2016 IEEE Intelligent Vehicles Symposium (IV)
Gothenburg, Sweden, June 19-22, 2016

Manoeuvre Segmentation Using Smartphone Sensors

Christopher Woo, *Graduate Student Member, IEEE* and Dana Kulić, *Member, IEEE*

Abstract— **In this paper, we propose a classifier-based approach for driving manoeuvre recognition from mobile phone data. We introduce a driving manoeuvre classifier using Support Vector Machines (SVM). We investigate the performance of a sliding window of velocity and angular velocity signals obtained using a smartphone as features for our classifier. Principal Component Analysis (PCA) is used for dimensionality reduction. The classifiers use a vehicle simulation for training data and experimental data for validation. A novel technique to extract the rotation matrix using PCA is presented to calibrate the smartphone's orientation. A classifier performance of 0.8158 average precision and 0.8279 average recall was achieved resulting in an average F1 score of 0.8194. Balanced accuracy was calculated to be 0.8874.**

Index Terms— **Support vector machines, Principal component analysis, Intelligent vehicles, Global Positioning System, Gyroscopes**

I. Introduction

This paper proposes a novel classification-based approach for driving manoeuvre detection based on mobile phone data. For classifier-based approaches, a major impediment is the collection of training data for the classifier [1] [2]. To alleviate the need for cumbersome data collection and labelling, we propose to use simulation data for classifier training. Automobiles are indispensable in today's society and are ubiquitous in personal transportation, commercial activities, and public services. However safety and environmental factors remain a large concern for consumers, businesses, and governments. According to [3], Canada's road fatality rates are statistically the 4th highest in the world with 7.18 fatalities per 1 billion kilometers traveled by a vehicle or vehicle-kilometers (VKM). The annual social costs of automobile accidents in terms of loss of life, medical treatment, lost productivity, and property damage are measured in tens of billions of dollars [3]. The development of tools to recognize driving manoeuvres can provide tools for driver monitoring and training, improving safety and fuel efficiency.

The advent of more accurate Global Navigation Satellite Systems (GNSS), affordable Inertial Measurement Units (IMU), and mobile devices has enabled real-time measurement of vehicle positions, from both vehicle-based and mobile platforms. GNSS such as the Global Positioning System (GPS) allows commercial navigation applications to be used. Mobile phones are ubiquitous in developing and developed countries making cheap GPS and IMU sensors already available to the typical North American driver [4]. Inexpensive IMU sensors can provide frequent accurate updates for the short term while correcting with GPS in the long term as proposed in [5].

In this paper, we propose a classifier-based approach for driving manoeuvre recognition from mobile phone data. While a telematic device can connect to the car directly to collect information relating to automobile control (eg. steering, accelerator position, and brake position), it is difficult to install for the average consumer and costly to distribute for companies tracking driver behaviour. Instead, a mobile device can be used to collect kinematic data as it is ubiquitous for an average consumer and contains sensors to accurately estimate a vehicle's trajectory [4].

Past studies have approached the manoeuvre segmentation problem using different methods. A simple approach is to analyze the angular velocity signals. Chen et al. present a method to detect lane changes and intersection turns using the gyroscope sensor on a smartphone [6]. A state machine was developed to represent the driver's current manoeuvre using bump analysis. Thresholds were empirically determined to tune the bump detection algorithm. Horizontal displacement calculated by dead reckoning the gyro signal is used as a feature to distinguish between single lane changes, multiple lane changes, curved roads, and intersection turns. Heading angle displacement calculated by dead reckoning the gyro signal was also used to distinguish between intersection turns and U-turns. The accuracy of this approach was reported to be 100% for turns, 93% for lane changes and 97% for curvy roads. No false positive rates were reported. Only 2 routes were reported in their results, which may have been overfitted through extensive tuning and may not generalize well.

Machine learning algorithms have also been employed for manoeuvre classification. A study by Di Lecce and Calabrese introduces a neural network (NN) to label longitudinal acceleration and turns using a GPS and biaxial accelerometer [7]. Seven regions are defined to represent different acceleration conditions using a scatter plot of the biaxial accelerations. Their method uses a tuple of lateral/longitudinal accelerations, velocity, radius of curvature, and orientation sampled at 1 Hz for 3 seconds. Learning techniques such as the standard gradient descent, variable learning-rate gradient descent, Levenberg Marquardt, and Bayesian Regularization were employed during training. The best classifier observed was a 2 layer NN trained using the LM back-propagation, achieving an accuracy of 88% after 200 epochs.

C. Woo and D. Kulić are in the Electrical and Computer Engineering Department, University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1

This research was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and Intelligent Mechatronic Systems (IMS).

978-1-5090-1820-8/16/$31.00 ©2016 IEEE

A study conducted by Johnson and Trivedi introduces a Dynamic Time Warping (DTW) algorithm to detect Lane Changes and Intersection Turns using IMU and GPS sensors on a smartphone [8]. Their method uses a moving average filter to first threshold the rotational energy. Measurements were taken from the accelerometer, gyro, and magnetometer.

It was observed that classifiers trained on the accelerometer and gyroscope features obtained an accuracy of 77% and 79% respectively. After combining features from all 3 sensors, the accuracy improved to 91% with a false positive rate of about 5%. This result demonstrates that a combination of IMU sensors will help in achieve better classification results.

Using the smartphone's camera to obtain a video stream has also been proposed in several studies [9] [10]. However as noted in [6], cameras have limitations as they depend on visibility. Bad lighting at night, low visibility weather with rain and snow, pavement wear, and poor camera placement all contribute to lowering the effectiveness of camera dependent solutions.

Classifiers in past works have attempted to perform segmentation on manoeuvres such as accelerations, turns, and lane changes. In this paper we focus on segmentation of stop, acceleration, deceleration, left turn, and right turn manoeuvres.

In this study, a method is proposed to first use a simulated car kinematic model as training data for a classifier. Since only simulation data is used, there is no need to collect training data and training data can be generated with minimal computation. To input features for classifier training a sliding window is applied on the simulated sensors and PCA is used for dimensionality reduction. The simulation ground truth states provided supervised training targets to train an SVM. Measurements were performed on a smartphone to collect test data. A novel method applying PCA to the gyroscope signal was used to calibrate the gyroscope's rotation matrix. Our approach allows phone orientation detection in the vehicle.

The remainder of the paper is organized as follows: Section II details methods used to generate training data, preprocess data, and train our classifier. Our methods for obtaining test measurements are described in Section III. Classifier performance is presented in Section IV. Concluding remarks and future work are made in Section V.

II. Methodology

Measurements simulating mobile phone sensors are generated via simulation and features are created using a sliding window technique. Since data in the sliding windows are heavily correlated [11], PCA is used as a feature reduction technique to pre-process features prior to training. A SVM classifier is trained using the reduced features and known labels. To reduce the sensitivity to phone orientation, a new technique is introduced to determine the phone to car rotation matrix when the phone is mounted on a vehicle.

TABLE I
Randomized simulation parameters

Random Variable	Mean (μ)	Std Dev (σ)
Speed Limit	15	2
Turn/Lane Change Acceleration	0.5	0.25
Lane Change Velocity	5.7	0.4
Intersection Turn Velocity	5.7	0.4

A. Simulation

The simulation emulates a trial of a vehicle in normal traffic conditions for 1500 seconds (25 minutes) with a model update rate of 100 Hz totalling 150,000 data points per trial. In a normal driving scenario on the road, the driver's main focus is controlling the steering and acceleration. When driving in a single lane, the steering should centre the car in the lane. However, the acceleration control is situation dependent as the driver needs to decide whether to increase velocity towards the speed limit, control velocity to follow the car in front, or slow to a stop for an intersection or turn. To emulate the transition between the various manoeuvres, the driver's decisions are modelled using a Markov Decision Process (MDP). The speed limit was set to about 15 $\frac{m}{s}$ which is equivalent to 54 $\frac{km}{h}$. This speed was selected as it was close to Ontario's standard speed limit of 50 $\frac{km}{h}$ if no speed limits are posted [12] with an added 10% tolerance for added speed. The resulting MDP is illustrated in Figure 1. This technique to generate the ground truth targets for classifier training has an advantage over training with real data as the ground truth of the manoeuvre is known during the simulation and therefore no manual labelling is required.

Once the manoeuvre is selected, it should be emulated by the car's simulation model. A proportional controller was developed (Equation 1) to control the car velocity during lane keeping manoeuvres. The acceleration (a) is calculated as the product of the controller gain k and the difference between the desired velocity (v_{ref}) and the current velocity (v). Past works characterizing lane changes [13] [14] and intersection turns [6] were used to generate open loop signals for respective manoeuvres as seen in Figure 2. Control parameters for lane keeping velocities (v_{ref}), lane change times (t_{lane}), intersection turn times (t_{turn}) were randomized over a Gaussian distribution tabulated in Table I. Gaussian noise was added to simulate sensor noise. Road curvature to replicate curvy roads, observed in trips described in Section III, was simulated by adding random Gaussian PDFs to non-stop portions of the signal. The simulation updated the model and generated observations at 100 Hz.

$$a = k_v * (v_{ref} - v) \tag{1}$$

B. Measurement Calibration

The aim of this research is to identify driver manoeuvres from a mobile device, which cannot be assumed to be located with a fixed and known orientation relative to the vehicle. A novel technique was developed to extract the rotation matrix between the mobile device and the vehicle.

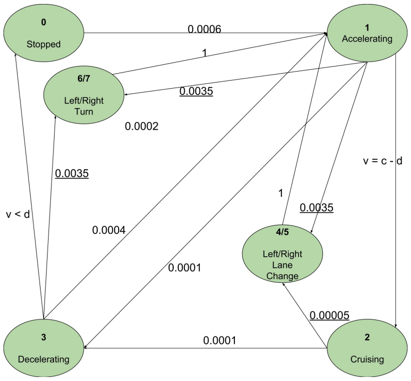

Fig. 1. Markov Decision Process to simulate driver decisions

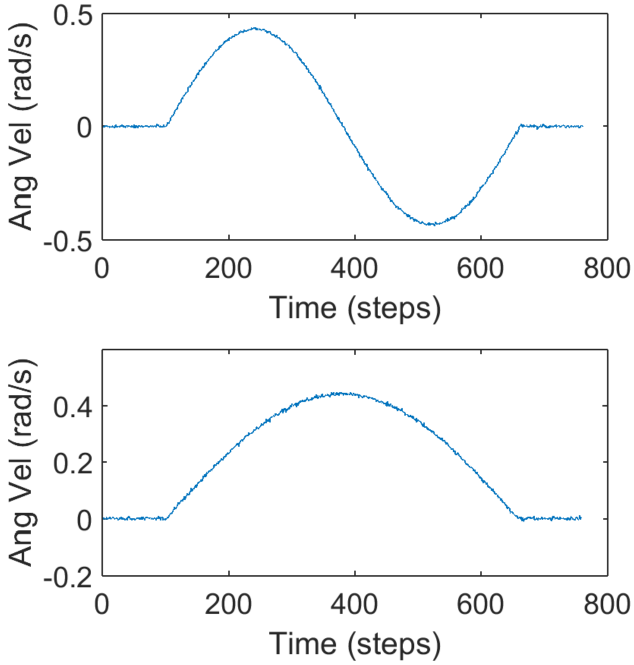

Fig. 2. Simulated Lane Change Manoeuvre (top) and Intersection Turn Manoeuvre (bottom)

Principle Component Analysis (PCA) was applied to the 3-axis gyroscope signal to calculate the directions of maximum variance in the phone frame.

We observed that the Principle Components (PC) corresponded to axes describing the motion of the vehicle in the car frame as shown in Table II. For all the trips analyzed, the highest direction of variance, i.e. PC1, corresponded to the yaw axis of the car frame, which was the axis describing turning actions. For the detection of turning manoeuvres, the angular velocity about the yaw axis of the vehicle frame is of interest.

TABLE II

PRINCIPLE COMPONENT TO AXIS MAPPING

PC (#)	Axis	Description
1	z	Yaw
2	x	Pitch
3	y	Roll

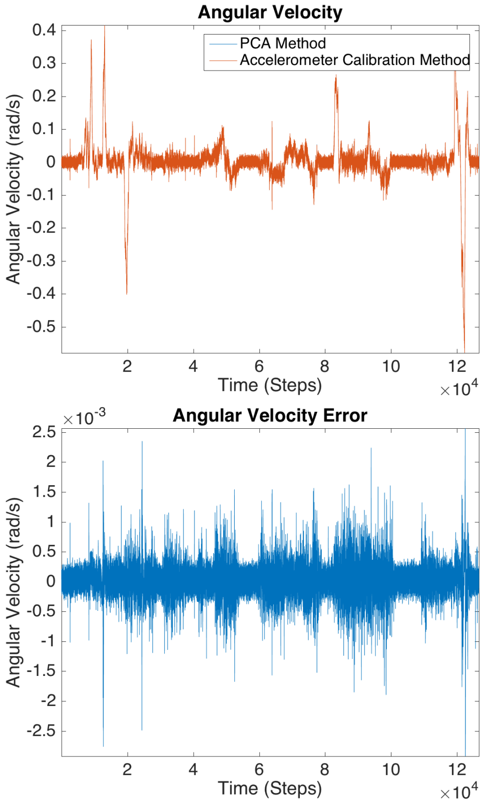

Fig. 3. Performance (top) and error (bottom) between the PCA rotated gyro signal and the acc rotated gyro signal

Since PCA produces an orthonormal rotation matrix, we can directly use this rotation matrix as the phone to car rotation matrix. The PCA algorithm produces an ambiguity in terms of direction of the rotation frame [15]. To resolve the ambiguity of the z-axis direction, one accelerometer reading was used to determine the dominant direction corresponding to gravity.

To demonstrate the proposed algorithm, we obtained measurements from a smartphone wedged at known Euler angle offsets in the vehicle during driving. Lane changes, intersection turns, and highway driving were investigated in this dataset. The phone was placed in a standard flat position to observe if any significant error was introduced using this method for 7 trips. An additional 5 trips with non-zero Euler angles were used to test the effectiveness of this procedure. For the second set of trips, the car frame yaw signals were calculated using the proposed PCA rotation matrix and the traditional accelerometer calibration approach. The performance for an exemplar trip is presented in Figure 3. RMS error between the PCA-estimated gyro signal about the vehicle yaw axis and the accelerometer based calibration for all trips was found to be 0.0026 with a SNR of 1538.1. The low error achieved demonstrates this method generalizes across different trips.

C. Features

To build the feature vector for training and classification, the velocity and angular velocity signals were selected as relevant features that could help to classify the manoeuvres. A sliding window was used on the Gyro PCA signal described in Section II-B and the vehicle forward linear velocity signal. A window size variable r was used to denote the window size in time steps as seen in Equations 2 and 3. To ensure the values from both sensor types are of comparable magnitudes during the variance computation, the data from each sensor were scaled to the maximum measurement value of the entire simulation. The velocity and angular velocity windows are then concatenated and the feature vector for time step t is presented in Equation 4.

$$V_t = \begin{bmatrix} v_{t-r} & \ldots & v_t & \ldots & v_{r+t} \end{bmatrix}^T \tag{2}$$

$$\dot{\Theta}_t \begin{bmatrix} \dot{\theta}_{t-r} & \ldots & \dot{\theta}_t & \ldots & \dot{\theta}_{t+r} \end{bmatrix}^T \tag{3}$$

$$F = \begin{bmatrix} V_t/v_{max} & \dot{\Theta}_t/\dot{\theta}_{max} \end{bmatrix}^T \tag{4}$$

A sliding window has several advantages over a single measurement. Temporal context of the signal is observable before and after the current time point. More data allows to detect situational context such as lane changes vs road curvature as investigated in [6]. The disadvantage of a sliding window is the high dimensionality and highly correlated features involved with this technique. PCA aims to reduce the dimensionality of these highly correlated features by finding the directions of maximum variance. Furthermore, different window size could be selected. To address this issue, grid search can also be employed to find optimal window size.

Elements in temporal sliding windows contain highly correlated features as signals vary by small values from one time step to the next. Feature reduction methods such as PCA can be used to reduce the feature vector's dimensionality. The 5th eigenvalue in the scree plot presented in Figure 4 was observed to be an inflection point and was used to truncate the signal to the top 5 PCs. The explained variances for the first 5 PCs are 79.95%, 7.311%, 6.006%, 3.967%, and 1.597%. The total explained variance of the first 5 PCs is 98.831%.

The first 5 PC are presented in Figure 5. PC1 has a strong bias on the velocity features, effectively taking the mean of the velocity. PC2 takes the derivative of the velocity in the left half of the features. On the right half, it has a bias to construct the average of the angular velocity and a peak at the centre of the right half to represent peaks in the angular velocity signal. PC3-PC5 are higher order derivatives of both velocity and angular velocity signals.

D. Training

The Libsvm extension for Matlab [16] is used for SVM classifier training. A grid search was used to find the optimal SVM training parameters. SHARCNET[1], was used to train SVMs using parameters linearized by the grid search's bounds. C-Support Vector Classification (C-SVC) was used as the training SVM type and a Radial Basis Function (RBF) kernel was used. Cost, γ, and sliding window size were varied as tabulated in Table III.

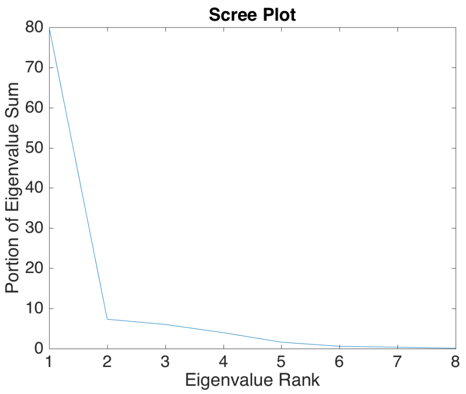

Fig. 4. Scree plot of feature vector

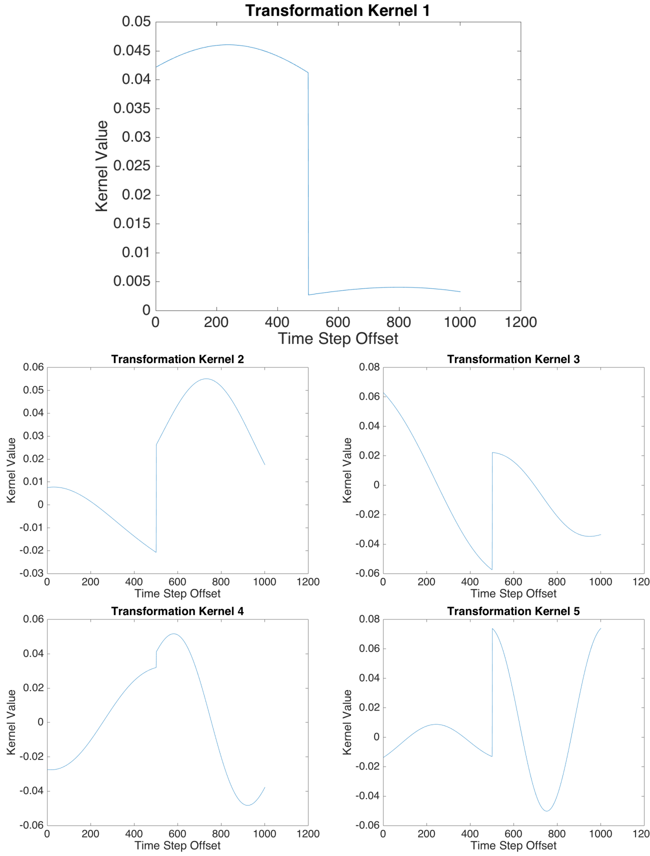

Fig. 5. Top 5 principle components of feature vector for a 5 second window. The first 500 features correspond to normalized linear velocity and the second 500 features correspond to normalized angular velocity.

III. Measurements

The proposed approach is tested with data collected from driving experiments. To collect data during driving, an Android app was developed using the Android SDK to collect GPS and IMU data for this study. The measurement types are listed in Table IV.

[1] University of Waterloo is a member of SHARCNET (www.sharcnet.ca), a consortium of universities and colleges operating a network of high-performance compute clusters in south western, central and northern Ontario.

TABLE III

PARAMETER AND RESPECTIVE GRID SEARCH

Parameter	Min	Max	Subdivisions	Subdivision Scale
Cost	10^{-7}	1	5	Log
γ	10^{-5}	1	10	Log
Sliding Window Size	0.5	5	5	Linear

TABLE IV

MEASUREMENT FROM THE CARRECORDER MOBILE APP

Measurement	Sensor	Max Update Rate (Hz)
Latitude	GPS	1
Longitude	GPS	1
Speed	GPS	1
Heading	GPS	1
Acceleration	Accelerometer	100
Angular Velocity	Gyroscope	100
Magnetic field	Magnetometer	100

In this study, we analyze 12 trips operated by one driver using one vehicle. The trips include straight and curved roads, busy and light traffic, city and suburban driving conditions. The total length of all trips combined was 53 minutes. Trips were recorded using a Samsung Galaxy S4 mobile phone running Android 4.4.2. The phone was placed in the vehicle cupholder during the data collection. The car started to operate at least 5 seconds after a GPS fix ($<$ 10m accuracy) was obtained for each trip. Since the reported update rates are the sensor's maximum rates, sensor updates vary and are not at a constant interval. To maintain a constant measurement interval and consistency of sample rates during training, the measurements were re-sampled at 100 Hz and linearly interpolated.

IV. RESULTS

The classifiers were trained using the methodology described in Section II-D and were tested on the datasets collected in Section III. Table V tabulates classifiers with the top 10 F_1 scores including parameters and performance. The performance metrics in the table are a mean of the metric aggregate for all manoeuvres.

The top performing classifiers had cost parameters between 0.017783 and 1 with a mean of 0.8035. The classifiers had γ parameters in the magnitudes 10^{-3} to 10^{-1} with a mean of 0.0764. It was observed that the top classifiers generally worked better when the range was smaller than 2.5 seconds. The window size of the top classifiers had a mean of 2.6 and a median of 2.25.

Our proposed method achieves an average F_1 score of 0.8194 across all manoeuvres. To analyze the results of the unbalanced testing dataset, the average balanced accuracy was also calculated and achieved 0.8874. Finally, precision and recall performance was 0.8158 and 0.8279 respectively. These results demonstrate the proposed approach is able to achieve good results.

TABLE V

TURN AND STRAIGHT MANOEUVRE CLASSIFICATION PERFORMANCE

F_1	BA	P	R	C	γ	r
0.8194	0.8874	0.8158	0.8279	1	0.021544	1.5
0.8018	0.8655	0.8417	0.7904	0.017783	0.077426	2.5
0.7867	0.8671	0.8072	0.7978	1	0.00046416	2.5
0.7854	0.8624	0.7958	0.7913	1	0.021544	2
0.7850	0.8651	0.8028	0.7940	1	0.27826	1
0.7758	0.8566	0.8176	0.7799	1	0.0016681	3.5
0.7758	0.8505	0.8301	0.7691	1	0.0016681	5
0.7746	0.8493	0.8322	0.7674	1	0.0059948	5
0.7744	0.8575	0.7937	0.7824	0.017783	0.27826	1
0.7742	0.8550	0.8160	0.7764	1	0.077426	2

TABLE VI

CONFUSION MATRIX OF THE OPTIMAL CLASSIFIER

	Stop	Acc	Dec	L Turn	R Turn
Stop	0.9995	0.0002	0	0	0.0003
Acc	0.0803	0.6651	0.1868	0.0193	0.0486
Dec	0.0639	0.1405	0.7494	0.0328	0.0133
L Turn	0.0369	0.0422	0.0388	0.8739	0.0083
R Turn	0.0362	0.0415	0.0700	0.0006	0.8517

The confusion matrix for the top classifier is illustrated in Table VI. In the table, the rows are the ground truth and the columns are the classes predicted by the classifier. Recall rates for turns were found 0.8739 for left turns and 0.8517 for right turns. These results achieve similar performance when compared with [7] and [8], which use non-simulation data to train their classifiers. Despite obtaining a lower recall rate compared to [6], their study may not generalize well as only 2 routes were used, which may have led to overfitted time and signal thresholds. In this study, each of our tests used different routes providing results for more generalized naturalistic driving.

It was observed that the classification performance for each manoeuvre was sensitive to the γ parameter. From the trained classifiers, it was observed that top straight manoeuvre classifiers used a larger γ $(10^{-3} - 1)$ and a larger window $(1 - 2.5$ seconds). Intersection turns were observed to perform better with a lower γ $(10^{-5} - 10^{-3})$ and a smaller window $(0.5 - 2$ seconds).

The different γ value leads to the classifier performing well on one type of manoeuvre but not the other or the classifier performs poorly on both. This trend as well as the range of γ parameters in the overall top performers indicate the performance metric is prioritizing straight manoeuvres over turn manoeuvres. This explains the confusion between intersection turn and straight manoeuvres.

A second observation is that acceleration and deceleration are confused with stop manoeuvres. When reviewing the labelled data, the confusion with these labels were found to be near transitions between accelerating, decelerating, and when the car shifts forward near an intersection.

Finally, it was observed that left turns consistently achieved better classification performance than right turns. Despite higher angular velocity amplitudes, left turn manoeuvres tend to take longer to execute allowing them to be more temporally distinguishable.

V. CONCLUSIONS AND FUTURE WORK

In this paper, a method was proposed to use a 2D car kinematics simulation as training data for a manoeuvre classifier. Using this data, a sliding window and PCA was used to obtain input features for classifier training. The simulation ground truth states provided supervised training targets to train an SVM, removing the need for costly training data collection and tedious and error prone manual labelling. A grid search was employed for parameter tuning. We are able to account for arbitrary phone orientations by rotating the gyroscopic signals into PCs corresponding to the principal directions of turning.

The scree plot indicated the top 5 PCs should be used when using PCA for a 5 second window. The classifier was found to have a 0.8158 average precision rate and a 0.8279 average recall rate across all manoeuvres resulting in an average F_1 score of 0.8194 on the dataset.

In this study, promising accuracy rates were achieved when classifying multi-class manoeuvres using only simulated data for training. Simple features such as velocity and angular velocity are useful for identifying driving manoeuvres. These results will help support further research into manoeuvre segmentation and analysis of driver behaviour.

There are limitations to using the PCA method to estimate the directions of turning for the gyroscope. The proposed method allows the mobile phone to be placed in an arbitrary position and orientation with respect to the vehicle, but assumes that the phone remains in a constant pose with respect to the vehicle for the duration of the trip. If the phone moves during the trip, the principal directions of turning would need to be re-estimated.

A few aspects of the proposed algorithm can be improved. The simulator may increase transition probabilities between manoeuvres to simulate velocity peaks and generate training data for manoeuvres that occur in quick succession. Trip data may also be used to train the classifiers as a baseline comparison. The method should also be more extensively tested with data from multiple drivers with different driving styles.

Libsvm contains additional parameters to tune the classifier training. For instance, varying the weighted costs of each class when training the SVM in the Libsvm package may improve performance and mitigate the unbalanced data problem mentioned in Section IV. Additional features can also be added to evaluate if they improve on these results. Features such as radius of curvature can be calculated [7] and input as a feature to the SVM. Another potentially useful feature is the magnetometer sensor which could be combined with velocity and gyroscope similar to [8].

Highway and non-highway driving behaviours are different according to [17]. The work investigates driver behaviour classification at different speeds. Training a highway vs. non-highway classifier will support distinguishing between these two scenarios. As mentioned in their work, on- and off-ramp behaviour should also be analyzed.

Finally, the feasibility of using hierarchical classifiers or ensemble methods can be investigated. Hierarchical classifiers may account for any behavioural changes between scenarios. Furthermore, ensemble methods may be tuned for specific manoeuvres allowing certain classifiers to specialize in a subset of manoeuvres (i.e. turning or straight manoeuvres).

References

[1] S. Klauer, T. a. Dingus, V. L. Neale, J. Sudweeks, and D. Ramsey, "The Impact of Driver Inattention On Near Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data," *Analysis*, no. April, p. 226, 2006.

[2] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving Behavior Analysis with Smartphones : Insights from a Controlled Field Study," 2012.

[3] Transport Canada, "Road Safety in Canada," 2011.

[4] J. Almazan, L. M. Bergasa, J. J. Yebes, R. Barea, and R. Arroyo, "Full auto-calibration of a smartphone on board a vehicle using IMU and GPS embedded sensors," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 1374–1380, 2013.

[5] H. Liu and G. Pang, "Evaluation of RTK-GPS Performance with Low-cost Single-frequency GPS Receivers," in *Proceedings of IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, Tokyo, Japan, oct 1999, pp. 435–440.

[6] D. Chen, K.-t. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible Sensing of Vehicle Steering with Smartphones," *Mobile Systems, Application and Services*, pp. 1–13, 2015.

[7] V. Di Lecce and M. Calabrese, "NN-based measurements for driving pattern classification," *2009 IEEE Intrumentation and Measurement Technology Conference*, no. May, pp. 259–264, may 2009.

[8] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1609–1615, 2011.

[9] P. Liu, A. Kurt, and U. Ozguner, "Trajectory Prediction of a Lane Changing Vehicle Based on Driver Behavior Estimation and Classification," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 942–947.

[10] J. McCall, D. Wipf, M. Trivedi, and B. Rao, "Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 431–440, sep 2007.

[11] T. G. Dietterich, "Machine Learning for Sequential Data: A Review," Tech. Rep., 2002.

[12] Ontario Ministry of Transportation. (2010, jun) Driving The Speed Limit.

[13] W. van Winsum, D. de Waard, and K. A. Brookhuis, "Lane change manoeuvres and safety margins," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 3, pp. 139–149, sep 1999.

[14] H. M. Mandalia and D. D. Salvucci, "Using Support Vector Machines for Lane-Change Detection," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 22, pp. 1965–1969, sep 2005.

[15] M. Tipping and C. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[17] J.-H. Hong, B. Margines, and A. K. Dey, "A Smartphone-based Sensing Platform to Model Aggressive Driving Behaviors," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14, pp. 4047–4056.