Acoustics based Terrain Classification for Legged Robots

Joshua Christie[1], Navinda Kottege[2]

***Abstract*— Legged robots offer a more versatile solution to traversing outdoor uneven terrain compared to their wheeled and tracked counterparts. They also provide a unique opportunity to perceive the terrain-robot interactions by listening to the sounds generated during locomotion. Legged robots such as hexapod robots produce rich acoustic information for each gait cycle which includes the foot fall sounds and feet pushing on the terrain (support phase), as well as the sounds produced when the feet travel through the air (stride phase). Interpreting this information to perceive the terrain it is traversing makes available another valuable sensing modality which can feed in to higher level systems to facilitate robust and efficient navigation through unknown terrain. We present an online real-time terrain classification system for legged robots that utilise features from the acoustic signals produced during locomotion. A 32-dimensional feature vector extracted from acoustic data recorded using an on-board microphone was fed in to a multi-class Support Vector Machine (SVM). The SVM was trained on 7 different terrain types and the results of the experimental evaluations are presented. The system was implemented using the Robotic Operating System (ROS) for real-time terrain classification. A classification time-resolution of 1 s was achieved by capturing acoustic signals of two steps, and the results show a true positive rate (sensitivity) of up to 92.9%. We also present a noise subtraction technique which removes servo noise and improves the sensitivity up to 95.1%.**

I. Introduction

Legged robots have the capability of traversing difficult outdoor terrain which can prove challenging to wheeled or tracked robots [1]. In many scenarios, these robots are required to operate on terrain with minimal or no prior information. Given that legged robots have the ability to adapt their gait patterns to different terrain types to maintain efficient locomotion [2], [3], the ability to perform terrain classification is important [4]. Among other modalities, acoustic sensing has been successfully used to perceive robot-terrain interactions [5]. For humans, listening to the sound of our foot steps is an intuitive way of obtaining information about what type of terrain is underfoot, even in the dark. Gathering inspiration from this as well as building upon existing work [6], [4], this paper presents an online real-time terrain classification system for legged robots using acoustic features. We experimentally evaluate the proposed system and present cross validation results showing a high true positive rate (sensitivity). We also present a noise subtraction technique which removes the servo noise and further improves the sensitivity of the system.

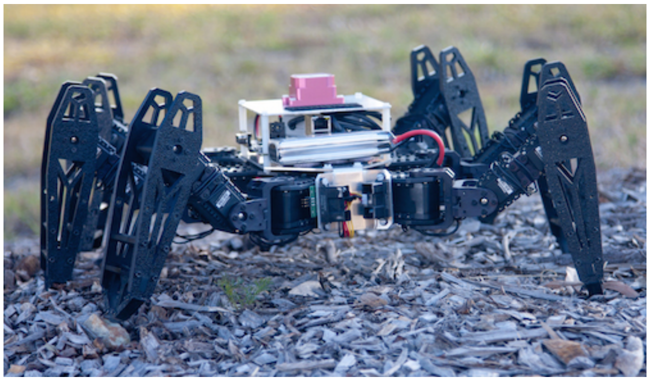


Fig. 1. A modified PhantomX Hexapod robot [7]

In the domain of robotic perception, integrating multiple sensing modalities is crucial to robustly and accurately perceive the surrounding environment. Examples of widely used robotic perception modalities are vision, sonar, lidar, and inertial sensing. However, acoustic sensing is a relatively underutilised method in robotics which can complement popular sensing modalities such as vision and lidar by being able to perform well in situations leading to failure modes in the latter sensors (e.g. varying lighting, darkness, dust, fog, smoke). The presented system has the potential to operate alongside other terrain perception methods, feeding in to higher level systems to facilitate effective and reliable navigation through difficult terrain with minimal prior knowledge.

Legged robots produce distinct sounds as a result of the legs interacting with the terrain they traverse on - both during the support phase and the stride phase of the gait cycle. By using a microphone, we sense these acoustic signals, extract a rich feature set and feed it in to a Support Vector Machine (SVM) to perform online real-time terrain classification. The novel contributions of this work can be listed as follows:

- A real-time terrain classification system with a 1 Hz update rate,
- A 32-dimensional feature vector with combined spectral and temporal features,
- Noise removal method to subtract servo noise to improve performance,
- 95.1% true positive rate with noise removal (92.9% in real-time trials) compared to 90% for offline legged robots [6].

The rest of this paper is organised as; section II reviews related work in the area of acoustic classification, section III describes the overall classification system, section IV de-

[1] J. Christie was a student at the Autonomous Systems Program, CSIRO, Pullenvale, Brisbane, QLD 4069 and at the School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane QLD 4072 during the time this work was carried out.

[2] N. Kottege is with the Autonomous Systems Program, CSIRO, Pullenvale, Brisbane, QLD 4069. Correspondence should be addressed to navinda.kottege@csiro.au

scribes the data collection method used while section V present the experiments discuss the results. Finally, section VI presents the final conclusions.

II. Related Work

In speech recognition applications a common feature that is used to classify audio signals is the Mel-Frequency Cepstral Coefficients (MFCC) [8]. As the non-linear Mel-frequency bands are optimised based on the human auditory system, applying a linearly spaced frequency band may better suit sound produced by robot-terrain interactions. Ozkul et al. [6] proposed the use of trapezoidal frequency bands instead of the Mel-frequency bands. We apply similar frequency bands and analyse their effects in the following sections.

In most applications that extract acoustic features, transformation from frequency to time domain is performed by applying the Discrete Fourier Transform (DFT) [5], or its computationally efficient form, the Fast Fourier Transform (FFT). With the transformation into the frequency domain, the signal power spectrum estimate could be used to extract features such as the spectral flux, spectral centroid, spectral roll-off, spectral spread, spectral kurtosis and short-time energy (STE) [9], [5].

Libby and Stentz [5] presented a method for classifying vehicle terrain interactions and hazardous interactions in outdoor environments. The features used in their work were the zero-crossing rate (ZCR), STE, spectral roll-off, spectral centroid, spectral flux, spectral centroid, spectral skewness and spectral kurtosis. A Support Vector Machine (SVM) was used to classify the terrains to their respective classes. They achieved the best accuracy of 78% for 3 hazardous vehicle-terrain interactions and three terrain classes. After applying a 2 s smoothing filter to reduce noise, accuracy improved to 92%. Our approach aims to investigates spectral subtraction as an alternative noise reduction technique.

Durst and Krotkov [10] proposed a method for classification of single impact sounds where the surface of an object was struck by an aluminium cane. The resulting acoustic emissions were then examined to classify the object that was struck. Ozkul et al. [6] was able to classify terrains using the acoustics generated from the naturally occurring steps of a hexapod robot. A tripod gait configuration was used, meaning that three single-impact sounds occurred at the same time. Ozkul et al. predominately extracted features from frequency domain including the frequency band coefficients, delta-features and the ZCR. However, the ZCR is a time based feature that gives a representation of the frequency of the changing positive-negative signs. Ozkul et al. also proposed a method of continually classifying for single-impact sounds without knowing when the impacts will occur. This was achieved by windowing the acoustic signal into frames and extracting the features. A number of windows were then averaged at a certain time interval to form the training set. Their offline classifier achieved a performance of up to 90% for six surface classes, which is the current state of the art for acoustic terrain classification for legged robots.

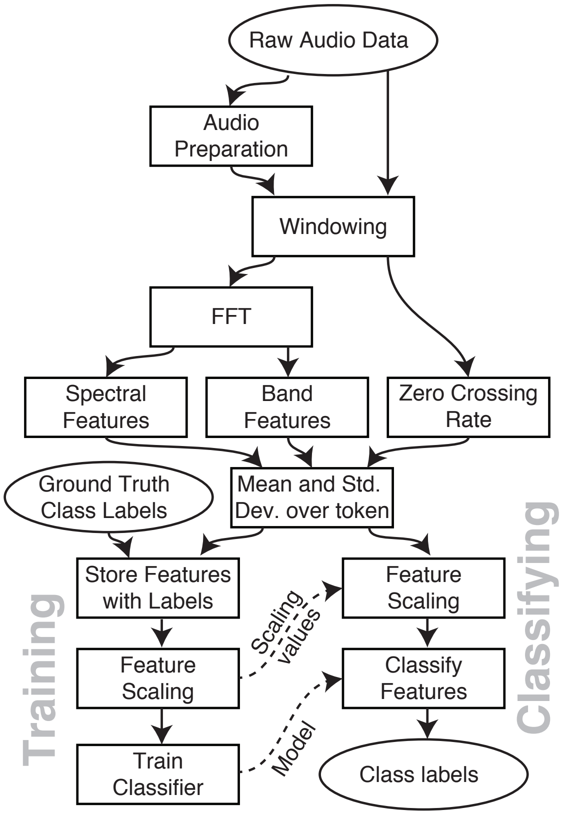


Fig. 2. System overview

Valada et al. [11] utilised a Deep Convolutional Neural Network (DCNN) that learns features to classify the vehicle-terrain interactions for a wheeled robot. Their system yielded an overall accuracy of 99.57% which is the current state of the art for acoustic vehicle-terrain classification for wheeled robots. Our work presents and implements an online real-time classification algorithm that exceeds current legged robot performance.

III. Real-time Terrain Classification System

A block diagram of the proposed acoustic terrain classification algorithm is shown in Figure 2. We first collected audio data of the robot's interactions during traversal on different terrains which is further described in Section IV. A few pre-processing steps were carried out on the audio data, including data conversion, windowing, audio inspection and noise removal. For each window, 32 features from the frequency and time domains were extracted to compile a training data set. The feature vectors were then fed into a multi-class Support Vector Machine (SVM) to generate a classification model. Our algorithm was implemented offline in Matlab and online real-time in the Robotic Operating System (ROS). The Matlab implementation served as an experimental environment to test and validate our results. ROS provided parallel execution of data collection, feature extraction and classification by splitting the algorithm into several nodes (processes). The following sections will continue to describe each component of the algorithm in detail.

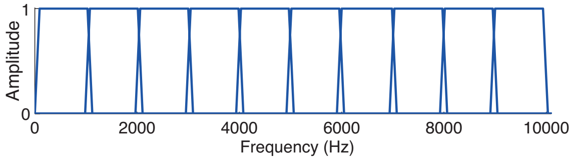


Fig. 3. Linear Power Band Filters

A. Data Overview and Pre-processing

To capture the audio data, we used the open source software HARK-ROS and Hark designer [12] in Ubuntu 12.04. A ROS node was configured to publish audio data messages which could either be saved as a bag file for offline processing or used directly for online real-time classification. During recording, each of the bag files were labelled with the terrain type and the location. Labelling the location was imperative in validating the classifier's performance with new locations, this is explained further in Section V-C. The labelled bag files were then converted to MAT-files for generating the training set in Matlab.

Overfitting may occur if the model is too general and describes noise instead of terrain specific characteristics [13]. We attempted to prevent overfitting by listening to each audio recording, and manually removing files that contained noticeable environment noise such as car sounds or bird calls. Additionally, there was servomotor noise present in recordings, which was difficult to eliminate without knowing when each of the legs impacted the surface. However, we investigated the use of a noise removal using spectral subtraction [14]. Using this approach, the noise generated from the servos was subtracted from the original signal.

The next stage was to split each sequence in the time domain into short windows. Each window contained 50% overlap between each successive window. Features were extracted from the windows, and windows were grouped to give token windows [6]. Each token was then used as a training vector to give one classification prediction. The time region containing the majority of terrain information was when the robot's legs impacted the surface. As the leg impact time was unknown, the minimum token size needed to incorporate the acoustic characteristics of at least one impact. Hence, our minimum token size was set to 0.5 s when the hexapod was traversing at $9\,\mathrm{cms}^{-1}$. In section V we experiment and analyse the effects of different window and token sizes.

B. Feature Extraction

The next stage consisted of extracting time and frequency domain features from each window. To transform the signal into the frequency domain, we applied a Hanning window to each window partition, effectively smoothing the signal and reducing truncation effects. We then calculated the FFT for each window. The FFT size was set to equal the window size. We kept the magnitude of the power signal and ignored the phase. Due to the symmetry of the FFT, only the first half of the windowed signal was used.

The MFCCs are widely used features in speech recognition algorithms [8]. As proposed in Ozkul et al. [6], a variation from the Mel-frequency scale is to use evenly spaced band-pass filters. Analysing the spectrograms of each of the terrain classes showed that the majority of the spectral power was between 0 to 15 kHz. However, after experimenting with different frequencies it was found that 0 to 10 kHz contained the majority of distinctive information. The power spectrum was multiplied by 10 successive band filters over 0 to 10 kHz with an overlap of 100 Hz between each band, as shown in Figure 3. The mean and standard deviation across the token was calculated, giving a 20-dimensional feature set. We also observed the differences in filtered spectral powers between each successive window, namely the delta-features. The mean and standard deviation of the delta-features were calculated, giving an additional 20 dimensions.

A number of spectral features that characterise the shape of the distribution were extracted in the frequency domain, including the spectral roll-off, spectral centroid, spectral kurtosis, spectral skewness and spectral flux [15], [16], [17]. The equations for calculating the spectral features are defined in Table I where, N is the window size, $X(i)$ is the i^{th} sample of the power spectrum, F_k is the k^{th} frequency band-filter, n is the reference window, S is the equivalent to half the window size $(N/2)$, μ is the mean across a window, θ is the standard deviation across a window, c is a percentage threshold, $f(i)$ is the i^{th} frequency bin, $y(i)$ is the time domain signal, α emphasises the noise estimate, Γ is a magnitude or power subtraction factor and P_N is the averaged noise estimate.

In calculating the spectral roll-off, which gives the frequency bin below a certain percentage (c) of the magnitude distribution, we choose the percentage to be 95% empirically. The spectral centroid indicates the centre of mass of the power frequency distribution. Spectral kurtosis measures the peak of a frequency distribution and its similarity to a Gaussian distribution. Spectral skewness measures the distribution's symmetry with respect to it's magnitude about the mean. Spectral flux measures the spectral change between each successive window. A time based feature that exhibits frequency characteristics is the ZCR, which measures the number of zero crossings in the time domain [17]. For each of these features the mean and standard deviation was calculated across the token window adding 12 more elements to the feature vector.

C. Classification

Support Vector Machines (SVM) provide a non-linear high dimensional model that is generally less prone to overfitting than other classifiers [18]. We implemented SVMs offline and online using the open-source libSVM library [19]. Prior to classification we scaled the features to avoid attributes with large numeric ranges dominating attributes with small numeric ranges [20]. We linearly scaled our feature vectors between 0 and 1. The same scaling factors were used in both training and testing.

TABLE I

Feature and Spectral Subtraction Equations

Feature name	Feature Equations
Band	$B_k = \sum_{i=1}^{N/2} X(i)F_k(i), \;\; k = 1, 2, \ldots, 10$
Delta	$\Delta_k = B_k(n) - B_k(n-1), \;\; k = 1, 2, \ldots, 10$
Spectral Skewness	$SS = \frac{1}{S}\sum_{i=1}^{N/2}\left(\frac{X(i)-\mu}{\theta}\right)^3, \quad \text{where} \quad \mu = \frac{1}{S}\sum_{i=1}^{N/2} X(i) \quad \text{and} \quad \theta = \sqrt{\frac{1}{S}\sum_{i=1}^{N/2}(X(i)-\mu)^2}$
Spectral Kurtosis	$SK = \frac{1}{S}\sum_{i=1}^{N/2}\left(\frac{X(i)-\mu}{\theta}\right)^4 - 3$
Spectral Roll-off	$SR = k, \quad where \sum_{i=1}^{k} X(i) = \frac{c}{100}\sum_{i=1}^{N/2} X(i)$
Spectral Flux	$SF = \frac{\sqrt{\sum_{i=1}^{N/2}(X(i,n) - X(i,n-1))^2}}{N/2}$
Spectral Centroid	$SC = \frac{\sum_{i=1}^{N/2} f(i)X(i)}{\sum_{i=1}^{N/2} X(i)}$
Zero-crossing Rate	$Z_n = \frac{1}{N}\sum_{i=1}^{N/2} \lvert sgn[y(i)] - sgn[y(i-1)]\rvert, \quad \text{where} \quad sgn[x(m)] = \begin{cases} 1, x(n) \geq 0 \\ -1, x(n) < 0 \end{cases}$
Spectral Subtraction	$P(i) = (X(i) - \alpha P_N(i))^{1/\Gamma}$

The choice of kernel is an open question, as we are dealing with complex acoustic signals, and overlaps are therefore expected to occur in the feature space. Radial Basis Function (RBF) and the linear kernel have been shown to effectively characterise acoustic terrain data [5]. RBF kernels are used to map attributes into high dimensional space when the relationship between class label and attributes are nonlinear. Nonlinear mapping tends to not improve the performance for higher dimensional feature vectors, meaning that a linear kernel is sufficient [20]. Our tests compared the performance of both the linear and RBF kernels. As a benchmark comparison, a k-Nearest Neighbour (k-NN) classifier was run in the WEKA environment [21].

We built a multi-class classifier with the "one-versus-one" approach, which assigns each binary SVM a class based on a max-wins voting strategy. To determine the optimal parameters C and γ for the kernels, 10-fold cross validation and a grid search was applied. In 10-fold cross validation the data is partitioned in to N subsets of equal size. Progressively one subset trains the model and the $N-1$ subsets validate the model. The predicted labels are compared to the actual labels and an accuracy metric is calculated. The goal of the grid search is to identify the parameters that achieve the highest classification accuracy for new terrain classes. Grid search iteratively trains the classifier with a new pair of C and γ and compares the highest cross validation accuracy of each. Using a finer grid, the process is repeated around the parameters with the highest accuracy in order to give the optimal parameters.

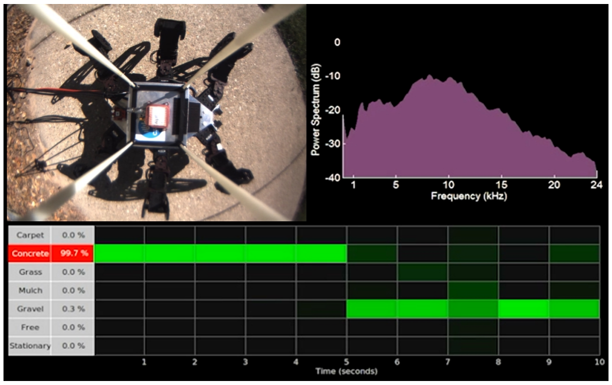


Fig. 4. Real-time terrain classification of the hexapod walking on gravel and concrete. The bottom array displays the current classification (in red) and the confidence level (brighter green indicates more confidence)

IV. Data Collection

Our tests were conducted on a modified PhantomX Hexapod robot, consisting of 18 Dynamixel AX-12 smart servomotors (3 per leg), as shown in Figure 1. The standard alternating tripod gait was used at a controlled speed of $9\,\text{cms}^{-1}$. In the alternating tripod gait each step consists of 3 simultaneous leg impacts. The leg height and stride length were set at 45 mm to allow traversal on a wider range of terrains. An omnidirectional Knowles microphone (WP-23849-C36) was attached to the bottom of the hexapod facing towards the ground to capture the hexapod-terrain interactions. A Point Grey Chameleon (CMLN-13S2C-CS) camera was mounted by four rods above the hexapod and positioned to give a bird's-eye view of the hexapod as it

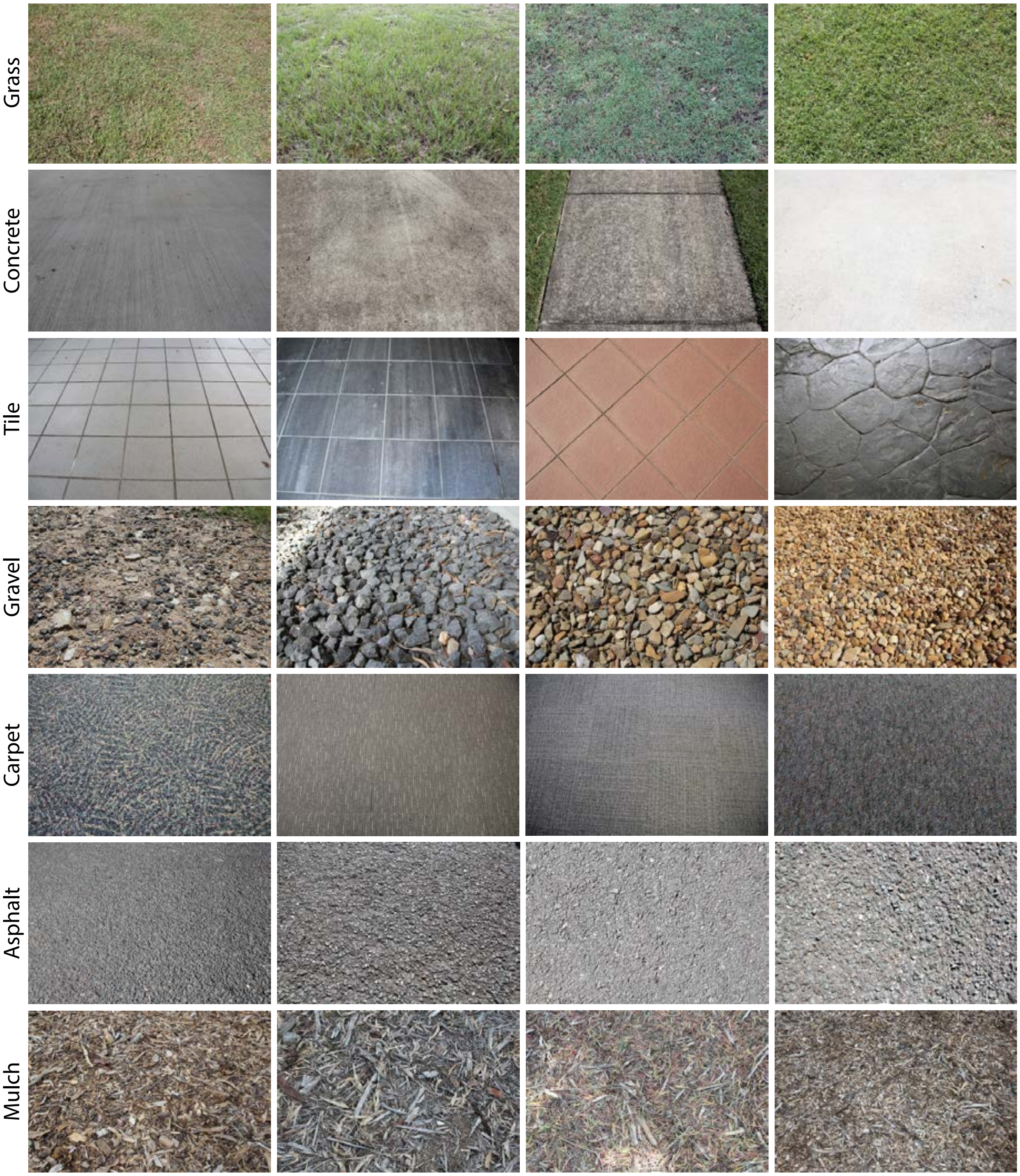


Fig. 5. Examples of terrain types used for training and testing the system

traversed on the different terrain. Attached to the camera was a fisheye lens with a 185° field of view. The camera was only used during real-time classification to verify the predictions. The Fly Capture 2.0 software and the open-source PGR camera package was used to process the image messages. Figure 4 shows the GUI of the online real-time terrain classification system.

Data was collected over a number of terrains including carpet, grass, mulch, concrete, tiles, asphalt and gravel. For each of the terrain types, data was collected from different locations to provide variations within each terrain type. Training with multiple locations helps prevent the classifier from overfitting, meaning that the model is better able to predict new locations [4]. All of the datasets were recorded on flat surfaces. In total we recorded approximately 5 mins worth of acoustic interactions for each terrain type.

Figure 5 shows the terrain classes used during training. For grass terrain, the locations differed from cut to uncut grass. For the mulch terrain, the locations contained bark cuts with varying thickness, leaf coverage and age. For the concrete terrain, the locations were recorded with varying surface roughness, slab size and age. For the tile terrain,

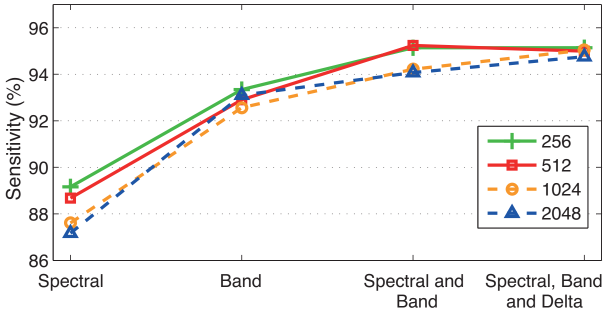


Fig. 6. Classification sensitivity of different window lengths and features

the locations contained ceramic tiles with varying shape and surface roughness. For the carpet terrain, the locations were varied by recording on high traffic areas, low traffic areas, different ages and different knit structures. For the gravel terrain, the locations varied with different grain sizes and coarseness. For the asphalt terrain, the locations were varied in roughness and fatigue, depending on the amount of traffic.

Additionally, two non-terrain specific classes were added to the real-time classification system: stationary and free. Stationary indicates when the hexapod was not moving, while free indicates when the hexapod was suspended above the ground and walking. These two non-terrain classes were used as reference classes, otherwise in these circumstances the real-time classifier would choose the next best terrain. Having these two classes further emphasised the classifier's ability to classify multiple terrain classes. Recordings of wooden bridges were conducted for earlier experiments, however this terrain class was discarded as only a limited number of different locations for wood could be found.

V. Experiments and Results

This section describes the results of the acoustic terrain classification algorithm. For each classification test, the number of training points in every class was lowered to match that of the class with the least points, in order to preserve symmetry. We then compared the effects of different SVM kernels, parameters, token sizes and window sizes. We also investigate the effects of removing noise from the signal using spectral subtraction.

A. Feature Selection

We experimented with two kernels: the RBF and the linear kernel. A grid search was performed with a coarse grid range from 2^{-5} to 2^{11} for C and γ, and a finer grid was done on the neighbourhood of the best parameter of the coarse grid. Each classification model's performance was evaluated by 10-fold cross validation. Confusion matrices, where columns show the actual classes and rows show the predicted classes for each trial, was then calculated and tabulated in Figure 9a. From the confusion matrices we derived useful statistical measures in Table II such as accuracy, precision (positive-predictive value), sensitivity (true-positive rate) and specificity [22].

In order to determine the optimal window length, the SVM was trained with an RBF kernel and a token length of 1 s. The results in Figure 6 show that across all feature combinations, a smaller window size was able to capture sufficient information for new predictions. Hence, the following experiments were trained with a window size of 256 samples.

Evaluating the token size presented a time-resolution trade-off. That is, the trade-off between minimising the time that tokens are sampled and maximising the classification accuracy obtained with a greater token size. The minimum token size that should be considered is equivalent to the minimum time to take each step. Figure 7 shows the results of token sizes from 0.5 s (1 step) to 2.5 s (5 steps). The results show that larger token sizes tend to increase the sensitivity. However, due to the trade-off we used a 1 s token size which gave a 1.8% performance increase over the 0.5 s token size.

Figure 8 shows the performance results of two SVM kernels and the k-NN classifier. Both kernels had similar classification performance across all features, however, the RBF kernel lead on average by 0.5%. In this case both kernels are viable, however, we opted to use the RBF due to its nonlinear separation benefits.

B. System Performance

A comparison of the feature sets' performance is shown in Figures 6, 7 and 8. In preliminary studies we found that integrating the mean and standard deviation of each feature improved the average performance by 4%. Hence, in the following comparisons the results yield an additional 4%

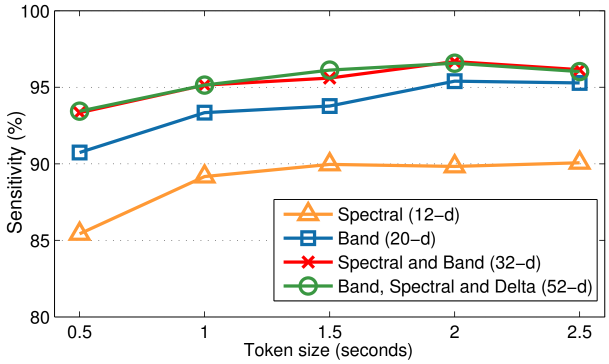


Fig. 7. Classification sensitivity of different token sizes and features

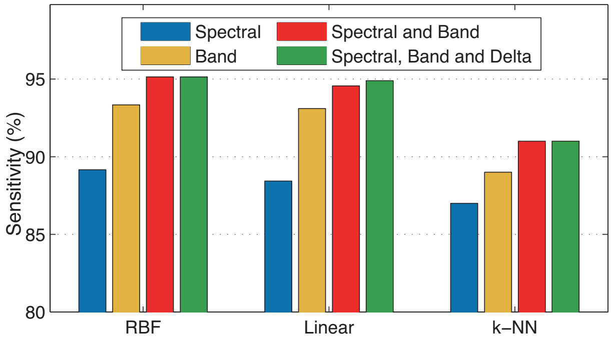
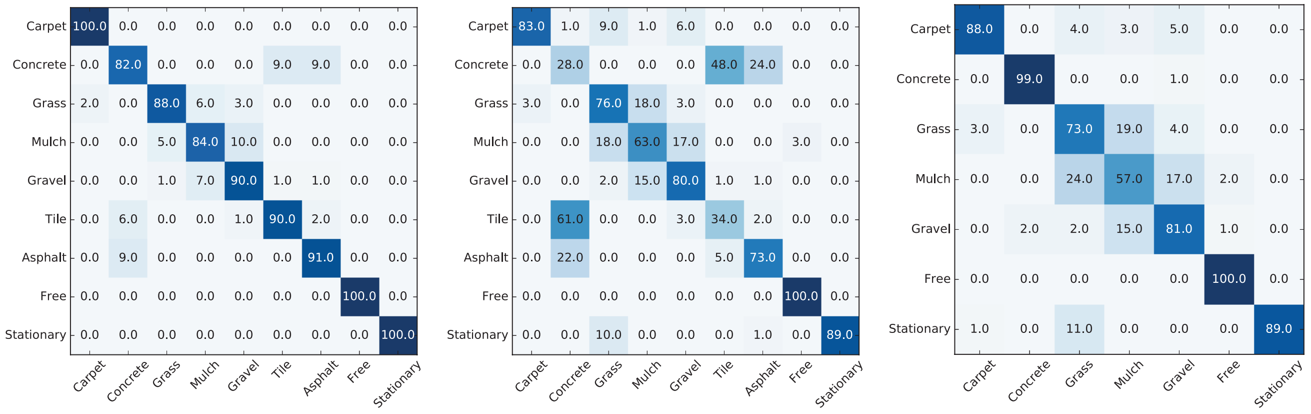


Fig. 8. Comparison of the Radial Basis Function, the linear kernel and k-Nearest Neighbour



(a) (b) (c)

Fig. 9. Normalised confusion matrices of (a) 10-fold crossvalidation with all terrains, (b) leave-one-out crossvalidation with all terrains and (c) leave-one-out cross validation with asphalt and tile combined in to concrete. Background shading of each cell represents the relative number of predictions.

performance for the feature sets proposed in the respective literature.

The spectral features which were a selection of the best performing features in the work by Libby and Stentz [5] had the worst performance. The band features generated from the MFCCs presented in the work by Ozkul et al. [6] had comparatively better results, increasing performance by 4.2%. Combining the Spectral and Band features we get an improvement of 6% over just using the Spectral features. The addition of delta-features presented by Ozkul et al. had less than 0.5% increase on the performance. Applying Occam's razor, we dropped the delta-features and selected the 32-dimensional feature set consisting of Spectral and Band features.

Summarising the highest performing attributes, our system uses a 1 s token length, a window size of 256 samples and the Spectral and Band features (means and standard deviations). Disregarding the additional classes stationary and free as they achieved 100% classification and would not be encountered in practical applications, our system yielded an overall sensitivity of 92.9% on terrain classes.

C. *New terrains leave-one-out cross validation*

Best et al. [4] addresses the issue of location data being included in the training set during k-fold cross validation. The proposed method was the *leave-one-out* cross validation, which involves iteratively testing with one location and training with all other locations. The confusion matrix showing the new validation results are shown in Figure 9b for all the terrain classes.

There was considerable confusion in the asphalt, tile and concrete terrains. These firmer terrains likely have similar acoustic interactions and overlap in the feature space, decreasing the classifier's ability to discriminate between classes. There was also confusion between the grass, mulch and gravel terrains. Figure 9c shows the results after combining the asphalt, tile and concrete classes where the performance increased to over 95% sensitivity and precision (Table II).

D. *Spectral Subtraction*

Figure 10 shows the classification results after spectral subtraction with all terrain classes and with the combined class of asphalt, tile and concrete. We extracted 20 s of servo noise from each terrain class, resulting in approximately 140 s of noise data. A grid search ranging from 0.1 to 2 for the parameters α and Γ was performed to give the optimal parameters. Our average signal-to-noise ratio was -0.85 dB prior to spectral subtraction, meaning that the noise dominated the signal. This further emphasises our system's ability accurately classify in the presence of considerable noise. Spectral subtraction significantly increased the sensitivity of the grass by over 5%, a 4% improvement in mulch and a small improvement in all other terrain classes. Overall, the spectral subtraction increased the average sensitivity to 95.1% from the original signal's 92.9% sensitivity.

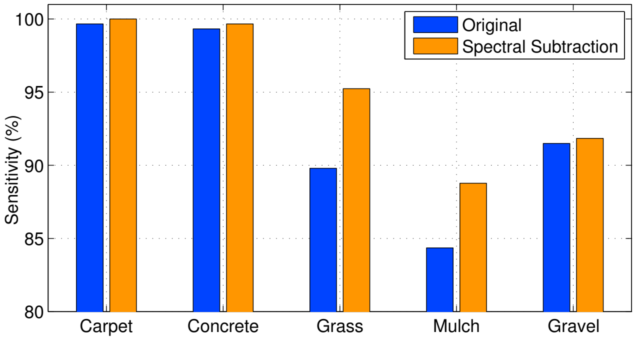


Fig. 10. Performance comparison of classifying with the original acoustic signal and a noise removed signal

VI. Conclusions

We presented an acoustic feature based real-time terrain classification system for legged robots operating at 1 Hz and presented an experimental evaluation of the system. The feature vectors consisted of combined spectral and temporal features that accurately represented the sounds produced by robot-terrain interactions. We also presented a noise removal method which improved performance, especially in terrains such as grass and mulch. The SVM which was trained on

TABLE II

ACCURACY, PRECISION, SENSITIVITY AND SPECIFICITY OF 10-FOLD CROSS VALIDATION WITH ALL TERRAINS, LEAVE-ONE-OUT CROSS VALIDATION WITH ALL TERRAINS AND (*) LEAVE-ONE-OUT CROSS VALIDATION WITH ASPHALT AND TILE COMBINED IN TO CONCRETE.

	10-fold				Leave-one-out				Leave-one-out*			
Terrain	Accu.	Prec.	Sens.	Spec.	Accu.	Prec.	Sens.	Spec.	Accu.	Prec.	Sens.	Spec.
Carpet	99.7	97.7	99.7	99.7	97.7	96.0	82.7	99.6	97.8	95.6	88.4	99.3
Concrete*	96.3	84.0	82.0	98.0	82.5	24.5	27.6	89.4	99.7	98.3	99.3	99.7
Grass	98.0	93.5	88.1	99.2	93.1	66.8	75.9	95.3	90.4	64.3	73.5	93.2
Mulch	96.8	86.4	84.4	98.3	92.1	64.7	62.9	95.7	88.6	60.8	57.5	93.8
Gravel	97.3	86.1	90.5	98.2	94.5	72.8	80.3	96.3	93.5	75.7	80.6	95.7
Tile	97.8	89.9	90.5	98.7	86.6	38.4	33.7	93.2				
Asphalt	97.7	89.0	90.8	98.6	93.9	72.3	72.8	96.5				
Free	99.9	99.3	100.0	99.9	99.6	97.0	99.7	99.6	99.7	97.7	100.0	99.6
Stationary	100.0	100.0	100.0	100.0	98.8	100.0	89.5	100.0	98.4	100.0	88.8	100.0

7 different terrain types performed better than the state of the art for legged robots. The overall sensitivity of the real-time system was 92.9% which improved to 95.1% with noise removal. Therefore, the output of the proposed system can be effectively used by higher level systems on the robot to facilitate robust and efficient navigation through unknown terrain.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Pravin Kumar and Dylan Williamson for their support in preparing and conducting experiments.

REFERENCES

[1] M. H. Raibert, "Legged robots," *Communications of the ACM*, vol. 29, no. 6, pp. 499–514, 1986.

[2] J. D. Weingarten, G. A. Lopes, M. Buehler, R. E. Groff, and D. E. Koditschek, "Automated gait adaptation for legged robots," in *proceeding of the IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 3. IEEE, 2004, pp. 2153–2158.

[3] N. Kottege, C. Parkinson, P. Moghadam, A. Elfes, and S. Singh, "Energetics-informed hexapod gait transitions across terrains," in *IEEE International Conference on Robotics and Automation (ICRA '15)*, May 2015, pp. 5140–5147.

[4] G. Best, P. Moghadam, N. Kottege, and L. Kleeman, "Terrain classification using a hexapod robot," in *Australasian Conference on Robotics and Automation (ACRA '13)*, 2013.

[5] J. Libby and A. T. Stentz, "Using sound to classify vehicle-terrain interactions in outdoor environments," in *2012 IEEE International Conference on Robotics and Automation (ICRA 2012)*, May 2012.

[6] M. C. Ozkul, A. Saranli, and Y. Yazicioglu, "Acoustic surface perception from naturally occurring step sounds of a dexterous hexapod robot," *Mechanical Systems and Signal Processing*, vol. 40, no. 1, pp. 178 – 193, 2013.

[7] Trossen Robotics, "PhantomX AX Hexapod kit [Online]," Available: http://www.trossenrobotics.com/phantomx-ax-hexapod.aspx, accessed: 10-09-2015.

[8] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing: a guide to theory, algorithm, and system development*. Upper Saddle River, NJ: Prentice Hall, 2001.

[9] H. Jiang, J. Bai, S. Zhang, and B. Xu, "SVM-based audio scene classification," in *proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE '05)*, Oct 2005, pp. 131–136.

[10] R. Durst and E. Krotkov, "Object classification from analysis of impact acoustics," in *proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, August 1995, pp. 90–95.

[11] L. S. Abhinav Valada and W. Burgard, "Deep feature learning for acoustics-based terrain classification," in *Proceedings of the International Symposium on Robotics Research (ISRR)*, September 2015.

[12] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, "Design and implementation of robot audition system'hark'open source software for listening to three simultaneous speakers," *Advanced Robotics*, vol. 24, no. 5-6, pp. 739–761, 2010.

[13] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. The MIT Press, 2010.

[14] M. Karam, H. F. Khazaal, H. Aglan, and C. Cole, "Noise removal in speech processing using spectral subtraction," *Journal of Signal and Information Processing*, vol. 5, no. 2, pp. 32–41, 2014.

[15] A. Lerch, *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*, 1st ed. Wiley-IEEE Press, 2012.

[16] M. C. Wellman, N. Srour, and D. B. Hillis, "Feature extraction and fusion of acoustic and seismic sensors for target identification," pp. 139–145, 1997.

[17] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence content classification using audio features," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, G. Antoniou, G. Potamias, C. Spyropoulos, and D. Plexousakis, Eds. Springer Berlin Heidelberg, 2006, vol. 3955, pp. 502–507.

[18] W. Duch, N. Jankowski, and T. Maszczyk, "Make it cheap: Learning with o(nd) complexity," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, June 2012, pp. 1–4.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] C. Hsu, C. Chang, and C. Lin, "A practical guide to support vector classification," Department of Computer Science, National Taiwan University, Tech. Rep., 2010.

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[22] N. Kottege, F. Kroon, R. Jurdak, and D. Jones, "Classification of underwater broadband bio-acoustics using spectro-temporal features," in *proceedings of the 7th ACM International Conference on Underwater Networks and Systems*, ser. WUWNet '12. New York, NY, USA: ACM, 2012, pp. 19:1–19:8.