Acceleration based Collision Detection with a Mobile Robot

Patrick Speleers and Marc Ebner
Ernst Moritz Arndt Universität Greifswald, Institut für Mathematik und Informatik
Walther-Rathenau-Str. 47, 17487 Greifswald, Germany
{ps123430,marc.ebner}@uni-greifswald.de

***Abstract*—Data from acceleration sensors can be used for collision detection in autonomous mobile robots. A mobile robot was equipped with a three axis accelerometer. Three different methods for collision detection are evaluated: a simple threshold on the acceleration signal along the driving direction, a running median filter and a frequency based averaging filter. The three methods were first evaluated in an artificial setting and then tested while the robot was driving around in an office environment. The frequency based averaging filter performed best.**

I. Introduction

Autonomous mobile robots [1] are used for a variety of tasks, e.g. transportation of goods within a building. During operation, these robots will eventually collide with obstacles. Even though robots are equipped with sensors (visual, infrared, sonar or laser) collisions cannot be completely avoided.

Several researchers have used acceleration sensors for collision detection. He et al. [2] compute the standard deviation of the acceleration signal using median filtered acceleration data obtained from a running window of 7 consecutive values. This standard deviation is then compared against a threshold to detect collisions. The collision angle is computed from the angle of the largest force that impacted on the robot. Nadarajan and Sridharan [3] classify sensory input vectors (consisting of acceleration and gyroscope data) using a support vector machine to detect instability in an Aldebaran Nao robot. Acceleration sensors are also used to deploy airbags in cars [4]. Barfield Jr. and Welch [5] have used historical sensor data from potential vehicle collisions to learn a model for collision detection. Moorits and Usk [6] used data from a 3 axis solid state accelerometer for collision detection in navigational marine buoys.

II. Arduino-based Mobile Robot

For our experiments we are using an Arduino based mobile robot. It consists of a Dagu Robot T'Rex Tank chassis. Sensors and controller were mounted on the robot using a MakerBeam structure. We have used an Arduino Mega 2560 microcontroller to control the robot and to process sensor data. Power to the motors (left and right) were provided through a T'Rex controller. The robot was powered by a 11.1V LiPo battery. Three infrared sensors were used for collision avoidance. Two Sharp GP2D12 infrared sensors with a 10-80 cm measuring range were mounted at a height of 26 cm on the robot. The two sensors were mounted at a 30° angle and were used to navigate the robot around. A Sharp GP2Y0A60SZ infrared sensor with a 10-150 cm measuring range was mounted at a height of approximately 25 cm. This sensor was facing to the front.

A BNO055 acceleration sensor was used for collision detection. This sensor was mounted approximately in the center of the robot's base. This sensor provides acceleration data (with or without the gravity vector) at a rate of 100Hz. The sensor was oriented such that the y axis points to the back of the robot, the x axis to the left and the z axis upwards. Due to the amount of processing required, acceleration data was only obtained at an actual rate of 67 Hz.

III. Collision Detection

The accelerations occurring during a car crash are considerably larger than the accelerations occurring during normal driving behavior [4]. However, mobile robots are able to accelerate quite fast and are also able to decelerate quickly. This makes it difficult to separate accidental collisions of the robot from normal driving behavior. If a robot is coming to a stand still while moving forward, it will experience a large acceleration along its y axis. Therefore, it should be possible to use a simple threshold to detect a collision. Let a_y be the acceleration along the y axis. Let a_{thresh} be the threshold. Then we can use $a_y > a_{\text{thresh}}$ to detect a collision. The threshold should be low enough to detect collisions that do not stop the robot completely. On the other hand, the threshold should also be large enough in order to detect only actual collisions.

In order to better separate normal driving behavior and actual collisions, it is possible to apply a filter to the acceleration data. We evaluate the method developed by He et al. [2] and the method developed by Moorits and Usk [6].

IV. Experiments

We first collect data during normal driving behavior. The robot first accelerates to a certain velocity, maintains this velocity for some time and finally decelerates to a complete stop. We obtain this data for several different driving velocities. Collision data was obtained by placing the robot infront of an obstacle and having him collide with a certain velocity with the obstacle. The robot then drives backwards to prepare the robot for another collision test. Fig. 1 shows the acceleration data for all three methods obtained during normal driving behavior for three different velocities (140, 160, 180 sent

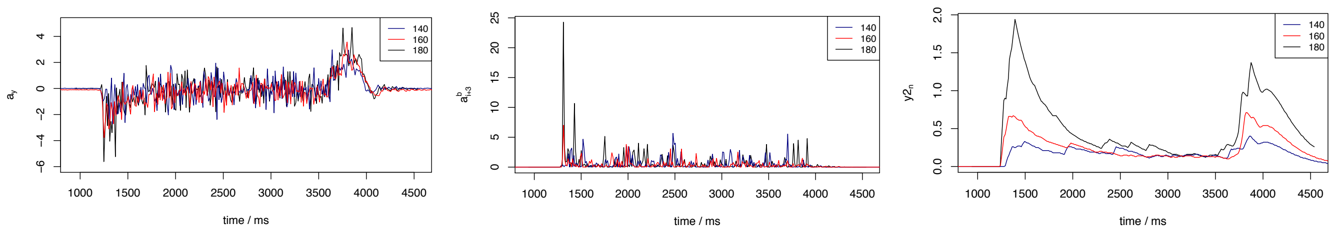

Fig. 1. Acceleration data a_y ($\frac{m}{s^2}$), processed acceleration data a^b_{i+3} and $y2_n$ for three different velocities 140, 160 and 180 for normal driving behavior.

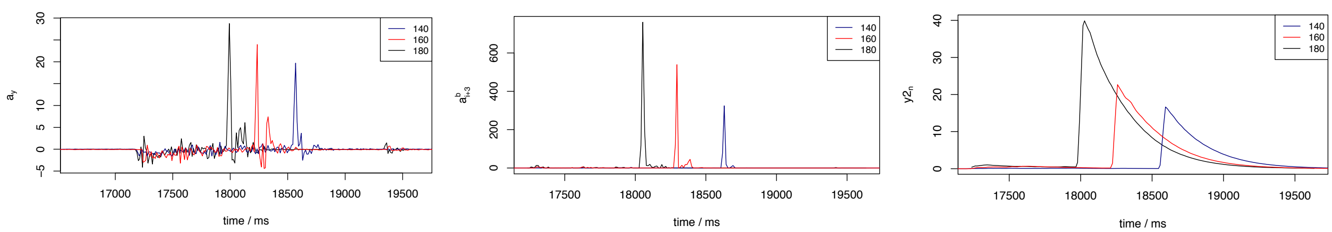

Fig. 2. Acceleration data a_y ($\frac{m}{s^2}$), processed acceleration data a^b_{i+3} and $y2_n$ for three different velocities 140, 160 and 180 during crash tests.

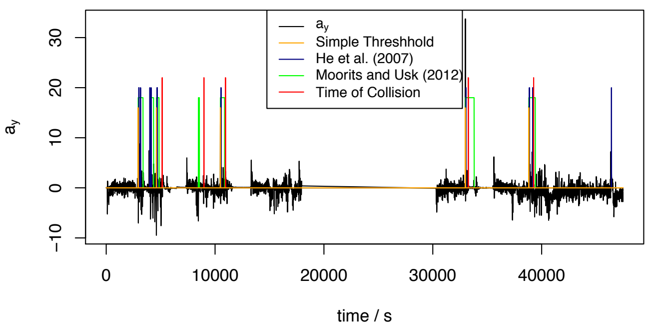

Fig. 3. Collision detection results during normal driving using the centering algorithm inspired from bees. The time of collision was marked manually (red peaks). Results are shown for three different methods: a simple threshold operation on a_y, He et al. [2], and Moorits and Usk [6].

to the T-Rex controller corresponding to a maximum power of 55%, 63% and 71%). The plot on the left shows the unprocessed acceleration data $a_{y,i}$. The plot in the middle shows the averaged median filtered acceleration data a^b_{i+3} using the method of He et al. [2]. The plot on the right shows the filtered data $y2_n$ using the method of Moorits and Usk [6].

Fig. 2 shows the acceleration data for all three methods obtained during crash tests. Comparing the acceleration between the three methods, we see that a^b_{i+3} as well as $y2_n$ show larger differences between the data observed during the crash tests and the data observed during normal driving behavior.

Using the results from these preliminary experiments, we have used a threshold of 8 for the raw acceleration data, a threshold of 40 for the method of He et al. and a threshold of 3.8 for the method of Moorits and Usk.

The three thresholds were tested using normal driving behavior of the robot. If a collision was observed by a human standing nearby, then a button was pressed to indicate the fact that a collision has occurred. Fig. 3 shows the resulting data for all three methods. Due to human response time, the button was always pressed with a slight delay after the actual collision. The method of Moorits and Usk is the only method that is able to accurately detect all collisions. The second collision was not accurately detected by the other methods. Other than this one collision, the other methods were also able to correctly detect the remaining collisions. The method of He et al. also detected a collision where none happened. The collisions at the beginning of the experiment occurred because the robot was caught in between the legs of a chair and repeatedly collided with them. These collisions were correctly identified even though no button was pressed by the human observer.

V. Conclusion

An autonomous mobile robot was equipped with a three axis acceleration sensor. The acceleration sensor was used for collision detection. Three different methods for collision detection were evaluated. A simple threshold on the raw acceleration data, filtering the data with a running median and a frequency based averaging filter. The frequency based averaging filter worked slightly better.

References

[1] P. Corke, *Robotics, Vision and Control*. Berlin: Springer, 2011.

[2] F. He, Z. Du, L. Sun, and R. Lin, "Collision signal processing and response in an autonomous mobile robot," *Neural, Parallel and Scientific Computations*, vol. 15, pp. 319–334, 2007.

[3] K. Nadarajan and M. Sridharan, "Online detection of instability for robust teamwork in humanoid soccer robots," in *Proceedings of the 5th Workshop on Humanoid Soccer Robots, Nashville,TN*, 12 2010.

[4] C.-Y. Chan, *Fundamentals of crash sensing in automotive air bag systems*. Warrendale, PA: Society of Automotive Engineers, Inc., 2000.

[5] J. R. Barfield, Jr. and S. C. Welch, "Automatic vehicle crash detection using onboard devices," *United States Patent US 9,392,431 B2*, 7 2016.

[6] E. Moorits and A. Usk, "Buoy collision detection," in *Proceedings ELMAR, 54th international Symposium, Zadar, Croatia*, J. Božek and M. Grgić, Eds. IEEE, Sep. 2012.