

# A Robust Statistical Collision Detection Framework for Quadruped Robots

Tekin Meriçli, Çetin Meriçli, and H. Levent Akin

Department of Computer Engineering  
Boğaziçi University, Istanbul, Turkey  
{tekin.mericli,cetin.mericli,akin}@boun.edu.tr

**Abstract.** In order to achieve its tasks in an effective manner, an autonomous mobile robot must be able to detect and quickly recover from collisions. This paper proposes a new solution to the problem of detecting collisions during omnidirectional motion of a quadruped robot equipped with an internal accelerometer. We consider this as an instance of general signal processing and statistical anomaly detection problems. We find that temporal accelerometer readings examined in the frequency domain are good indicators of regularities (normal motion) and novel situations (collisions). In the course of time, the robot builds a probabilistic model that captures its proprioceptive properties while walking without obstruction and uses that model to determine whether there is an abnormality in the case of an unfamiliar pattern. The approach does not depend on walk characteristics and the walking algorithm used, and is insensitive to the surface texture that the robot walks on as long as the surface is flat. The experiments demonstrate quite fast and successful detection of collisions independent of the point of contact with an acceptably low false positive rate.

## 1 INTRODUCTION

One of the most important skills for an autonomous mobile robot is to safely and robustly navigate in both known and unknown environments, such as planetary surfaces [1, 2] and disaster areas [3]. Different tasks and operation environments require different mobility configurations. Every mobility configuration has its advantages and disadvantages; however, all have a possibility of suffering from unforeseen failure events, such as collisions.

Detection of and fast recovery from failures are crucial for autonomous systems since improper treatment of such situations may cause loss of control and harm the system or the environment. Inability to detect collisions may lead to a failure of the whole mission. For instance, collision of a mobile service robot with a low doorstep will prevent it from reaching the destination hence accomplishing its goal, or collision of a soccer playing robot with another robot on the field will probably lead to a failure in its main task which is to chase the ball and score goals. Such situations require the mobile robot to be able to detect the unexpected/abnormal event quickly and to infer the cause of the event in order to react appropriately [4].

Obstacle detection and collision avoidance problems are usually solved by processing the data obtained from dedicated hardware such as laser range-finders and touch/bump sensors. However, the task becomes much harder when those sensors are not present. This paper considers a case of a quadruped robot, Sony Aibo ERS-7 [5], with accelerometers and a color camera. It is able to perform vision-based localization. In this case, one possible approach is to use vision to detect collisions; especially in the environments where the structure is known by the robot which can localize itself in that environment [6–8]. The difference between vision-based and odometry-based pose (location and orientation) estimates may give an idea about collisions since the pose of the robot will not change as expected if the robot is blocked by an obstacle when trying to move. However, the objects that are used for localization are not always in the field of view of the robot, hence vision-based methods may easily fail. Failure in detecting collisions usually has a severe impact on the pose estimate of the robot when only the odometry feedback is used to update the estimate due to the lack of visual information.

This paper proposes a new approach for detecting collisions during omnidirectional motion of an Aibo robot. There is a limited number of sensors on the robot and those available do not include touch/bump sensors that can be used for detecting collisions directly. Hence, our proposed method utilizes internal accelerometer readings, which are obtained in the time-amplitude domain, and processes this signal in the frequency domain since the robot generates a noisy but regular pattern while walking. Approaching the problem from a statistical novelty detection point of view, our method can discriminate between what is normal and what is not in terms of accelerometer signal patterns. The main advantages of our method are

- being collision-point-independent,
- detecting collisions with an *acceptably low* false positive rate,
- being computationally cheap and fast, which is essential considering limited processing power on Aibo robots that is shared between several processes.

The rest of the paper is organized as follows. In Section 2, previous work on collision detection in literature is discussed. Section 3 restates the problem to be solved and elaborates on our approach. Experiments are described and results are discussed in Section 4. Section 5 summarizes and concludes the paper, providing possible extensions to our work.

## 2 RELATED WORK

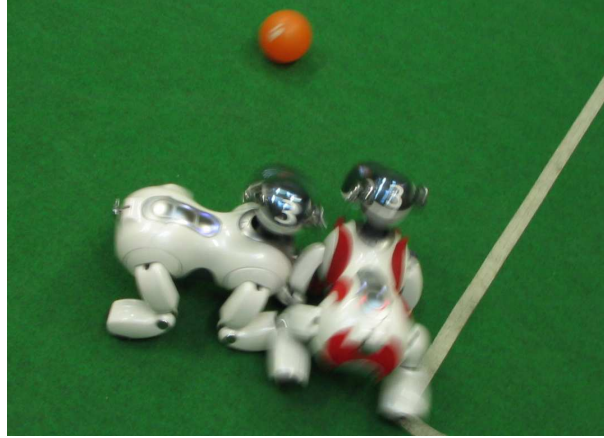
There has been limited research on collision and slippage detection using Aibo as the robotic platform. Vail and Veloso [9] used the accelerometer readings for surface detection and velocity prediction. In their work they used a supervised learning approach, feeding a decision tree with labeled data collected while the robot was marching in place on different types of surfaces. Quinlan *et al* [10, 11] showed that it is possible to learn normal motion of the limbs for distinct

walking directions and used this to detect slippage of the legs and collisions of the robot with its environment. The main drawbacks of this method are that it is not generalized for omnidirectional walking and that it is not independent of collision point. This method will not be able to detect the collision when the robot touches the obstacle with its chest while the legs are freely moving. Also, they used ERS-210 model in their experiments; however, ERS-7 has much more powerful servo motors and therefore the paws can slip on the surface even though the robot is obstructed, in which case this approach will again fail in detecting collisions. Hoffman and Göhring [12] proposed an approach that compares the desired motion of the actuators; that is the actuator command, with actual motion calculated from servo readings. Their work also considers very limited walking speeds and directions for training; therefore, it has almost the same shortcomings of the work of Quinlan *et al.*

### 3 APPROACH

Having a perfect map of the environment with all the static obstacles are marked, makes the life of a mobile robot very easy in terms of avoiding collisions even if it has a very limited number of sensors. In such a case the robot may not even need collision detection capability. However, when there are dynamic objects around, it becomes crucial for the robot to be able to detect collisions since getting stuck may result in the complete failure in the robot's task. Robot soccer environments are perfect examples of such dynamic environments with many robots on a small field [13]. Since 2005, RoboCup 4-legged league games have been played on a  $540\text{cm} \times 360\text{cm}$  field with 4 robots in each team [14]. With that many robots on that small field, it is therefore inevitable to have collisions which may be between two or more robots or between a robot and a static object such as one of the goals or a beacon. Not having the ability of detecting such situations may have significant negative impacts. For instance, consider the case where the robot gets the ball and tries to make a  $180^\circ$  turn to aim the opponent goal. During its motion, if it collides with another robot and is not able to see sufficient number of objects to correctly update its pose estimate, hence relying only on its odometry-based update, it may very well end up with shooting towards an arbitrary direction, even towards its own goal. More generally, while the robot is walking on the field and updating the pose estimate based on odometry feedback, colliding with another object without noticing may cause mislocalization, which results in improper behaviors. Figure 1 shows two robots colliding with each other where neither of them can proceed in the desired direction. This is one example of a set of situations which we want the robot to detect and avoid using our proposed method.

The problem is specified using Aibo details to make it concrete, but the method we propose is general and can be applied to any legged robot with an accelerometer. It is essentially an analysis of the signal obtained by keeping a history of accelerometer readings. The internal accelerometers of Aibo consist of a mass/spring system located in the core of the body and they provide real-valued



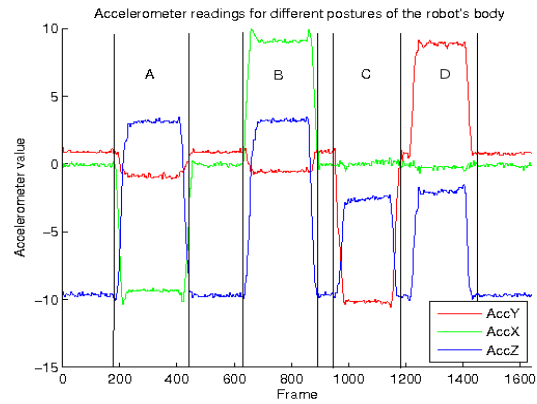
**Fig. 1.** One robot colliding with another, which is a very common situation in a robot soccer game. Both are unable to proceed in the desired direction; hence, the result is failure in the task.

estimates of the acceleration of the body in  $x$ ,  $y$ , and  $z$  axes. These values are read at 30Hz, which is the decision making rate in our implementation, although new values from the sensors can be read at 125Hz. Positive  $y$  is defined as the direction the robot is facing, positive  $x$  as directly away from the right side of the robot, and positive  $z$  as up from the floor. Figure 2 shows the accelerometer readings for different body postures of Aibo.

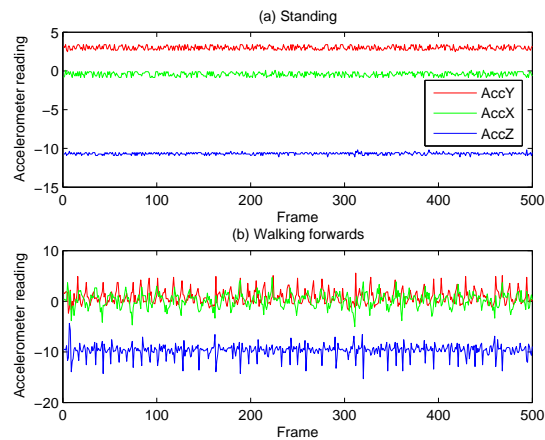
As it is apparent from Figure 2, the readings can provide useful information when there is a substantial change in the robot's body posture, e.g. when it falls over. However, due to the noisy characteristics of the accelerometers, it may be quite challenging to identify minor changes. Figure 3 shows the accelerometer data collected for 500 frames while the robot is (a) standing still and (b) walking forwards with a constant velocity without any obstruction.

Since the default motion system built by Sony for Aibo robots is inadequate in terms of speed and versatility which are necessary to be a good soccer player, most of the RoboCup 4-legged league teams developed inverse kinematics-based motion modules for better locomotion. In our parameterized and flexible motion engine, the loci defined by 3D polygons are traced by the paws in a certain number of intermediate steps and diagonally opposite legs move simultaneously, resulting in a trot gait. Different speeds are obtained by stretching or squeezing the polygons, and walking in different directions is achieved by rotating the polygons around the  $z$  axis accordingly. In either case the number of intermediate steps to trace the whole polygon stays the same. This is what makes the motion of the body, hence the accelerometer signals, regular in a certain way independent of speed and direction of the motion.

During the analysis of the accelerometer readings for omnidirectional walking, we observed that the characteristics of those signals change when there is a

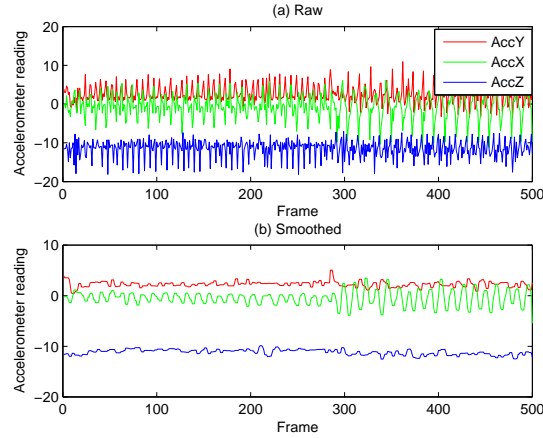


**Fig. 2.** Accelerometer readings for different body postures. In region A, the robot lays on its left on the ground, in region B, it lays on its right, in region C, it is sitting on its back (i.e. its tail), in region D, it is standing upside down on its head. All the rest correspond to stand still on the paws.



**Fig. 3.** Accelerometer readings while the robot is (a) standing still and (b) walking forwards with a constant velocity. The positive value on the y axis is a result of the robot's default pose that slightly tilts forward. As it can be seen from the plots, the readings are quite noisy.

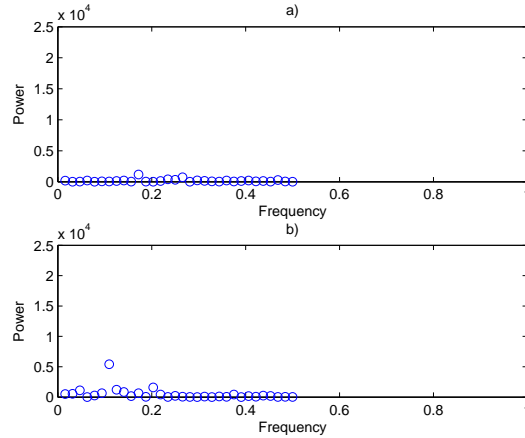
collision and a different pattern emerges. Those changes, however, are not very obvious or easy to characterize using simple threshold values. Figure 4 illustrates the three signals obtained from the accelerometers while the robot is walking forwards and starts colliding with an obstacle at 300<sup>th</sup> frame. In order to make the change easier to see, a 4253h-twice smoother [15] was applied (Figure 4b) to the raw signals (Figure 4a).



**Fig. 4.** Accelerometer readings ((a) raw and (b) smoothed using 4253h-twice smoother) during forward walking where the robot collides with an obstacle starting from frame number 300.

As it can easily be seen from the plot, the magnitudes of the signals, especially for the  $x$ -axis signal, are different during collision compared to regular walk. The reason that  $x$ -axis is affected the most is due to the trot gait style, which -by default- makes the robot swing around the  $y$ -axis. When the robot collides with an object, it tries to climb over it, and that motion results in a greater magnitude of swinging. Same effect arises when the robot's paws get stuck with the carpet that it walks on. Therefore, we decided to examine the patterns generated on the  $x$ -axis.

Buffered temporal readings of  $x$ -axis values are transformed to the frequency domain via the application of Fast Fourier Transform (FFT) operation. In order to expose the patterns in the recent readings, FFT operation is applied only to a window keeping the recent history of the readings. Since the buffer is filled in a certain number of frames that is specified by the length of the window that we slide over the buffer, keeping the window size as small as possible is crucial in responding to collisions quickly. Considering our decision making rate of 30 frames/second and necessity to supply an array with a size of a power of 2 to the FFT function, we decided to use a sliding window of size 64, which was enough for FFT to better extract the frequency characteristics of the accelerometer signal.



**Fig. 5.** Frequency/Power representation for a) signal portion recorded under unobstructed walking, b) signal portion recorded under collision situation.

The change in the frequency/power distribution is obvious as in Figure 5, but since different motion commands produce different signal characteristics and different frequency/power distributions, representing the change in distribution over time is a challenging task. To cope with this problem, we defined Sum of Squared Differences (SSD) of two consecutive frequency/power distributions as the sum of squared differences of each power value and its corresponding previous value. By doing so, we are able to represent the change in frequency/power distribution between two consecutive frames with a single scalar value. Since the frequency/power characteristics differ drastically between normal (unobstructed walking) and novel (collision) situations, we expect the corresponding SSD values to be different as well. The major problem here is that collecting collision data for all possible collision situations and for all possible walk speeds and directions is impractical; hence, formulating this problem as a supervised learning problem is neither practical nor flexible enough to capture most of the abnormal occurrences (i.e. collisions). Therefore, we decided to formulate the problem as an instance of statistical anomaly/novelty detection problem.

Novelty detection is the identification of new or unknown data or signal that a machine learning system is not aware of during training [16]. Although it is not possible to train a system on all possible failure situations, it is relatively easy to train a system on known situations and reject the ones that substantially diverge from what is expected. In the collision detection case, it is much easier to train the robot on the normal walking behavior rather than trying to simulate all possible collision cases to provide as negative examples.

We used hypothesis testing which is a well-known statistical technique for answering a *yes-or-no* question about a population and assessing the probability that the answer is wrong. In our problem, we are seeking the answer of the ques-

tion whether the sample of SSDs consisting of SSD values calculated for a certain number of recent consecutive frames are drawn from the population of SSD values calculated under normal circumstances. Formulation of such questioning as a hypothesis testing problem is shown in Equation 1

$$\begin{aligned} H_0 : \mu_{sample} &= \mu_{regular} \\ H_1 : \mu_{sample} &\neq \mu_{regular} \end{aligned} \tag{1}$$

where,  $\mu_{sample}$  is the mean of a certain number of recent consecutive SSD values and  $\mu_{regular}$  is the learned mean of SSD values calculated under normal walking conditions. Here, the null hypothesis is that the mean of the recent SSD values and the learned mean of SSD values are equal, i.e., the robot is walking without any obstruction. The alternative hypothesis states that the means are different; that is, the recent SSD values are calculated under a novel situation.

Based on the *Central Limit Theorem* [17], we assumed that the SSD values calculated under regular walking conditions come from a normal distribution. A one-tailed *t*-test [17] is used to test the hypothesis in order to be able to use small ( $N < 30$ ) samples and hence to be able to respond quickly in case of an unfamiliar pattern. The formula for *t*-test is shown in Equation 2 where  $\sigma_{regular}$  is the standard deviation of the regular SSD values population and  $N$  is the sample size. *Type I errors*, or *false positives* are crucial for a mobile robot since such errors fire false collision signals resulting in improper behaviors. Depending on the application domain, there is always a trade-off between false positives and false negatives. In the robot soccer case, Type II errors (false negatives; i.e. the robot not detecting collision while colliding) are preferred over false positives (i.e. the robot detecting collision while walking without obstruction); therefore, we applied a conservative test and rejected the null hypothesis with an expected false rejection probability of  $p = 0.001$ .

$$T = \frac{\mu_{sample} - \mu_{regular}}{\sigma_{regular}/\sqrt{N}} \tag{2}$$

The pseudo-code summarizing the operations from reading accelerometer values to concluding about collision status is given in Figure 6.

## 4 EXPERIMENTS

Experiments are conducted on a regular RoboCup 4-Legged field using an Aibo ERS-7 robot. Experimental study consists of two phases:

- Building a statistical proprioceptive sensory model
- Testing the performance



```

function updateCollisionDetector()
    read recent accelerometer values
    slide the window over the buffer
    perform FFT on the window
    calculate SSD values
    update sample mean for the last N SSD values
    apply one-tailed t-test
    if testResult > tValue
        set collision flag = true
    else
        set collision flag = false
    end
end
end

```

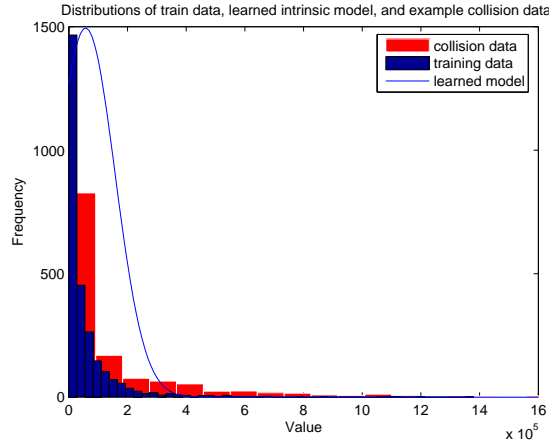
**Fig. 6.** Pseudo-code of the collision detector.

#### 4.1 Building a Statistical Proprioceptive Sensory Model

In order to develop a model of frequency changes under normal walk conditions, we considered two most common situations: self positioning and ball chasing. First, we let the robot go to its predefined initial position on the field omnidirectionally, so that it would walk in various directions with various speeds. After self positioning, we let the robot chase the ball for nearly 1.5 minutes. Ball was manually moved across the field in such a way to make sure that the robot walked omnidirectionally in a symmetric manner. This setup provided us more stratified data. During the execution, a total of 4150 frames were processed and a Gaussian distribution was fit on the data. The distribution of gathered train data, learned model, and the distribution of a test sequence containing collisions are illustrated in Figure 7. As it can be seen in the figure, although the gathered data is not in normal form, learned model is a good estimator of it. The collision data contains normal parts (the left-most bin) and the collision parts. As we mentioned in Section 3, our domain specific preference about Type I and Type II errors is in favor of Type II errors; therefore, we used a very conservative hypothesis testing and considered only extremely far values from the mean as the indicators of novel situations (collisions). We used a sample size of  $N = 10$  and an expected false rejection probability  $p = 0.001$ , and we applied a one-tailed  $t$ -test with 9 degrees of freedom yielding a  $T$  value limit of 4.297.

#### 4.2 Testing the Performance

Using our code base, a simple behavior for handling the collision situations was implemented for assessing the performance of the collision detection mechanism. This behavior makes the robot walk in the exact opposite of the direction that it was walking when it encounters a collision situation, then makes it to walk sideways to avoid the obstacle, and finally resets the collision detector buffers to the learned sample mean value. The reason behind resetting is to provide the



**Fig. 7.** Distribution of gathered train data, learned model, and the distribution of a test sequence containing collisions.

robot empty buffers since the buffers are implemented as sliding windows over a history of accelerometer readings and they still keep the readings obtained during collision even after the collision is detected and handled.

The performance was tested for both collisions during self positioning and ball chasing. For testing collision detection performance during self positioning we placed stationary robots on the field and recorded the average response time during collisions. Experimental setup regarding ball chasing was consisting of a stationary robot placed in front of a ball, 50 cm away from it. Two test robots running the same soccer player code were released from the same point 5 times each and the average time spent in between touching the obstacle for the first time and touching the ball was recorded. Only one of the robots had our proposed collision detection algorithm, all other code being equal.

Compared to the one that does not have the collision detection module, the robot featuring our collision detection algorithm reduced the measured time nearly by a factor of 3, having an average time to reach the ball of 6.52 seconds while it took 19.82 seconds on the average for the other robot to push the obstacle completely and reach the ball. Excluding the time needed to walk from the starting point to the obstacle and from the obstacle to the ball, the average time to detect collisions, which is the time between the first touch to the obstacle and the execution of the retreat behavior, was 0.64 seconds. The detailed results are given in Table 1. Although we cannot claim that the proposed approach results in a zero false positive rate, we did not observe any false positives during our experiments.

**Table 1.** Test results comparing the performances of robots featuring collision detection algorithm and not featuring collision detection algorithm.

	Collision detection algorithm <b>ON</b> / <b>OFF</b>	
<b>Trial</b>	<b>Total Time</b>	<b>Obstacle Contact Time</b>
1	7 / 20.3	0.1 / 17
2	6.5 / 20.7	0.3 / 19
3	6.2 / 22.5	1.8 / 20
4	6.9 / 18.6	0.7 / 17
5	6 / 17	0.3 / 14
<b>Mean</b>	<b>6.52 / 19.82</b>	<b>0.64 / 17.4</b>
<b>Std. Dev.</b>	<b>0.43 / 2.1</b>	<b>0.24 / 2.3</b>

## 5 CONCLUSIONS AND FUTURE WORK

Detecting collisions and recovering from them quickly is crucial for a mobile robot to achieve its tasks effectively. Collision detection on mobile robots that are not equipped with special hardware such as bump sensors is a problem that is challenging yet not studied deeply. In this paper, a robust statistical collision detection framework for a quadruped robot equipped with an accelerometer sensor is presented. The proposed method considers the problem as an instance of general signal processing and statistical anomaly/novelty detection problem.

The main contribution of this paper is proposing a novel collision detection framework for legged robots which is:

- collision-point-independent,
- walk speed and direction independent
- walk surface texture independent as long as the surface is flat
- fully autonomous in building statistical proprioceptive sensory model
- and, computationally cheap and fast

Although it was tested only on four-legged robots, the method can be applied to any kind of legged robot with a periodical walking pattern.

The method is tested on a soccer playing legged robot, specifically on a Sony Aibo ERS-7 four-legged robot. Experimental results showed that the proposed approach provided fast recovery from collisions, hence reducing the time necessary for accomplishing the task. By building a statistical proprioceptive sensory model fully autonomously for normal walk conditions in less than two minutes, the robot was able to detect collisions with a zero false-positive rate during experiments.

Applying this method to bipedal humanoid robots, investigating the underlying properties of the SSD distribution, and developing an online version of the proposed method are among the possible future works.

## References

1. K. Yoshida, H. Hamano, and T. Watanabe. Slip-based traction control of a planetary rover. In *8th International Symposium on Experimental Robotics (ISER)*, volume 5 of *Springer Tracts in Advanced Robotics*, pages 644–653. Springer, 2002.
2. S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains, functions and integration. *International Journal of Robotics Research*, 21(10-11):917–942, 2003.
3. The RoboCup Rescue official website. <http://www.robocuprescue.org/>.
4. C. Plagemann, D. Fox, and W. Burgard. Efficient failure detection on mobile robots using particle filters with gaussian process proposals. In *IJCAI*, pages 2185–2190, 2007.
5. The Sony Aibo robots. <http://support.sony-europe.com/aibo/>.
6. T. Röfer and M. Jünger. Vision-based fast and reactive monte-carlo localization. In Daniel Polani, Andrea Bonarini, Brett Browning, and Kazuo Yoshida, editors, *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*, pages 856–861. IEEE, 2003.
7. M. Sridharan, G. Kuhlmann, and P. Stone. Practical vision-based monte carlo localization on a legged robot. In *IEEE International Conference on Robotics and Automation*, April 2005.
8. K. Kaplan, B. Çelik, T. Meriçli, Ç. Meriçli, and H. L. Akın. Practical extensions to vision-based monte carlo localization methods for robot soccer domain. In *RoboCup*, volume 4020 of *Lecture Notes in Computer Science*, pages 624–631. Springer, 2006.
9. D. Vail and M. Veloso. Learning from accelerometer data on a legged robot. In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.
10. M. J. Quinlan, C. L. Murch, R. H. Middleton, and S. K. Chalup. Traction monitoring for collision detection with legged robots. In Daniel Polani, Brett Browning, Andrea Bonarini, and Kazuo Yoshida, editors, *RoboCup*, volume 3020 of *Lecture Notes in Computer Science*, pages 374–384. Springer, 2003.
11. M. Quinlan, S. Chalup, and R. Middleton. Techniques for improving vision and locomotion on the sony aibo robot. In *Australasian Conference on Robotics and Automation*, Brisbane, 2003.
12. J. Hoffmann and D. Göhring. Sensor-actuator-comparison as a basis for collision detection for a quadruped robot. In *RoboCup*, pages 150–159, 2004.
13. RoboCup International Robot Soccer Competition. <http://www.robocup.org>.
14. The RoboCup Standard Platform League. <http://www.tzi.de/spl>.
15. P. F. Velleman. Definition and comparison of robust nonlinear data smoothing algorithms. *Journal of the American Statistical Association*, 75(371):609–615, September 1980.
16. M. Markou and S. Singh. Novelty detection: A review - parts 1 and 2. *Signal Processing*, 83:2481–2521, 2003.
17. R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye. *Probability and Statistics for Engineers and Scientists*. Prentice-Hall, Inc., New Jersey, 8<sup>th</sup> edition, 2007.