learn to walk slowly on a simulator [10]; coronal plane control using an actuated passive walker [11]; point feet foot placement [12] [13] [14] [12]; learning central pattern generator parameters [15]; and CMAC function approximation to learn the parameters of a swing-leg policy [16].

For the type of robot of concern in this paper, related approaches include the hand-coded gyroscope feedback and pause reset control for the sagittal and coronal planes respectively [17]. Uncannily, the gyroscope feedback controller is almost identical to the controller developed in this paper using reinforcement learning. This paper will elucidate why this type of control is effective. Another approach is the use of analytic methods [1], where control is asserted using an iterative calculation based on an inverted pendulum model to adjust the placement of the swing-leg. The ankle-joint does no seem to be directly actuated as a control variable, but by keeping the foot flat the ground, there is some implicit control to counteract unplanned movements.

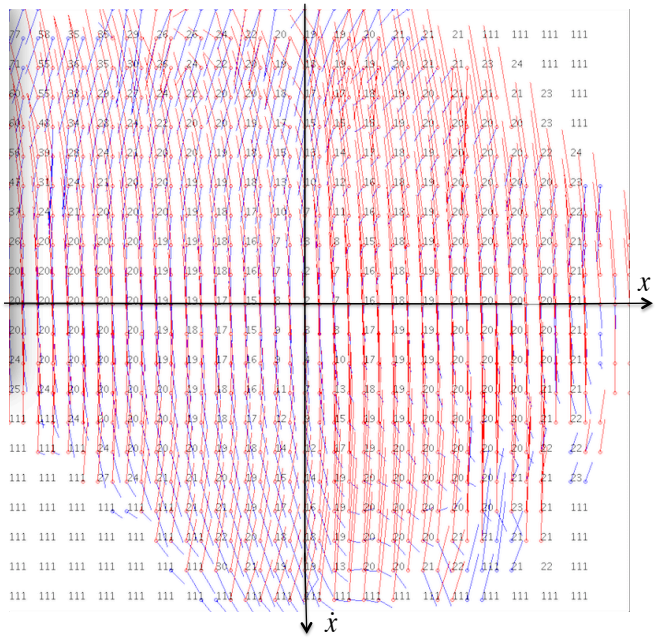
III. Reinforcement Learning Sagittal Plane Disturbance Rejection

The simulator for the Nao was built using the Open dynamics Engine (ODE) after discovering that the Bullet physics engine sometimes behaves erratically modelling the 0.01 second duration state transitions. While the simulated robot was composed by simply linking boxes as shown in Figure 1, the dimensions, joint-positions, and masses were taken from the manufacturer's specification and believed to accurately reflect those of the physical robot.

A. Reinforcement Learning

RL is based on an underlying Markov Decision Problem (MDP) given by a tuple $\langle S, A, P, R \rangle$. S is a set of states. A is a set of actions. $P : S \times A \times S \rightarrow [0, 1]$ is a state transition function giving the probability, $P(s, a, s')$, of moving to state $s' \in S$ after the next time-step starting in state $s \in S$ and taking action $a \in A$. $R : S \times A \rightarrow \mathbb{R}$ is the expected reward value $R(s, a)$ for the next time-step when in state s and taking action a. For episodic problems the objective can be to maximise the sum of future rewards until termination. The optimal policy $\pi^* : S \rightarrow A$ is a function from states to actions that achieves this objective. In reinforcement learning it is useful to learn the optimal action-value function $Q^* : S \times A \rightarrow \mathbb{R}$ that is defined as the sum of rewards received stating in state s, taking action a and then following the optimal policy $\pi^*(s)$. The optimal action a^* in state s can be derived from the Q function: $a^* = \pi^*(s) = argmax_a Q^*(s, a)$.

To model sagittal stabilisation using ankle-tilt control as a reinforcement learning problem, we let $S = (x, \dot{x}, a)$ where x represents the position of the Centre-of-Mass (CoM) of the robot in the forward-back direction with the origin at the point of the CoM with the robot stationary and torso upright, $\dot{x}$ is the velocity of the CoM, and a is the last action taken. The last action is included in the state description to better approximate a Markov state, as there is a lag of about one time-step before the action is fully implemented. $A = \{-0.03, -0.015, 0.0, 0.015, 0.03\}$ radians and represents the ankle-tilt as a variation from the upright position. Actions are indexed 0, 1, 2, 3, and 4. The transition function is learned using the simulator by randomly selecting a new action from A with a 10% probability at each time-step. This exploration policy ensures that durative actions are explored as well as rapidly changing actions. The reward function shapes the policy and is -1.0 at every time-step with an extra penalty proportional to the size of the action change, and a penalty if the action is not 0.0. The idea is to reduce the wear on the robot by minimising ankle-tilt changes with a preference for the upright position. The problem is terminated in a goal state when x and $\dot{x}$ are close to zero. The function approximator linearly interpolates Q values between sample points in the continuous $(x, \dot{x})$ state-space. The model is represented and learned by storing the transition and reward functions. The policy is learned using the model to solve the MDP [18].



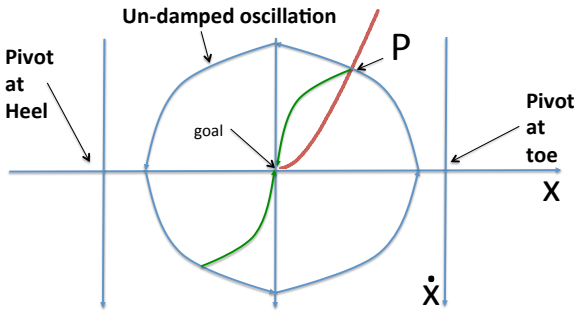


Fig. 2. TOP: Optimal value function sampled over $(x, \dot{x})$ state space showing cost to the goal reducing closer to $(0.0, 0.0)$ as expected. Blue and red lines show the state transition for different actions. BOTTOM: Schematic showing undamped transitions when swaying forwards and backwards, and bringing the sway to a halt by pivoting at the heel at point P.

Figure 2 (top) shows the state transitions stored for toe-heel ankle-tilt actions sampled at grid points in the state-space by the exploration policy on the simulator. The value function of the sum of future cost (negative reward) is overlaid and can be seen to reduce towards the goal. Figure 3 shows the optimal policy as action indices that arrest the movement by spiralling the state trajectory towards the goal, $(x = 0, \dot{x} = 0)$. The interpretation is that as the CoM of the robot moves towards the origin, the ankle-tilt is activated to move the CoP towards the toe or heel of the foot, just at the right time to decelerate robot to an upright and stationary position. Figure 2 (bottom) shows the idea schematically. An un-damped robot without ankle-tilt control would oscillate back and forwards

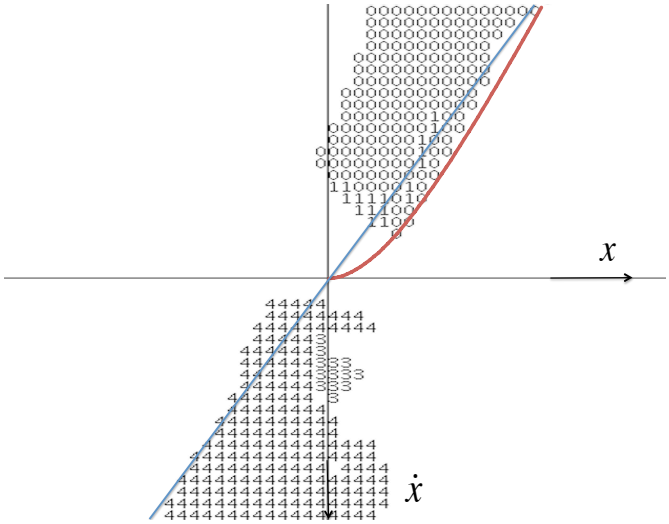


Fig. 3. Optimal policy sampled over $(x, \dot{x})$ state space. The numbers refer to action identifiers in A of the MDP with a blank representing action 2 for clarity, i.e. action 0.0. The policy shows that non-zero ankle-tilt actions are used as the robot approaches the upright state at which time it tries to arrest the motion.

(blue trajectory). If at point P the ankle is tilted so that the CoP/pivot moves to the toe, then the forward motion would be arrested and the robot sway stops dead (green trajectory), at which time the ankle-tilt is adjusted back to keep the robot upright.

B. Idealised Flat Footed Linear Inverted Pendulum Model

A post-hoc algebraic analysis with a simple inverted pendulum model confirms the optimal ankle-tilt control policy from RL. Using the inverted pendulum equations, the point (P in Figure 2) at which to apply the ankle-tilt force to arrest the sway is derived. From [1]:

$$x(t) = x_0\, cosh(kt) + \dot{x}_0\, sinh(kt)/k \quad (1)$$

$$\dot{x}(t) = x_0\, sinh(kt)\, k + \dot{x}_0\, cosh(kt) \quad (2)$$

where $k = \sqrt{g/h}$, g is the acceleration due to gravity, h is the height of the CoM of the pendulum, and x_0 and $\dot{x}_0$ are the position and velocity of the CoM at time = 0 relative to the pivot of the pendulum. Superscript f is used to denote the variables when the pivot is located at the heel or toe. Given an initial velocity $\dot{x}_0^f$ and the distance to the toe or heel, f, the aim is to find the initial point x_0^f so that the pendulum comes to rest, $\dot{x}_t^f = 0$, and the robot is upright, $x^f(t) = f$, hence $\dot{x}(t) = 0$ and $x(t) = 0$. Figure 4 shows the meaning of the variables.

Solving the equations simultaneously yields:

$$x_0^f = f\, cosh(kt) \quad (3)$$

where $t = sinh^{-1}(-\dot{x}_0^f/fh)/k$. Figure 3 shows the plot of $x = x_0 = x_0^f - f$ for different values of $\dot{x} = \dot{x}_0 = \dot{x}_0^f$ (brown curve), corroborating the results from reinforcement learning. The (brown) control curve is also shown in Figure 2. Point P is the point at which the ankle-tilt control action is initiated.

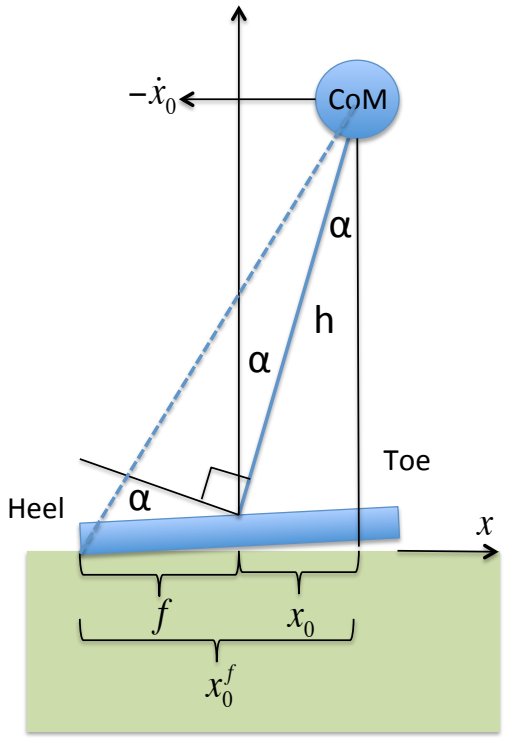


Fig. 4. Diagram showing an ankle-tilt α where the pivot of the foot of the robot is at the heel slowing down the backward velocity of the robot. Variables used in the algebraic analysis are also shown.

C. Stabilisation Policy Implemented on the Nao

The final implementation of the sagittal controller on the Nao adjusts the ankle-tilt α proportional to the smoothed value of the gyroscope y-axis reading y.

$$\alpha = Ky \tag{4}$$

A gyroscope measures the rate of change of angle in radians per second. For small angles the velocity of the torso $\dot{x}$ is proportional the y-gyro, and the ankle-tilt, α, is proportional to the CoM displacement of an upright robot, x. The simple linear controller (Equation 4) is justified because we can approximate the curve in Figure 3 by a line through the origin (in blue) as shown. The raw gyroscope values are filtered using a fixed-gain Kalman filter after being calibrated to read zero when the robot is stationary.

The CoM for the upright posture while walking is repositioned to be about halfway between the toe and the heel to ensure that the Nao can leverage the flat foot equally to brake the sway both forwards and backwards. We stand the robot up with knees locked and the "stiffness" significantly reduced to rest the robot when it is not walking. In this posture the CoM must be moved back over the ankle-joints so that the robot does not fall forwards under its own weight.

Figure 5 shows the sway of the simulated Nao, with and with out the controller, when met with an impulse force

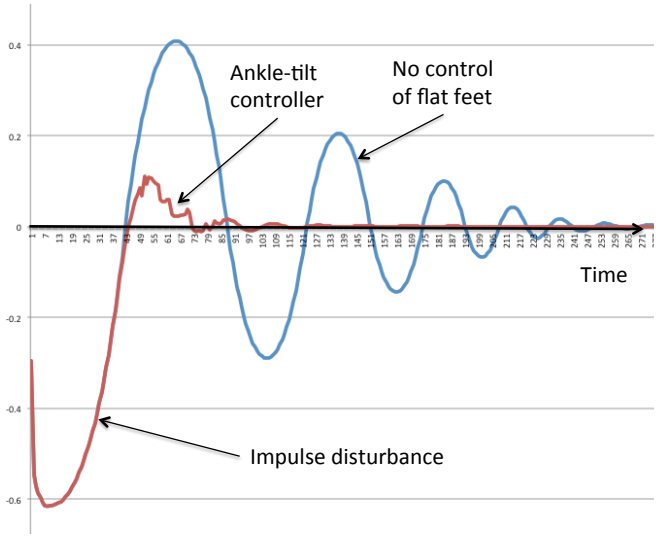


Fig. 5. Before and after effect of the ankle-tilt policy controller following an impulse force in simulation.

disturbance while stationary. The controller shows some jitter from the discretised actions used by the reinforcement learner.

IV. Integrated Nao Bipedal Locomotion

A more elaborate description of the integrated bipedal locomotion for the Nao and code walk-through has been documented in a report [19] as a part of UNSW 2014 RoboCup SPL code release. We next describe some of the salient features.

A. Coronal Plane Control

The coronal rock is stabilised by synchronising the onset of the leg-lift motion with the switch in swing and support feet. We switch the swing and support feet by observing the zero-crossing of the measured CoP in the sideways y-direction using the Noa's foot pressure sensors. The CoP is calculated in the coronal plane with the origin in the middle of the robot between the feet. It is negative when the robot stands on the right foot and positive when it switches to the left foot. The period that the robot spends on the support foot cannot be determined precisely when there are disturbances such as uneven surfaces, play in motor gears, dirt on the ground, and bumping by other robots. The zero-crossing point of the CoP indicates that the robot is in the process of shifting its weight to the other leg. We use it to reset the starting time for both the left and right swing phases of the walk-cycle.

The controller running on the real Nao produces the time-series for the CoP and leg-lift as shown in Figure 6. The real Nao was tested on a felt carpet which may explain the ragged edges on the CoP measurement over the 8 foot sensors.

While the gyro controller was not used for controlling the sideways rock of the robot, this type of controller was used for kicking, to balance the Nao on one foot in the coronal plane.

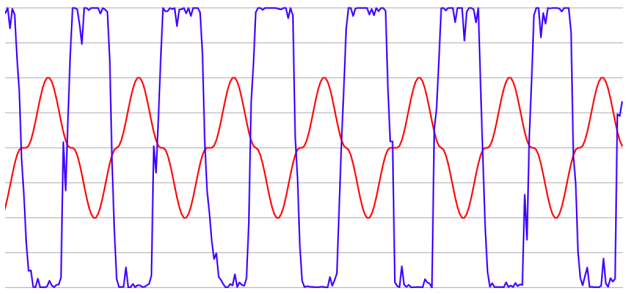


Fig. 6. Real Nao closed-loop coronal rock using the CoP zero-crossing point. CoP is in blue, right/left leg-lift is in red with +ve and -ve values used just to show the values for the different feet.

B. Footstep Response to Change in Walk Parameters

Omni-directional locomotion is achieved by providing the walk-engine with concurrent forward, sideways and turn parameters at about 30 Hz, the maximum frame-rate of the camera. The change in walk parameter settings takes effect immediately at the start of the next walk phase. There is only a delay of between 0 and 0.23 seconds from the time the command is given by behaviour.

Foot positioning is optimised when changing direction, for example, when switching from walking left to walking right. This means that the walk does not necessarily transition through a state where the feet are both together, but may rock with the feet apart when changing direction. We next illustrate the change in walk variables in response to a change in walk commands one at a time. In combination they operate concurrently, in the same manner as they would independently, to achieve omni-directional locomotion.

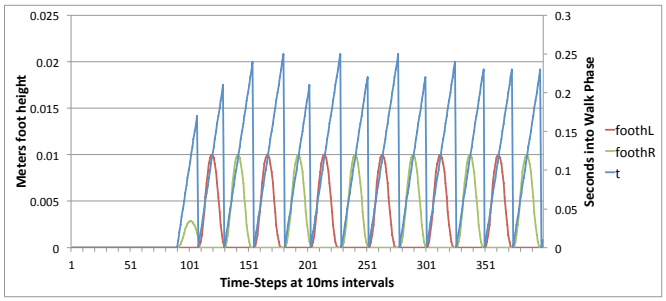


Fig. 7. Basic walking pattern showing the phase timing (blue) regulated by the leg-lift period. The red and green plots show the leg-lift for the left and right foot respectively.

Figure 7 shows the initiation of a walking pattern from a standing start with the robot just marking time. The blue sawtooth graph shows the time t elapsed after each phase change from when the walk is initiated. It resets at the beginning of each phase of the walk when the centre-of-pressure changes sign indicating a change in support foot. Each phase is of a slightly different duration due to noise in the rocking behaviour. The red and green parabolic-like plots show the lift in the left and right foot respectively. When the walk is initiated the foot is only lifted to about 30% of it walking height with the effect that it initiates the sideways rocking motion. This also makes the initial phase shorter than the following ones.

The robot does not shift the CoM sideways using a hip motion. This has the advantage that we do not need to resynchronise a hip sway with the change in support foot, and it provides greater acceleration form side to side allowing the period of the walk to be adjusted.

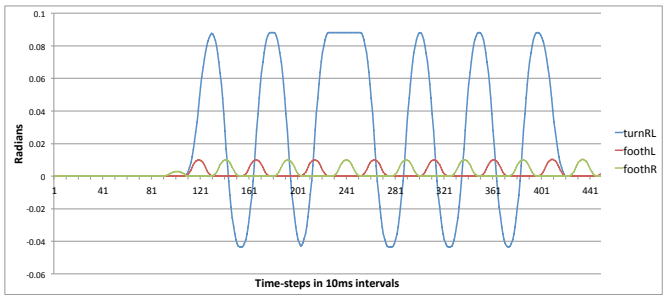


Fig. 8. Turning counterclockwise and then clockwise.

Figure 8 illustrates behaviour commanding the walk to turn counterclockwise and then reverse direction at the same turn speed for several steps. The blue graph shows the value of the hip-yaw joint in radians, in relation to the foot-steps in green and red from the previous graph for reference. The turn is not activated until after the sideways rock is initiated. The outward turn of the feet is greater than the inward turn of the feet. As can be seen in the middle of the graph, when the direction of turn is reversed the hip-yaw joint is not moved to zero first. The feet are kept apart while the walk changes support foot. The turning motions start with both feet together and they are reset to this position at the conclusion of the turn.

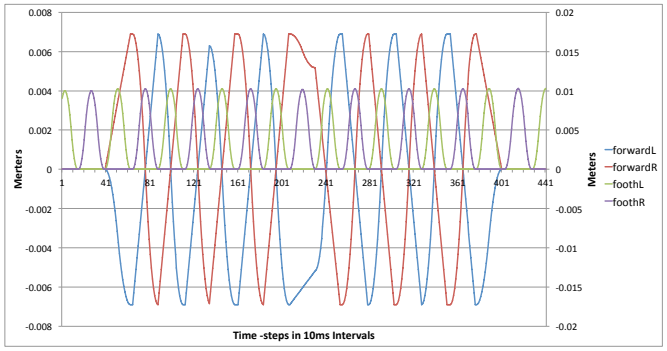


Fig. 9. Walk forward and then reverse.

Figure 9 shows a motion that first starts to walk forward and then reverses direction and walks backwards for a few steps. The walk variables indicating the position of the left and right foot are shown in blue and red respectively, with the left and right leg-lift shown in green and purple for reference. When reversing direction the walk makes a slight adjustment due to ratcheting the step-size but does not return the feet to a zero position. It is also evident from the graph that the swing foot moves in a parabolic fashion while the support foot moves

at a constant velocity with respect to the body of the robot. Starting and stopping the forward/backward motion is smooth and achieved within a single phase of the walk.

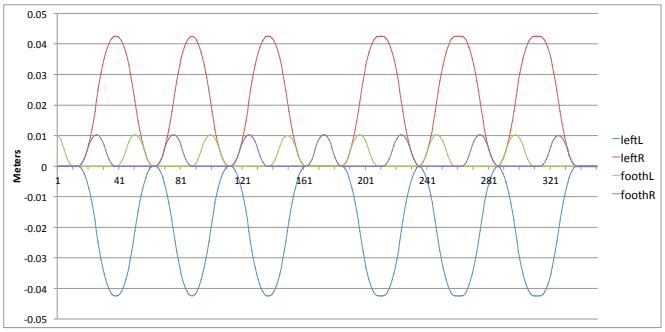


Fig. 10. Walk left and then right.

Figure 10 shows a sideways walk to the left followed by the sideways walk to the right. The red and blue time-series show left and right sideways displacements of the feet. In this case the timing of the reversal in direction is such that the optimum way to reverse direction is bring both feet together for one phase. If on the other foot at the time of reversal, the walk would pause with legs apart before resuming the walk in the other direction.

V. Conclusion

This paper has described reinforcement learning experiments and their subsequent interpretation and implementation on the real Nao robot to mitigate disturbance rejection in the sagittal plane and coronal plane when kicking. Post-hoc algebraic analysis has shown that the controller could be improved by fitting a hyperbolic ankle-tilt function derived from Equation 3, instead of the cruder linear approximation. Omni-directional locomotion is achieved by adding feedforward control, implicit in maximising future reward, to an open loop generated bipedal gait. Concurrent changes in forward, sideways and turn walk parameters are implemented by coordinating the left and right footsteps to optimise the speed of the change. The bipedal locomotion described in this paper was used by the University of New South Wales, Australia team in the RoboCup SPL competition in 2014 and 2015. This workshop paper is accompanied by a video showing the response of the simulator to sagittal disturbances, and the controller in action on the real Nao robot during competition.

Acknowledgment

The author would like to thank the UNSW RoboCup SPL team members who provided feedback on the effectives of the bipedal locomotion described in this paper during its development and implementation on the Nao.

References

[1] C. Graf and T. Röfer, "A closed-loop 3d-lipm gait for the robocup standard platform league humanoid," in *Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots*, C. Zhou, E. Pagello, S. Behnke, E. Menegatti, T. Röfer, and P. Stone, Eds., 2010.

[2] J. Needham, *Science and Civilisation in China: Volume 2, History of Scientific Thought*, ser. Science and Civilisation in China. Cambridge University Press, 1956. [Online]. Available: http://books.google.com.au/books?id=yaOe-jblVEYC

[3] T. McGeer, "Passive dynamic walking," *I. J. Robotic Res.*, vol. 9, no. 2, pp. 62–82, 1990.

[4] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005. [Online]. Available: http://www.sciencemag.org/content/307/5712/1082.abstract

[5] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Boca Raton: CRC Press, 2007, vol. 1.

[6] A. Ames and R. Gregg, "Stably extending two-dimensional bipedal walking to three dimensions," *American Control Conference*, pp. 2848–2854, 2007.

[7] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet, "3d bipedal robotic walking: Models feedback control, and open problems," *8th IFAC Symposium on Nonlinear Control Systems*, 2010.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, and K. H. K. Yokoi, "Biped walking pattern generation by using preview control of zero-moment point," in *in Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 1620–1626.

[9] 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," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pp. 4489–4496, 2010.

[10] H. Benbrahim and J. A. Franklin, "Biped dynamic walking using reinforcement learning," *Robotics and Autonomous Systems*, vol. 22, pp. 283–302, 1997.

[11] R. Tedrake, "Stochastic policy gradient reinforcement learning on a simple 3d biped," in *Proc. of the 10th Int. Conf. on Intelligent Robots and Systems*, 2004, pp. 2849–2854.

[12] J. Morimoto, G. Cheng, C. Atkeson, and G. Zeglin, "A simple reinforcement learning algorithm for biped walking," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 3, april-1 may 2004, pp. 3030 – 3035 Vol.3.

[13] S. Wang and J. Braaksma, "Reinforcement learning control for biped robot walking on uneven surfaces," in *Proceedings of the 2006 International Joint Conference on Neural Networks*, 2006, pp. 4173–4178.

[14] J. Morimoto, J. Nakanishi, G. Endo, G. Cheng, C. G. Atkeson, and G. Zeglin, "Poincaré-map-based reinforcement learning for biped walking." in *ICRA'05*, 2005, pp. 2381–2386.

[15] T. Mori, Y. Nakamura, M.-A. Sato, and S. Ishii, "Reinforcement learning for a cpg-driven biped robot," in *Proceedings of the 19th national conference on Artifical intelligence*, ser. AAAI'04. AAAI Press, 2004, pp. 623–630.

[16] C.-M. Chew and G. A. Pratt, "Dynamic bipedal walking assisted by learning," *Robotica*, vol. 20, pp. 477–491, September 2002.

[17] F. Faber and S. Behnke, "Stochastic optimization of bipedal walking using gyro feedback and phase resetting," in *2007 7th IEEE-RAS International Conference on Humanoid Robots, November 29th - December 1st, Pittsburgh, PA, USA*, 2007, pp. 203–209. [Online]. Available: http://dx.doi.org/10.1109/ICHR.2007.4813869

[18] B. Hengst, "On-line model-based continuous state reinforcement learning using background knowledge," *Twenty-Fifth Australasian Joint Conference on Artificial Intelligence (AI12)*, 2012.

[19] ——, "rUNSWift Walk2014 report," 2014, http://cgi.cse.unsw.edu.au/"tilde"robocup/ 2014ChampionTeamPaperReports/ 20140930-Bernhard.Hengst-Walk2014Report.pdf.