Real Time Pavement Classification Using an Embedded Neural Network

Filipe do Ó Cavalcanti, Waslon T. A. Lopes and Fabrício B. S. de Carvalho

***Abstract*— The identification of the pavement can be an important feature for the automotive market, as it impacts on different characteristics of the vehicle and its navigability. Considering the importance to identify the pavement, this paper describes the design and implementation of a system capable of data acquisition and real time classification of two types of road pavement: asphalt and paving stone. The system is based on a real-time operating system that polls data from a triaxial accelerometer and GPS at a fixed frequency and offloads it to a computer. An Artificial Neural Network is trained in Python with 92% accuracy and the model is exported to the embedded system for real-time classification while driving.**

***Keywords*— Pavement Classification, RTOS, Accelerometer, Artificial Neural Network, Embedded systems.**

I. Introduction

Modern vehicles have a rigorous testing process where all aspects of their behavior are analyzed: from idle vibrations to bumps in gear change, accelerometers are used in conjunction with the data bus for overall analysis of the vehicle under stress conditions. This activity requires the vehicle to be travelling in a smooth surface, so that captured vibrations are associated only to the vehicle and not to the test track. On the other hand, accelerometers are also used to classify roads and road quality.

In Brazil, it is common to use paving stone due to low cost installation and maintenance [1]. This type of pavement is not as comfortable as asphalt due to excessive vibrations which are transferred across the entire suspension to the driver and passengers, while forcing the vehicle to drive at slower speed to avoid damaging the vehicle. By analyzing this vibration data, it is possible to map where this type of pavement is present on a city, classify road quality or even control a vehicle adaptive suspension.

In this context, the main goal of this paper is to develop an embedded application to classify roads between asphalt and paving stone in real time, using a triaxial accelerometer as input and an Artificial Neural Network (ANN) model as the classifier.

The rest of this paper is organized as follows: Section II analyzes some of the similar works by other authors. Section III describes the equipments necessary for the proposed system. In Section IV the implementation of the peripherals, the data flow in the real-time operating system (RTOS) and the test setup are discussed. Section V describes the results from the data acquisition and training of the ANN that will be used for real time prediction. Finally, Section VI is devoted to the conclusion.

Filipe do Ó Cavalcanti, Waslon T. A. Lopes and Fabrício B. S. de Carvalho, Programa de Pós-Graduação em Engenharia Elétrica, Centro de Energias Alternativas e Renováveis, Universidade Federal da Paraíba (UFPB), João Pessoa-PB. E-mails: filipe.cavalcanti@cear.ufpb.br, waslon@cear.ufpb.br, fabricio@cear.ufpb.br.

II. Literature Review

The work in [2] uses a Support Vector Machine (SVM) to classify the road among seven types of pavement, using only an accelerometer on the vertical axis that is coupled to a small transport cart limited to a speed of 1 m/s. The work compares statistical metrics of the temporal data to metrics obtained from the frequency domain, such as Power Spectral Density (PSD) and the Fast Fourier Transform (FFT). The authors conclude that training the model for seven types of pavement there are some improvements when using metrics from both frequency and time domains. However for only three types, there is no advantage and it is better to use temporal metrics only.

Some papers try to detect the occurence of potholes or general anomalies instead of using an Artificial Intelligence (AI) implementation for a more detailed classification of the terrain. The approach in [3] is focused on detecting potholes and compares some techniques used by other authors. The Z-THRESH, Z-DIFF and STDDEV algorithms were compared to the proposed G-Zero algorithm. Z-THRESH will detect an anomaly when the Z axis acceleration goes above a certain threshold. Z-DIFF will seeks two consecutive values with the greatest difference in amplitude, showing when a sudden change in acceleration occurs. STDEV evaluates the standard deviation in a set of data and it will classify a pothole if it crosses a threshold. In addition to the three methods mentioned above, the proposed G-Zero algorithm analyzes moments where the three accelerometer values approach 0 g, which can be an indication of a free fall. Among the analysis, it was stated that Z-DIFF provides the best performance when comparing the rate of true positives.

Similar to the approach in [3], the work on [4] seeks to identify anomalies on a surface. The data was obtained from accelerometers available in smartphones. The author uses SVM and compares it with the techniques proposed by others authors. The comparison occurs in 30 different streets and it is concluded that the proposed SVM has better F1 score but the STDEV technique proposed in [3] shows better result among the analyzed works.

A Z axis accelerometer was used on a truck and compared a Artificial Neural Network (ANN), SVM and Naive Bayes for terrain classification, while searching for a speed independent classification model. This work also used a vehicle model for simulation [5]. On different terrains, data acquisitions at speeds

of 20, 30 and 40 km/h were performed and two main metrics were compared: FFT and Fast Wavelet Transform (FWT). The work suggests that if a classification independent of speed is desired, the metrics generated from the FWT will yield a better output.

In addition to the use of triaxial accelerometer, the work on [6] also has a triaxial gyroscope and geolocation by the use of the Global Positioning System (GPS). The proposal detects anomalies on the ground up to 2 cm long but makes an analysis based on length of asphalt, where every 100 m provides data for a classification. The proposed classification method is based on the Pavement Condition Index (PCI), an index heavily dependent on manual survey of the pavement condition where a grade is assigned to a length of asphalt that indicates its quality. By using this equipment, the work creates an equivalent scale to the PCI proposal. This approach provides different metrics from the ones proposed in other works, such as the peak of the moving variance.

In a previous work [7], SVM classifier was developed to classify between two types of pavement: asphalt and paving stone. The features contain metrics in time and frequency domain for classification with precision up to 97%. Now, this work seeks to continue what was previously implemented, aiming at increased robustness of the acquisition system. The triaxial accelerometer is kept in its position near the center of gravity of the vehicle and a new implementation in the embedded system is developed: a real-time operating system (RTOS) is used to ensure data acquisition accuracy and an ANN is trained to also be embedded in this system, allowing the pavement classification in real time.

III. Equipment and Tools

The proposed system is composed by a GPS module, a triaxial accelerometer and an ARM Cortex-M4 development board.

The GPS Module is the ublox NEO 6M, a cost-effective device that is used during data acqusition by polling current geolocation data twice a second. The data is received via UART communication in the NMEA 0183 format. Acceleration measurements from X, Y and Z axis are obtained from the MPU6050 at a sampling rate of 100 Hz. This device contains a 16 bit Analog-to-digital converter (ADC) in each axis, which can be setup in a configurable working range from $\pm$ 2 g to $\pm$ 16 g. The communication is implemented using an Inter-Integrated Circuit (I2C) bus. All information is processed on a EK-TM4C1294XL development board that has an ARM Cortex-M4 operating with a clock frequency of 120 MHz. This microcontroller will be carrying the Free RTOS kernel for the implementation of a RTOS. Beyond the main devices, there is also a push button for user input when necessary and some LEDs are used for status monitoring. All data is sent to a laptop via serial communication.

IV. System Overview

A. GPS Data Request

The proposed architecture interfaces with 4 external devices. The GPS module and laptop communication are based on UART, configured for a baud rate of 11520 bits/s. The accelerometer requires I2C communication and the external pushbutton is connected to a General Purporse Input-Output (GPIO) pin and triggers an interrupt handler.

The GPS data is collected by using a polling operation requested by the Offload Task which happens two times per second. A GPS message is polled by writing the UBX,00 message on the serial bus and a NMEA formatted message will be returned containing the most recent data position available. The message contains 23 data fields which includes the required information, such as UTC Time (field 2), latitude and longitude (fields 3 to 6) and speed in km/h (field 11).

Once the polling returns data, the GPS Read Task will receive all data through the serial bus and the Offload Task will be temporarily suspended to ensure all information is correctly received. Upon receiving the end of message control sequence, the Offload Task resumes and the GPS Parse task is triggered. The GPS Parsing is a lower priority task and it is responsible for parsing the data fields and calculating the checksum before making this new position data available for the Offload Task.

Data reading and parsing from the GPS were separated in two tasks due to importance and different requirements: the GPS data is not time sensitive in the post processing and a 0.5 seconds delay will not influence on the data analysis, since the main information to be extracted is the vehicle speed.

B. Accelerometer Data

The most time critical implementation of this system is the Accelerometer reading. The Accelerometer Task has the same priority as the GPS Read Task. However, it is triggered at a constant rate of 100 Hz. Each time this task is called, the X, Y and Z axis acceleration are read and put into the queue to be used by the Offload Task. Accelerometer reading is carried using the I2C bus and a burst read sequence of 6 registers (two for each axis). Each register read returns 8 bits of data and a complete axis read will be a 16 bit float value in two's complement.

C. Data Offload

Data offloading to the laptop is executed via UART and each line of data sent consists of ten data fields in a tabulated format. The offload operation happens when an accelerometer read is available, which is 100 times per second. The GPS data is updated at a much slower rate, so the latest available data is repeatedly sent. The buffer variable that allocates all data described in Table I has a fixed size of 85 bytes, which includes all accelerometer data, GPS data, sample number, general tabulation and plus and minus signs. At a frequency of 100 Hz this requires a UART connection capable of 68000 bits/s for adequate message throughput. The UART baud rate is set at 115200 bits/s, which is more than enough to accomodate a sampling frequency up to 150 Hz. An example of the data received by the laptop is available on Figure 1.

D. Dataflow between Tasks

Moving data between tasks can be a complex process if global variables are used. To simplify this operation, the Free

TABLE I

ACQUISITION MODE MESSAGE HEADERS.

Header	Description
Sample	Number of acquisitions. Can be reset by pressing the pushbutton.
AccX	Acceleration on the longitudinal axis.
AccY	Acceleration on the lateral axis.
Accz	Acceleration on the vertical axis.
Terrain	Actual terrain the vehicle is driving.
Lat	Latitude information.
DirLat	North/South indicator.
Long	Longitude information.
DirLong	East/West indicator.
Spd	Speed over ground in km/h.

File Edit View Search Terminal Help

Sample	AccX	AccY	AccZ	Terrain	Lat	DirL	Long	DirL	Spd
1	0.0130	0.0001	1.0387	0	2254.0461	S	04703.233	W	0.220
2	0.0115	-0.0138	1.0150	0	2254.0461	S	04703.233	W	0.220
3	0.0264	-0.0119	1.0091	0	2254.0461	S	04703.233	W	0.220
4	0.0166	-0.0260	1.0238	0	2254.0461	S	04703.233	W	0.220
5	0.0227	-0.0229	1.0362	0	2254.0461	S	04703.233	W	0.220
6	0.0152	-0.0258	1.0355	0	2254.0461	S	04703.233	W	0.220
7	0.0374	-0.0153	1.0138	0	2254.0461	S	04703.233	W	0.220
8	0.0164	0.0011	1.0113	0	2254.0461	S	04703.233	W	0.220

Fig. 1. Example of a data acquisition file.

RTOS API provides queues, which are simple message buffers in a First in, First Out (FIFO) format that can be used to transmit messages between tasks [8]. Each queue has a fixed number of items that it can hold and it will allocate enough RAM to fit those items based on the data types it will carry. When data is inserted in the queue, it will be allocated to the first available slot. To retrieve data from the queue, a task can poll the queue receive function and evaluate if a message is available to be read. Reading a message requires that data will be copied from the queue to a variable and when this happens, the received message will be removed from the queue, clearing space for new messages.

The proposed system requires three queues for data acquisition which can be seen in Figure 2. The common queue between those modes is the accelerometer queue that moves X, Y and Z acceleration samples from Accelerometer Task to the Offload Task. The other queues are the GPS Read queue that moves the raw GPS data from the GPS Read Task to the GPS Parse task and finally the GPS Ready queue, that will move the parsed geolocation data from the GPS Parse Task to the UART Offload Task.

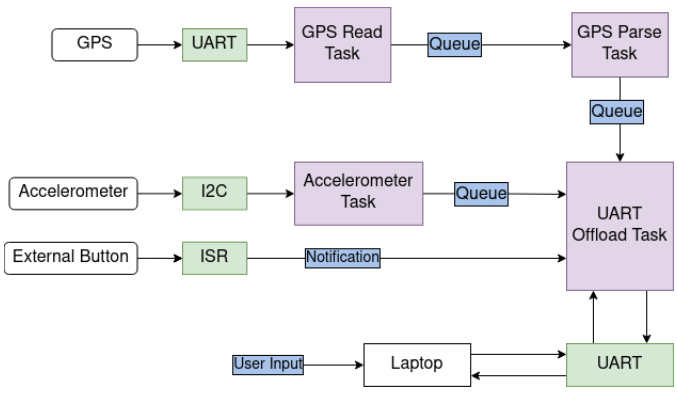

Fig. 2. Proposed dataflow for the acquisition mode.

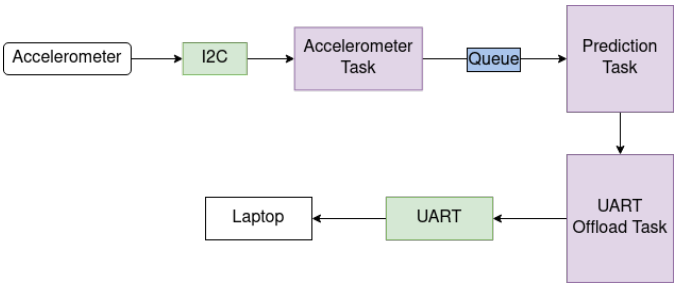

Fig. 3. Proposed dataflow for the prediction mode.

Figure 3 shows how data is moved in the prediction mode. It does not require the GPS task since speed or location are not features used for pavement classification and no user input is necessary. All prediction results are shown on the laptop screen.

E. Prediction

When compiled in prediction mode, the system will acquire 128 samples that are written in an array for each accelerometer axis. By acquiring 128 samples, the classification procedure is triggered, and the arrays are passed to a function that will calculate the required metrics for prediction. Only after the metrics are available, the prediction will take place.

Classification of the terrain requires heavy mathematical processing for metrics and feedforward of the multi-layer perceptron. Some metrics used were defined in [7] for a Support Vector Machine. This work required an ANN due to easier implementation in an embedded scenario and to achieve this, the training was executed in a computer and used the following features, which are extracted from each second of acquisition: standard deviation of the Z axis moving average (as proposed in [3]), peak of the rolling variance [6], maximum Z axis peak-to-peak amplitude, low frequency (0-15 Hz) and high frequency (16-50 Hz) FFT amplitude average. Those metrics show good accuracy and are enough for a proof of concept of the real time classification.

The algorithm for classification was built using the GNU Scientific Library (GSL) [9] that has all the mathematical functions necessary to calculate the features that will be used in classification. The advantage of using GSL is that it provides the necessary mathematical functions and also dynamic allocated vector and matrices which can be helpful when dealing with large arrays and matrices.

The output of the classification is returned as a floating point value representing the confidence in classification, where values close to zero represent asphalt and values close to one represent paving stone.

F. Test Setup

A standardized setup was developed to put all peripherals into a single board. It contains a slot for accelerometer placement, GPS connector and push button. The prototype on Figure 4 was used to validate that the system was working before mounting it on the vehicle.

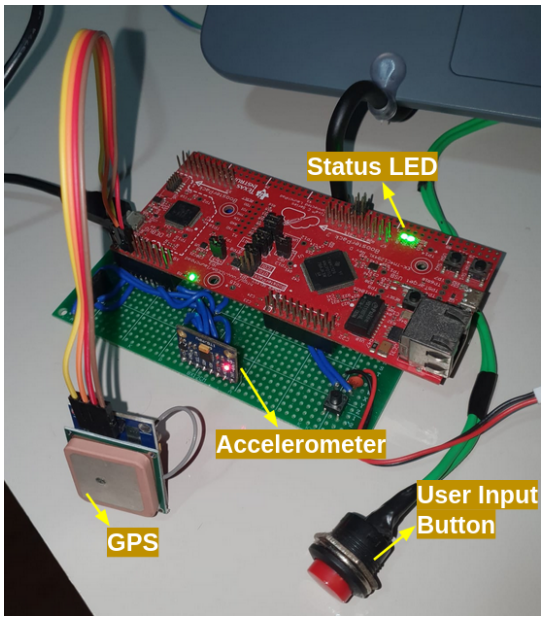

Fig. 4. Setup for validation before testing in vehicle.

V. Results and Discussion

After all tests were executed in the test bench, the system was ready to be installed in the vehicle. The mouting point is the passenger seat rail due to its proximity to the center of gravity and direct connection to the vehicle chassis [7]. A calibration procedure is necessary before starting acquisition due to the placement of the accelerometers XY plane not being fully parallel to the ground. This procedure requires that the vehicle be in a planar surface and it will acquire a few seconds of accelerometer data, which will be averaged and an offset will be calculated. This offset will set a virtual ground reference that should read close to 0 g in the X and Y axis and -1 g on the Z axis.

A. Data Acquisition

A total of 38 minutes were recorded in the region of Cambuí, in Campinas, state of São Paulo, Brazil. The total amount of collected data was 14 MB containing the acceleration, terrain, speed and position of each sample. Statistics on each type of terrain is available in Table II, where the last three columns represent the actual terrain where the acquisition took place (Invalid represents conditions such as vehicle stopped or speed bump).

TABLE II
Statistics of each dataset obtained during a 38 minute acquisition session.

Dataset #	Time [min]	Paving Stone	Asphalt	Invalid
1	5.46	0.00%	95.60%	4.40%
2	3.79	0.00%	73.70%	26.30%
3	4.61	51.30%	35.20%	13.50%
4	3.01	6.70%	84.70%	8.60%
5	8.83	43.90%	36.60%	19.50%
6	2.15	0.00%	66.20%	33.80%
7	10.54	25.60%	39.20%	35.20%

The obtained data was processed in Python and generated a total of 840 metrics from 3 files that contained over 25% of samples in paving stone. Among them, 737 were actually used after removing unwanted metrics when the speed was below 10 km/h or above 50 km/h.

B. ANN Training and real-time classification

A simplified ANN was trained to be used in the proposed system. Older acquisition data obtained in [7] were used to test this model and it showed prominent results. This data was shuffled and split 80% for training and 20% for testing. Figure 5 shows the metrics defined in Section IV-E on the top plot (where Z_Max_Pk2pk was scaled down tenfold for better visualization) and the prediction on the bottom, where the black line is the actual terrain and the pink line is the predicted terrain. The Predicted Terrain plot shows that the predicted value contains some peaks that could be due to many factors such as potholes or the vehicle speed. Filtering the metrics can have a positive effect on those outliers.

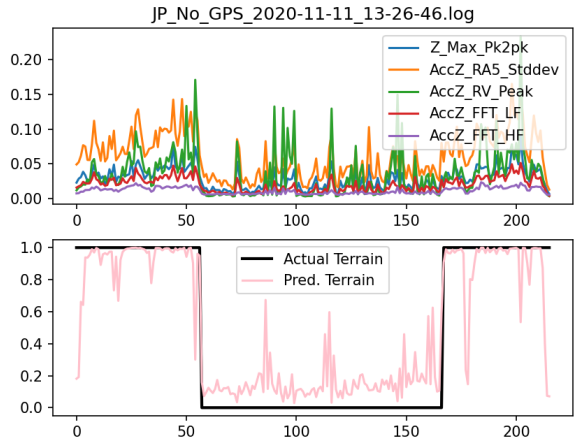

Fig. 5. Prediction graph of a dataset.

The ANN model uses the five metrics displayed on the top plot of Figure 5 as input, four hidden layers, an output layer with a total of 44 neurons and learning rate of 0.001. The Keras framework is used to develop and train the model, which used binary cross-entropy as loss function, *ReLu* activation on all layers except the output layer, which used *Sigmoid*.

The weights and biases of this model were exported to a header file that was imported by the firmware of the proposed system to be used in real time classification. Figure 6 shows a screenshot of the real time classification while driving, where each "Pred" field represents the confidence in a classification of the last 128 samples, where values close to 0 are asphalt and close to 1 are paving stone. This data is not available to post-processing due to the current implementation not being able to export the real-time classification data.

VI. Conclusions

This work presented the implementation of a RTOS capable of classifying road pavements based on accelerometer

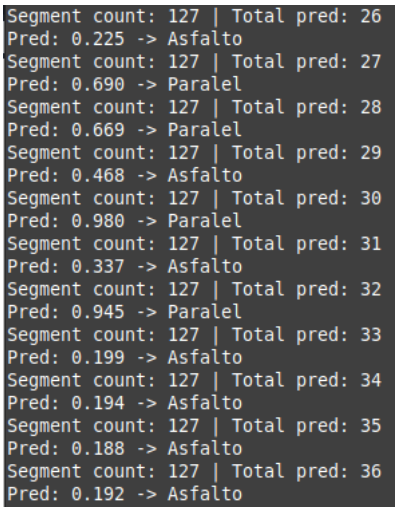

Fig. 6. Screenshot of the prediction mode.

readings. The system can work in two different modes: one capable of acquiring accelerometer data, GPS data and terrain for dataset generation while offloading the data to a computer and the other mode classifies accelerometer data in two types of pavement: asphalt or paving stone. This requires the ANN to be trained using data acquired by the acquisition mode. This work was validated by mounting the accelerometer in the passengers' seat rail of a vehicle and acquiring over 30 minutes of data. The data obtained was processed in Python and used as the training set for the ANN that would later be exported to this system, making it capable of real-time classification.

In future works, some improvements will be executed on the real-time classification algorithm for faster analysis of the terrain and better metrics. The firmware must also be upgraded to have both real-time classification and data logging capabilities (currently real-time data is not saved), allowing data to be post-processed and analyzed in a computer. This implementation has the potential to improve the performance of the ANN and allows for more terrains to be classified, such as dirt or even speed bumps. Using an ANN was necessary due to its simplicity to implement in an embedded environment when compared to the SVM. However, comparing the current implementation to the SVM in [7] requires improvements on the current classification method. This system can be expanded to classify asphalt quality by analyzing the prediction output on false positives, where an asphalt road is misclassified as paving stone due to its quality.

ACKNOWLEDGMENT

The authors would like to express their thanks to Coordination for the Improvement of Higher Education Personnel (CAPES) and to National Council for Scientific and Technological Develpoment (CNPq) for the financial support of this work.

REFERENCES

[1] M. B. Megier, J. R. Hammarstron, B. M. D. Quevedo, R. D. Palmeira, and B. F. Finckler, "Análise comparativa de pavimento asfáltico, pavimento em alvenaria poliédrica e pavimento intertravado em bloco de concreto," *2018: Salão do Conhecimento UNIJUÍ*, 2018.

[2] C. Weiss, H. Frohlich, and A. Zell, "Vibration-based terrain classification using support vector machines," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4429–4434, 2006.

[3] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using android smartphones with accelerometers," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pp. 1–6, 2011.

[4] M. R. Carlos, M. E. Aragón, L. C. González, H. J. Escalante, and F. Martínez, "Evaluation of detection approaches for road anomalies based on accelerometer readings—addressing who's who," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3334–3343, 2018.

[5] S. Wang, R. Khushaba, and S. Kodagoda, "Towards speed-independent road-type classification," in *2012 12th International Conference on Control Automation Robotics Vision (ICARCV)*, pp. 614–619, 2012.

[6] M. Meocci, V. Branzi, and A. Sangiovanni, "An innovative approach for high-performance road pavement monitoring using black box," in *Journal of Civil Structural Health Monitoring (JCSHM)*, vol. 11, p. 485–506, 2021.

[7] F. Cavalcanti, W. T. A. Lopes, and F. B. S. de Carvalho, "Sistema classificador de pavimentação utilizando acelerômetros e máquina de vetores de suporte," *XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2021)*, 2021.

[8] D. Ibrahim, *ARM-Based Microcontroller Multitasking Projects*. Newnes.

[9] M. Galassi et al, *GNU Scientific Library Reference Manual (3rd Ed.)*.