Time Series Classification of IMU Data for Point of Impact Localization

Richard Krieg and Marc Ebner†

Universität Greifswald, Institut für Mathematik und Informatik

Walther-Rathenau-Str. 47, 17487 Greifswald, Germany

Email: {s-rikrie, ebner}@uni-greifswald.de

Orchid ID †: 0000-0003-2725-2454

***Abstract*—Collision detection is a crucial part of every mobile robot system. The field of collision detection has received a lot of attention in recent years. Proper handling of a collision event involves many challenges. Once a collision has occurred, the robot needs to decide on how to proceed. However, prior to taking action it is important to localize the point of impact. This can be done efficiently and accurately using machine learning methods. We show how the recent method FRUITS can be used for point of impact localization using IMU data on a mobile robot. We also compare it with the very efficient algorithm ROCKET. Our results show that both methods are able to accurately identify discrete points of impact but FRUITS has a quicker response time.**

***Index Terms*—Collision Detection, Point of Impact, Time Series Classification**

I. INTRODUCTION

Mobile robots are increasingly deployed inside home environments. A robot needs to detect if a collision has occurred [1]. It also needs to determine where that collision took place, i.e. what body part collided with an obstacle in order to properly react to this collision event. With this contribution we address the second problem, our work focuses on the localization of the point of impact from IMU (inertial measurement unit) data. McMahan et al. [2] used a support vector machine to localize a contact point on a robot arm. Wisanuvej et al. [3] used multiple accelerometers and were also able to identify the material the arm collided with. We show that machine learning methods are also able to localize the point of impact on a mobile robot using IMU data.

II. EXPERIMENTAL SETUP

For all our experiments we have used a HCR robot (manufactured by DFRobot, dfrobot.com). The two wheeled robot (with a swivel wheel in the back) is controlled by a Sabertooth 2x12 motor controller. An Adafruit BNO055 Absolute Orientation Sensor is used to collect the time series data. All mechanical parts are operated by an Arduino Mega. The Arduino sends the data to a Raspberry Pi for further processing. The robot consists of 4 levels. Three bumpers, buttons, battery pack, motor controller, IMU and the two wheels are located on the first or ground level. The Arduino is located on the second, the Raspberry Pi on the third level. Yellow markers have been placed on all four levels marking possible collision points as shown in Fig. 1.

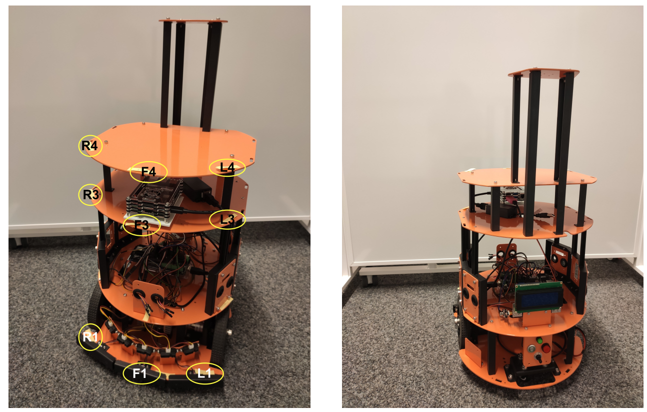

Fig. 1. HCR robot manufactured by DFRobot (dfrobot.com). Yellow markers show the location of the different points of impact.

TABLE I
RESULTS FOR DIFFERENT COMPONENTS OF THE TIME SERIES VECTOR.

	FRUITS		ROCKET	
DIMENSIONS	Accuracy	Time	Accuracy	Time
All	95.36%	$1.83s$	96.03%	$9.93s$
Linear Acceleration only	70.86%	$0.20s$	82.81%	$6.50s$
Angular Velocity only	93.83%	$0.20s$	97.89%	$6.47s$

III. POINT OF IMPACT CLASSIFICATION

Some points of impact might be easier to classify than others. If a collision happens on either the right or left side, we see a change of the angular velocity. If a frontal collision occurs then this change might not be as prominent. Classifying the height of the impact is more difficult.

We use two methods for time series classification to transform the data obtained from the IMU unit (acceleration and rotational velocity): ROCKET [4] and FRUITS [5] (github.com/alienkrieg/fruits). The acceleration data as well as the rotational velocity is normalized by dividing each series by the maximum vector norm. We apply FRUITS to the original data sequence and also to its first derivative. FRUITS computes 1174 features (maximum values) from the transformed data. A ridge classifier is used to classify the transformed data.

IV. RESULTS

We have analyzed the performance of ROCKET and FRUITS on the entire dataset and also on subsets of the dataset.

TABLE II
EXPERIMENTAL RESULTS ON SUBSETS OF THE DATA.

	FRUITS		ROCKET	
CLASS LABELS USED	Accuracy	Time	Accuracy	Time
All	95.36%	1.83s	96.03%	9.93s
(1) L1,F1,R1	99.17%	1.45s	100.0%	4.30s
(2) L3,F3,R3	96.67%	1.44s	98.75%	4.26s
(3) L4,F4,R4	100.0%	1.43s	100.0%	4.26s
(4) F1,F3,F4	97.08%	1.54s	97.08%	4.31s
(5) L1,L3,L4	93.33%	1.45s	96.67%	4.26s
(6) R1,R3,R4	94.17%	1.43s	95.83%	4.26s
(7) L1,F1,R1,L3,F3,R3	95.83%	2.00s	97.92%	8.32s
(8) L1,F1,R1,L4,F4,R4	99.17%	2.00s	100.0%	8.32s
(9) L3,F3,R3,L4,F4,R4	94.17%	2.00s	94.58%	8.40s

ROCKET is a state-of-the-art algorithm. It achieves a mean accuracy of 96.03% over 50 runs with different splits of the dataset into training and testing data. FRUITS achieves nearly the same mean accuracy of 95.36% over the same training and testing data splits (see Tab. I). We have used 72 instances for testing and 288 for training for all experiments. We have used the same number of examples from each class for both sets.

It is noteworthy that ROCKET needs approximately ten seconds for one iteration of the dataset while FRUITS achieves the classification task in about 1.8 seconds on the same machine. All experiments were done on a Windows PC with a Ryzen 7 5800X CPU and 16 GB of RAM.

One obvious question when looking at the collected data is whether all six dimensions are really necessary for the classification or if either linear acceleration or angular velocity alone is sufficient to achieve a comparable accuracy. The results from experiments analyzing this question are shown in Tab. I. We can see that a lot of information actually comes from the gyroscope data, as the classification accuracy of both methods on the reduced dataset is comparable. In the case of ROCKET it is even higher compared to the entire dataset.

This is not surprising as ROCKET tends to work better on lower dimensional data because the method only switches between the dimensions of one time series in convolutions with different kernels. However, the number of features is always the same. As the number of dimensions is reduced, more details and patterns can be analyzed in a single dimension.

Some positions of impact are easier to classify compared to others simply by looking at the tilt or the rotation of the robot. Experimental results to confirm this hypothesis are shown in Tab. II. We restricted the dataset to different combinations of class labels. Linear acceleration and angular velocity was used. We see that better average accuracy is obtained in experiments (1) to (3) compared to experiments (4) to (6). Experiments (1) to (3) vary only the position of impact on one level of the robot (front, left or right). It is easier for both methods to separate those classes than it is to distinguish between the different levels (experiments (4), (5) and (6) vary the height of the position of impact). Experiments (7), (8) and (9) show that the classification task gets easier as the classes are further apart from each other, which is on par with our first intuition.

We have also tested how well the two methods are able to generalize. To do this, we have trained the models on all of the 360 recorded time series, i.e. all experiments with different distances between the robot and the obstacle. The speed of the robot was always set to the same value. We also performed experiments where the robot was not moving and was manually hit. Datasets where the speed was set to 160% of the original value were also done. Tab. III shows the results. FRUITS seems to perform better than ROCKET on the dataset with a stationary robot. Keep in mind that the baseline for guessing the correct class label is $\frac{1}{9} \approx 11.1\%$. It is important to mention that experiments with a high speed robot were often challenging to control because the robot also moved a bit after the impact and sometimes revolved around the obstacle. This led to motion trajectories that did not occur at lower speeds.

TABLE III
COMPARISON RESULTS WITH DIFFERENT SPEEDS.

	FRUITS		ROCKET	
TEST SET	Accuracy	Time	Accuracy	Time
Stationary	57.78%	3.09s	46.67%	17.02s
160% Speed	47.78%	3.11s	52.22%	16.49s

We also combined the original 360 time series with the 180 time series with different velocities and again calculated the mean accuracy over 50 runs of different permutations of this data into training and testing set. FRUITS achieves an overall accuracy of 89.85% (time to completion was 2.25 seconds) and ROCKET accurately predicts 90.76% (time to completion was 14.72 seconds) of the testing data.

V. CONCLUSION

Nine different points of impact were carefully selected on a HCR mobile robot platform from DFRobot. The robot was set to a constant velocity and then collided with a stationary obstacle. For some experiments we also manually hit the robot (simulating a collision with a human). We applied the two algorithms FRUITS and ROCKET to transform the acceleration time series data. A ridge classifier was used to classify the transformed data. Both methods achieve an accuracy of over 95% on the entire dataset. ROCKET was more accurate in general. But the computational effort of FRUITS is lower leading to a quicker response time. Even though ROCKET is slower than FRUITS, both methods are sufficiently fast to allow for real time localization of the point of impact.

REFERENCES

[1] F. Becker and M. Ebner, "Collision detection for a mobile robot using logistic regression," in *16th Int. Conf. on Inform. in Control, Autom. and Rob., Prague, Czech Rep.*, Science and Tech. Pub., 2019, pp. 167–173.

[2] W. McMahan, J. M. Romano, and K. J. Kuchenbecker, "Using accelerometers to localize tactile contact events on a robot arm," in *Workshop on Advances in Tactile Sensing and Touch-Based Human-Robot Interaction, ACM/IEEE Int. Conf. on Human-Robot Interaction, Boston, MA*. 2012.

[3] P. Wisanuvej, J. Liu, C.-M. Chen, and G.-Z. Yang, "Blind collision detection and obstacle characterisation using a compliant robotic arm," in *IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 2249–2254.

[4] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, 34(5), pp. 1454–1495, 2020.

[5] R. Krieg, "Klassifikation von Zeitreihen mithilfe iterierter Summen," *Bachelor's Thesis, Universität Greifswald*, 2021.