Terrain Classification Using Vibration Sensors: Theory and Methods

Edmond M. DuPont[†*], Emmanuel G. Collins, Jr.[†*],
Eric J. Coyle[†*], Rodney G. Roberts[†+]

[†]Center for Intelligent Systems Control, and Robotics
[*]Department of Mechanical Engineering
[+]Department of Electrical and Computer Engineering
Florida A&M University - Florida State University
2525 Pottsdamer St. Tallahassee, FL 32310

Abstract

To improve the response of an autonomous ground vehicle (AGV) as it seeks to complete an outdoor mission, the control system of the vehicle should vary as it encounters specific terrains such as sand, gravel, mud, or ice. To determine when to switch the control system it is beneficial to have accurate and fast online terrain classification. The earliest terrain classification algorithms were vision-based. However, vision sensors may lead to inaccuracy in certain situations, for example, when the terrain that the vehicle experiences cannot be visually detected due to superficial ground cover such as leaves, dry grass, or even a thin layer of water. Hence, vision-based terrain classification should be accompanied by vibration-based classification, similar to how a human driver perceives the terrain using both sight and feel.

This chapter presents both theory and methods for terrain classification using vibration sensors. First, it uses a specific result from terrain classification using 2D structured light sensors to experimentally demonstrate that the signature of a particular terrain is given by the magnitude of the spatial frequency response of the system. Next, it is shown that the speed of the vehicle and the vibration transfer function of the system define a map from the spatial frequency response to the frequency responses of the vibration sensors. Hence, the magnitudes of the latter frequency responses serve as speed-dependent terrain signatures. An algorithm based on a Probabilistic Neural Network is then developed for terrain classification using this vibration signature. The online computational burden of this algorithm is substantially reduced by using Principal Component Snalysis to obtain a lower dimensional signature vector. The resulting algorithm is demonstrated using data

from two very different AGVs, a relatively small ATRV-Jr with no suspension system and a much larger Experimental Unmanned Vehicle with a suspension system.

The remainder of this chapter introduces new concepts in terrain classification. First it is assumed that the vehicle's vibrational transfer function is known. Then the vibration measurements can be mapped back to the surface inputs of the vehicle tires. The result is an algorithm that is theoretically speed independent and is also capable of detecting when the vehicle's left and right tire sets are traveling on two different surfaces at once. Second, a staged classification approach is presented to improve the classification results using the previously presented algorithms. The chapter closes with future directions for research.

1 INTRODUCTION

Autonomous ground vehicles (AGVs) are expected to operate in collaboration with manned vehicles in operations including agricultural applications, search and rescue missions, reconnaissance and surveillance, supply and logistics, etc. These situations may require the vehicle to traverse a variety of off-road terrains that can adversely affect the vehicle's performance. Therefore, terrain-based adjustments to the vehicle's driving rules [1, 2] and control system [3] can improve its performance and prevent the robot from becoming disabled. As an illustration of off-road driving rules, consider an AGV traversing loose sand. By limiting the turn radius of the vehicle and not allowing the vehicle to lose momentum, the tires will be less likely to dig into the sand or experience a large amount of slip [1]. A terrain-dependent control system is currently available for the Land Rover LR3 [3]. This *Terrain Response System* allows the driver to manually adjust the vehicle's control system, including traction control, anti-lock braking, throttle response, transmission shift schedule and differential locking to achieve improved vehicle handling and performance on four sets of terrain groups: asphalt, grass/gravel/snow, mud and ruts, and sand.

Prior to automating terrain-dependent driving rules and control systems for AGVs, an automated terrain classification procedure must be integrated to identify the terrain. Initially, research for terrain classification focused almost exclusively on applying vision-based algorithms to infer the traversability and class of the terrain [4, 5]. The purpose was to identify the terrain prior to traversal to avoid hazardous terrains and obstacles. For example, Howard and Seraji introduced the Fuzzy Traversability Index Algorithm, which used visual intensity levels to determine the terrain characteristics such as the roughness, slope, and discontinuity [4]. These characteristics are used to determine whether the terrain is traversable in relation to the presence of ditches, rocks, or extreme slopes. Detection of the terrain characteristics becomes difficult and unreliable when shadows affect the ability to distinguish between the terrain

surface and obstacles located on the terrain. Another classification method by Vandapel avoids the effects of variability in light intensity by the use of statistical properties of 3D ladar sensor data to determine the surrounding terrains [5]. This method applied off-line statistical analysis techniques to ladar sensor data to segment a scene into three classes: vegetation, terrain rocks, and thin wires or tree branches.

Another approach to classifying the terrain is through the reactive responses felt by the AGV as it traverses the local terrain. Past research in terrain classification through "feel" characterized the terrain by estimating its cohesion and internal friction from measured parameters of the vertical load, torque, wheel sinkage, wheel angular speed, and wheel linear speed [6, 7]. These measurements were used in simplified forms of classical terramechanics equations to estimate terrain parameters for planetary rovers, which were compared to stored parameters of known terrains. The equations on which the method is based are effective at low speeds and may encounter inaccuracies in parameter measurements, such as wheel sinkage in non-granular type terrains. There have been efforts to improve the estimation for use on higher speed vehicles by incorporating vision and auditory based classification with the terrain parameter estimation technique [8].

This chapter details approaches that characterize the terrain by the measured frequency response of the vehicle vibrations. It demonstrates that the vehicle vibrations are correlated to the terrain type and a terrain signature is given by extracted magnitude frequency responses measured from vibration sensors. The frequency response approach, initially suggested in [8] and first developed in [9, 10, 11], treated the vehicle as a particle with only one degree of freedom. This classification was carried out statistically by applying the Probabilistic Neural Network (PNN) classifier. Similar results for classifying sand, dirt, and gravel were obtained by applying linear discriminant analysis to the power spectral density of vibration data [12]. However, treating the vehicle as a particle results in poor performance at certain speeds and on certain terrains, especially when the vibration amplitudes are low.

Recent research also classified and characterized the terrain using individual types of sensor modalities [13]. The experiments were conducted using Activmedia's skid steered Pioneer 3-AT robot platform, with sensor modalities that included an inertial navigation system, motor current sensors, encoders, and a microphone sensor. These sensors measured the vehicle responses as the robot traversed grass, sand, gravel, and pavement. Experiments were conducted using data collected as the robot traversed along a 4 m $\times$ 4 m square path at a single speed of 30 cm/s. A multilayered feed-forward network classification system was applied to the extracted discrete Fourier transform frequency components from each individual sensor modality to return a terrain type. Implementation of this approach requires a separate neural network for each sensor modality. In this aspect, each network would return a separate terrain type, which requires an additional classification fusion algorithm to make

a final determination of the underlying terrain type. The fusion of data from all sensor modalities into a single multilayered feed-forward neural network results in a considerable increase in memory and computation requirements, which will result in slow classification.

This chapter is organized as follows. Section 2 identifies the input terrain signature applied to the wheels of an AGV using a 2D structured light sensors and describes its relationship to the robot's reactive vibrations. Section 3 uses the measured reactive vibrations to develop a speed dependent classification system to determine the terrain at a single trained speed. Section 4 improves this classification system by developing a multiple speed classification system that enables computationally efficient classification for a given number of speeds. Section 5 then presents a preliminary algorithm for multiple speed classification that uses interpolation to reduce the number of speeds used in training. Section 6 compares the algorithms of Sections 3, 4, and 5 in terms of memory requirements and computational time, important metrics for on-line implementation. Section 7 proposes a new approach to terrain classification in which the reactive terrain signatures are mapped back to each input wheel of the AGV using the vibrational transfer function. This approach theoretically leads to classification at multiple speeds while training at only one speed and also has the ability to determine whether the left-side and right-side tires are traversing different terrains. Section 8 discusses future work. Finally, Section 9 presents conclusions.

2 TERRAIN SIGNATURE RESPONSE

As an AGV traverses a terrain at a given speed, it exhibits a distinct vibrational reaction based on this underlying terrain. As the terrain or vehicle speed changes, these vibrations change. This section provides an explanation of this phenomena based on the spatial Fourier transform of the surface countour of the underlying terrain. This countour was experimentally observed in this analysis for various terrains using the Carnegie Mellon University Laser Line Striper [14], a high resolution 2D structured light sensor. The analysis then proceeds by considering a model of the vibrational transfer function of the AGV.

Though the purpose of using vision sensors here is to provide the fundamental justification for vibration-based classification, the results of this section can actually be used for vision-based terrain classification. However, it should be kept in mind that vision-based classification becomes unreliable, such as, cases when the underlying terrain is overlaid with superficial ground coverings, such as leaves. Hence, it should be complemented with vibration-based classification, the subject of this chapter.

2.1 Terrain Profile

Capturing the surface profile of a terrain can provide significant features that can be used for classification of the traversed terrain. A 2D structured light sensor, specifically the Laser Line Striper, developed at Carnegie Mellon University (Figure 1) was mounted on Activmedia's Pioneer 3-AT mobile robot. As this robot traversed various terrains that included sand, grass, gravel, and asphalt, images are captured from the Laser Line Striper. An

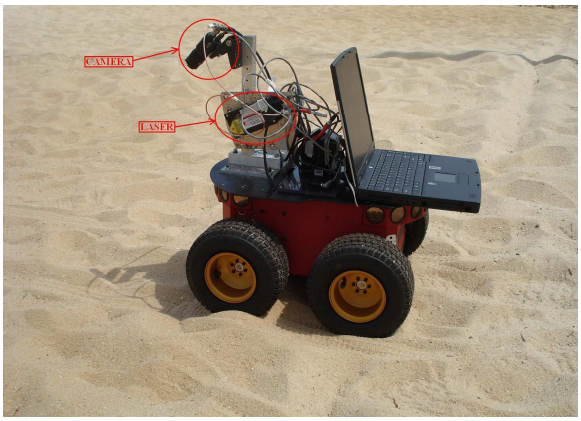
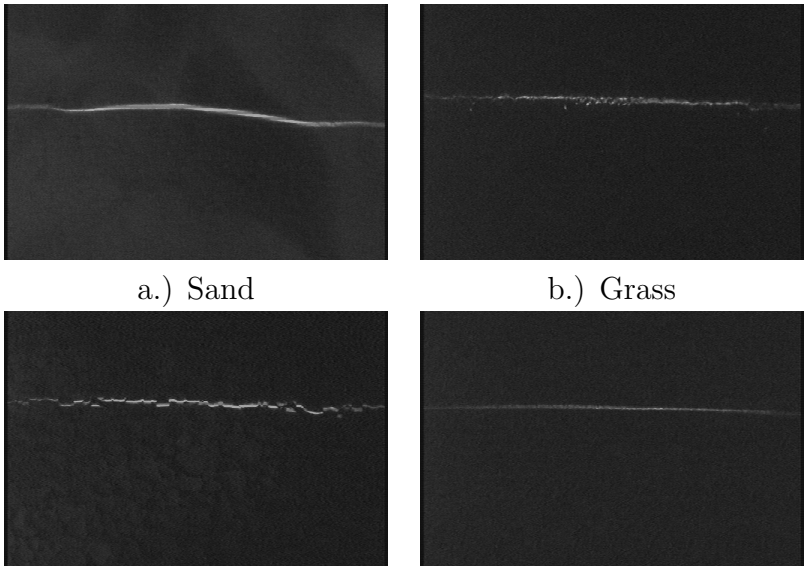


Figure 1: The CMU Laser Line Striper Mounted on Activmedia's Pioneer 3-AT Mobile Robot

example of the raw image data collected from the Laser Line Striper as it traversed the terrains is shown below in Figure 2. The surface profiles were extracted from each image in Figure 2 using the image data and triangulation of the known mount location of the laser and camera. Effectively this results in each pixel being spaced by approximately 0.83mm. The resulting surface profiles are shown in Figure 3. Applying the Fast Fourier Transform (FFT) on the resulting profiles, the magnitude spatial frequency response of these profiles was extracted as shown in Figure 4. From this figure the unique signature of each terrain can be easily distinguished by observing the differences in frequency response magnitude for each terrain. It can be shown that in general, these signatures are consistent along a given terrain and are sufficient for classification purposes [15].

2.2 Relationship Between Vehicle and Terrain

To illustrate the concept of how the terrain contour affects the reactive vibrations of an AGV, consider an AGV modeled as a linear, open-loop vibrational system with transfer function $G(s)$. $X(t)$, the system inputs, are the surfaces under each wheel and $Y(t)$, the system outputs, are the corresponding AGV vibrations. This relationship is stated in both the Laplace and frequency



a.) Sand b.) Grass

c.) Gravel d.) Asphalt

Figure 2: The Characteristic Images of the Contours of Four Terrains

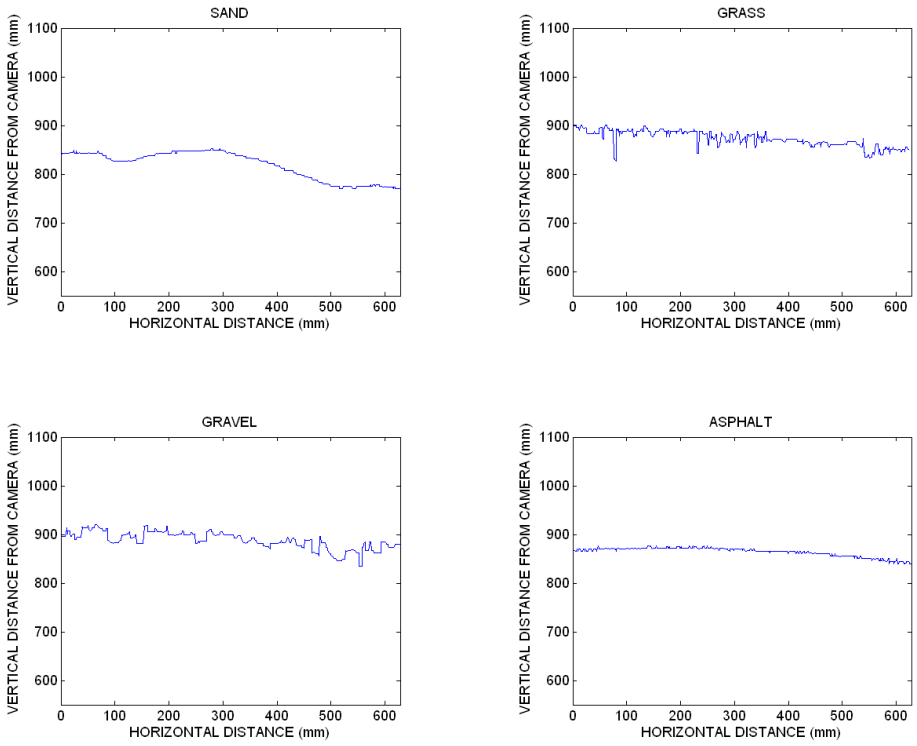


Figure 3: The Extracted Profiles of the Characteristic Images in Figure 2

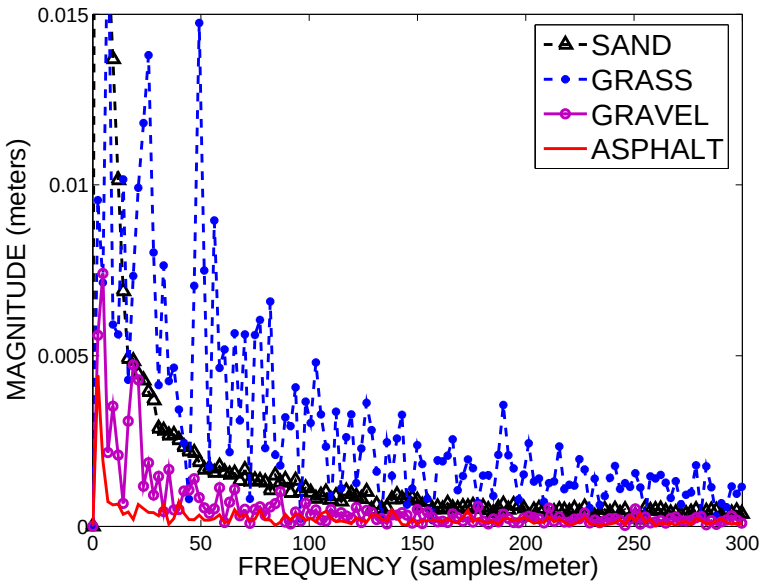


Figure 4: The Terrain Magnitude Spatial Frequency Response

domains as follows:

$$Y(s) = G(s)X(s), \tag{1}$$
$$Y(j\omega) = G(j\omega)X(j\omega). \tag{2}$$

The system's vibration outputs will contain characteristics resulting from the signatures in the surface profile of each terrain (input frequency response). It should be noted that these *spatial* input signatures of the terrain surface profile remain relatively consistent irrespective of changes in vehicle speed. However, changes in speed does influence the output reactive vibrations felt by the system, due to the influence of the vehicle model on the inputs.

A depiction of the model used to represent a four-wheeled AGV is shown in Figure 5. To develop a vibration model for the system using traditional modeling techniques, as in [16], several assumptions were used. First, the frame of this vehicle was treated as a rigid body with a center of mass in the geometric center. Second, the geometry of the tires was ignored and each tire was modeled as the combination of a linear spring and damper. Next, it was assumed that the wheels never lose contact with the ground and the direction of the contact force is assumed to always be in the vertical direction. Next, small angle approximations were used in order to make the model linear. For simplification the robot will be assumed to be moving in a straight line at constant speed. This allows the affects of turning, accelerating, and decelerating on the vertical velocity and angular rates to be ignored and all the wheels to

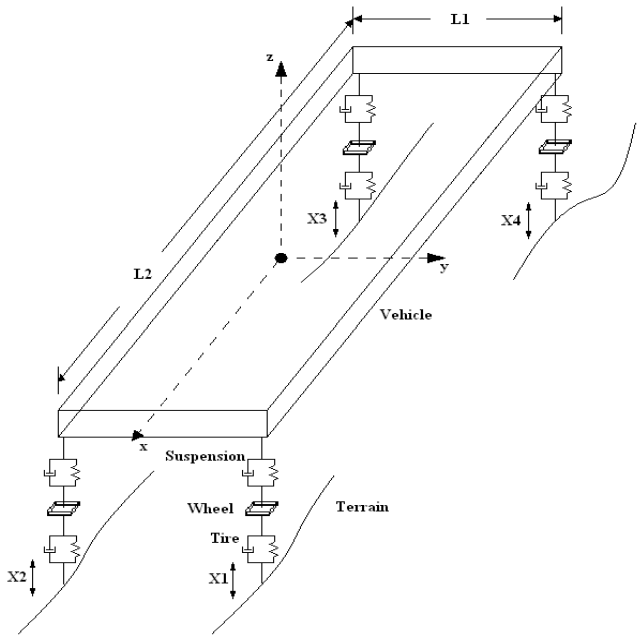


Figure 5: The AGV Model

be assumed to be moving at the same speed. The resulting model is given by

$$\begin{bmatrix} \ddot{z}(s) \\ \omega_y(s) \\ \omega_x(s) \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11}(s) & g_{12}(s) & g_{13}(s) & g_{14}(s) \\ g_{21}(s) & g_{22}(s) & g_{23}(s) & g_{24}(s) \\ g_{31}(s) & g_{32}(s) & g_{33}(s) & g_{34}(s) \end{bmatrix}}_{G(s)} \begin{bmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \\ X_4(s) \end{bmatrix}, \tag{3}$$

where the vertical displacements $X_1(s)$, $X_2(s)$, $X_3(s)$, and $X_4(s)$ of the wheels are linearly mapped to the z axis acceleration $\ddot{z}(s)$, the y axis rotation rate $\omega_y(s)$, and the x axis rotation rate $\omega_x(s)$ by the system transfer function $G(s)$.

Due to the irregular surface of the terrain, time-varying vertical displacements $X_1(t)$, $X_2(t)$, $X_3(t)$, and $X_4(t)$ occur at each of the wheels. If these displacements can be measured, then they can be used for terrain classification. However, it is impossible or extremely difficult to directly measure the vertical displacements using proprioceptive sensors. Instead the vibration of the vehicle as a single body is measured using proprioceptive sensors, such as an internal navigational unit [17]. The vibration due to the vertical wheel displacements is completely characterized by the vertical, roll, and pitch motions and here are considered to be the vertical acceleration $\ddot{z}(t)$, the pitch rate $\omega_y(t)$, and the roll rate $\omega_x(t)$. As a result, features from the output frequency responses of the vibration measurements will be used to train and test a classifier used for terrain classification.

2.3 Vibration-Based Terrain Features

This section explains the basic concepts and assumptions for vibration-based classification. It details the structure of the terrain signature features that are extracted and used for classifying the terrain.

Consider an AGV modeled as a vibrating system [18] with base excitation as illustrated in Figure 5, which shows the body-centered x-y-z axes. The reactive signature features are contained within the robot's translational acceleration $\ddot{z}(t)$, and angular velocities $\omega_y(t)$ and $\omega_x(t)$. A classifier can be developed to train on these signature features to later identify the terrain type. In the context of terrain classification, training involves the presentation of sample terrain induced sensor data from different terrain types recorded at different speeds. The sensor data, representative of the robot's reactive response along the terrain, is collected or extracted into time segments. Therefore, samples of the robot's translational acceleration $\ddot{z}(t)$, and angular velocities $\omega_x(t)$, $\omega_y(t)$ are collected into short time segments. The magnitude frequency components are extracted from each time segment sample and stored within a column vector. For example, consider one segment of time domain data along one terrain with extracted magnitude frequency components between the range of f_{min} and f_{max} for the l-dimensional vectors $\mathbf{t}_{\omega_x}$, $\mathbf{t}_{\omega_y}$, and $\mathbf{t}_{\ddot{z}}$ (l is the selected number of magnitude frequency components between f_{min} and f_{max}). A d-dimensional terrain signature vector $\mathbf{t}$ comprised of these magnitude frequency components is defined as

$$\mathbf{t} \triangleq \begin{bmatrix} \mathbf{t}_{\omega_x}^T & \mathbf{t}_{\omega_y}^T & \mathbf{t}_{\ddot{z}}^T \end{bmatrix}^T, \tag{4}$$

where $d = 3l$. Recall that since $\mathbf{t}$ is dependent upon the unknown transfer function $G(s)$, changes to the body of the vehicle, such as load variations or tire pressure, will lead to changes in $G(j\omega)$. This will in turn lead to a change in the signature vector $\mathbf{t}$. As an illustration, consider the case in which results from [19] are used to choose the mass, stiffness and damping parameters to correspond to the Pioneer 3-AT in Figure 1, which has a specified maximum load capacity of 25 kg. Figure 6 shows the variations in the magnitude of four of the elements of $G(j\omega)$ as the carried load, assumed to be concentrated above the center of gravity, varies from 0 kg to 25 kg. Obviously, these variations have the potential to affect classification based on $\mathbf{t}$.

In classifying the traversed terrain there are two possible ways to address effects such as changing load, tire pressure, etc. The most straightforward approach is to collect additional samples to train the algorithm under multiple expected load conditions. However for a robot with a large carrying capacity this could result in an unreasonably large training set. A second approach is to amplify or attenuate the elements of $\mathbf{t}$ based on the known load and tire pressures. If a reliable analytical model of the vehicle is known, this approach is preferable since the training set would require fewer samples, therefore resulting in faster online classification.

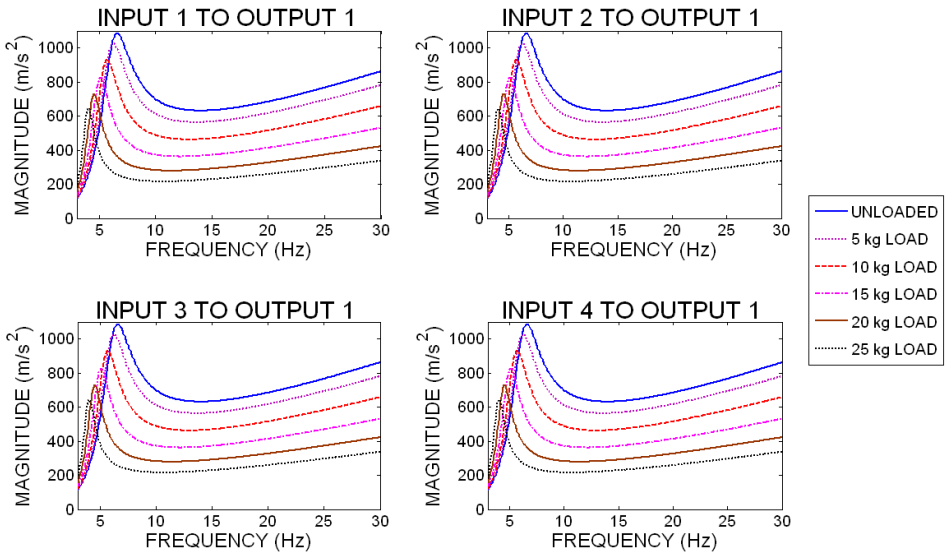


Figure 6: The Effects of Load (i.e., Mass) Changes on the Magnitude Frequency Response of the Vibrational Transfer Function $G(s)$

3 SPEED DEPENDENT TERRAIN CLASSIFICATION

As detailed in the previous section, the terrain signature is contained in the magnitude of the spatial frequency response of the terrain and the terrain determines the time-varying inputs to the wheels of the vehicle. The frequency responses of these inputs are directly related to the spatial frequency response of the terrain and are mapped via the vehicle's vibration transfer function to the responses measured by the vibration sensors. The results of this mapping are magnitude frequency responses from measured vehicle vibrations that serve as speed-dependent terrain signatures. This section assumes that the vibration transfer function is unknown and focuses on terrain classification using the responses measured by the vibration sensors.

In contrast to previous research, the algorithm developed here models the vehicle as a vibration system with three degrees of freedom. The FFT is applied to extract the magnitude frequency components, which serve as the terrain feature vector. Classification is carried out statistically on these signature features using a Probabilistic Neural Network (PNN) classifier, which can be implemented online with relatively fast computations. The developed classification algorithm is implemented experimentally on data collected from two autonomous ground vehicle platforms, which will be described later, at multiple speeds to show the efficacy of the proposed approach.

3.1 The Classification Algorithm

In determining a method to classify the terrains, the classification algorithm was developed using the statistical classification approach. The four fundamental elements of statistical based classification are the choice of the sensor inputs, preprocessing of the sensor data, feature extraction, and classification [20]. As previously mentioned, the sensor inputs were the vibration measurements ($\ddot{z}(t)$, $\omega_y(t)$, and $\omega_x(t)$) measured from an inertial navigation unit over a fixed time interval at a constant velocity. The preprocessing of the input involved extracting the magnitude frequency response components of the vibration measurements to create the feature vector $\mathbf{t}$. An unknown terrain feature vector is given as an input to the PNN classifier, which uses the previously stored training data to classify the feature vector as one of the trained terrains. A set of n training samples is taken at each speed s_j along

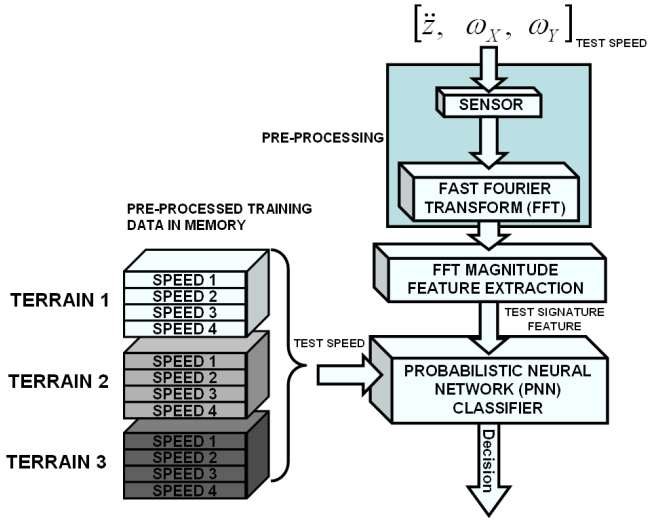


Figure 7: The Speed Dependent Classification System

each terrain t_i and stored in the training matrix

$$\mathbf{P}_{s_j,t_i} = \left[\ \mathbf{t}_{1,s_j,t_i} \quad \mathbf{t}_{2,s_j,t_i} \quad \dots \quad \mathbf{t}_{n,s_j,t_i}\ \right]. \tag{5}$$

Since models of the terrain and the robot are unknown, it is impossible to determine the expected response of the system. Therefore, multiple samples of the robot's response along each terrain are collected to provide an estimated model of the system's response. It is impossible to determine the exact number of samples required, as it is dependent on the classification problem. The training matrix is stored in the system's memory for use in the online training of the PNN classifier.

An illustration showing sample magnitude frequency responses of different terrains traversed by the robot platform is given in Figure 8. In particular, this

figure displays examples of the extracted magnitude responses obtained for an AGV traveling at a constant speed of 0.5 m/s over six surfaces that include packed gravel, loose gravel, sparse grass, tall grass, asphalt, and beach sand. Each terrain produced a unique FFT magnitude in the frequency range [3,30] Hz. However, for visual clarification the figures only show the responses in the interval [3,12] Hz. The FFT magnitudes of Figures. 8a, 8b, and 8c respectively define the vectors $\mathbf{t}_{\omega_x}$, $\mathbf{t}_{\omega_y}$, and $\mathbf{t}_{\ddot{z}}$, the components of the single terrain feature vector $\mathbf{t}$ defined by (4). The online classification procedure corresponds to determining a best fit for a new unknown input test vector $\mathbf{t}$ within a region of a high dimensional vector space. To enable some visualization of the

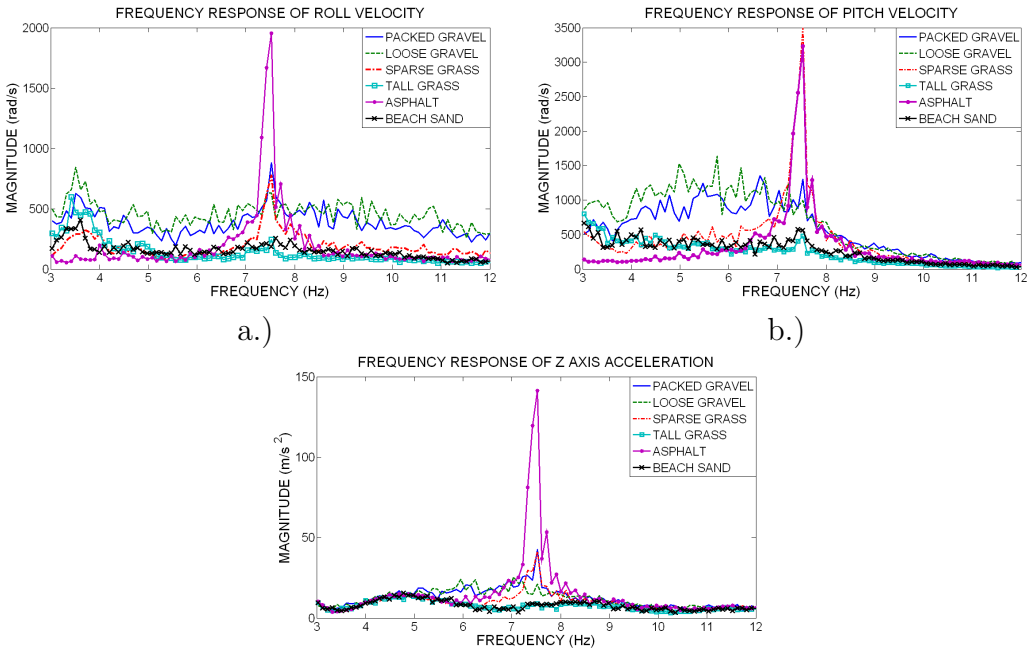


Figure 8: The Magnitude Frequency Response Of a) x-Axis Rotation Rate, b) y-Axis Rotation Rate, and c) z-Axis Acceleration Traversing Six Terrains.

classification process, Figure 9 shows an extracted index of the feature vectors $\mathbf{t}_{\omega_x}$, $\mathbf{t}_{\omega_y}$, and $\mathbf{t}_{\ddot{z}}$ corresponding to the frequency of 7.5195 Hz; this frequency was chosen because the magnitude frequency responses of Figures. 8a, 8b, and 8c tend to have a high amplitude at this frequency. (This phenomena is dependent the specific robot platform and this frequency is simply used here for visualization purposes.) Figure 9 presents a scatterplot representation for the six terrains with five samples for each terrain and clearly shows data clustering for each of the terrains. There is some separation between terrains, which is especially evident for asphalt, packed gravel, loose gravel, and sparse grass. It is also evident that beach sand and tall grass appear to be closely

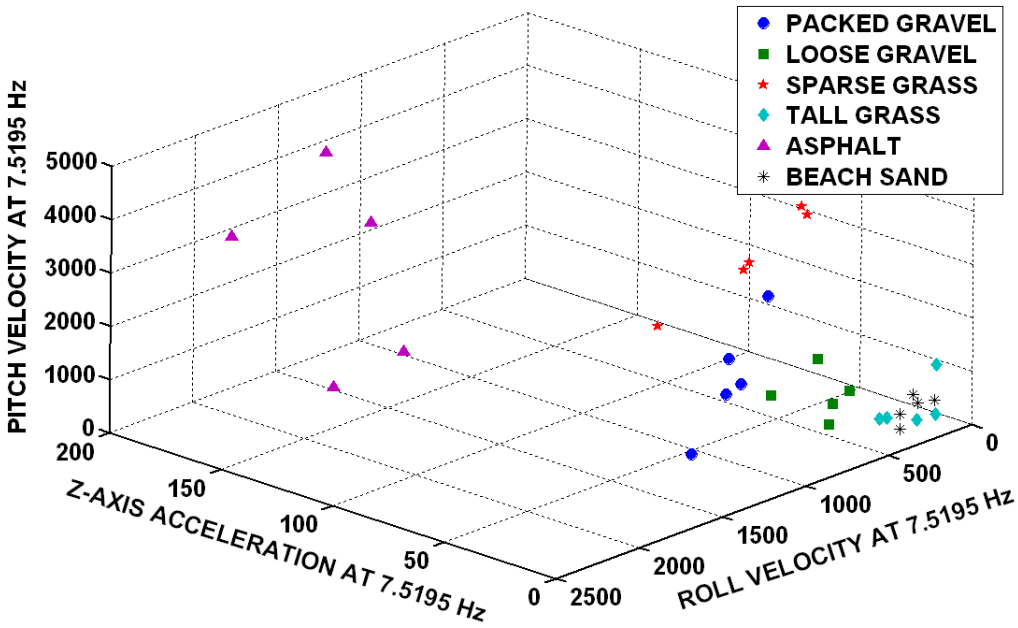


Figure 9: The Terrain Vector Space Point Cloud Representation Of the Six Terrains at The Dominant Frequency 7.5195 Hz.

adjoined. Also note that the actual dimension of the feature vector $\mathbf{t}$ is 831 ($= 277 \times 3$) for this data. In general, more separation between these two terrains is evident in the higher dimensional space with the inclusion of more distinguishing features.

3.2 Probabilistic Neural Network Classification

This section describes how the Probabilistic Neural Network (PNN) was used to classify an unknown test feature vector $\mathbf{t}$ as a particular terrain. Several tools such as the backpropagation multilayer perceptron, support vector machines, and other similar methods for pattern classification could have been used [21, 22]. The PNN was chosen because of its simplicity, robustness to noise, and online training [23].

The PNN [23, 24] is a pattern classifier that applies classical Bayesian optimal decision theory. The Bayes decision rule states that an unknown vector $\mathbf{t}$ is assigned to class c_i of a two category case as

$$h_i l_i P(c_i|\mathbf{t}) > h_j l_j P(c_j|\mathbf{t}), \quad i \neq j, \tag{6}$$

where h_k is the prior probability of occurrence, l_k is the loss associated with misclassifying an object to class k, and $P(c_k|\mathbf{t})$ is the probability density function (pdf) of $\mathbf{t}$ belonging to class k. The PNN simplifies this decision rule by assuming that the prior probability h_k and the loss function l_k are equal for all

categories; therefore, the decision is based entirely on the pdf, which reduces (6) to

$$P(c_i|\mathbf{t}) > P(c_j|\mathbf{t}), \quad \forall\, i \neq j. \tag{7}$$

A known limitation of (7) is that the pdf of $\mathbf{t}$ for all classes is not known, meaning (7) cannot be directly implemented. The PNN avoids this problem by applying the Parzen-windows approach, which uses random samples of the classes in conjunction with the unknown test sample to approximate the probability distribution functions [25, 26, 27, 28]. The feed-forward neural network structure of the PNN classifier, shown in Figure 10, consists of an input layer, pattern layer, summation layer, and output layer. The input layer

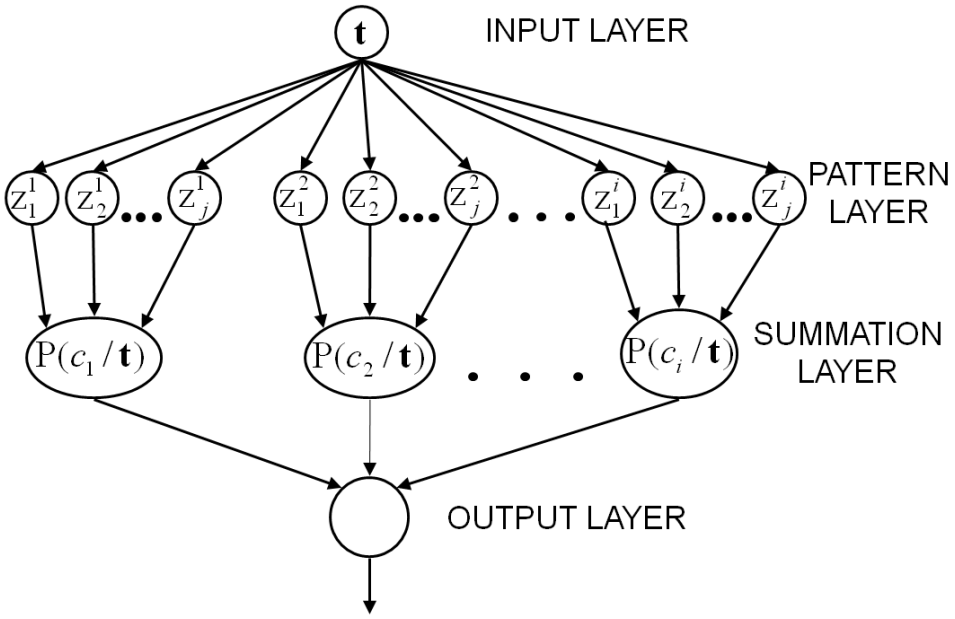


Figure 10: The Probabilistic Neural Network Structure

distributes the unknown test input vector $\mathbf{t}$ to the pattern layer after applying a weight $\mathbf{t}_j^i$ that is essentially the training vector j of class i yielding

$$z_j^i = \mathbf{t}^T \mathbf{t}_j^i, \tag{8}$$

where z_j^i represents the input to each pattern neuron j of class i.

The exponential function $\exp\left[-(z_j^i - 1)/\sigma^2\right]$ is applied to the input z_j^i as the nonlinear activation function instead of the commonly known sigmoid function used by typical multi-layered feed-forward networks. This activation function represents a Gaussian distribution centered at each training vector for each class.

The summation layer applies a unit scaled weight to each neuron received from the pattern layer and sums the inputs corresponding to the class of the

training features. The summation layer applies the Parzen-window estimator to compute the probability of a given input $\mathbf{t}$ belonging to a class c_i, with the nonlinear exponential as the window function. This results in the approximation,

$$P(c_i|\mathbf{t}) = \frac{1}{(2\pi)^{\frac{N}{2}}\sigma^N n_i} \sum_{j=1}^{n_i} \exp\left[\frac{(z_j^i - 1)}{\sigma^2}\right], \tag{9}$$

where $\sigma > 0$ is known as the smoothing parameter defining the window width, N is the dimension of the input vectors, and n_i is the number of sample patterns in class i. This represents an average of density estimates for each class that is then scaled by the reciprocal of the window function volume. The output layer uses the calculation of the pdf from the summation layer, and applies the decision rule of (7) to select the class with highest probability.

An issue associated with applying the PNN classifier is choosing a suitable estimate of the smoothing parameter σ. This parameter defines the decision boundaries between the trained categories. As σ increases, the decision boundaries becomes highly linear and can result in an overlap of closely related classes. On the other hand, a value of σ too small results in very non-linear boundaries behaving as the nearest neighbor classifier, which does not result in optimal class separation. It is not difficult to find a suitable value of σ and the misclassification rate is not significantly impacted with small changes in σ [29]. There exists different approaches to choosing a suitable smoothing parameter σ such as the cross-validation approach, in which σ is selected to maximize the performance across all classes within a predefined range for given test inputs. The inputs then consist of cyclically removing one training data sample to use as a testing input or having an entirely separate cross-validation dataset. In these initial experiments, the smoothing parameter was chosen empirically, but future experiments will implement the cross-validation technique.

3.3 Field Experimentation

The classification algorithm described in Section 2.3 was programmed in Matlab and implemented off-line. It was first applied to data collected from the ATRV-Jr, a differentially steered vehicle that has no suspension system, weighs about 50 kg, and has a maximum traversal speed of 1.4 m/s. Next, the classification algorithm was applied to data collected on the much larger XUV, a four wheel steered vehicle with an independent suspension system on each wheel that weighs approximately 1150 kg and has a maximum speed of about 40 mph, although the experiments were limited to speeds not exceeding 20 mph.

3.3.1 Experimental Results for the ATRV-Jr

The ATRV-Jr, shown in Figure 11a, was equipped with an onboard computer powered by a Pentium 3 800 MHz processor running the Red Hat Linux 6.2 operating system. Although various sensors were mounted on the ATRV-Jr robot platform, this classification research focused on data measured from the inertial measurement unit (IMU). As shown in Figure 11b, the IMU is mounted within the body of the robot centered underneath the onboard computer; it measures the vehicle's translational accelerations and the rotational velocities of the vehicle about the body fixed coordinate axes. As shown in Figure 12,

(a) (b)

Figure 11: a) The iRobot ATRV-Jr Robot Platform; b) The Mounted Crossbow IMU Sensor

the robot was commanded to traverse six different terrains: asphalt, packed gravel, loose gravel, tall grass, sparse grass, and beach sand. For each terrain the IMU recorded $\ddot{z}$, ω_y, and ω_x at ground speeds of 0.5 m/s and 1 m/s in 10-second intervals at a sample rate of 200 Hz resulting in 2000 time domain samples. A uniform windowed FFT was applied to each 10-second time signal and the magnitude frequency components were extracted. The transformation produced magnitudes at 277 of the total 2048 frequency points along each axes within the frequency range [3,30] Hz. This frequency range contained the dominant signature magnitude components across all the terrains. This resulted in a 831-dimensional feature vector for each terrain. These vectors were stored in memory as the training data for the PNN classifier.

Testing the algorithm consisted of inputting additional data of the robot traversing the test terrains for 150 seconds and measuring $\ddot{z}$, ω_y, and ω_x at the same sample rate as the training data. This data was also divided into 10 second segments and the frequency components were extracted similar to that of the training data. Each of the test samples were applied as inputs to the classification system to calculate the probability that the test feature vector belongs to a particular trained class. The algorithm produced 15 resulting

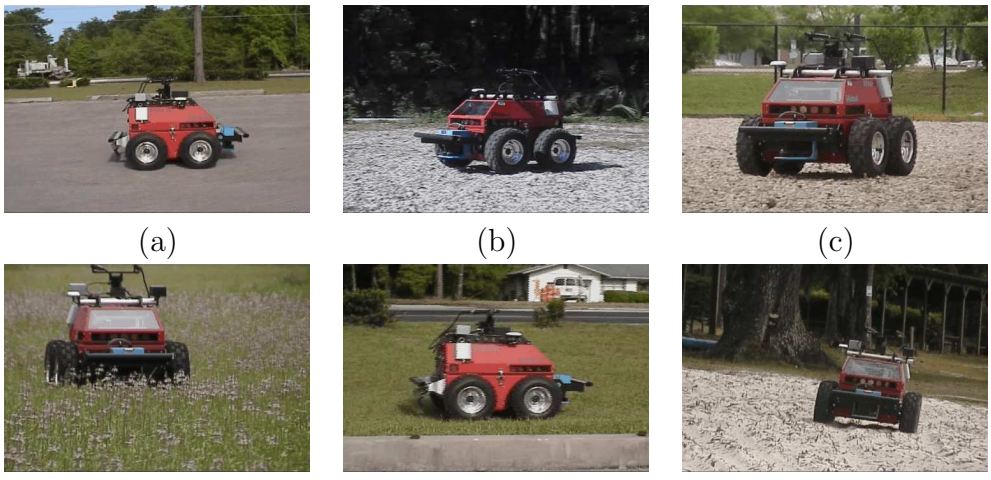


Figure 12: The ATRV-Jr Traversing a) Asphalt, b) Packed Gravel, c) Loose Gravel, d) Tall Grass, e) Sparse Grass, f) Sand

terrain classifications, given that the 150 second feature vector is evaluated every 10 seconds.

Tables 1 and 2 present the classification results for all of the terrains at the speeds of 0.5 m/s and 1 m/s, respectively. The results show a high ability to distinguish between the trained terrains. For example, at 0.5 m/s speed, the algorithm classified five of the six terrains with greater than 90% accuracy. At 1.0 m/s these same five terrains were classified with greater than 85% accuracy. Tall grass misclassified one of the input samples as beach sand at 0.5 m/s, but provided a perfect classification at 1 m/s. Although at 0.5 m/s, one sample of the sparse grass was misclassified as tall grass, the algorithm did classify this sample as a grass terrain. Similar classifications results were evident in distinguishing between the gravel terrains. In general the loose gravel provided more tire sinkage than the packed gravel. However, this effect was not consistent throughout the loose gravel area. As a result, the classification algorithm sometimes confused the two types of gravel surfaces. The asphalt and sand terrains resulted in perfect classification performance.

It should be noted that since the goal of this research is to alter the robot's driving and control based on terrain, it is not necessary to distinguish between all terrain types. Instead, it is sufficient to distinguish between terrain classes that require unique driving rules and control. For example, since the loose and packed gravel are nearly identical, they may not require dissimilar control. Nevertheless, they were treated separately in these experiments to test the algorithm's ability to distinguish between terrains that are nearly identical in appearance and roughness.

Table 1: Terrain Classification Results at 0.5 m/s.

	Detected Terrain (Accuracy Rate = 94.4%)					
Tested Terrain	Packed Gravel	Loose Gravel	Sparse Grass	Tall Grass	Asphalt	Sand
Packed Gravel	100% (15)					
Loose Gravel	26.7% (4)	73.3% (11)				
Sparse Grass			100% (15)			
Tall Grass				93.3% (14)		6.7% (1)
Asphalt					100% (15)	
Sand						100% (15)

Table 2: Terrain Classification Results at 1 m/s.

	Detected Terrain (Accuracy Rate = 92.2%)					
Tested Terrain	Packed Gravel	Loose Gravel	Sparse Grass	Tall Grass	Asphalt	Sand
Packed Gravel	86.7% (13)	13.3% (2)				
Loose Gravel	26.7% (4)	73.3% (11)				
Sparse Grass			93.3% (14)	6.7% (1)		
Tall Grass				100% (15)		
Asphalt					100% (15)	
Sand						100% (15)

3.3.2 Experimental Results for the XUV

The required XUV data was provided by its Inertial Reference Unit (IRU), which measured the rotation rates and accelerations about the three axes, at an update rate of 50 Hz. The XUV traversed asphalt, gravel, grass, mud, and dirt at speeds of 5 mph (2.2 m/s), 8 mph (3.6 m/s), 11 mph (4.9 m/s), 14 mph (6.2 m/s), and 17 mph (7.6 m/s), and 20 mph (8.9 m/s). With the goal of classifying terrains that require dissimilar control strategies, the grass and gravel terrains were combined into one terrain type, consistent with the Land Rover Terrain Response System [3]. Hence, the number of terrain types considered was actually four: asphalt, gravel/grass, mud, and dirt. The data collection for the XUV differed substantially from the data collection for the ATRV-Jr in that some of the data was collected while the vehicle was accelerating. This was necessitated by limits on the traversable distance at the experimental test grounds. Figure 14 shows the results of applying the classification system on the data collected and divided into 2 second samples. The number of samples varied depending on the speed and terrain traversed.

Figure 13: The XUV Traversing a) Asphalt, b) Mud, c) Dirt, d) Gravel, e) Grass.

The terrain classification accuracy was in the range [70%,100%] for each of the terrain/speed combinations except one (i.e., dirt at 20 mph which was classified at 65% accuracy). The reduction in classification performance on the XUV can be attributed to a reduced amount of terrain training samples used for classification compared to the ATRV-Jr. The classification of mud improved as the speed increased, possibly because on mud at the higher speeds the magnitude of the vehicle vibrations increased and hence the classification features became more distinct.

In this classification system, the classifier can sufficiently determine terrains only at speeds that were trained. Ideally, this is impractical since it would require the training data to encompass every conceivable speed along the various terrains. This is not only insufficient, it will drastically degrade the performance and classification time. The next section addresses this issue by applying a linear transformation on training data at multiple speeds and extracts a reduced set of features along all the terrains.

4 TERRAIN CLASSIFICATION AT MULTIPLE SPEEDS

The previous approach to classifying terrains yielded high accuracy classification, but only when classifying at the single speed that was trained. Based on the structure of the PNN classifier, this classification system computation time significantly increases as training data from multiple speeds is incorpo-

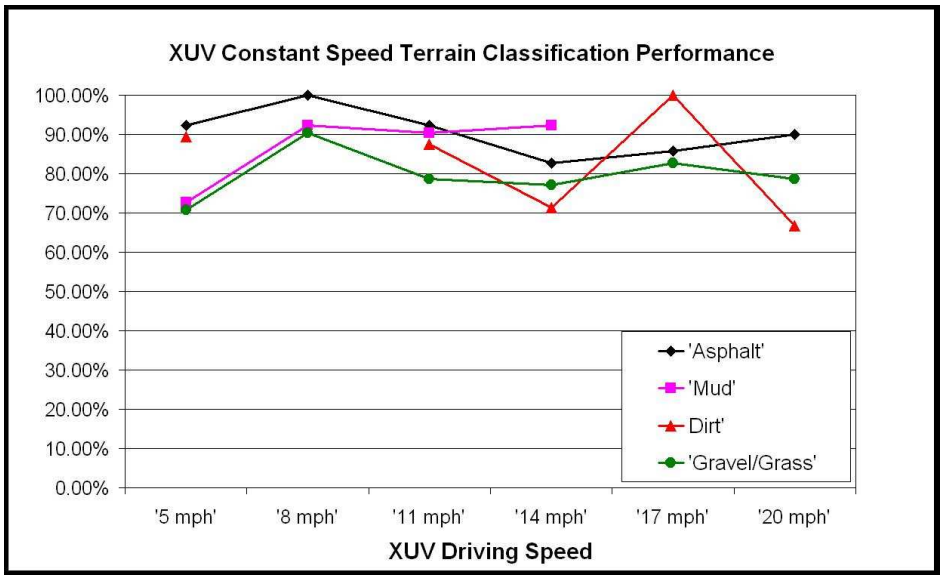


Figure 14: The Speed Dependent Classification System Results from the XUV.

rated into the PNN. For slow moving systems such as planetary rovers, which have relatively small speed ranges, this limitation is not necessarily a problem; however, for full scale vehicles with higher top speeds, the need to train at a large number of speeds, while retaining fast, accurate classification, is imperative. The algorithm presented below addresses this problem by applying Principal Component Analysis (PCA) [30] to the matrix of training feature vectors in order to obtain feature vectors of reduced dimension. The remainder of this section details this algorithm and presents experimental results on data collected from the ATRV-Jr and XUV.

4.1 Multiple Speed Classification System

The addition of an eigenspace feature extraction method to the classification system extends the off-line computations as illustrated in Figure 15. A feature extraction method called Principal Component Analysis (PCA) was applied to project all of the training magnitude feature vectors onto a new reduced orthogonal subspace. The training features for this system incorporates terrains at multiple speeds instead of a single speed.

4.1.1 Off-line Training

In the algorithm of Section 3 each terrain and speed combination was treated separately both in the training and classification phases. Hence as the number of speeds increases the online computational requirements for classification can increase dramatically. It is desirable to develop an on-line clas-

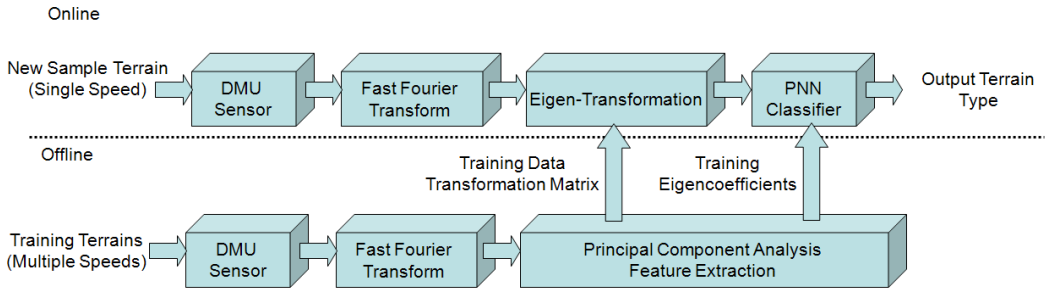


Figure 15: The Multiple Speed Classification System

sification algorithm that does not inherently increase in computation time as training data from multiple speeds are added.

Consider a set of n redundant samples of the terrain signature vector $\mathbf{t}$ collected for the terrain t_i at a select speed s_j, resulting in the matrix

$$\mathbf{P}_{s_j,t_i} = \left[\begin{array}{cccc} \mathbf{t}_{1,s_j,t_i} & \mathbf{t}_{2,s_j,t_i} & \dots & \mathbf{t}_{n,s_j,t_i} \end{array} \right]. \tag{10}$$

When m multiple speeds are collected per terrain, a matrix $\mathbf{T}$ for each terrain t_i becomes

$$\mathbf{T}_{t_i} = \left[\begin{array}{cccc} \mathbf{P}_{s_1,t_i} & \mathbf{P}_{s_2,t_i} & \dots & \mathbf{P}_{s_m,t_i} \end{array} \right], \tag{11}$$

where $\mathbf{P}_{s_j,t_i}$ represents samples at speeds s_j of terrain t_i. Combining all of the terrain matrices $\mathbf{T}_{t_i}$ $(t_i = t_1, t_2, \dots, t_\nu)$ results in the training matrix

$$\mathbf{D} = \left[\begin{array}{ccccc} \mathbf{T}_{t_1} & \mathbf{T}_{t_2} & \mathbf{T}_{t_3} & \dots & \mathbf{T}_{t_\nu} \end{array} \right] \in \mathbb{R}^{d \times r}, \tag{12}$$

where $r = nm\nu$ is the multiplication of the number of samples n, speeds s_m, and terrains t_ν and $d = 3l$ is the dimension of the terrain signature vector with three sensor modalities of dimension l. Recall, the dimension d is dependent on the sampling frequency used to collect the sensor data and the range of extracted magnitude frequency components scaled by the number of sensors modalities.

With a large number of high dimensional feature vectors for each terrain and speed, the overall training matrix becomes very large, which can affect the online computation performance. Fortunately, PCA can be applied to yield reduced-order feature vectors. In implementing PCA, an unbiased set of the training data $\hat{\mathbf{D}}$ was created by subtracting the mean vector $\bar{\mathbf{d}} \in \mathbb{R}^{d \times 1}$ of the training matrix $\mathbf{D}$ from each column of the training data $\mathbf{D}$ matrix as

$$\hat{\mathbf{D}} = \mathbf{D} - \bar{\mathbf{d}}\mathbf{1}_{1 \times r}, \tag{13}$$

where $\mathbf{1}_{m \times n}$ is an $m \times n$ matrix of ones. Computing the singular value decomposition of this unbiased training data yields orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ with a nonnegative diagonal matrix $\mathbf{\Sigma}$ such that

$$\hat{\mathbf{D}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{14}$$

Each diagonal component of $\mathbf{\Sigma}$ corresponds to a principal component vector (i.e., a column) $\mathbf{u}_i$ of $\mathbf{U}$. The amount of energy ξ associated with the first k principal components is given by

$$\xi = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{d} \sigma_i^2}, \tag{15}$$

where σ_i are the diagonal components of the matrix Σ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$. The parameter k serves as the reduced-order dimension and is chosen such that ξ corresponds to 80% to 90% of the energy. Define

$$\mathbf{U}_k = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_k \end{bmatrix} \tag{16}$$

and let $\mathbf{Y} \in \mathbb{R}^{k \times r}$ be given by

$$\mathbf{Y} = \mathbf{U}_k^T \hat{\mathbf{D}}, \tag{17}$$

where the columns of $\mathbf{Y}$, denoted by $\mathbf{y}_i$ are called the eigenspace coefficients and serve as the reduced-order feature vectors for the training data. They are the projections of the unbiased feature vectors $\hat{\mathbf{d}}_i$ (the columns of $\hat{\mathbf{D}}$) onto the space spanned by $\mathbf{U}_k$. Hence, the eigenspace coefficients are reduced-order representations of the terrain signatures in the orthogonal basis defined by the column space of $\mathbf{U}_k$. Each $\mathbf{y}_i$ corresponds to a specific terrain type and speed.

Due to the eigenspace orthogonality described above, the first three elements of the $\mathbf{y}_i$ can be plotted on orthogonal axes in a 3-dimensional scatter plot, which represents the eigenspace coefficients for $k = 3$. For example, consider a robot traversing asphalt, gravel, and grass at training speeds of 0.5 m/s, 1 m/s, and 1.4 m/s. The eigenspace representation with $k = 3$ is shown in Figure 16, which gives a visual representation of how the reduced-order feature vectors cluster according to terrain type and speed.

4.1.2 On-line Testing

For online classification of an unknown terrain, the procedure is similar to the previous speed dependent classification system. The transformation matrix $\mathbf{U}_k$ is applied to the online procedure to transform the test feature vector $\mathbf{t}_{test}$ corresponding to the speed $s_{test} \in \{s_1, s_2, \dots, s_m\}$ onto the reduced eigenspace of the training data, i.e.,

$$\mathbf{y}_{test} = \mathbf{U}_k^T \mathbf{t}_{test}. \tag{18}$$

The PNN classifier determines the terrain by comparing $\mathbf{y}_{test}$ with the $\mathbf{y}_i$ that correspond to the speed s_{test}.

4.2 Field Experimentation

The multiple speed algorithm was implemented on data collected from both the ATRV-Jr and the XUV. The experiments involved the vehicles traversing grass, gravel, and asphalt terrains.

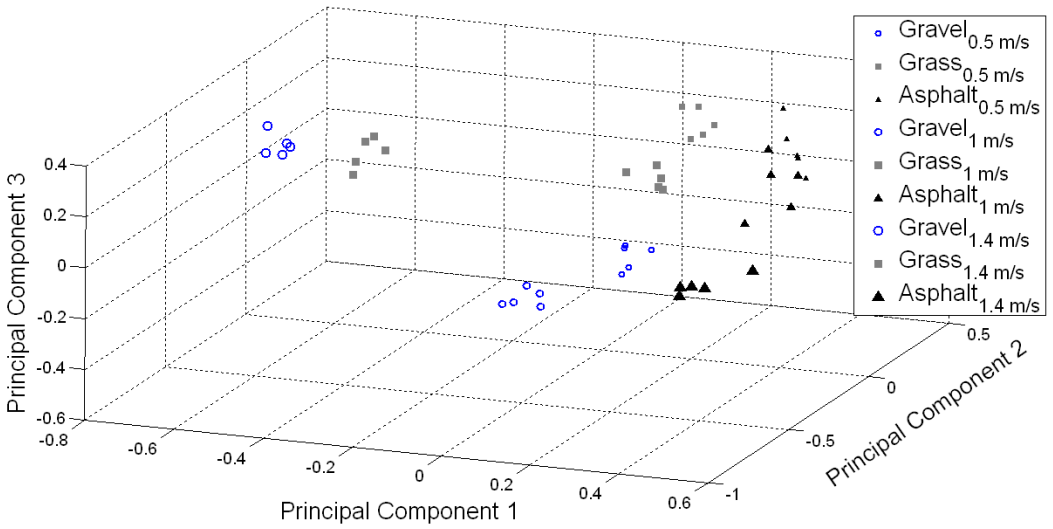


Figure 16: The Eigenspace Coefficients Of Three Terrains (Asphalt, Gravel, Grass) for $k = 3$.

4.2.1 Experimental Results for the ATRV-Jr

The experimental procedures were consistent with those of Subsection 3.3.1. However, in contrast to those experiments, the time segments of the data samples were 2 seconds instead of 10 seconds; this reduction reduces the overall computational time.

For each combination of speed and terrain 40 training samples were collected. An additional 35 samples were collected and used for implementation of cross validation to determine the smoothing parameter σ. The training data consisted of the eight defined speeds within the range [0.2 m/s,1.4 m/s] along each of the three terrain types. This resulted in a training matrix $\hat{\mathbf{D}} \in \mathbb{R}^{831\times960}$ and a cross validation dataset matrix in $\mathbb{R}^{831\times840}$. The eigenspace transformation matrix $\mathbf{U}_k$ and the eigenspace coefficient matrix $\mathbf{Y}$ were also computed. It is evident from Figure 16 that there exist some distinctions between the terrains at the different speeds. The addition of higher principal components improves this separation of the terrains but becomes difficult to depict for 3D visual evaluation. In our simulation experiments, the first $k = 25$ principal components were chosen to preserve approximately 90% of the training data energy. The resulting transformation matrix $\mathbf{U}_k$ then effectively reduced the training matrix $\hat{\mathbf{D}} \in \mathbb{R}^{831\times960}$ to the matrix of eigenspace coefficients $\hat{\mathbf{Y}} \in \mathbb{R}^{25\times960}$ using (17). Hence, the feature vectors were substantially reduced from dimension 831 to 25. The comparison of how this dimension reduction improved memory and computational time will be discussed later in Section 6.

In applying the offline cross validation, an estimated smoothing parameter of $\sigma = 0.15$ was determined to maximize the overall performance. The test samples consisted of 8 speeds within the speed range [0.2 m/s,1.4 m/s] along each of the three trained terrains. At each speed and terrain, a total of 75 unknown feature vectors $\mathbf{t}$ were computed, each representing time domain samples of 2 seconds each. During classification the positive classification and misclassifications were collected and are presented in Table 3. The rows of the table represent the classified terrains based on the input test terrain, represented as the columns of the table. For each of the 8 blocks in Table 3 the diagonal components correspond to positive classifications, while the off-diagonal components correspond to misclassifications. In each block the parenthetical entries in each column correspond to the number of samples and sum to 75 while the percentages sum to 100%. This table shows that for all speeds the the accuracy of the multiple speed classification system was in the range $[81.8\%, 97.8\%]$, which corresponds to efficient classification. There still

Table 3: Results of the Multiple Speed Classification System for the ATRV-Jr

Classified Terrains At Multiple Speeds	Input Terrains At Multiple Speeds												
		0.2 m/s (Accuracy Rate= 81.8%)			0.4 m/s (Accuracy Rate= 81.3%)			0.5 m/s (Accuracy Rate= 92.9%)			0.6 m/s (Accuracy Rate= 97.8%)		
		Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt
	Gravel	48.0% (36)			76.0% (57)			81.3% (61)	2.7% (2)		94.7% (71)		
	Grass	48.0% (36)	100.0% (75)	2.7% (2)	17.3% (13)	69.3% (52)	1.3% (1)	17.3% (13)	97.3% (73)		05.3% (4)	98.7% (74)	
	Asphalt	4.0% (3)		97.3% (73)	6.7% (5)	30.7% (23)	98.7% (74)	1.3% (1)		100.0% (75)		1.3% (1)	100.0% (75)
		0.8 m/s (Accuracy Rate= 84.9%)			1 m/s (Accuracy Rate= 86.7%)			1.2 m/s (Accuracy Rate= 90.2%)			1.4 m/s (Accuracy Rate= 89.8%)		
		Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt
	Gravel	89.3% (67)	25.3% (19)	2.7% (2)	90.7% (68)	30.7% (23)		90.7% (68)	16.0% (12)	1.3% (1)	88.0% (66)	14.7% (11)	4.0% (3)
	Grass	10.7% (8)	68.0% (51)		9.3% (7)	69.3% (52)		9.3% (7)	84.0% (63)	2.7% (2)	12.0% (9)	85.3% (64)	
	Asphalt		6.7% (5)	97.3% (73)			100.0% (75)			96.0% (72)			96.0% (72)

is some difficulty in distinguishing between the grass and gravel terrains. It is also seen that at certain speeds it may be challenging to distinguish asphalt from these two terrains.

4.2.2 Experimental Results for The XUV

The multiple speed classification system was also applied to data collected from the XUV robot platform. The robot traversed asphalt, gravel, and grass for 5 speeds within a range of [5 mph,17 mph] ([2.2 m/s,7.6 m/s]) in increments

of 3 mph. Due to the inability to collect many training samples for mud and dirt, these terrains were not included in these classification experiments. The training data was comprised of all the data within the range [2.2 m/s,7.6 m/s]. Both the training and testing data corresponded to 2 seconds of vibration measurements sampled at 50 Hz. The transformations (17) and (18) resulted in reducing the dimension of the feature vectors used for classification from 114 to 5.

Table 4: Results of the Multiple Speed Classification System for the XUV

Classified Terrains At Multiple Speeds		Input Terrains At Multiple Speeds														
		5 mph (2.2 m/s) Accuracy Rate = 75.9%			8 mph (3.6 m/s) Accuracy Rate = 73.5%			11 mph (4.9 m/s) Accuracy Rate = 80.0%			14 mph (6.2 m/s) Accuracy Rate = 66.7%			17 mph (7.6 m/s) Accuracy Rate = 86.7%		
		Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass
	Asphalt	100.0% (30)	40.0% (8)	13.8% (4)	94.4% (17)	15.4% (2)		90.0% (9)	12.5% (1)	08.3% (1)	57.1% (4)			60.0% (3)		
	Gravel		25.0% (5)			30.8% (4)	16.7% (3)		50.0% (4)			33.3% (2)			100.0% (4)	
	Grass		35.0% (7)	86.2% (25)	5.6% (1)	53.8% (7)	83.3% (15)	10.0% (1)	37.5% (3)	91.7% (11)	42.9% (3)	66.7% (4)	100.0% (8)	40.0% (2)		100.0% (6)

The classification performance is summarized in Table 4, which shows results for three terrains at the five trained speeds. At the slower speeds the asphalt terrain is more accurately classified than grass or gravel. The gravel terrain misclassified as either grass or asphalt at 5 mph (2.2 m/s), 11 mph (4.9 m/s), and 17 mph (7.6 m/s). This misclassification is probably due to the fact that there were fewer samples of the gravel terrain than for any of the other terrains; hence, there were not enough samples of gravel to accurately estimate the distribution function for this terrain, resulting in classification that is biased towards the other trained terrains. This can easily be overcome by simply taking more samples. Future experiments will include more terrains and multiple runs along the terrains to collect a larger number of samples. The use of different numbers of samples for different terrains tended to bias classification towards the asphalt terrain, which at all the speeds had more samples available than the other terrains. The classification accuracy of grass tended to increase as the speed increased, although the number of samples decreased. This shows that the vibration signature of grass may become more distinct as the speed increases. Although the performance along individual terrains at select speeds was inconsistent, the minimum accuracy rate was just over 65%. This shows that the classification system is able to classify with relatively good accuracy and including more samples should substantially increase the classification rates.

For the multiple speed classification system, it was shown that by transforming the feature matrix $\hat{\mathbf{D}}$ corresponding to the various terrains and multiple speeds using the eigenspace transformation $\mathbf{U}_k$, can yield a reduced order feature vector $\mathbf{Y}$ that retains the features needed for accurate classification.

This provides the ability to include a substantial number of speeds with relatively low online computations. The next section presents an interpolation scheme that can be used to reduce the number of training speeds.

5 MULTIPLE SPEED CLASSIFICATION USING INTERPOLATION

As an AGV traverses a terrain at speeds close to each other, the eigenspace coefficients $\mathbf{y}_i$ corresponding to these speeds are highly correlated. As a result, a discrete nonlinear manifold is formed for each terrain, representing the eigenspace coefficients as a function of speed. Using the eigenspace coefficients for a fixed number of speeds within a given range, a continuous manifold can be estimated for all speeds within this range using spline estimation. As illustrated by Figure 17, in this section the cardinal spline technique is applied to the means of the clustered data coefficients at the discrete speeds for each terrain in eigenspace. In this estimation technique, the PNN classifier is removed and replaced with a simple nearest neighbor based classifier that compares a test coefficient with the terrain manifolds.

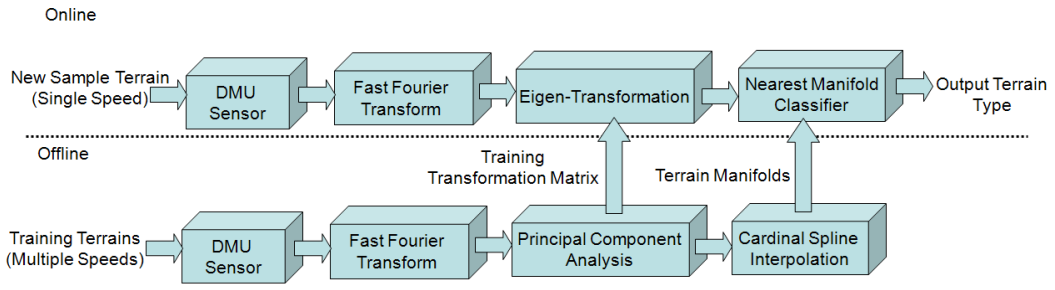


Figure 17: The Speed Estimation Classification Block Diagram

5.1 Classification Using Interpolation

The classification method presented here uses an approach presented by Murase and Nayar that determined the pose of an object within an image [31, 32]. This research utilized the pose and illumination characteristics to parameterize a 3-D object as it was rotated within a consecutive array of images. It was presented that consecutive images of the rotated object at relatively small angle changes were highly correlated and resulted in close proximity projection coefficients within eigenspace. The projection coefficients created a discrete manifold in eigenspace, in which a cubic spline or parametric estimation techniques [33] were then used to interpolate between the discrete points to create a hypersurface. For classification, a new unknown test object

image is projected and compared using the Euclidean distance to the different manifolds. Based on this premise, as an AGV traverses a terrain at relatively close speeds, the corresponding eigenspace coefficients of the extracted features should show correlation between the response of the vehicle to this change in speed.

5.1.1 Off-line Training

Recall the unbiased training data matrix $\hat{\mathbf{D}} \in \mathbb{R}^{d \times r}$, defined by (13), which is comprised of the magnitude frequency response feature vectors of multiple terrains at various speeds. Using PCA, the training data matrix is transformed onto a reduced eigenspace resulting in the matrix of eigenspace coefficients $\mathbf{Y} \in \mathbb{R}^{k \times r}$. The coefficients (i.e., the columns $\mathbf{y}_i$) of $\mathbf{Y}$ form numerous clusters relating samples taken at the same speed along the same terrain. In the preliminary results of this section the mean points were calculated and chosen to represent each of the respective clusters. These means represent the discrete points used to calculate the manifold estimated for each terrain.

The estimation of the manifold was applied using the cardinal (Catmull-Rom) spline technique [34]. This spline technique was chosen due to its implementation simplicity and the fact that it passes through all the given control points, which does not occur for all splines techniques. The cluster means are given as the control points of the cardinal spline technique. Figure 18 shows for three terrains an example of the spline estimated manifolds that pass through the cluster mean coefficients defined as the control points. (Note that Figure 18 corresponds to the first 3 elements of the principal components.)

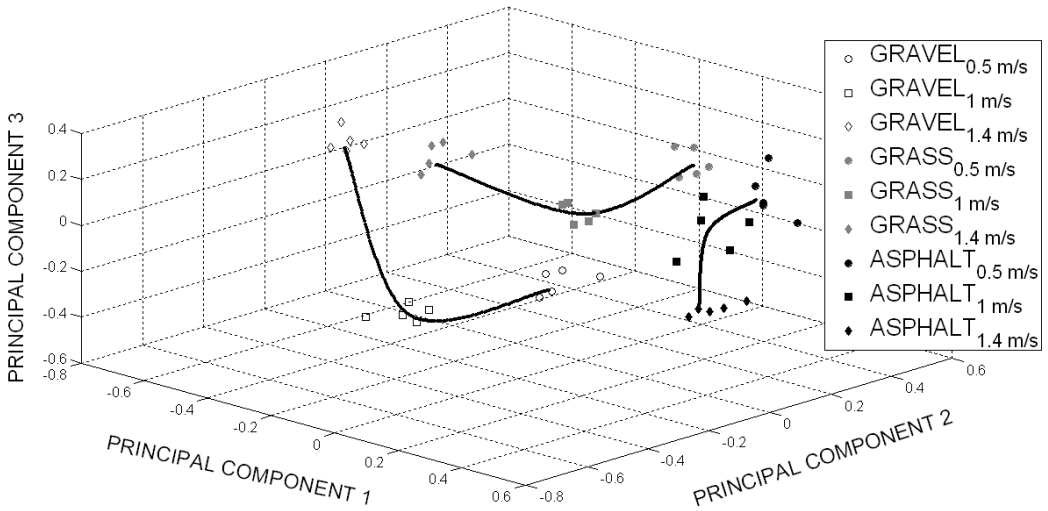


Figure 18: The Eigenspace Manifolds Corresponding to the Cluster Median Control Points of 3 Terrains.

5.1.2 Online Classification

The online classification procedure involves providing the vibration measurements $(\ddot{z}(t), \omega_y(t), \omega_x(t))$ over an unknown terrain for a specified time domain segment. As in the previous approaches, a magnitude frequency response feature vector $\mathbf{t}_{test}$ is calculated. Then, the reduced-order feature vector $\mathbf{y}_{test}$ is calculated using

$$\mathbf{t}_{test} = \mathbf{U}_k^T \mathbf{t}_{test}. \tag{19}$$

The vector $\mathbf{y}_{test}$ is compared to the points in the manifold curves for the various terrains corresponding to the estimated speed.

5.2 Field Experimentation

The multiple speed classification with interpolation was applied to data collected from the XUV. The system was trained at three speeds: 5 mph (2.2 m/s), 11 mph (4.9 m/s), and 17 mph (7.6 m/s). The results of classifying data at these speeds and the intermediate speeds of 8 mph (3.6 m/s) and 14 mph (6.2 m/s) are shown in Table 5.

Table 5: Results of Multiple Speed Terrain Classification with Interpolation for the XUV with Training at 5 mph (2.2 m/s), 11 mph (4.9 m/s), and 17 mph (7.6 m/s) (Shaded Columns)

Classified Terrains At Trained And Untrained Speeds		Input Terrains At Trained And Untrained Speeds														
		5 mph (2.2 m/s) Accuracy Rate = 62.0%			8 mph (3.6 m/s) Accuracy Rate = 71.4%			11 mph (4.9 m/s) Accuracy Rate = 73.3%			14 mph (6.2 m/s) Accuracy Rate = 76.2%			17 mph (7.6 m/s) Accuracy Rate = 66.7%		
		Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass	Asphalt	Gravel	Grass
	Asphalt	96.7% (29)	40.0% (8)	34.5% (10)	88.9% (16)		5.6% (1)	80.0% (8)		8.3% (1)	57.1% (4)			60.0% (3)		
	Gravel	3.3% (1)	60.0% (12)	37.9% (11)		46.2% (6)	22.2% (4)	10.0% (1)	75.0% (6)	25.0% (3)		83.3% (5)	12.5% (1)		25.0% (1)	
	Grass			27.6% (8)	11.1% (2)	53.8% (7)	72.2% (13)	10.0% (1)	25.0% (2)	66.7% (8)	42.9% (3)	16.7% (1)	87.5% (7)	40.0% (2)	75.0% (3)	100.0% (6)

The results of Table 5 clearly show the ability of the classification technique to classify terrains at speeds that do not correspond to the trained speeds. However, by comparing this table with Table 4 it is seen that the classification for the trained speeds is not as accurate as that achieved using the multiple speeds classification algorithm of Section 4. This is most likely due to the fact that the current interpolation-based algorithm does not explicitly use the clusters corresponding to each terrain and speed; instead, it simply uses the means of these clusters. Future work will consider extending this method to explicitly handle clusters.

6 Performance Comparison for On-Line Implementation

In addition to terrain classification accuracy, a classification algorithm must be suitable for online implementation, which is largely determined by the memory storage requirements and the online computational time. In this section the algorithms of Sections 3, 4, and 5 are compared using these two measures. It is assumed that each algorithm is used to classify three terrains at 8 speeds and the variable is the number of training samples for each combination of terrain and speed. Each of the algorithms was evaluated based on implementation in Matlab. The comparison is fair, but the actual numbers could be improved for each algorithm by using optimized code in a more efficient programming language such as C++.

Each classification algorithm requires different parameters to be stored for online computation. The speed dependent classification approach requires storage of the training matrices $\mathbf{P}_{s_j,t_i}$. The multiple speeds classification approach requires the reduced order matrix $\mathbf{Y}$ and the transformation matrix $\mathbf{U}_k$, while the multiple speeds classification with interpolation requires the parameters of the manifolds along with the transformation matrix $\mathbf{U}_k$. Figure 19 shows how increasing the number of terrain samples affects the memory requirements for the three classification approaches. (Note that the y-axis has a logarithmic scale.) The interpolation approach had an initial increase in storage requirements, but then leveled off, while the other two approaches continued to increase as the number of samples increased.

The time required to classify a terrain is extremely important for online implementation for an AGV because it determines the speed at which the control system may be adjusted for a new terrain. A comparison of the the online computational time for the three classification approaches is shown in Figure 20. The computational time of the interpolation approach is essentially constant with respect to the number of samples (although it will increase as the number of trained speeds is increased). In contrast, the computational time of the remaining two approaches increased as the number of samples increased. Although for the number of samples shown, the multiple speeds classification approach required smaller computational times than the interpolation approach, as the number of samples continues to increase the former approach would have greater computational requirements than the latter approach.

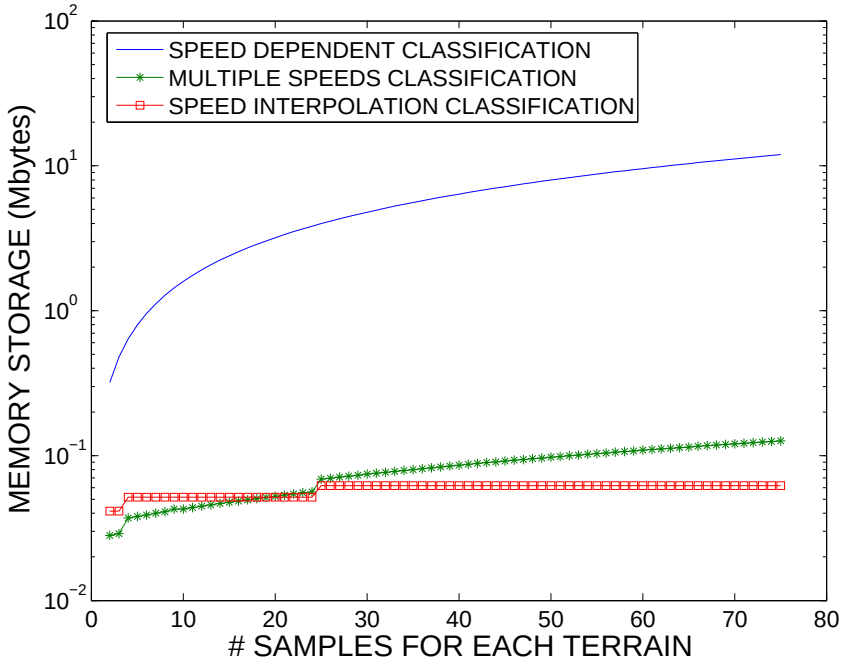


Figure 19: The Memory Required For Increasing Training Samples.

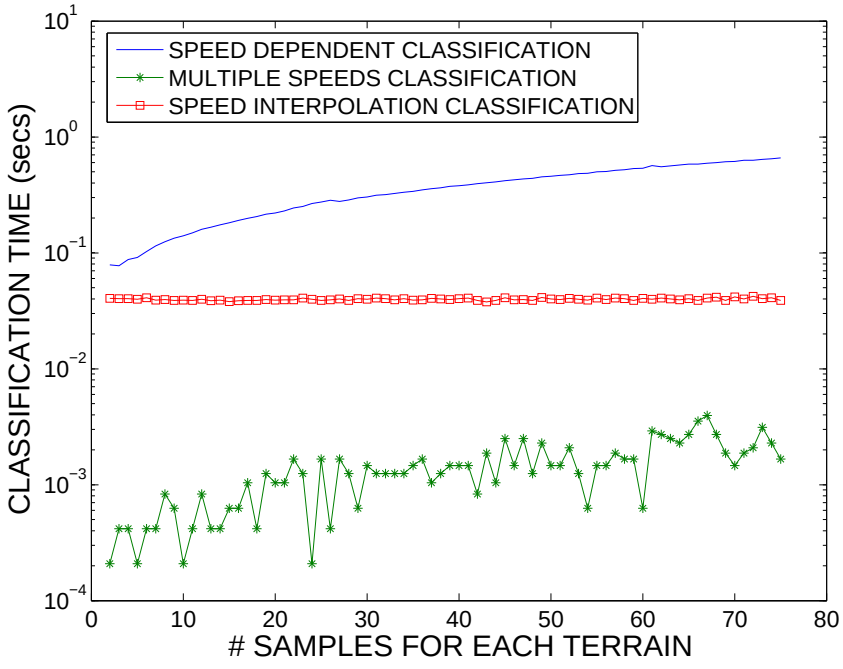


Figure 20: The Computation Time For Increasing Training Samples.

7 Terrain Input Classification

Each of the classification algorithms presented in the previous sections assumed an accurate vibrational transfer function model $G(s)$ of the AGV is not available. In contrast this section assumes that $G(s)$ is available and proposes classification based on mapping the sensor measurements of vibration back to the inputs to the wheels. The terrain is then classified based on the wheel inputs. Theoretically, this approach only requires training at one speed. In addition, it has the capability of detecting when the vehicle is traversing multiple terrains simultaneously (e.g., when the left wheels are off-road and the right wheels are on-road). Although simulation results are promising, this method has not yet been validated experimentally.

7.1 Basis for Terrain Input Classification

Recall that in Section 2.2 the frequency response of an AGV was modeled as

$$Y(j\omega) = G(j\omega)X(j\omega), \tag{20}$$

where $X(j\omega)$ is the frequency response of the inputs to the wheels resulting from traversing the terrain, $Y(j\omega)$ denotes the frequency response of the measurements of the sensor vibrations, and $G(j\omega)$ is the frequency response of the the vibrational transfer function. Now suppose that an input frequency response $X_a(j\omega)$ yields a corresponding output frequency response $Y_a(j\omega)$. At a different speed, the input becomes $X_b(j\omega) = \frac{1}{\alpha}X_a(j\frac{\omega}{\alpha})$, where $\alpha > 0$ represents the associated time scaling, i.e., $x_b(t) = x_a(\alpha t)$ in the time domain. The resulting output $Y_b(j\omega)$ is generally not a time scaled version of $Y_a(j\omega)$, i.e., $Y_b(j\omega) = \frac{1}{\alpha}G(j\omega)X(\frac{j\omega}{\alpha})$ instead of the time scaled version of $Y_a(j\omega)$, which is given by $\frac{1}{\alpha}Y_a(\frac{j\omega}{\alpha}) = \frac{1}{\alpha}G(\frac{j\omega}{\alpha})X_a(\frac{j\omega}{\alpha})$. This important observation concerning the speed dependence of the output frequency response $Y(j\omega)$ motivated the focus on classification at different speeds in the previous sections. However, it also suggests using $X(j\omega)$ for speed independent terrain classification. Since these inputs cannot be directly measured with vibration based proprioceptive sensors, they must be computed or approximated by using $G(j\omega)$.

If $G(j\omega)$ is a square matrix, where the number of inputs equals the number of outputs, then $X(j\omega)$ can be computed from (20) using

$$X(j\omega) = G(j\omega)^{-1}Y(j\omega). \tag{21}$$

If there are more outputs than inputs, i.e., (20) represents an over-determined system of equations, then the least squares solution may suffice. If there are fewer outputs than inputs, i.e., the system is under-determined, which is the case considered here, there are in general an infinite number of solutions $X(j\omega)$ that satisfy (20) and some means must be sought for finding one meaningful solution, a key problem in terrain input classification.

Recall the system transfer function, $G(s)$, for the four wheeled three output vibration model (see Figure 5 and (3)) is defined by

$$\begin{bmatrix} \ddot{z}(s) \\ \omega_y(s) \\ \omega_x(s) \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11}(s) & g_{12}(s) & g_{13}(s) & g_{14}(s) \\ g_{21}(s) & g_{22}(s) & g_{23}(s) & g_{24}(s) \\ g_{31}(s) & g_{32}(s) & g_{33}(s) & g_{34}(s) \end{bmatrix}}_{G(s)} \begin{bmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \\ X_4(s) \end{bmatrix}, \tag{22}$$

where the matrix elements of (22) (previously not described) are defined by

$$\begin{aligned} g_{11}(s) = g_{12}(s) = g_{13}(s) = g_{14}(s) &= \tfrac{bs^3+ks^2}{ms^2+4bs+4k}, \\ g_{21}(s) = g_{22}(s) &= \tfrac{\frac{L_2}{2}bs^2+\frac{L_2}{2}ks}{I_y s^2+L_2^2 bs+L_2^2 ks}, \\ g_{23}(s) = g_{24}(s) &= \tfrac{-\frac{L_2}{2}bs^2-\frac{L_2}{2}ks}{I_y s^2+L_2^2 bs+L_2^2 ks}, \\ g_{31}(s) = g_{34}(s) &= \tfrac{\frac{L_1}{2}bs^2+\frac{L_1}{2}ks}{I_x s^2+L_1^2 bs+L_1^2 ks}, \\ g_{32}(s) = g_{33}(s) &= \tfrac{-\frac{L_1}{2}bs^2-\frac{L_1}{2}ks}{I_x s^2+L_1^2 bs+L_1^2 ks}. \end{aligned} \tag{23}$$

The parameters of (23) were chosen to create a model representative of the Pioneer 3-AT AGV, shown in Figure 1. The spring and damping constants are computed based on a scaled car model from [19], while the remaining parameters were measured directly from the robot. Vibration outputs were then simulated using the resulting model and the previously determined surface profiles from each terrain, obtained using the CMU Laser Line Striper, as described in Subsection 2.2. The FFT was applied to extract the frequency response from the simulated outputs. Methods were then sought to find $X(j\omega)$ that satisfy or nearly satisfy (20) as described below.

One solution is to apply least squares, but since this is an under-determined system of equations, there are an infinite number of least squares solutions. One least squares solution often used in under-determined systems is the minimal least squares solution, calculated using the Moore-Penrose Inverse. This least squares solution minimizes both the squared error and the norm of the computed inputs. However, in these simulations, the minimal least squares solution was found to result in calculated inputs that are all of nearly the same magnitude. When all four inputs are of the same terrain, this is an advantageous property since terrain signatures are expected to have nearly the same magnitude. However, during *multiple terrain traversal*, i.e., when different wheels (usually left and right wheel pairs) are traversing different terrain, this property can make the terrain signature of each input difficult to distinguish; this is seen in Figure 21, where wheels traversing asphalt were paired with wheels traversing grass terrains and the least squares solution substantially distorts the asphalt signature, i.e., the dashed line is a distortion of the solid line. Thus a different method of finding a solution for $X(j\omega)$ was sought with the purpose of detecting multiple terrain traversal.

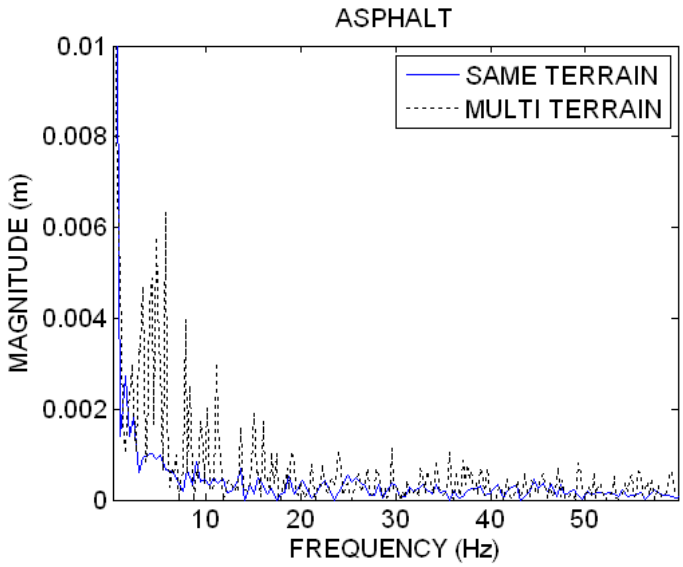


Figure 21: An Example Of Least Squares Solutions

Since inputs from the same terrain are expected to have approximately the same magnitude frequency response, nonlinear constraints could be used to find a more appropriate solution to (20). Since the front wheels are close in proximity to the back wheels, the front and back wheels on the same side of the AGV will typically experience the same terrain. Thus, in the four input, three output case in Figure 5 inputs $X_1(j\omega)$ and $X_4(j\omega)$ are expected to be from the same terrain and inputs $X_2(j\omega)$ and $X_3(j\omega)$ are expected to be from the same terrain. The resulting nonlinearly constrained optimization problem then becomes

$$\min_{X(j\omega)} \left[Y(j\omega) - G(j\omega)X(j\omega)\right]^T \left[Y(j\omega) - G(j\omega)X(j\omega)\right] \tag{24}$$

subject to

$$|X_1(j\omega)| = |X_4(j\omega)|, |X_2(j\omega)| = |X_3(j\omega)|. \tag{25}$$

Although the optimization problem (24) and (25) will be considered in future research, below less computationally intensive alternatives are presented.

When the vehicle is restricted to move in a straight line, the input to the back wheels will be the same input experienced by the front wheels, with a phase difference caused by a time delay. However, application of this concept is impractical when the system has a high amount of symmetry. To illustrate this, observe the resulting over-determined system given by

$$\begin{bmatrix} \ddot{z}(j\omega) \\ \omega_y(j\omega) \\ \omega_x(j\omega) \end{bmatrix} = \begin{bmatrix} g_{11}(j\omega) + g_{14}(j\omega)e^{-j\omega\tau} & g_{12}(j\omega) + g_{13}(j\omega)e^{-j\omega\tau} \\ g_{21}(j\omega) + g_{24}(j\omega)e^{-j\omega\tau} & g_{22}(j\omega) + g_{23}(j\omega)e^{-j\omega\tau} \\ g_{31}(j\omega) + g_{34}(j\omega)e^{-j\omega\tau} & g_{32}(j\omega) + g_{33}(j\omega)e^{-j\omega\tau} \end{bmatrix} \begin{bmatrix} X_1(j\omega) \\ X_2(j\omega) \end{bmatrix}, \tag{26}$$

where τ is the time delay in seconds. From (26) it can be seen that if $g_{11}(j\omega) \approx g_{14}(j\omega)$, $g_{12}(j\omega) \approx g_{13}(j\omega)$, and the time delay element $e^{-j\omega\tau}$ has a phase of 180 degrees, the first equation in the system (26) will essentially reduce to $\ddot{z}(j\omega) \approx 0$. Similar symmetry in the second and third equations can result in $\omega_y(j\omega) \approx 0$ and $\omega_x(j\omega) \approx 0$. The resulting system will ultimately cause the computed inputs to spike toward infinity at certain frequencies. These peaks make the terrain signatures almost indistinguishable. Thus in the presence of model symmetry another computationally efficient method of solving the constrained optimization problem must be sought.

Now, assume the nonlinear constraints of (25) are approximated by $X_1(j\omega) = X_4(j\omega)$ and $X_2(j\omega) = X_3(j\omega)$. In general, this will result in an over-determined system given by

$$\begin{bmatrix} \ddot{z}(j\omega) \\ \omega_y(j\omega) \\ \omega_x(j\omega) \end{bmatrix} = \begin{bmatrix} g_{11}(j\omega) + g_{14}(j\omega) & g_{12}(j\omega) + g_{13}(j\omega) \\ g_{21}(j\omega) + g_{24}(j\omega) & g_{22}(j\omega) + g_{23}(j\omega) \\ g_{31}(j\omega) + g_{34}(j\omega) & g_{32}(j\omega) + g_{33}(j\omega) \end{bmatrix} \begin{bmatrix} X_1(j\omega) \\ X_2(j\omega) \end{bmatrix}. \quad (27)$$

This then leads to the use of least squares to determine the inputs $X_1(j\omega)$ and $X_2(j\omega)$. However, due to the symmetry in the assumed model of (22) and (23) the transfer function matrix will not have full row rank and it may be better to simply ignore the redundant equation as was done to develop the below results.

Example inputs computed from (27) when the inputs are from the same terrain are shown in Figure 22. The true system inputs are shown as well for comparison purposes. Figure 22 shows relatively few difference in the computed and true inputs and most importantly, the terrain signatures seem to be preserved. Also, unlike classification with vibration outputs, terrain input classification can be used to classify computed terrain inputs at an observed speed, based on known inputs at a different reference speed. This comparison is performed by scaling the frequency range of the observed inputs based on the speed ratio ρ, which is defined by

$$\rho = \frac{v_{\text{reference}}}{v_{\text{observed}}}. \quad (28)$$

This speed ratio is used to compare the observed inputs $X(j\omega\rho)$ to the reference, instead of $X(j\omega)$. As seen in Figure 23 computed input signatures of drastically different speeds are very similar.

Based on the terrain signature preservation seen in Figures 22 and 23, high classification accuracy is possible using terrain input classification. A more detailed explanation of this approach as well as further simulation results that show high classification accuracy can be found in [35]. However, this terrain input classification methodology has not been tested experimentally and thus can only be presented as a theoretical method at this time. Additionally, several challenges are expected when moving from simulation to experimentation.

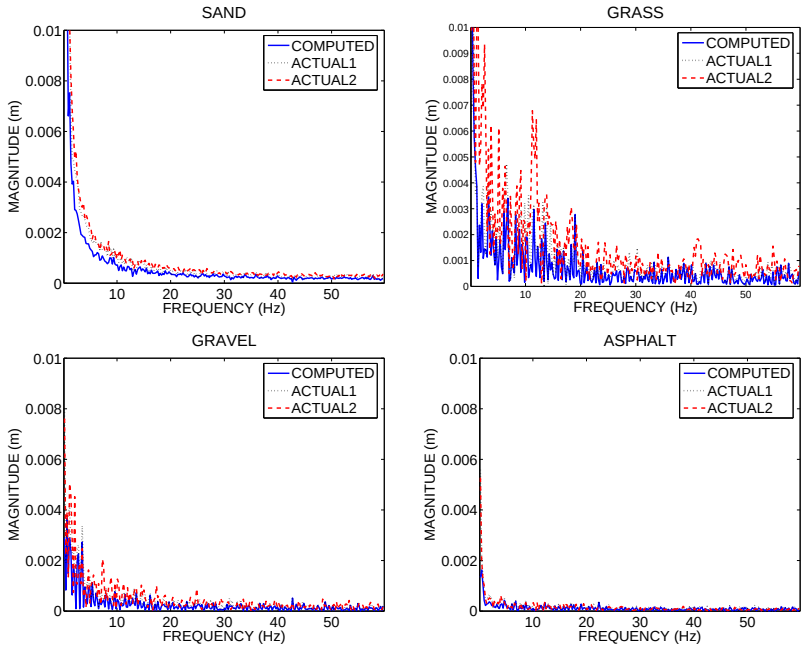


Figure 22: The Computed and Actual Terrain Signatures.

7.2 Current Challenges

The most obvious challenge in implementing this approach is determining a linear transfer function for an AGV system. Due to inaccurate or unknown parameters, analytical system models are expected to have insufficient model fidelity. But, a sufficiently high fidelity model can probably be obtained through system identification experiments. However, the range of functionality of such a model may still be somewhat limited by extreme circumstances such as high speed traversal or rugged terrains. The lower functionality would be the result of unexpected changes in the system transfer function due to loss of ground contact or other nonlinearities.

The other significant challenge is obtaining an accurate absolute velocity measurement. Due to constant wheel slip on many terrains, wheel encoder measurements on drive wheels could show large amounts of bias. However, alternative solutions include the use of encoder measurements on non-drive wheels (where rolling can be assumed), or velocity estimates using expensive IMU sensors and GPS readings. It may also be possible to estimate slip from other exteroceptive sensors, such as using stereo camera technology to correct the encoder measurements accordingly to give an accurate measurement of absolute velocity. Ultimately, the best method for obtaining an estimate of absolute velocity will be determined experimentally.

The current research emphasizes experimental implementation using various AGV platforms. The initial step in this process is determining the linear

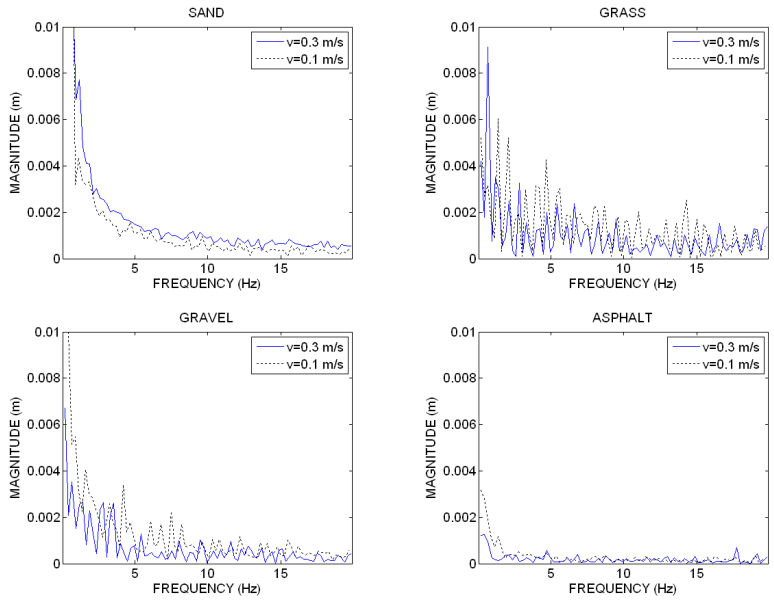


Figure 23: The Computed Terrain Signatures at Different Speeds.

transfer function of each AGV system. Current methodologies include using an excitation track to perform a "mechanical" sine sweep as well as excitation using an electrodynamic shaker. Once the system transfer function is obtained, terrain input classification will be tested using vibration data collected from multiple terrains.

8 FUTURE WORK

Vibration based terrain classification should be viewed as a complement to vision based terrain classification. An ideal terrain classification system requires the use of both methods to effectively identify the terrain. The ultimate goal is to fuse vision based classification with vibration based classification to develop a more robust system than can be achieved when either approach is used in isolation. Some research has already been performed to accomplish this fusion [36].

The area of vibration based terrain classification has great potential for growth and development. Research is currently being conducted to include additional terrain features. In particular, features such as slip relative to the wheel or the entire vehicle will be pertinent in distinguishing terrains such as sand, mud or snow and even terrain variations such as wet and dry grass.

When implemented in practical applications, terrain classification algorithms are expected to consider a large number of terrain types and conditions, including terrains that are similar yet distinguishable from a control perspec-

tive, such as wet asphalt and dry asphalt. Thus a more robust vibration-based classification algorithm must be able to classify larger numbers of terrains than have been considered in research to date. Current research in this direction includes the implementation of a multistaged classifier system, as illustrated in Figure 24; this methodology allows for the grouping of similar terrains to improve classification accuracy [37]. The use of the PNN classifier in this clas-

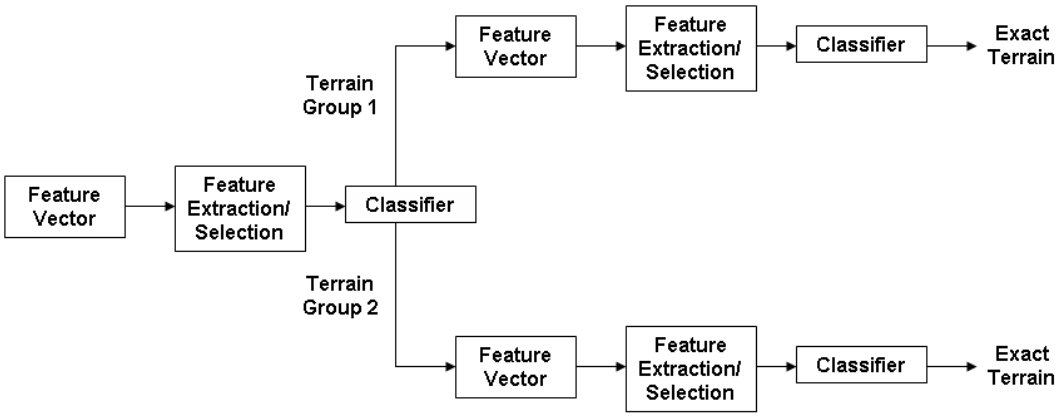


Figure 24: The Basic Structure Of A Multi-Stage Classification System

sification system provides the ability to tune the smoothing parameters σ for each terrain type and feature. Thus, instead of a single scalar smoothing parameter, a matrix can be developed, where each matrix element corresponds to a specific class and feature.

Although the PNN has been used in the research reported here, there is some evidence that support vector machines may provide even better classification [38]. Current research is also seeking to experimentally validate terrain input classification by using system identification to experimentally obtain the vibrational transfer function of an AGV.

In developing a terrain classification system, the objective is to ultimately improve the control of an AGV in unstructured and uncertain outdoor environments. A set of driving behaviors should be developed to improve the vehicle's control characteristics, such as traction and steering, on different terrains. Hence, current research is being conducted to quantify various qualitative expert driving rules for implementation on an AGV control system. Once the terrain classification algorithm identifies that an AGV has encountered a new terrain, the control system can then be adjusted to the appropriate rules.

9 CONCLUSIONS

This chapter presented vibration based terrain classification methods for an AGV. The discussion began by describing how the terrain profile influ-

ences the wheel inputs and hence, due to the vibrational transfer function of the vehicle, produces a response that has a unique signature for each terrain and speed. Samples of the vehicle's response at each speed were trained and classified using a Probabilistic Neural Network in a speed dependent terrain classification algorithm. This approach was improved in both computational time and the ability to train at a larger number of speeds by applying Principal Component Analysis to obtain reduced-order terrain signature vectors. Further refinement involved the use of spline interpolation to classify at speeds that are between two of the trained speeds. The interpolation results are preliminary but promising. The first three approaches were verified experimentally using data collected from both an ATRV-Jr and an eXperimental Unmanned Vehicle (XUV), while the third approach was verified using only XUV data. A fourth approach, terrain input classification, uses the vibration transfer function of the vehicle and theoretically provides vibration-based classification that only requires training at one speed. However, it remains to be experimentally demonstrated. Ultimately, these vibration based methods will be used in conjunction with vision based methods for more robust terrain classification.

References

[1] J. Allen. *Four-Wheeler's Bible.* MotorBooks, St Paul, MN, 2002.

[2] B. Delong. *4-Wheel Freedom: The Art of Off-Road Driving.* Paladin Press, Boulder, CO, 2000.

[3] D. Vanderwerp. What does terrain response do?, 2005. http://www.caranddriver.com/features/9026/what-does-terrain-response-do.html.

[4] A. Howard and H. Seraji. Vision-based terrain characterization and traversability assessment. *Journal of Robotic Systems*, 18(10):577–587, 2001.

[5] N. Vandapel, D. F. Huber, A. Kapuria, and M. Herbet. Natural terrain classification using 3-d ladar data. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5117–5122, New Orleans, LA, April 2004.

[6] K. Iagnemma, H. Shibly, and S. Dubowsky. Terrain parameter estimation for planetary rovers. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3142–3147, Washington, DC, May 2002.

[7] K. Iagnemma, S. Kang, H. Shibly, and S. Dubowsky. Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers. *IEEE Transactions On Robotics*, 20(5), October 2004.

[8] K. Iagnemma and S. Dubowsky. Terrain estimation for high speed, rough-terrain autonomous vehicle navigation. In *Proc. of the SPIE Conference on Unmanned Ground Vehicle Technology*, pages 256–266, Orlando, FL, May 2002.

[9] D. Sadhukan and C. Moore. Online terrain estimation using internal sensors. In *Proc. of the Florida Conference on Recent Advances in Robotics*, Boca Raton, FL, May 2003.

[10] D. Sadhukhan. *Autonomous Ground Vehicle Terrain Classification Using Internal Sensors*. Master's thesis, Dept. of Mechanical Engineering, Florida State University, Tallahasee, FL, 2004.

[11] E. M. DuPont, R. G. Roberts, C. A. Moore, M. F. Selekwa, and E. G. Collins, Jr. Online terrain classification for mobile robots. In *Proc. of the ASME International Mechanical Engineering Congress and Exposition Conference*, Orlando, FL, November 2005.

[12] C. Brooks, K. Iagnemma, and S. Dubowsky. Vibration-based terrain analysis for mobile robots. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3142–3147, Barcelona, Spain, May 2002.

[13] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen. Terrain characterization and classification with a mobile robot. *Journal of Field Robotics*, 23(2):103–122, 2006.

[14] C. Mertz, J. Kosar, J. R. Miller, and C. Thorpe. Easy Safe Laser Line Striper For Outside Use. In *IV 2002, IEEE Intelligent Vehicle Symposium*, June 2002.

[15] L. Lu, C. Ordonez, E. G. Collins, Jr., and E. M. DuPont. Terrain classification for autonomous ground vehicles using 2-d laser stripe-based structured light sensors. *submitted to 2008 International Conference on Intelligent RObots and Systems (IROS)*, available at http://www.eng.fsu.edu/ciscor/publications.htm.

[16] H. Baruh. *Analytical Dynamics*. McGraw-Hill Companies Inc, 1999.

[17] S. Sukarrieh. *Low Cost High Integrity Aided Inertial Navigation Systems for Autonomous Land Vehicles*. Ph.D. Thesis, University of Sydney, Sydney, Australia, 2000.

[18] J. von Scheidt, R. Wunderlich, and B. Fellenberg. Random road surfaces and vehicle vibration. In L. Arkeryd, J. Bergh, P. Brenner, and R. Pettersson, editors, *Progress in Industrial Mathematics at ECMI 98*, pages 352–359. Teubner-Stuttgart, 1999.

[19] J.Y. Wong. *Theory of Ground Vehicles, Third Edition.* John Wiley & Sons, New York, NY, 2001.

[20] A. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–36, January 2000.

[21] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood, 1994.

[22] L. H. Tsoukalas and R. E. Uhrig. *Fuzzy and Neural Approaches in Engineering.* John Willey & Sons, New York, NY, 1997.

[23] D. F. Specht. Probabilistic neural networks. *Neural Networks*, 3(1):109–118, Jan 1990.

[24] D. F. Specht. Probabilistic neural networks and the polynomial adaline as complementary techniques for classification. *IEEE Transactions on Neural Networks*, 1(1):111–121, March 1990.

[25] R. Cacoullous. Estimation of a probability density. *Annals of the Institute of Statistical Mathematics (Tokyo)*, 18(2):179–189, 1966.

[26] V. K. Murthy. Estimation of a probability density. *Annals of Mathematical Statistics*, 36:1027–1031, 1965.

[27] V. K. Murthy. Nonparametric estimation of multivariate densities with applications. In P. R. Krishnaiah, editor, *Multivariate Analysis*, pages 43–56, New York, 1966. Academic Press.

[28] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[29] D. F. Specht. Probabalistic neural networks for classification, mapping, or associative memory. In *Proc. IEEE International Conference on Neural Networks*, pages 525–532, San Diego, CA, 1988.

[30] I. T. Jolliffe. *Principal Component Analysis.* New York: Springer-Verlag, 1986.

[31] H. Murase and S.K. Nayar. Detection of 3D objects in cluttered scenes using hierarchical eigenspace. *Pattern Recognition Letters*, 18(4):375–384, April 1997.

[32] H. Murase and S.K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal Of Computer Vision*, 14(1):5–24, January 1995.

[33] Lina, T. Takahashi, I. Ide, and H. Murase. Appearance manifold with covariance matrix for 3-D object recognition. In *IEICE Technical Report PRMU2006-135*, pages 31–36, 2006.

[34] E. Catmull and R. Rom. *A class of local interpolating splines in computer aided geometric design*. Academic Press, 1974.

[35] E. G. Collins, Jr., and E. Coyle. Vibration-based terrain classification using surface profile input frequency responses. *accepted for publication in 2008 International Conference on Robotics and Automation*, available at http://www.eng.fsu.edu/ciscor/publications.htm.

[36] I. Halatci, C. Brooks, and K. Iagnemma. Terrain Classification and Multi-Classifier Fusion Using Visual and Tactile Sensing for Planetary Rovers. In *Proc. of the 2007 IEEE Aerospace Conference*, 2007.

[37] E. Coyle, E. Collins, E. DuPont, D. Ding, H. Wangand, R. Cooper, and G. Grindle. Vibration-Based Terrain Classification for Electric Powered Wheelchairs. *accepted for publication in 2008 IASTED Conference in Assistive Technologies*, available at http://www.eng.fsu.edu/ciscor/publications.htm.

[38] C. Weiss, N. Fechner, M. Stark, and A. Zell. Comparison of Different Approaches to Vibration-based Terrain Classification. In *Proc. of the 3rd European Conference on Mobile Robots (ECMR 2007)*, pages 7–12, September 2007.