Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Vibration-based Terrain Analysis for Mobile Robots*

Christopher Brooks, Karl Iagnemma, and Steven Dubowsky
Field and Space Robotics Laboratory
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139
cabrooks, kdi, dubowsky@mit.edu

***Abstract*—Safe, autonomous mobility in rough terrain is an important requirement for planetary exploration rovers. Knowledge of local terrain properties is critical to ensure a rover's safety on slopes and uneven surfaces. This paper presents a method to classify terrain based on vibrations induced in the rover structure by wheel-terrain interaction during driving. Vibrations are measured using an accelerometer on the rover structure. The classifier is trained using labeled vibration data during an off-line learning phase. Linear discriminant analysis is used for on-line identification of terrain classes such as sand, gravel, or clay. This approach is experimentally validated on a laboratory testbed.**

***Index Terms*—Rough terrain, mobile robots, robot vibration sensing, pattern classification**

I. Introduction

Planetary exploration rovers are being proposed for missions to increasingly challenging terrains [1]. These may include craters, hills, and ravines, where rocky outcrops might yield glimpses of a planet's history. To safely traverse slopes and highly uneven terrain, knowledge of local terrain properties is critical. For example, a rover might climb a rocky slope with ease, but slide down a sandy slope of the same grade. Similarly, a rover traversing loose sand could become entrenched, where a rover crossing packed soil would face no such danger. Terrain class knowledge would allow a rover to adapt its control and/or planning strategy to safely and efficiently traverse terrain of varying composition.

Previous research on terrain classification has focused primarily on remote classification using vision or range data. Navigation systems employing this data have been demonstrated by researchers at the Jet Propulsion Laboratory and at the National Institute of Standards and Technology [2], [3]. Most systems have focused on navigating a robot on a semi-structured road surface. Algorithms designed for unstructured terrain emphasize detection of geometric obstacles such as rocks or steep slopes [4], [5]. These papers have not addressed the issue of the soil itself being a hazard. Such a hazard can be termed a *nongeometric hazard* [6]. Previous research at MIT has attempted to address this issue by developing algorithms to measure wheel sinkage into deformable terrain, identify soil cohesion and internal friction angles, and explicitly estimate terrain traversability [7]–[11].

Vibration-based terrain classification was suggested in 2002 by Iagnemma and Dubowsky [12]. Research with a similar focus was published by Sadhukhan and Moore [13], [14]. Their neural network approach focused on classification of terrain for a high-speed vehicle, and degraded at low speeds where vibration amplitudes were severely reduced. Thus, its applicability to planetary rovers, where speeds are expected to be under 15 cm/sec, would be limited.

This paper presents a method to identify terrain class based on vibrations in the rover structure induced by wheel-terrain interaction during driving. In this approach, an accelerometer is mounted to the rover structure. Vibrations recorded on-line during a traverse are classified based on their similarity to vibrations observed during an off-line supervised training phase. This algorithm employs principal component analysis and linear discriminant analysis to efficiently measure similarity.

II. Terrain Classification Algorithm Overview

The algorithm presented here takes a signal recognition approach to classifying terrain based on vibration signals. This is in contrast to an approach that might use a solid mechanics or finite element model to analytically predict how the rover structure will vibrate in response to interaction with terrain. The proposed algorithm learns to recognize distinct terrain types based on labeled vibration data provided during an off-line training phase. During training, the algorithm analyzes these data sets to form a low-dimensional representation of the signals corresponding to each labeled terrain. This information is stored in memory for use by an on-line classifier. During on-line classification, measured vibration signals are quickly classified as one of the labeled terrain classes. An overview schematic of the algorithm is shown in Fig. 1.

In this approach, vibration signals are first divided into short segments. These are then converted from time-domain voltage signals into power spectral densities. Further analysis is performed in the Fourier domain. Log scaling of the power spectral magnitude is used to reduce the dominating effect of high-magnitude frequency components.

With the signals represented as a time series of Fourier spectra, training is a matter of dividing a high-dimensional space (i.e. the Fourier coefficients) into regions associated with an individual terrain class. To reduce the dimensionality of the comparison, principal component

* This work is supported by the Mars Technology Program at the NASA Jet Propulsion Laboratory.

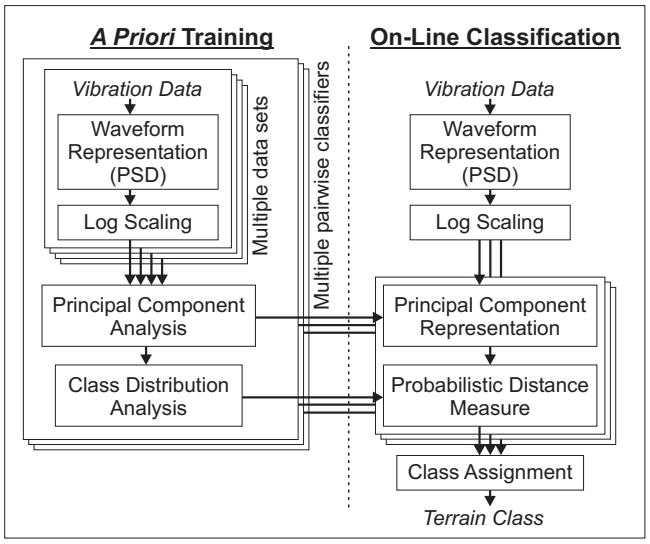

Fig. 1. Overview flowchart for vibration-based terrain classification algorithm

analysis is used [15]. Here only the first k components are retained. The value of k is chose empirically. Note that principal components are computed during the training phase. These same principal components are used during the classification phase.

To define class boundaries in this principal component space, linear discriminant analysis is used as a pairwise classifier [16], [17]. In this approach, subclassifiers are created to classify a sample signal as being associated with one of two possible terrains. Separate sub-classifiers are used for each possible pair of terrains. For example, for the three-terrain case of sand, gravel, and clay, one classifier would distinguish gravel from sand, another would distinguish gravel from clay, and a third would distinguish sand from clay. Linear discriminant analysis considers both the distribution of samples within a single terrain class and the separation between class means to compute an optimal vector along which to compare samples. Classification of a test sample can be done by projecting its principal component representation onto this vector. A number of simple classifiers are available to address the resulting one-dimensional classification problem [18], [19].

To accommodate classification of more than two terrains, a voting scheme is used. Each pairwise classifier can cast a "vote" for one of the two terrains it distinguishes, or remain "undecided." The winning terrain class is returned.

III. Terrain Classification Algorithm Detailed Description

The terrain classification algorithm may be broken into two separate phases, *a priori* training and on-line classification. *A priori* training is computationally intensive and is performed off-line. On-line classification is computationally efficient and is performed during a rover traverse. These phases are described below

A. A Priori Training

During the *a priori* training phase, the algorithm learns to recognize vibration signatures corresponding to various user-selected terrain types. These are chosen to correspond to terrains of interest that a robot might encounter during field operations.

The first step in *a priori* training is to collect vibration data from representative terrains. This data is in the form of a time series of the voltage output of the accelerometer. Data should be collected for the terrain under a range of conditions spanning those for which the classifier is expected to perform (for example, under varying speeds, wheel slip conditions, and wheel loads).

This time series is broken into short segments. The duration of these segments should be appropriately scaled to the physical scenario (i.e. a single segment should contain data from a travel distance that is scaled to the wheel diameter, spacing between grousers, spatial variations of terrain, etc.). The power spectral density (PSD) of each of these segments is then computed using Welch's method [20], and a log-scaled version of this PSD is stored in a matrix.

For illustration purposes, consider classification of sand and gravel. Data for sand would be stored in a matrix $\mathbf{Y}_{sand}$ as:

$$\mathbf{Y}_{sand} = \begin{bmatrix} y_{sand,fmin,1} & \cdots & y_{sand,fmin,n} \\ \vdots & \ddots & \vdots \\ y_{sand,fmax,1} & \cdots & y_{sand,fmax,n} \end{bmatrix}. \quad (1)$$

In this representation, each column corresponds to a single time segment and contains the log PSD components in a frequency range of interest. Each row corresponds to a single frequency and contains the log PSD components for all time segments. (Note that this matrix may be visualized as a spectrogram such as the one in Fig. 2, simply by assigning a grayscale intensity proportional to the value of each element. This plot is a convenient way to view the time-varying nature of the frequency components in a signal.)

A separate classifier is used to distinguish between terrain pairs. For each classifier, a discrimination vector and terrain class statistics are produced as follows.

The two matrices describing the training data for each class, $\mathbf{Y}_{sand} \in \Re^{m \times n_{sand}}$ and $\mathbf{Y}_{gravel} \in \Re^{m \times n_{gravel}}$, are combined to form a complete record of the data: $\mathbf{Y} = [\mathbf{Y}_{sand} \quad \mathbf{Y}_{gravel}]$,

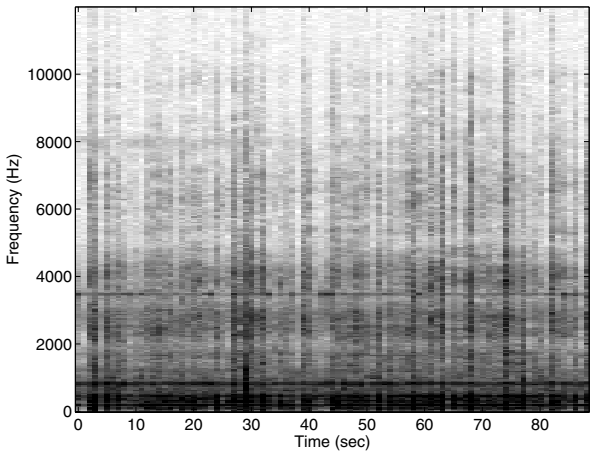

Fig. 2. Sample spectrogram of terrain vibration data

$\mathbf{Y} \in \Re^{m \times n}$, where m is the number of frequency components, and $n = n_{sand} + n_{gravel}$ is the total number of time segments. The rows of $\mathbf{Y}$ are then mean adjusted to form the matrix $\hat{\mathbf{Y}}$:

$$\overline{\mathbf{y}} = \frac{1}{n}\mathbf{Y}\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} \tag{2}$$

$$\hat{\mathbf{Y}} = \mathbf{Y} - \overline{\mathbf{y}}\begin{bmatrix}1 & \cdots & 1\end{bmatrix}. \tag{3}$$

The mean of each row of the matrix $\hat{\mathbf{Y}}$ is equal to zero. This matrix is used to analyze variation in the entire set of training data.

Singular value decomposition [21] is then used to separate $\hat{\mathbf{Y}}$ into three matrices, $\mathbf{U}_a$, $\mathbf{S}_a$, and $\mathbf{V}_a$:

$$\hat{\mathbf{Y}} \xrightarrow{\text{SVD}} \left(\mathbf{U}_a, \mathbf{S}_a, \mathbf{V}_a^{\mathbf{T}}\right). \tag{4}$$

Here, $\mathbf{U}_a \in \Re^{m \times m}$ is a unitary matrix with the principal components of $\hat{\mathbf{Y}}$ as columns. $\mathbf{S}_a \in \Re^{m \times n}$ is a diagonal matrix of singular values. $\mathbf{V}_a \in \Re^{n \times n}$ is a unitary matrix with the principal components of $\hat{\mathbf{Y}}^{\mathbf{T}}$ as columns.

The matrix $\mathbf{U}_a$ is assumed to be composed of orthogonal signal and noise subspaces: $\mathbf{U}_a = [\mathbf{U}_{signal} \quad \mathbf{U}_{noise}]$, such that $\mathbf{U}_{signal}{}^{\mathbf{T}}\mathbf{U}_{noise} = [0]$. To represent the signal subspace, the first k columns of $\mathbf{U}_a$ (i.e. the first k principal components of $\hat{\mathbf{Y}}$) are assigned to the matrix $\mathbf{U}_{signal}$. Similarly, the upper-left $k \times k$ block of $\mathbf{S}_a$ represents the singular values associated with the signal space, and will be referred to as $\mathbf{S}_{signal}$. Using too many principal components with a limited amount of training data can be detrimental, as this would overtrain the algorithm (i.e. train it to recognize noise in the training data, reducing its ability to classify new data). In practice we have used $k = 15$, as it appears to give good signal representation without overfitting. In experiments, the first 15 principal components account for approximately 90% of the signal variance.

The two matrices $\mathbf{U}_{\mathbf{signal}}$ and $\mathbf{S}_{\mathbf{signal}}$ can be considered to be a map from the full frequency space ($\Re^m$) to the signal space ($\Re^k$). Signal space mappings of the separate data sets $\mathbf{Y}_{sand}$ and $\mathbf{Y}_{gravel}$ are computed as:

$$\mathbf{W}_{sand} = \mathbf{S}_{signal}{}^{-1}\mathbf{U}_{signal}{}^{\mathbf{T}}\mathbf{Y}_{sand} \tag{5}$$

$$\mathbf{W}_{gravel} = \mathbf{S}_{signal}{}^{-1}\mathbf{U}_{signal}{}^{\mathbf{T}}\mathbf{Y}_{gravel}. \tag{6}$$

In this representation each column of $\mathbf{W}_{sand}$ and $\mathbf{W}_{gravel}$ corresponds to an individual time segment. Each row of $\mathbf{W}_{sand}$ and $\mathbf{W}_{gravel}$ corresponds to a principal component (i.e. a linear combination of frequency components). Taking each column as a vector to a point in k-dimensional space, $\mathbf{W}_{sand}$ and $\mathbf{W}_{gravel}$ represent point clouds with means $\overline{\mathbf{w}}_{sand}$ and $\overline{\mathbf{w}}_{gravel}$, respectively:

$$\overline{\mathbf{w}}_{sand} = \frac{1}{n_{sand}}\mathbf{W}_{sand}\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} \tag{7}$$

$$\overline{\mathbf{w}}_{gravel} = \frac{1}{n_{gravel}}\mathbf{W}_{gravel}\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}. \tag{8}$$

Fig. 3 shows sample point clouds plotted on the plane spanned by the second and third principal components. (The coefficient of the first principal component is not plotted here because it did not differ significantly between the two data sets.) Here it can be seen that the point clouds from the two data sets clearly lie in separate regions of the space.

Writing $\mathbf{W}_{sand}$ and $\mathbf{W}_{gravel}$ as linear combinations of principal components reduces the classification problem from m dimensions (typically 200 or more) to k dimensions (typically less than 20). To reduce the comparison further, a single scalar metric is desired which captures the difference between the two terrains. Such a scalar value is referred to as the discrimination metric. The discrimination metric presented here is the dot product with the vector between the class means, in a scaled version of the signal space.

To determine an appropriate scaling of the k-dimensional signal space, it is important to examine the distribution of points within a terrain class. This is accomplished by performing a second singular value decomposition, this time on a matrix $\hat{\mathbf{W}}$ formed by merging mean-adjusted matrices $\hat{\mathbf{W}}_{sand}$ and $\hat{\mathbf{W}}_{gravel}$:

$$\hat{\mathbf{W}}_{sand} = \mathbf{W}_{sand} - \overline{\mathbf{w}}_{sand}\begin{bmatrix}1 & \cdots & 1\end{bmatrix} \tag{10}$$

$$\hat{\mathbf{W}}_{gravel} = \mathbf{W}_{gravel} - \overline{\mathbf{w}}_{gravel}\begin{bmatrix}1 & \cdots & 1\end{bmatrix} \tag{11}$$

$$\hat{\mathbf{W}} = [\hat{\mathbf{W}}_{sand} \quad \hat{\mathbf{W}}_{gravel}] \tag{12}$$

$$\hat{\mathbf{W}} \xrightarrow{\text{SVD}} \left(\mathbf{U}_b, \mathbf{S}_b, \mathbf{V}_b^{\mathbf{T}}\right). \tag{13}$$

Here, each column of $\mathbf{V}_b^{\mathbf{T}}$ is associated with a particular time segment; each row corresponds to a combination of principal components (i.e. a linear combination of frequency components within the signal space). More importantly, due to the unitary nature of $\mathbf{V}_b^{\mathbf{T}}$ the norm of each row of $\mathbf{V}_b^{\mathbf{T}}$ is equal, so the standard deviations are equal across all k dimensions. The scaling of the space which produced $\mathbf{V}_b^{\mathbf{T}}$ is therefore appropriate for classifying the data. $\mathbf{V}_b^{\mathbf{T}}$ can be written as

$$\mathbf{V}_b^{\mathbf{T}} = \mathbf{S}_b{}^{-1}\mathbf{U}_b{}^{\mathbf{T}}\mathbf{W}. \tag{14}$$

Thus, the transformation from the signal space to this scaled signal space can be written as $\mathbf{S}_b{}^{-1}\mathbf{U}_b{}^{\mathbf{T}}$.

The discrimination metric $d(\mathbf{w})$ is thus defined using a dot product in this scaled signal space:

$$d(\mathbf{w}) = (\mathbf{S}_b{}^{-1}\mathbf{U}_b{}^{\mathbf{T}}(\overline{\mathbf{w}}_{sand} - \overline{\mathbf{w}}_{gravel})) \cdot (\mathbf{S}_b{}^{-1}\mathbf{U}_b{}^{\mathbf{T}}\mathbf{w}) \tag{15}$$

for an arbitrary vector $\mathbf{w}$ in the signal space. Writing the discrimination metric in terms of an arbitrary vector $\mathbf{y}$ in the frequency space (where $\mathbf{y}$ can be the log PSD of a data segment):

$$d(\mathbf{y}) = (\mathbf{S}_b{}^{-1}\mathbf{U}_b{}^{\mathbf{T}}(\overline{\mathbf{w}}_{sand} - \overline{\mathbf{w}}_{gravel})) \cdot (\mathbf{S}_b{}^{-1}\mathbf{U}_b{}^{\mathbf{T}}(\mathbf{S}_{signal}{}^{-1}\mathbf{U}_{signal}{}^{\mathbf{T}}\mathbf{y})). \tag{16}$$

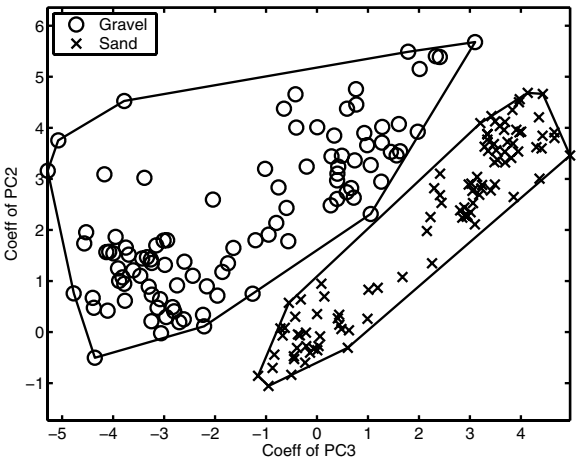

Fig. 3. Sample gravel and sand training data plotted on the plane spanned by the second and third principal components, shown with convex hull of each data set

Rewriting the dot product as a matrix multiplication yields

$$d(\mathbf{y}) = (\overline{\mathbf{w}}_{sand} - \overline{\mathbf{w}}_{gravel})^{\mathrm{T}} \mathbf{U}_b \mathbf{S}_b^{-1} \mathbf{S}_b^{-1} \mathbf{U}_b^{\mathrm{T}} \mathbf{S}_{signal}^{-1} \mathbf{U}_{signal}^{\mathrm{T}} \mathbf{y} \quad (17)$$

of which all but the last multiplication may be precomputed without knowledge of the vector to be classified. This precomputed row vector is labeled $\mathbf{d}$, and is a compact representation of the *a priori* training data for a pair of terrain classes:

$$\mathbf{d} = (\overline{\mathbf{w}}_{sand} - \overline{\mathbf{w}}_{gravel})^{\mathrm{T}} \mathbf{U}_b \mathbf{S}_b^{-1} \mathbf{S}_b^{-1} \mathbf{U}_b^{\mathrm{T}} \mathbf{S}_{signal}^{-1} \mathbf{U}_{signal}^{\mathrm{T}}. \quad (18)$$

Here, $\mathbf{d}$ will be referred to as the "discrimination vector." This is the linear combination of Fourier coefficients which best discriminates between the two terrain classes. Writing the discrimination metric in terms of $\mathbf{d}$ gives

$$d(\mathbf{y}) = \mathbf{d}\ \mathbf{y}. \quad (19)$$

The last step in the *a priori* analysis is to compute the statistics of the discrimination metric for the training data. Row vectors of discrimination metrics are computed as:

$$d(\mathbf{Y}_{sand}) = \mathbf{d}\ \mathbf{Y}_{sand} \quad (20)$$

$$d(\mathbf{Y}_{gravel}) = \mathbf{d}\ \mathbf{Y}_{gravel}. \quad (21)$$

The means and standard deviations of these metrics are computed as $\overline{d}_{sand}$, σ_{sand}, $\overline{d}_{gravel}$, and σ_{gravel}.

The discrimination vector and the terrain class statistics are stored for use in the on-line classification phase of the algorithm.

B. On-Line Classification

During a rover traverse, short segments of vibration sensor data are collected, of the same duration as those use in *a priori* training. For each segment the power spectral density is computed, and the magnitude is log-scaled and stored in a vector $\mathbf{y}$.

Pairwise classifiers then compute the discrimination metric $d(\mathbf{y})$ corresponding to the vibration to be classified. The Mahalanobis distances [22] from $d(\mathbf{y})$ to the terrain class means, e.g. $\overline{d}_{sand}$ and $\overline{d}_{gravel}$, are then computed as:

$$md_{sand}(\mathbf{y}) = \frac{\left| d(\mathbf{y}) - \overline{d}_{sand} \right|}{\sigma_{sand}} \quad (22)$$

$$md_{gravel}(\mathbf{y}) = \frac{\left| d(\mathbf{y}) - \overline{d}_{gravel} \right|}{\sigma_{gravel}}. \quad (23)$$

If the difference between the Mahalanobis distances is less than one (i.e. $\left| md_{sand}(\mathbf{y}) - md_{gravel}(\mathbf{y}) \right| < 1$), the pairwise classifier labels the vibration as "unknown." Otherwise, the pairwise classifier labels the vibration as the terrain with the smaller Mahalanobis distance.

A voting scheme merges the results of the various pairwise classifiers. In this approach, each pairwise classifier may return a terrain label, or it may return "unknown." If a pairwise classifier returns a positive vote for a terrain class, the alternative terrain class receives a negative vote. If the pairwise classifier returns "unknown," both classes receive an "unknown" vote.

For a terrain to be positively identified, it must 1) receive more positive votes than any other terrain class, 2) receive only positive and "unknown" votes, and 3) receive more positive votes than "unknown" votes. These rules were chosen to provide a conservative estimate that would not become drastically more or less conservative with an increased number of terrain classes. This is based on the belief that returning "unknown" is preferable to returning the wrong terrain class.

IV. Experimental Results

The algorithm presented above was experimentally validated on the Field and Space Robotics Laboratory (FSRL) Wheel-Terrain Interaction Testbed [8]. The FSRL Wheel-Terrain Interaction Testbed, shown in Fig. 8, consists of a driven wheel mounted on an undriven vertical axis. The wheel-axis assembly is mounted on a driven carriage, so the wheel forward velocity and angular velocity can be controlled independently. These testbed experiments were conducted using a wheel from the FIDO rover [23] supplied by the Jet Propulsion Laboratory. For these experiments,

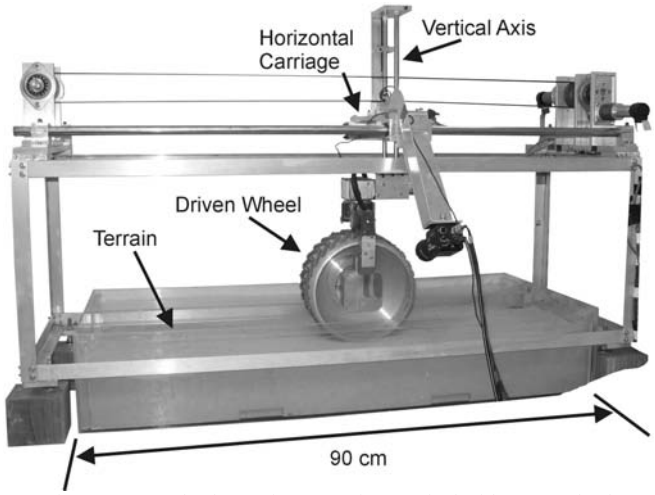

Fig. 8. FSRL Wheel-Terrain Interaction Testbed with FIDO wheel

three terrains were used: landscaping gravel, JSC Mars-1 Soil Simulant [24], and washed beach sand. Landscaping gravel is a mixture of small rounded pebbles ranging in size from 0.5 cm to 2 cm. JSC Mars-1 Soil Simulant is a dry glassy volcanic ash, developed by Johnson Space Center to represent the Martian soil as observed by Viking Lander 1. It contains fine particles as well as larger solid particles ranging up to 4 cm. Washed beach sand is a homogeneous fine-grained dry sand.

In these experiments, wheel forward velocity ranged from 0.5 cm/sec to 5 cm/sec, with these values chosen to be similar to planned rover missions. Forward velocity was set at a constant value for each trial. The wheel slip ratio i, was varied from 0 to 0.5. (The slip ratio is defined as $i = 1 - V / r\omega$, where V is the wheel forward velocity, r is the wheel radius, and ω is the wheel angular velocity.) The vertical load on the terrain was varied as well, from 30 N to 50 N including the weight of the wheel. This variation captures the effect of weight distribution among rover wheels due to travel over uneven terrain.

Vibration signals were sensed using an accelerometer mounted to the wheel frame, as shown in Fig. 9. These signals were collected using a desktop computer with a sound card. Sixteen-bit samples were collected at a frequency of 44.1 kHz. A single data set consisted of vibration data recorded over a full traverse of the wheel across the 90-cm-long testbed, at a specified load, forward velocity, and angular velocity.

Once all data were collected, the algorithm was tuned using the leave-one-out approach [25]. Tuning consisted of selecting appropriate values for 1) the range and spacing of frequency components for spectral representation, 2) the number of principal components used to represent the signal space, and 3) the discrimination thresholds for the pairwise classifiers. A single combination of tuned parameters is shared among all pairwise classifiers.

Once the parameters were tuned the classification accuracy was evaluated. First the vibration data was randomly divided into training data and test data sets. For each of the three terrains, ten data sets were randomly chosen as test data. This represents approximately 25% of the total data. The remaining data sets were chosen for training.

A three-terrain classifier was trained using the labeled training data sets. Here, a 1-second segment length was used. After the classifier was trained, it was used to classify the test data sets. Classification results are presented in Table I. Values shown are counts of 1-second-long vibration segments. These results are plotted in Fig. 10.

These results show the algorithm's ability to distinguish between multiple terrain types. When attempting to identify gravel-induced vibrations, the algorithm misclassified less than 1% of the test data as Mars-1 or sand. Similarly, when classifying Mars-1 and sand vibration data, less than 1% was misclassified as gravel. This clearly demonstrates the ability of the algorithm to identify terrains which induce obviously distinct vibrations.

The more challenging distinction was between Mars-1 and sand. These two terrains are alike in the fact that they contain small particles which may damp out vibrations in the wheel. Despite this similarity, less than 2% of the Mars-1 vibration data was misidentified as being sand. The difficulty of this distinction reveals itself in the amount of sand vibration data being misidentified as Mars-1. Nevertheless, these misclassifications comprise less than 20% of the sand vibration data, while most of the data is correctly classified.

Considering the inverse problem—having confidence that the actual terrain matches the classification result—the algorithm performs quite well. Given equal prior likelihoods of the above three terrains, the algorithm is more than 98% confident that terrain identified as gravel is actually gravel. Similarly, the algorithm is more than 97% confident that

Fig. 9. Vibration sensor mounted on the FIDO wheel in the FSRL Wheel-Terrain Interaction Testbed

TABLE I
CLASSIFICATION RESULTS FOR FSRL WHEEL-TERRAIN INTERACTION TESTBED VIBRATION DATA

		Classification Result				
		Gravel	Mars-1	Sand	Unknown	Total
Actual Terrain	Gravel	302	2	0	8	312
	Mars-1	5	208	3	61	277
	Sand	0	51	139	86	276

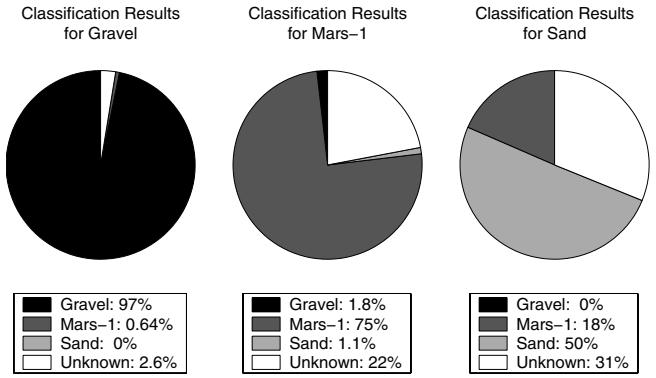

Fig. 10. Classification results for FSRL Wheel-Terrain Interaction Testbed vibration data

terrain identified as sand is truly sand. The confidence for Mars-1 is almost 80%.

It should be noted that these results are based solely on 1-second samples of vibration data, and incorporate no memory of prior classifications. An intelligent algorithm on a rover might incorporate an estimate of the likelihood of a transition from one terrain to another to improve overall classification results. Another way to improve terrain classification accuracy would be to combine the vibration-based classification with visual classification methods.

The algorithm required nine minutes of computation on a Pentium III 933 MHz computer, for the *a priori* training phase, using 43 minutes of labeled training data. On-line classification was much quicker, taking 202 ms per 1-second vibration segment, using unoptimized Matlab code. An optimized version of the code would be several times faster in both training and classification.

Additional tests on the Wheel-Terrain Interaction Testbed demonstrated the algorithm's capability to detect subsurface changes in terrain. Specifically, the algorithm detected the presence of a rock under a thin layer of sand. Such terrain features greatly affect the mobility of a rover and cannot be detected through traditional visual sensing.

V. CONCLUSION

An algorithm has been presented for classifying terrain based on the vibrations in the rover structure induced by wheel-terrain interaction. This algorithm uses linear discriminant analysis to distinguish between each pair of terrain classes, and uses a voting algorithm to arbitrate between pairwise classifiers. The terrain class is returned if the algorithm can uniquely identify one; otherwise an "unknown" result is returned.

Experimental results demonstrate the classification accuracy using vibration data. Classification of three distinct terrain types was demonstrated on the FSRL Wheel-Terrain Interaction Testbed. Additional experiments have shown the capability of detecting subsurface terrain changes with this algorithm.

These results show the potential for vibration-based classification as a standalone terrain classifier, or as a complement to current vision-based terrain classification approaches. The presented algorithm is a simple, inexpensive, and computationally efficient method for extracting terrain class information from vibration data. It may be used as a component of a meta-classifier, combining data from multiple sensors along with a memory of past classification results, or a similar approach may be used to define terrain class probabilities in a Bayesian classifier. This is an area of current research.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of Drs. Richard Volpe and Samad Hayati at NASA's JPL.

REFERENCES

[1] R. Volpe, "Rover Functional Autonomy Development for the Mars Mobile Science Laboratory," in *Proc. 2003 IEEE Aerospace Conf.,* Big Sky, MT, 2003. pp. 2:643-652.

[2] P. Bellutta, R. Manduchi, L. Matthies, K. Owens, and A. Rankin, "Terrain Perception for DEMO III," in *Proc. IEEE Intelligent Vehicles Symposium*, Dearborn, MI, 2000, pp. 326-332.

[3] C. Rasmussen, "Combining Laser Range, Color, and Texture Cues for Autonomous Road Following," in *Proc. 2002 IEEE Conf. On Robotics & Automation,* Washington, DC, 2002, pp. 4320-4325.

[4] S. Singh and B. Digney, "Autonomous Cross-Country Navigation Using Stereo Vision," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-99-03, January 1999.

[5] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies, "Fast and Reliable Obstacle Detection and Segmentation for Cross-country Navigation," in *Proc. IEEE Intelligent Vehicle Symposium, 2002*, France, 2002, pp. 610-618.

[6] B. H. Wilcox, "Non-Geometric Hazard Detection for a Mars Microrover," in *Proc. AIAA Conf. on Intelligent Robotics in Field, Factory, Service and Space*, vol. 2., Houston, TX, 1994, pp. 675-684.

[7] K. D. Iagnemma, C. A. Brooks, and S. Dubowsky, "Visual, Tactile, and Vibration-Based Terrain Analysis for Planetary Rovers," in *Proc. 2004 IEEE Aerospace Conf.*, Big Sky, MT, 2004.

[8] C. A. Brooks, "Terrain Identification Methods for Planetary Exploration Rovers," Master's thesis, Dept. Mech. Eng., Massachusetts Inst. of Tech., Cambridge, MA, 2004.

[9] K. D. Iagnemma, H. Shibly, and S. Dubowsky, "On-Line Terrain Parameter Estimation for Planetary Rovers," in *Proc. 2002 IEEE Intl. Conf. on Robotics & Automation,* Washington, DC, 2002, pp. 3142-3147.

[10] M. G. Bekker, *Theory of Land Locomotion.* Ann Arbor: University of Michigan Press, 1956.

[11] S. Kang, "Terrain parameter estimation and traversability assessment for mobile robots," Master's thesis, Dept. Mech. Eng., Massachusetts Inst. of Tech., Cambridge, MA, 2003.

[12] K. D. Iagnemma and S. Dubowsky, "Terrain Estimation for High Speed Rough Terrain Autonomous Vehicle Navigation," in *Proc. SPIE Conf. Unmanned Ground Vehicle Technology IV*, Orlando, FL, 2002.

[13] D. Sadhukhan and C. Moore, "Online Terrain Estimation Using Internal Sensors," in *Proc. Florida Conf. on Recent Advances in Robotics*, Boca Raton, FL, 2003.

[14] D. Sadhukhan, "Autonomous Ground Vehicle Terrain Classification Using Internal Sensors," Master's thesis, Dept. Mech. Eng., Florida State Univ., Tallahassee, FL, 2004.

[15] I. T. Jolliffe, *Principal component analysis*. New York: Springer-Verlag, 1986.

[16] S. Balakrishnama, A. Ganapathiraju, (1998). Linear Discriminant Analysis—A Brief Tutorial. [Online] Available: http://www.isip.msstate.edu/publications/reports/isip_internal/1998/linear_discrim_analysis/lda_theory.pdf

[17] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annal of Eugenics*, vol. 7, 1936, pp. 179-188.

[18] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis.* San Diego: Academic Press, 1979.

[19] G. J. McLachlan, *Discriminant analysis and statistical pattern recognition*. New York: Wiley, 1992.

[20] P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method based on Time Averaging Over Short, Modified Periodograms," *IEEE Trans. Audio and Electroacoustics,* vol. AU-15, pp. 70-73.

[21] G. H. Golub and C. F. Van Loan, "The Singular Value Decomposition and Unitary Matrices," in *Matrix Computations*, 3rd ed., Baltimore, MD: Johns Hopkins University Press, 1996, pp. 70-71 and 73.

[22] P. C. Mahalanobis, "On the Generalized Distance in Statistics," in *Proc. National Institute of Sciences of India,* vol. 2(1), 1936, pp. 49-55.

[23] P. S. Schenker, "FIDO: a Field Integrated Design & Operations Rover for Mars Surface Exploration," in *Proc. 6th Intl. Symposium on Artificial Intelligence, Robotics, and Automation in Space,* Montreal, Canada, 2001.

[24] C. Allen, R. Morris, K. Jager, D. Golden, D. Lindstrom, M. Lindstrom, and J. Lockwood, "Martian Regolith Simulant JSC Mars-1," in *Proc. 29th Lunar and Planetary Science Conf.*, Houston, TX, 1998.

[25] T. Jaakolla and D. Haussler, "Probabilistic kernel regression models," in *Proc. 7th Intl. Workshop on Artificial Intelligence and Statistics,* San Francisco, 1999.