Perceiving Forces, Bumps, and Touches from Proprioceptive Expectations

Christopher Stanton[1], Edward Ratanasena[1], Sajjad Haider[2], and Mary-Anne Williams[1]

[1] Innovation and Enterprise Research Laboratory, University of Technology, Sydney
cstanton@it.uts.edu.au, edward.ratanasena@student.uts.edu.au, mary-anne.williams@uts.edu.au

[2] AI Lab, Institute of Business Administration, Karachi
sajjad.haider@khi.iba.edu.pk

Abstract. We present a method for enabling an Aldebaran Nao humanoid robot to perceive bumps and touches caused by physical contact forces. Dedicated touch, tactile or force sensors are not used. Instead, our approach involves the robot learning from experience to generate a proprioceptive motor sensory expectation from recent motor position commands. Training involves collecting data from the robot characterised by the absence of the impacts we wish to detect, to establish an expectation of "normal" motor sensory experience. After learning, the perception of any unexpected force is achieved by the comparison of predicted motor sensor values with sensed motor values for each DOF on the robot. We demonstrate our approach allows the robot to reliably detect small (and also large) impacts upon the robot (at each individual joint servo motor) with high, but also varying, degrees of sensitivity for different parts of the body. We discuss current and possible applications for robots that can develop and exploit proprioceptive expectations during physical interaction with the world.

Keywords: motor learning, Nao, robot soccer, anticipation, collision detection.

1 Introduction

Most animals have a rich sense of touch to provide feedback during interaction with the world. By fusing touch, proprioception, and other sensations they are able to perceive collisions and forces upon their body. Robots however often have hard protective shells, and lack equivalent tactile sensors. For robots to interact meaningfully with the environment, they will need to be capable of detecting expected and unexpected collisions with other objects, people, and themselves. Our aim is to model perceptions related to contact forces on robots that do not possess a dense array of dedicated touch, tactile or force sensors. Examples of such robots include the Sony AIBO or the Aldebaran Nao. We achieve this

T. Röfer et al. (Eds.): RoboCup 2011, LNCS 7416, pp. 377–388, 2012.

by identifying discrepancies in sensed angular motor position values caused by impact forces between the robot's limbs and objects within its environment.

This paper is structured as follows: Section 2 describes the application problem domain - robot soccer, and in particular the Standard Platform League (SPL). Section 3 describes other approaches for detecting physical contact with similar robots (i.e. robots without dedicated touch/tactile sensors). Next, we describe our approach based on proprioceptive motor sensory expectations, and how it differs from other relevant techniques, and the advantages it offers. Our implementation details are described in Section 5. Section 6 describes the results of experiments used to test our approach, and we conclude by discussing the current and future applications of the approach.

2 Problem Domain: The Standard Platform League

Robot soccer, as per human soccer, is a contact sport. It is dynamic, with frequent collisions and subsequent falls (perhaps more so in robot soccer, given the current state of the art). Such collisions can occur between multiple players, between players and referees, between players and the ground, players and the ball, players and the goal posts, and so forth. Players limbs also make (often accidental) impact with other parts of their own body (a self-collision).

In the RoboCup Standard Platform League (SPL), all teams participate with "identical"[1] robots. In the past the league used the Sony Aibo four-legged robot, and today the SPL uses the Aldbaran Nao humanoid robot. In SPL soccer matches, unwanted and undesirable collisions with other robots and obstacles is a common occurrence. With Aibos it could result in "leg-locking" [7], and with Naos it often results in a fall. In both cases, major damage to the robots can occur.

The Aldbaran Nao robots are equipped with sensors such as head mounted colour cameras, front-facing ultrasonic distance sensors, force sensors in the soles of the feet, bump detectors on the toes of each feet, accelerometers, and gyroscope. While the "pushing"[2] rules in the SPL are designed to encourage robots to avoid colliding with each other, collisions still occur with regular frequency. This is due to a number of factors. Both vision-based and sonar-based object avoidance is hampered by the small field of view of both these devices, with many robot-robot collisions occurring when one robot hits another from behind or from the side. Other techniques for detecting collisions involve the use of accelerometers and gyroscopes, but this involves detecting a collision after it happens (often at a point when the robot has already become unstable). Instead, it would be more useful to anticipate collisions, and to react to unexpected collisions almost instantaneously. Thus, a better solution is required.

[1] This is the ideal - in practice, teams use similar robots provided by the same manufacturer.

[2] A robot deemed by the referee to be guilty of pushing another robot is removed from the field of play for a period of time.

3 Background and Related Work

Detecting collisions is important for most autonomous robots. For robots to work safely alongside and with humans, they must be capable of detecting physical contact and/or responding appropriately. Many approaches to achieving safe human robot interaction (HRI) focus on building compliant robots through the use of variable stiffness actuators, which can absorb collision energies through compliant mechanisms, while achieving precise joint positioning through variation of stiffness gains. Compliance can be implemented in different ways - mechanically, e.g. through the use of internal springs as a potential energy buffer [9]), or through the use of intelligent software controllers (e.g. that dynamically adjust their commands in response to joint torque sensors [3]). Collision prevention strategies can also be imposed on the controller, e.g. by imposing forbidden workspace area constraints [1].

In the RoboCup domain, approaches to detecting collisions have focused on processing data from accelerometers or joint position sensors. Approaches utilising the accelerometer involve discriminating between patterns of data produced under different gaits and/or environmental conditions. For example, Vail and Veloso [8] demonstrate how the accelerometer sensor of a Sony Aibo can be used to identify different surfaces upon which the robot is walking through a decision tree. In soccer matches, Mericli et al. [6] demonstrate how a Sony Aibo's accelerometer readings can be analysed statistically to probabilisticly discriminate between "normal" walking, and walking in which a collision has occurred.

Another strategy for detecting collisions is to process the joint position data for each degree of freedom on the robot, with the aim of distinguishing normal, collision-free sensor readings, and sensor readings produced when the robot is experiencing a collision. Quinlan et al. [7] describe a fault detection system operating on the Sony Aibo, in which slip and collisions are detected by measuring the normal variation in the robot's motors during normal locomotion tasks, and comparing this with data collected in which collisions and slips occur. After training, a collision or slip is detected if multiple consecutive readings of a sensed motor position value are outside a range of two standard deviations from an expected value. Hoffman and Gohring [5] describe a collision detection process (also implemented on the Sony Aibo) in which they compare command data with sensor data. The difference between command values and sensor values are measured over a period of time (96ms, 12 frames). If the robot's range of motion is impaired because of a collision with another object, the difference between command value and sensor value increases. If the error is above a certain threshold for a particular movement, a collision is detected.

Each of these approaches have their particular strengths and weaknesses. Accelerometers allow the robot to detect collisions independent of the point of contact (useful on a robot without tactile sensors), but require disturbing the robot's body (e.g. if an obstacle hits the robots chest, the accelerometer would detect this, even though there is no motor in the chest). However, using the accelerometer to detect a collision requires the collision to be of sufficient force that it disrupts the stability of the robot (which is a problem on a humanoid robot, such as the Nao as some collisions are detected too late for the robot avoid a fall). Conversely, accelerometers might fail to detect collisions in which a motor's range of freedom is impaired, but the particular motor impedance is not sufficient to effect the stability of the robot. Approaches which rely on detecting collisions through impedances of motors rely on a collision restricting the range of motion of one of the robot's limbs. While this approach may be capable of detecting some gentle collisions, it will fail to detect any collision that does not impede the robot's range of motion. All approaches however, regardless of their input sensory device, rely on discriminating between sensor input during collision-free movement and collision-impaired movement.

4 Our Approach

Our goal was to use motor position sensors to provide the Nao with a sense of "touch" that provided more sensitivity and scope than existing methods. We aimed to build a system to perceive not only the strong, forceful bumps and collisions that occur during robot soccer, but the gentler interactions that might occur when an autonomous robot is interacting with objects and people in its environment - for example, a human gently pushing or impeding a robot's arm or head.

While the previous approaches of [7] and [5] had demonstrated detecting collisions between robots using motor position sensors is possible, these approaches focused mainly on detecting the (sometimes brutal) types of collisions that occur only in robot soccer games[3]. With both approaches, calibration of detection thresholds is done for each category of motion - thus requiring each new motion be calibrated. Also, their detection triggers rely on finding differences that exceed the maximum found in all previous training data for each particular category. Thus, sensitivity and responsiveness can only be improved by creating new categories of motion, and hand-tuning threshold parameters for each particular motion.

4.1 Sensor Command Difference (SCD)

Intuition suggests that previous commands issued to the motor will effect the current sensor reading, and that the difference between the most recent command value and most recent sensor value will not be constant. The motor, being a mechanical device, is subject to physical forces of varying degrees caused by the effects of friction , inertia, momentum, and gravity, as well as the effects of these forces upon the limbs to which it is attached. Impeding forces will likely create a bigger difference between sensor and command had they not occurred, while other forces may push the motor towards a target position, reducing sensor-command difference.

[3] Even in robot soccer, it would be advantageous to detect smaller forces earlier, as a soft touch can act as an early warning system to give the robot time to avoid a more serious impact.

We term the difference between the last command value and the current sensor value the "sensor command difference" (SCD). We examined how the SCD varied for each of the Nao's 21 degrees of freedom throughout the course of a robot soccer game. Data was collected for each motor and statistically analysed. As can be seen in Figure 1, predictable periodic patterns can be observed in the SCDs, which probably correspond to foot-ground impacts. Similar patterns were seen for all motors.

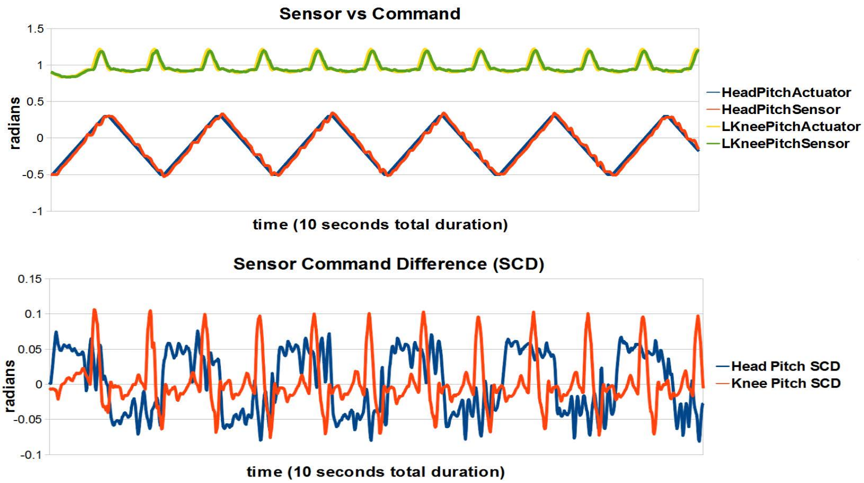

Fig. 1. Top: the sensor and command values for the head pitch and left knee pitch motors are displayed. Below, the corresponding calculated SCDs for head pitch and left knee pitch are displayed. The data was collected while the robot was chasing a soccer ball. Our aim is to predict the SCD every 10ms, and to use discrepancies between SCD estimates and the sensed SCD to infer the experience of an unexpected force/impact.

To further investigate how the SCD values of the different motors of the Nao robot were effected by walking, we instructed the robot to walk forwards (with no rotation and no strafe)[4]. There was no discernible differences between left and right motors. However, as can be seen in Table 1 the magnitude of SCD varied significantly for different parts of the body during the walk. In the legs, the greatest variances and magnitudes of SCD were found in the knees and hip pitch motors, while in the upper body both the shoulder pitch and head pitch experienced large SCD values. The fact that pitch motors (as opposed to yaw and roll) were most effected by walking suggests that impacts between the ground, and also the weight of the robot's body have large effects on SCD.

We speculated that other factors besides walking may also affect SCD values, such as the velocity and acceleration with which a limb is currently moving

[4] Using the Aldebaran walk engine, at full forwards velocity, and with motor stiffness set to 80 percent.

Table 1. Variance of SCD for the robot's motors while walking forwards (ordered by greatest variance, top to bottom)

Motor	Max SCD (radians)	Std Dev SCD (radians)
Knee Pitch	0.140	0.040
Shoulder Pitch	0.080	0.050
Hip Pitch	0.060	0.040
Ankle Pitch	0.050	0.020
Head Pitch	0.050	0.020
Hip Roll	0.040	0.020
Hip Yaw	0.020	0.010
Head Yaw	0.020	0.010
Shoulder Roll	0.010	0.005

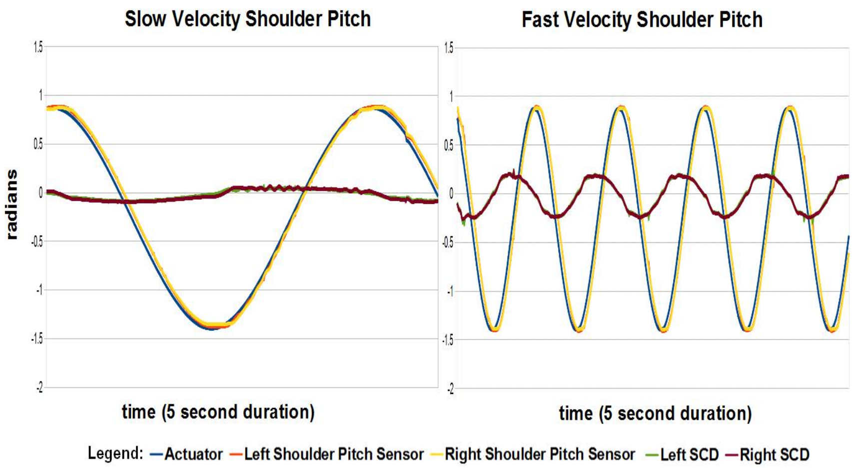

Fig. 2. Slow velocity (left) versus fast velocity (right). The SCD of the high velocity motion is greater in magnitude than the slow velocity motion.

when it is requested to move to a new position. To test the effects of velocity and acceleration, we instructed the robot to move its shoulder pitch and roll motors at varying velocities between two points near the extremities of its range of motion. As can be seen in Figure 2, requesting the motors to travel at higher velocities increases SCD.

5 Implementation

Our aim is to enable the robot to accurately estimate the SCD every 10ms[5]. We then test this estimate as a means of perceiving unexpected forces.

[5] On each DCM callback event.

5.1 Design

We train one neural network per degree of freedom to estimate the SCD for that motor every 10ms.

5.2 Data Collection and Processing

We programmed our robots to log all motor position commands and position sensor values every 10ms while performing their normal autonomous duties (e.g. soccer). This data was then processed to calculate instantaneous velocity and acceleration, and also the SCD. Training data is characterised by the absence of the forces we want the robot to perceive. Care is taken that the robot does not receive bumps or pushes from other robots or people. If the robot experiences such events, the training data is discarded. Unavoidable events that may effect SCD, such as foot-ground contact or shifts in the robot's centre of mass are included.

5.3 Learning

Each neural network is trained to approximate a function which predicts the SCD based on the command (c), velocity (v), and acceleration (a), i.e $SCD = f(c,v,a)$. The function is represented by a matrix of values which represents the weights of each node on each other. These weights are optimised to approximate the function (training) as closely as possible using particle swarm optimisation. Once the neural network has been sufficiently trained, the weights are then loaded onto the robots. Using the pre-trained weights, the robot can accurately predict the SCD by calculating its joint command, velocity, and acceleration and feeding it into a neural network with the same configurations as the neural network trained off-line.

5.4 Detection Triggers

We deem an abnormal motor event to have occurred if the predicted SCD differs significantly from the sensed SCD. This comparison is made every 10ms by the robot for every motor of the robot. Since the neural network only approximates the function and considering stochastic errors, our system looks for runs of consecutive discrepancies between prediction and measured SCD to infer the motor is experiencing an unusual force[6,7].

[6] To date, we have found 5 to be the best figure.

[7] Initially we modelled the velocity and acceleration over 50ms to reduce the effects of noise. However our neural network is then slow to respond to sudden changes of command.

6 Experiments and Results

Each SCD prediction function learnt by the corresponding neural network was evaluated visually, as per Figures 3 and 4. Neural nets were also evaluated by the maximum discrepancy between their estimate of SCD and the actual SCD.

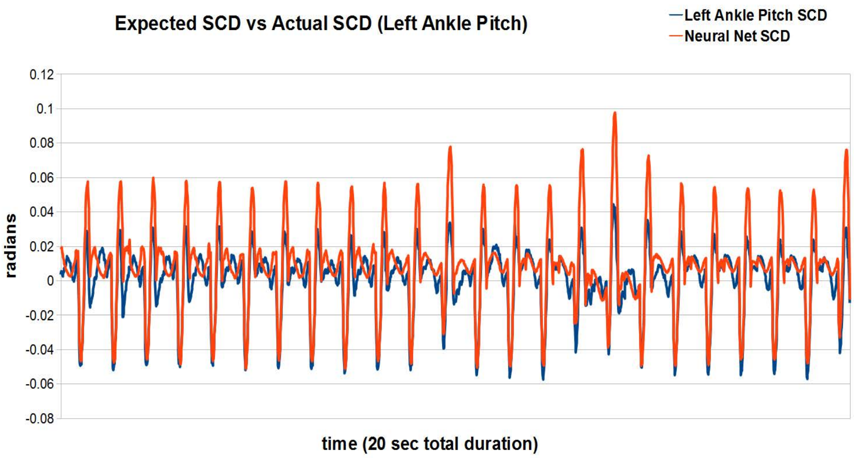

Fig. 3. Left Ankle Pitch: neural net estimate of SCD (displayed in red) versus the actual SCD (blue) experienced by the robot for the left ankle pitch motor for a 20 second duration from a robot playing soccer

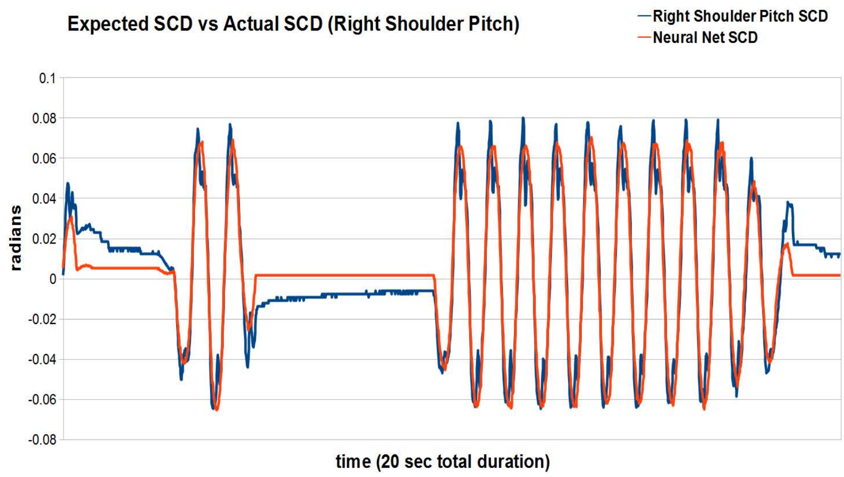

Fig. 4. Right Shoulder Pitch: the neural net estimate of SCD (displayed in red) versus the actual SCD (blue) experienced by the robot for the right shoulder pitch motor for a 20 second duration from a robot playing soccer

We calibrated detection thresholds during a period of experimentation in which we would use our hands to gently grab, push and impede a soccer playing robot. The soccer playing robot used the Aldebaran Walk Engine, and also performed two different kicking motions (both which require the robot to balance on one leg). The human would make contact with the robot at different points of the robot's body. The robot would speak the name of any motors for which it perceived 5 consecutive discrepancies between estimated SCD and the sensed SCD. If more than one motor perceived the impact, the name of the motor with the greatest discrepancy between estimated SCD and sensed SCD would be uttered by the robot.

Our system was calibrated so that it could be used in RoboCup soccer matches - as such false positives can be more damaging (strategically) than false negatives, as it is preferable to chase the ball than avoid a non-existent obstacle. Table 2 displays thresholds which eliminated nearly all false-positives. Unfortunately, we have so far been unable to empirically quantify the magnitude of the force required to trigger a discrepancy. One difficulty we faced in evaluating false-negative error rates is applying forces with a known magnitude. A false-negative can be easily induced with a weak touch, and conversely always avoided with strong touch or bump. The system is very sensitive to touches of the arms and head - only light touches are required (they are gentle enough that that there is little or no disturbance to the robot's movement. The system is less sensitive when it comes receiving bumps to the legs, as these forces need to be of sufficient strength that they disrupt the motion of the leg, and thus the stability of the robot. This may be due to the nature of the motors in these parts of the robots body[8]- it may also be the upper body motors are exposed to less SCD noise caused by walking than the motors in the legs. Also with regards to sensitivity, the further a limb was pressed away from the motor controlling that limb, the less force was required to generate an impact detection, most likely as a result of the leverage provided by the limb against the motor. We attempted to detect the presence or absence of the ball at the point of impact when kicking by instructing the robot to practice kicks without the ball, but the SCD discrepancy caused by ball-to-foot contact with our current kicking action and stiffness settings is so small that this appears impossible[9]. Table 3 describes various types impacts that were used to test our approach, and the motors that would identify the impeding forces.

We are currently using the system in our robot soccer team to detect unexpected impacts. The system is always operating, regardless of higher-level behaviour changes (e.g. a new walking gait). So far the system has proved quite robust to changes in behaviours and walking surfaces. We speculate this is because of the large effect velocity and acceleration have on SCD over a small time-scale (we are calculating velocity and acceleration for the last 20ms), and

[8] The motors in the legs appear stronger than the motors in the upper body.

[9] Perhaps with changes to our kicking motion and motor stiffness settings it may be possible to detect foot-ball contact.

Table 2. Threshold values are displayed (left), which were calibrated to remove almost all false positives

Motor	Threshold(radians)
Head Pitch	0.05
Head Yaw	0.05
Shoulder Pitch	0.05
Shoulder Roll	0.03
Hip Yaw	0.04
Hip Pitch	0.05
Knee Pitch	0.04
Ankle Pitch	0.05
Ankle Roll	0.03

Table 3. Impact types and the motors that typically identify them

Description of Forces	Motors typically identified by the robot
Impedance of the head	Head Pitch (vertical impedance), Head Yaw (lateral impedance)
Pushing down on the shoulders of the robot	Hip Pitch, Ankle Pitch, Ankle Roll
Grabbing robot's hand	Shoulder Pitch, Shoulder Roll, Elbow Pitch
Grabbing robot's upper arm	Shoulder Pitch, Shoulder Roll, but requires stronger force than if holding the robot's hand
Robot's foot is impeded while walking (e.g. by a heavy book)	Ankle Pitch, Ankle Roll, Hip Pitch, Hip Roll, Head Pitch, Head Yaw
Robot is unsteady (in danger of falling) but still trying to walk	Ankle Pitch, Ankle Roll, Hip Pitch, Hip Roll, Head Pitch, Head Yaw

that these relationships occur in all types of motions. However, further investigation is required.

Bumps and touches are detected regardless of whether the robot is stationary or moving. Changes to robot configuration are often detected. For example, a robot trained without a shoulder pad would then detect (via the head yaw motor) contact between the head and the shoulder pad when the shoulder pad was replaced. A sensor failure has also been detected via this system[10]. With regards to the collisions that occur in the robot soccer domain, our current results suggest this approach will be very useful for detecting arm-to-arm contact on the Nao robots, and many other collisions that occur in robot soccer matches. When the robot becomes severely unstable, this can be detected by a variety of motors, including head pitch and head yaw. Accelerometer-based approaches [6] would also make an excellent complementary approach.

[10] The robot repeatedly uttered "left ankle pitch" - closer investigation revealed the sensor value was a constant value, regardless of the position of the foot.

7 Discussion and Conclusion

We have demonstrated an effective approach to collision detection that relies on "detecting the unexpected". Sensor and command data is collected from a robot in which undesirable impacts do not occur. Machine learning is used to generate a proprioceptive expectation - an estimate of each motor's sensor position, based upon previous commands. Physical contact is inferred when a sensed motor position value differs significantly from a predicted value. Our approach allows the robot to perceive when its limbs physically contact other objects, despite the robot not having any dedicated tactile or force sensors at the point of impact.

Our results suggest interpreting SCDs can provide an estimate of force upon the motors of the robot. Violent actions produce large SCDS; smooth, slow movements small SCDs. We are yet to examine whether this approach can be used to perceive the direction of the force (e.g. is it impeding or pushing?), or whether it can provide a measure of the magnitude of a force.

Future work will extend our approach by fusing motor position values with the sensed electric current to each motor. We also aim to develop appropriate behavioural responses to unexpectedly large SCD stimuli. Currently, our robots only produce a verbal response to unexpectedly large SCD stimuli, and do not adjust or re-plan their motor instructions. As the Nao allows control of the stiffness of the robot's motors (via electric current), a simple approach would be to dynamically adjust motor stiffness in response to, or the anticipation of, large SCDs.

Other extensions to investigate include using neural networks that generate expectations for different types of sensors, such as the robot's accelerometers and the electric current sensors in each of the robot's motors. Also, if accelerometer information is provided to our robots using our current approach, this may assist them in predicting large SCDs, while also detecting collisions in which no range of motion is impeded.

Lastly, in future work we aim to use SCD measures for improving fine motor control. For example, if a walk engine dynamically reduces stiffness in anticipation of large SCDs, this may produce a smoother walk. While unsupervised machine learning has been applied to skills such as walking [2] and kicking [4] in the SPL, feedback for such behaviours is provided through the robot's visual system (e.g. a measure of ball distance in the case of a kick, or recognising land marks in the case of determining the speed at which a robot has walked). We are not aware of any motor learning research in this domain in which proprioceptive expectations are used to provide feedback to improve motor control.

Acknowledgements. We would like to acknowledge Nathan Kirchner and Benjamin Johnston for their useful suggestions.

References

1. Calinon, S., Sardellitti, I., Caldwell, D.: Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In: Intelligent Robots and Systems (IROS), pp. 249–254. IEEE (2010)

2. Chalup, S., Murch, C., Quinlan, M.J.: Machine learning with AIBO robots in the Four-Legged League of RoboCup. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37, 297–310 (2007)
3. Chen, W., Sun, Y., Huang, Y.: A collision detection system for an assistive robotic manipulator. Life System Modeling and Intelligent Computing 97, 117–123 (2010)
4. Hausknecht, M., Stone, P.: Learning Powerful Kicks on the Aibo ERS-7: The Quest for a Striker. In: Ruiz-del-Solar, J., Chown, E., Ploeger, P.G. (eds.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 254–265. Springer, Heidelberg (2010)
5. Hoffmann, J., Göhring, D.: Sensor-Actuator-Comparison as a Basis for Collision Detection for a Quadruped Robot. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 150–159. Springer, Heidelberg (2005)
6. Meriçli, T., Meriçli, Ç., Akın, H.L.: A Robust Statistical Collision Detection Framework for Quadruped Robots. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 145–156. Springer, Heidelberg (2009)
7. Quinlan, M.J., Murch, C.L., Middleton, R.H., Chalup, S.K.: Traction Monitoring for Collision Detection with Legged Robots. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 374–384. Springer, Heidelberg (2004)
8. Vail, D., Veloso, M.: Learning from accelerometer data on a legged robot. In: Proc. IAV 2004. Springer, Lisbon (2004)
9. Visser, L., Carloni, R., Stramigioli, S.: Energy efficient control of robots with variable stiffness actuators. In: Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems, NOLCOS 2010, Bologna, Italy, pp. 1199–1204. IFAC (2010)