Traction Monitoring for Collision Detection with Legged Robots — Source link ↗

Michael Quinlan, Craig L. Murch, Richard H. Middleton, Stephan K. Chalup

Institutions: University of Newcastle

Published on: 05 Jul 2003 - Robot Soccer World Cup

Topics: Legged robot, Robot, AIBO and Collision detection

Related papers:

- Traction monitoring for collision detection with legged robots

- A real-time low-computation cost human-following framework in outdoor environment for legged robots

- Real-time obstacle detection for legged robots using the Kinect sensor

- Vision-servoed localization and behavior-based planning for an autonomous quadruped legged robot

- Practical Vision-Based Monte Carlo Localization on a Legged Robot

Traction Monitoring for Collision Detection with Legged Robots

Michael J. Quinlan, Craig L. Murch, Richard H. Middleton, and
Stephan K. Chalup

School of Electrical Engineering & Computer Science
The University of Newcastle, Callaghan 2308, Australia
{mquinlan,cmurch,rick,chalup}@eecs.newcastle.edu.au
WWW home page: http://robots.newcastle.edu.au

Abstract. With the introduction of commercially available programmable
legged robots, a generic software method for detection of abnormalities
in the robots' locomotion is required. Our approach is to gain satisfac-
tory results using a bare minimum amount of hardware feedback; In
most cases we are able to detect faults using only the joint angle sen-
sors. Methods for recognising several types of collision as well as a loss
of traction are examined. We are particularly interested in applying such
techniques to Sony AIBO robots in the RoboCup legged league environ-
ment. This investigation provided us with a technique that enabled us
to detect collisions with reliable accuracy using limited training time.

1 Introduction

An important goal of research in adaptive robotics is to develop robots which
can navigate efficiently and robustly in different environments. This includes
driving or walking on a variety of surfaces such as for example, sand, ice, grass
or carpet, and dealing with obstacles such as stairs or rocks. These skills would
be required by robots which explore changing or unknown environments, for
example, disaster areas [12] or the surface of Mars [6]. The same methods would
also help to improve some transport machines which currently have only a very
restricted ability to move such as wheelchairs [4].

Wheeled, tracked and legged robots seem to have their advantages and dis-
advantages in different environments. On even and hard surfaces wheeled robots
are usually faster, while legged machines would typically be better in dealing
with stairs or similar obstacles. However, some six-legged robots can also walk
very fast [1,5] and some tracked tank-like machines [7] have been developed
which can climb stairs and jump over rocks with remarkable speed.

It is not yet completely clear what would be the practical advantages of
4–legged or biped robots over 6– or 8–legged robots and what could be useful
applications other than entertainment. Nonetheless, major effort is currently put
into the development of 4–legged and bipedal robots with the aim to achieve some
similarity to dogs, cats, and humans [2,3].

Traction monitoring is a fundamental task in traction control and achieving collision detection for satisfactory robot movement in particular with legged or wheeled robots. If the wheels spin in sand or the legs slip on ice the machine would not be able to move forward and may even fall over. Similarly, if the robot crashes into a wall or into other robots or if its legs get entangled in some smaller obstacles, its movement could be impaired.

In the present study we investigate methods to monitor traction measures and employ them for collision detection with 4–legged AIBO robots [2] in the environment of the legged league of RoboCup [11]. The idea is that detection of an abnormal situation should be used to alter the robot's behaviour—for example, if a collision while walking forwards is detected, the robot should either walk backwards, strafe or turn to avoid the obstacle. Particular goals are to increase the speed of the robots and to find a good strategy to deal with situations where the legs of two robots get entangled (*leg-lock*), see figure 1.

Fig. 1. Leg-locked Robots

In section 2, we give a brief overview of the Sony Legged League robot soccer environment followed by a discussion on gait control in section 3. In section 4, we describe a direct statistical approach to fault detection. Section 5 illustrates the different types of collisions which may occur and techniques to detect them, while section 6 deals with the detection of a loss of traction (*slip*). The discussion in

section 7 addresses possibilities for integrating our results into behaviour control as well as future developments. The paper concludes with a summary in section 8.

2   The Legged League Robot Soccer Environment

The hardware used for this project is the Sony AIBO Entertainment Robot, model ERS-210(A). The robots are programmed in a C++ software environment. They have an internal 64-bit processor and PC Card slot allowing communication via Wireless LAN.

The legs each have three degrees of freedom to enable walking and kicking. The joints in the legs are termed the rotator, abductor and knee. The *rotator* is the shoulder joint responsible for leg movement parallel with the body (along the length axis), while the *abductor* is the shoulder joint involved in leg movement perpendicular to the body. The *knee* functions in principle like a dog's knee in nature, allowing 174° of movement in only one dimension.

The robots have a CMOS colour image sensor (camera), microphones, speakers, a temperature sensor, an acceleration sensor, pressure sensors (on the head, back, chin and legs), LEDs on the head and tail and an infrared distance sensor. The robot is powered by a lithium ion battery pack.

The dimensions of the robot (width × height × length) are 154 mm × 266 mm × 274 mm (not including the tail and ears) and the mass is approximately 1.4 kg (including battery and memory stick).

The Sony AIBO is a state of the art entertainment robot. With a good development environment and a robust design, the AIBO provides a solid platform for investigations into robotics and AI. Further information on the Sony AIBO can be found at [14].

The dimensions of the soccer field are 270 cm × 420 cm. The walls are angled at 45°. The playing surface itself is carpeted to protect the robots and to allow better grip (although the friction of the carpet seems to vary from location to location). Each team has four robots. The rules are only loosely based on soccer, but the objective of the game is identical. More detailed rules and specifications are available at the RoboCup Legged League web site [13].

3   Gait Control

Sony's 4–legged AIBO robots come equipped with a default motion system which is adequate for general use. However, it lacks the speed and versatility required to effectively play soccer.

Instead, legged league RoboCup teams typically use a system based on inverse kinematics to achieve more efficient locomotion. The motion engines used in competition tend to be highly parameterised and extremely flexible. They are mostly based on the walk developed by the legged league team of the University of New South Wales (UNSW) in Australia [8].

In the original UNSW "ParaWalk" system, each leg follows a roughly rectangular trajectory in world space. These world space coordinates are then converted

to joint angles using inverse kinematics and sent to the effectors. Inclining the trajectory planes of different legs to the side allows omnidirectional motion to be achieved. Diagonally opposite legs are raised simultaneously (as in a trot).

Many variations on this general method have since been developed. The system employed for the traction monitoring described in this paper is based on the use of ellipsoidal trajectories. Note however that the fault detection methods presented below are independent of the particular locomotion system used.

4   Method for Fault Detection and Training

Current methods in legged robots use a complicated array of sensors (360° range finders, multiple cameras, sonar) [9] to minimize the chance of a robot colliding with an obstacle. In addition more sensors (torque sensors etc) [10] are used to detect a collision if one occurs. Such approaches are only possible if the developer has access to the workings of the robot. In a situation where the hardware is fixed, a technique that uses only the provided hardware is needed. We were confronted with this problem in the legged league of RoboCup. Here all teams are restricted to the use of unmodified Sony AIBO ERS-210(A) robots.

The camera and infrared distance sensor on the ERS-210(A) don't provide enough support in avoiding obstacles unless the speed of the robot is dramatically decreased. Even in the case that the robot avoids obstacles, the unpredictable movements of other robots mean that collisions are likely to occur. Again the camera and infrared distance sensor generally can't be used for detection as the majority of collisions occur outside the field of view of these sensors. For these reasons we have chosen to use the joint sensors (i.e. the angle of the joint) as the only input to our fault detection system.

The historical data needed to accurately determine the "normal" motion of the limbs is acquired using on-line training (described below). In our case the parameterised walk allows for an infinite number of parameter combinations. To minimise the number of possible combinations we have identified three key walk parameters, *backStrideLength*, *turn* and *strafe*. A fourth parameter is also important, the time parameter, t. It increases from 0 to 1 as the leg moves along its trajectory from start to finish. The parameter space was further divided into the following intervals:

Parameter	Units	Min Value	Max Value	Num. of Intervals
backStrideLength	mm	-100	100	20
turn	degrees	-25	25	12
strafe	mm	-80	80	20
t	-	0	1	20

Training involves letting the robot walk without collision or slip for a period of time, gathering sensor information from each joint. In our experiments we train the robot for about ten minutes. We then calculate the mean μ and standard deviation σ for each joint with all possible parameter combinations. To save

memory while calculating the variance (1) we store only the total of the inputs, the total of the squared inputs, and the number of inputs n, rather than the complete set or sensor data $\{y_1, y_2, ..., y_n\}$. This miserly approach to memory usage allows us to perform these calculations on the limited hardware of the robot.

$$\sigma^2 = \frac{1}{(n-1)}(\sum_{i=1}^{n} y_i^2) - \frac{(\sum_{i=1}^{n} y_i)^2}{(n-1)n} \tag{1}$$

Figure 2 shows the joint sensor as the step progresses (that is, as t moves from 0 to 1 on the horizontal axis) for six steps with the identical parameters. The parameters used for the steps shown are $backStrideLength \in [90,100]$, $turn \in [0,4.166)$ and $strafe \in [0,16)$.
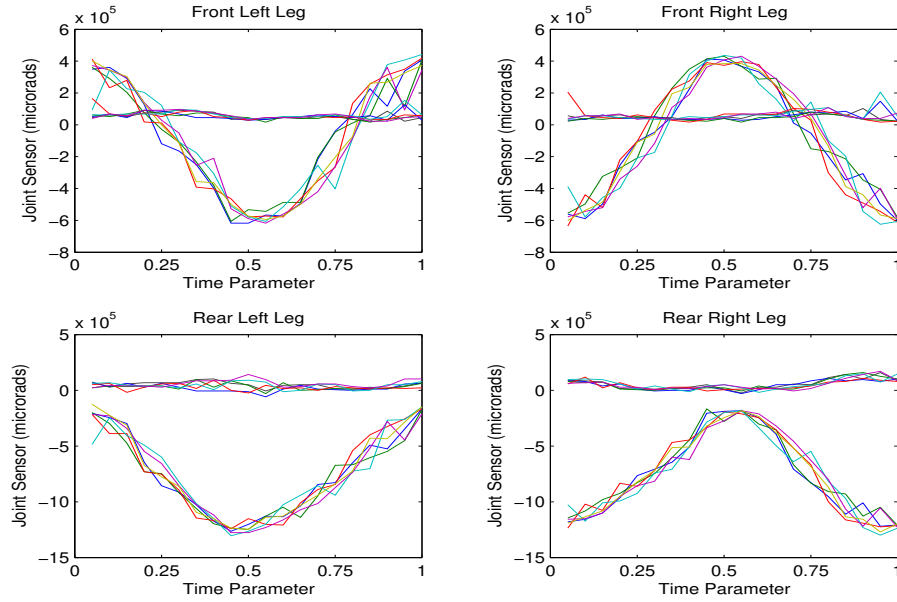

Fig. 2. Movement of legs through six forward steps with identical parameters. This shows the natural variation in "normal" motion.

Although only six steps are shown, the figure gives an indication of the natural variation in "normal" motion that occurs. The roughly horizontal lines are the abductor joints (which move very little during a forward motion), while the curved lines shown are the rotators. It can be seen that the trot gait for a forwards walk leads to the rotators on the right legs being 180° out of phase from the left legs. The unequal default positions of the rotator joints lead to the extreme values of the front and back rotators differing.

5 Collision Detection

Detection of a collision involves observing a joint position substantially differing from its expected value. In our case using an empirical study we found two standard deviations to be a practical measure. Initially we would have considered a collision to have occurred if a single error is found, but further investigation has shown that finding multiple errors (in most cases, three) in quick succession is necessary to warrant a warning that can be acted upon by the robot's behaviour system. It should be noted that if the locomotion engine was more reliable finding only one error may provide accurate results.

For RoboCup a more domain specific diagnosis system is used to provide detailed information about exactly what occurred. This section is further divided into the different types of collisions that occur and the distinguishing features that can be used to identify them.

In the experiments we collected data during different types of collisions; Walking forwards, turning and walking backwards into the field boundary. We assumed this data is representative for all other types of collisions that could occur during a soccer match, such as dog-to-dog collisions.

5.1 Walking Forwards

Our forward walking motion extracts most of its drive from the rear legs - or more precisely, their associated rotator joints. This means that collisions on the front legs result in the most noticeable change occurring on these rear rotators.

Figure 3 indicates the movement of the rear rotators. The graph plots the joint sensor values on the vertical y–axis in microrads against t on the horizontal x–axis. The bold line shows the path of a step that involved a collision. The dotted line represents the mean "normal" path of the joint (that is, during unobstructed motion), with the error bars indicating two standard deviations above and below.

It can be noticed that the clearest differences occur at roughly the midpoint of the time interval. Furthermore, the first few samples of joint positions in each step tend to be influenced by transitions between walk types. It is therefore prudent to deem a collision to have occurred only if an error occurred on the rear rotators with $t \in [0.30, 0.70)$.

Data was gathered in a two step process. First the robot was placed in a situation where no collisions would occur - this test was designed to find false positives (detecting a collision that we deemed not to have happened). False positives occurred in fewer than 1% of steps. In the second test we placed the robot directly in front of the field boundary to test the success rate of detecting a collision. In this case about 98% of collisions were detected. It should be noted that there is a level of human interpretation (and therefore human error) in the gathering of data. The output of the system is compared against what we perceived to have occurred. So if the system did not trigger an expected fault this may be the result of an incorrect human assumption.
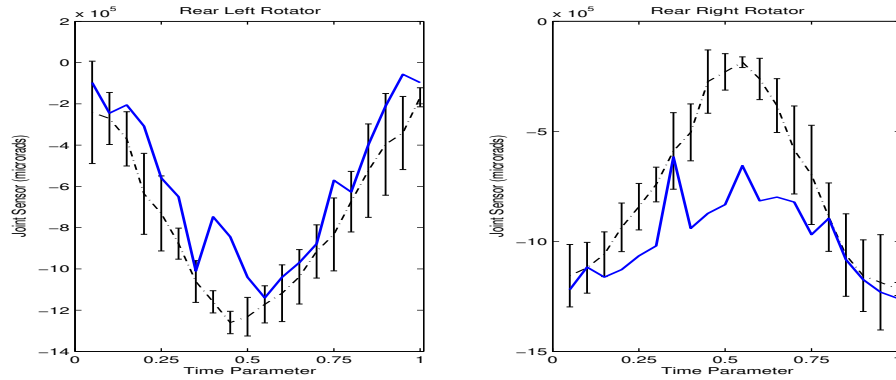
Fig. 3. Rear Rotators for a forwards walking boundary collision on both front legs, front right leg hitting first. The bold line shows the path of a collided motion. The dotted line represents the mean "normal" path of the joint (that is, during unobstructed motion), with the error bars indicating two standard deviations above and below.

5.2 Turning

Detecting a collision during a turn is not unlike detecting a forward collision, except a better conclusion about the impact point and therefore the position of the obstacle may be made. Our current turn motion uses only the rear legs, keeping the front legs stationary so they may be used for controlling the ball. Unfortunately, this also means that any obstruction to the rear legs severely affects the robot's ability to turn. Recognising that a collision has occurred may therefore be very useful, as we could then choose to reemploy the front legs to assist with the turn.

Figure 4 shows the values of the rear abductors for turning to the right when the left rear leg impacts with the boundary. In this case the obstruction can easily be seen on the left rear rotator while the values for the right rear rotator remains within the standard bounds. These are reversed when turning to the left (and a collision occurs on the right rear leg).

Data was gathered in similar manner to that of a forward collision, except for the second test the robot was positioned next to the field boundary in such a way that it would collide when turning. False positives occurred on 6% of unobstructed steps while 100% of collisions were detected. These results differ from forwards collisions because the turn movement is more sensitive, thus detection of a collision is easier but the step is also more likely to generate a fault under normal operation.

5.3 Walking Backwards

Detection of a collision while walking backwards requires a slightly different mechanism compared to a collision while turning or walking forwards. Here,
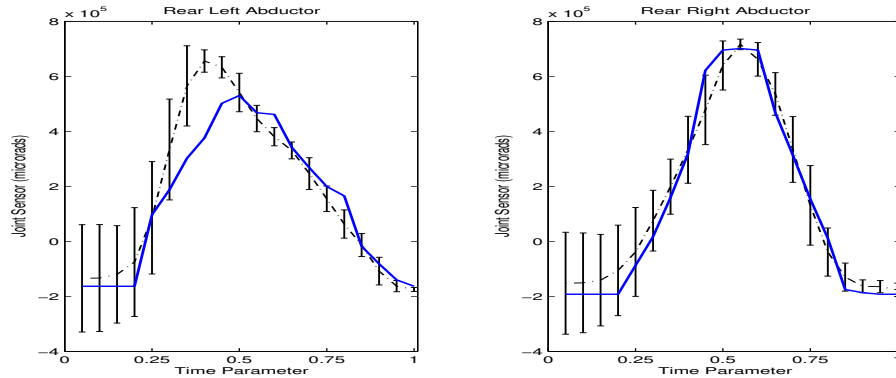
Fig. 4. Rear Abductors for turning to the right with the left rear impacting with the boundary. The bold line shows the path of a collided motion. The dotted line represents the mean "normal" path of the joint (that is, during unobstructed motion), with the error bars indicating two standard deviations above and below.

we found it is useful to examine the gradient of the sensor values between two specific time locations. Generally when this type of collision occurs the impact of the first leg is enough for the second leg to make little or no contact with the obstacle, meaning the motion of that leg is unaffected. Again the rear rotators give the most reliable indication that a collision has transpired.

Figure 5 shows the standard and obstructed movement of the left rear rotator when the left rear leg makes the first impact with the wall. The key time interval for the left leg is between $t \in [0.30, 0.35)$ and $t \in [0.35, 0.40)$, if a collision took place the gradient will be more than one standard deviation less than that of the average step. For the right leg the key time interval is between $t \in [0.70, 0.75)$ and $t \in [0.75, 0.80)$.

We experienced no false positives while still detecting 71% of the expected collisions. Limiting detection to a single time interval decreased the accuracy of detecting a collision but ensures a low rate of false positives.

5.4 Leg-Lock

The aim of this section is to detect when the legs of two robots are locked (Fig.1). This situation occurred frequently during the 2002 RoboCup competition, and appears to be a side effect of the low forward leaning stance adopted by most teams. Robots involved in a leg-lock can take several minutes to free themselves and are useless for the game during this time.

We are able to detect when leg-lock occurs in exactly the same manner as we detect a regular forward collision (see section 5.1). Unfortunately, the two problems appear to be virtually identical as far as the joint paths are concerned. The front abductors (shown in Fig.6) do appear to show a greater deviation
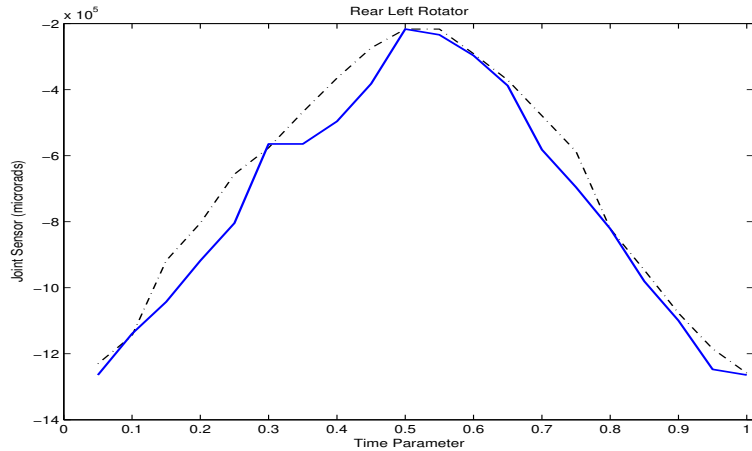
Fig. 5. Rear Left Rotator for walking backwards into the boundary, left rear leg hitting first. The bold line represents the path of a collided motion and the mean path of the joint is shown.

during leg-lock, but with our current approach it is insufficient to accurately discriminate between the two types of collision.

6 Slip Detection

We approached the detection of a slip in the exact same manner as detection of a collision. As with leg-lock, recognising a fault is simple but distinguishing a slip from a regular collision can be challenging. To simulate a slip occurring for the purposes of training, we simply allowed the robot to move around on a slippery tarpaulin on which it is almost completely unable to gain traction.

We only considered slipping during a forward walk. The key indicator of a collision is a substantial change in the rear right rotator, while the left rotator remains mostly unchanged (Fig.3). When a slip took place, the exact opposite occurred (Fig.7). We also added an additional constraint as a precaution against false diagnosis: A "slip" is only recognised if the left rear rotator is outside two standard deviations and the rear right rotator never exceeded two standard deviations anytime during the time interval, $t \in [0.30, 0.70]$. This extra constraint prevents us from detecting a slip during the middle of a step - we must instead wait until the critical part of the step has completed (that is, t must reach 0.70).

7 Discussion

The techniques described above provide us with results that enable our behavior system to make previously impossible decisions. We can now detect a possible
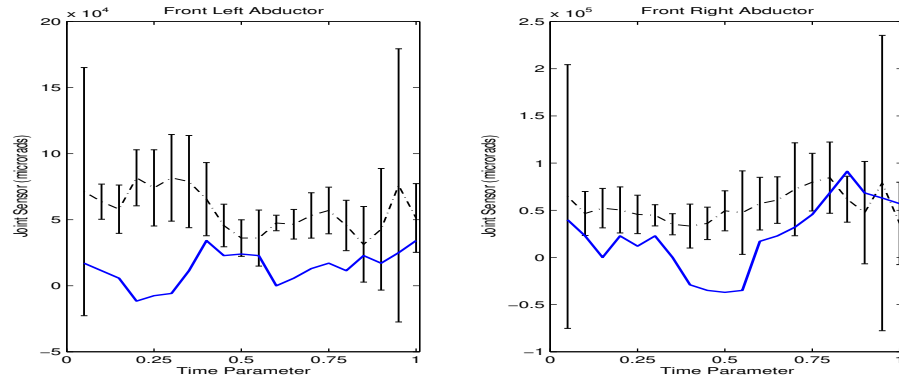
Fig. 6. Front Abductors for a leg-lock on the right front leg. The bold line shows the path of a leg-locked motion. The dotted line represents the mean "normal" path of the joint (that is, during unobstructed motion), with the error bars indicating two standard deviations above and below.
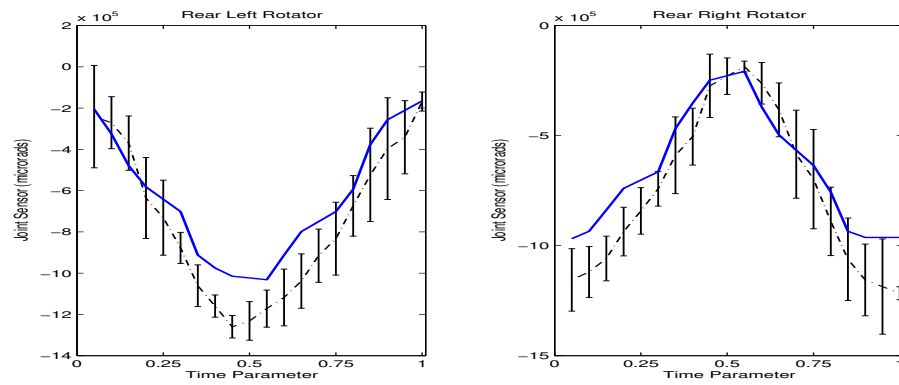

Fig. 7. Rear Rotators for a forwards step that slipped. The bold line shows the path of a slipped motion. The bold line shows the path of a collided motion. The dotted line represents the mean "normal" path of the joint (that is, during unobstructed motion), with the error bars indicating two standard deviations above and below.

collision in locations that were once blind spots, for example while walking backwards or collisions while turning. This extra information enables us to vary our behaviour accordingly.

The most prominent use of the system is during a collision while chasing a ball. Basically our behaviour is designed so only one dog will chase at any given time. In the past the only input on whether to chase was the vision distance to the ball. You could deem that a collision occurred if you were attempted to move towards the ball yet the ball distance appears to be steady. This approach was flawed - obstruction of the ball, bad lighting or even the ball moving away at a speed equal or greater then your speed results in failure. Having the collision data enables us to quickly detect the problem and allow another un-obstructed dog to chase the ball.

In addition, we are able to improve the reliability of odometry data by taking collisions into account. This allows us to better determine the robot's location in the world.

Future improvements to the system include an unsupervised learning approach to train and classify the data. This would eliminate the human observation errors that occur and hopefully lead to a more precise and robust classification.

8   Summary

In this study, a traction monitoring system was developed in software using a minimum of hardware feedback. The techniques of the present study were developed and tested using Sony's AIBO robots. Collisions and leg-lock were detected with a very high degree of reliability. Additionally, slips are recognised with reasonable accuracy.

The experiments show that traction monitoring for a legged robot is feasible without hardware modification. Such a system provides the possibility of a future locomotion system able to adapt to changes in the environment while maintaining a high level of stability.

After appropriate modifications it should be possible to employ these techniques for different types of robots and in more general environments as well.

References

1. Clarke, J.E., Cham, J.G., Bailey, S.A., Froehlich, E.M., Nahata, P.K., Full, R.J., Cutkosky, M.R.: Biomimetic Design and Fabrication of a Hexapedal Running Robot. IEEE International Conference on Robotics and Automation (2001) 3643–3649
2. Fujita, M., Kitano, H.: Development of an Autonomous Quadruped Robot for Robot Entertainment. Autonomous Robotics **5** (1998) 7–18

3. Dario, P., Gugliemelli, E., Laschi,C: Humanoids and personal robots: Design and Experiments. Journal of Robotic Systems **18**(12) (2001) 673–690

4. Krovi, V., Kumar, V.: Optimal Traction Control in a Wheelchair with Legs and Wheels. Proceedings of the 4th National Applied Mechanisms and Robotics Conference (1995) AMR-030

5. Ritzman, R.E., Quinn, R.D., Watson, J.T., Zill, S.N.: Insect Walking and Biorobotics: A Relationship with Mutual Benefits. BioScience **50**(1) (2000) 22–33

6. Yoshida, K., Hamano, H., Watanabe, T.: Slip-Based Traction Control of a Planetarty Rover. 8th International Symposium on Experimental Robotics (ISER'02) (2002)

7. Matthies, L., Xiong, Y., Hogg, R., Zhu, D., Rankin, A., Kennedy, B., Hebert, M., Maclachlan, R., Won, C., Frost, T., Sukhatme, G., McHenry, M., Goldberg, S.: A Portable, Autonomous, Urban Reconnaissance Robot. Robotics and Autonomous Systems **40**(2-3) (2002) 163–172

8. Hengst, B., Pham, S.B., Ibbotson, D., Sammut, C.: Omnidirectional Locomotion for Quadruped Robots. RoboCup 2001: Robot Soccer World Cup V (2002) 368–373

9. Everett, R.B.: Sensors For Mobile Robots : Theory and Application. A K Peters Ltd (1995)

10. Todd, D.J.: Walking Machines - An Introduction to Legged Robots. Korgan Page Ltd (1985)

11. RoboCup web site. `http://www.robocup.org`

12. RoboCup Rescue web site. `http://www.isd.mel.nist.gov/RoboCup2003`

13. RoboCup Legged League web site. `http://www.openr.org/robocup/index.html`

14. The Sony AIBO Entertainment Robot web site. `http://www.aibo.com`