Auton Robot (2009) 26: 33–46
DOI 10.1007/s10514-008-9105-8

Classification-based wheel slip detection and detector fusion for mobile robots on outdoor terrain

Karl Iagnemma · Chris C. Ward

Received: 7 April 2008 / Accepted: 6 October 2008 / Published online: 21 October 2008

Abstract This paper introduces a signal-recognition based approach for detecting autonomous mobile robot immobilization on outdoor terrain. The technique utilizes a support vector machine classifier to form class boundaries in a feature space composed of statistics related to inertial and (optional) wheel speed measurements. The proposed algorithm is validated using experimental data collected with an autonomous robot operating in an outdoor environment. Additionally, two detector fusion techniques are proposed to combine the outputs of multiple immobilization detectors. One technique is proposed to minimize false immobilization detections. A second technique is proposed to increase overall detection accuracy while maintaining rapid detector response. The two fusion techniques are demonstrated experimentally using the detection algorithm proposed in this work and a dynamic model-based algorithm. It is shown that the proposed techniques can be used to rapidly and robustly detect mobile robot immobilization in outdoor environments, even in the absence of absolute position information.

Keywords Mobile robots · Wheel slip · Outdoor terrain

1 Introduction

Mobile robot position estimation systems typically rely (in part) on wheel odometry as a direct estimate of displacement and velocity (Borenstein et al. 1996; Fuke and Krotkov 1996). On high-traction terrain and in combination with periodic GPS absolute position updates, such systems can provide an accurate estimate of the robot's position. However, when driving over low-traction terrain, deformable terrain, steep hills, or during collisions with obstacles, an odometry-based position estimate can quickly accumulate large errors due to wheel slip. With ineffective odometry, periodic absolute position updates can cause large "jumps" in a robot's position estimate. In addition, between updates an odometry-based system is unable to differentiate between a robot that is immobilized with its wheels spinning and one that is driving normally. Autonomous robots should quickly detect that they are immobilized in order to take appropriate action, such as planning an alternate route away from the low-traction terrain region or implementing a traction control algorithm. Additionally, robust position estimation is required for accurate map registration.

Wheel slip can be accurately estimated through the use of encoders by comparing the speed of driven wheels to that of undriven wheels (Gustafsson 1997) however this does not apply for all-wheel drive robots or those without redundant encoders. Ojeda and Borenstein have proposed comparing redundant wheel encoders against each other and against yaw gyros as an indicator of wheel slip, even when all wheels are driven (Ojeda et al. 2004), however this technique does not estimate the degree of wheel slip (i.e. whether the robot is fully immobilized). Ojeda and Borenstein have also proposed a motor current-based slip estimator (Ojeda et al. 2006), however this technique requires accurate current measurement and terrain-specific parameter tuning, with proposed tuning techniques requiring either an accurate absolute positioning device or a robot with at least four driven wheels. In Angelova et al. (2006) visual odometry (VO) is used to estimate robot velocity and slip for a slip

K. Iagnemma (✉) · C.C. Ward
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA
e-mail: kdi@mit.edu
url: http://web.mit.edu/mobility/

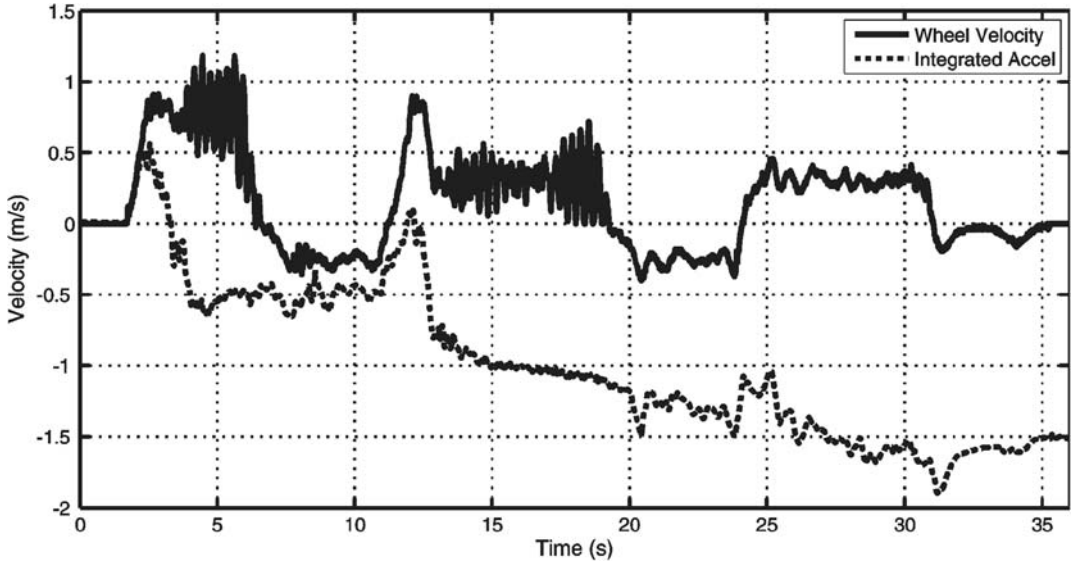

Fig. 1 Inaccuracy of estimating robot body velocity by integrating measured acceleration

prediction algorithm. Although VO can be accurate on average over time, the authors report VO errors of $\sim 12\%$ on short time scales. In addition, the performance of VO can be degraded in near-featureless environments, such as sand. It should be noted that a body of work exists in the automotive community related to traction control and anti-lock braking systems (ABS), however this work generally applies to systems operating at significantly higher speeds than is typical for many autonomous robots.

A large amount of work has utilized Kalman filters with inertial and absolute measurements to enhance dead reckoning and estimate lateral slip. In Barshan and Durrant-Whyte (1995) a navigation system is proposed that uses inertial measurements combined with a sensor error model in a Kalman filter to increase measurement accuracy. In Fuke and Krotkov (1996), Kelly (1994) traditional dead reckoning accuracy is improved by including inertial measurements. In Anderson and Bevly (2004), Wadda et al. (2000) absolute position updates from GPS are fused with a model-based Kalman filter to estimate vehicle sideslip and improve position estimation accuracy, and in Ryu et al. (2002) this work is extended to consider the effects of vehicle roll and pitch. The notion of an effective tire radius, which can indirectly compensate for some longitudinal slip, is presented in Julier and Durrant-Whyte (2003). None of this work, however, explicitly considers the effects of longitudinal wheel slip or robot immobilization.

A potentially simple approach to detecting robot slip and immobilization is to analyze GPS measurements. In open terrain, GPS can provide accurate position and velocity measurements; however, nearby trees and buildings can cause signal loss and multipath errors, and changing satellites can cause position and velocity jumps (Sukkarieh et al. 1999; Hofmann-Wellenhof et al. 2001). Additionally, GPS provides low frequency updates (e.g. typically near 1 Hz GPS 16/17 Series Technical Specifications 2005) making GPS alone too slow for immobilization detection, where as close to instantaneous detection as possible is desired to avoid excessive position errors. Dissanayake et al. (2001) propose a method of aiding the inertial estimate using vehicle constraints, however this method is not well suited for operation on uneven, low-traction terrain.

Another potentially simple slip detection technique is to estimate robot body velocity by integrating acceleration measurements (after subtracting gravitational acceleration due to robot pitch), then comparing this estimate to wheel velocity, thereby estimating wheel slip (Wong 2001). Figure 1 compares experimental data for wheel velocity and estimated body velocity of a wheeled mobile robot traveling on grassy, rolling outdoor terrain. It can be observed that at low speeds accelerometer errors dominate, causing the velocity estimate to quickly diverge. In this case a detector based on this estimate would erroneously signal an immobilized condition for the majority of the data set. Because the body velocity estimate error is essentially a random walk, in some cases such a detector would estimate the body velocity to always be larger than the wheel velocity, thus never detecting immobilization.

In this paper a method is presented for detecting robot immobilization using a classification approach. Machine learning/classification techniques have been employed in various mobile robotics applications including vibration-based terrain classification (Brooks and Iagnemma 2005) and self-supervised vision-based road detection (Dahlkamp et al. 2006), as well as other applications such as speech recognition (Ganapathiraju et al. 2004). In addition, an extensive

body of work exists in terrain classification. This includes methods for classification of terrain based on color, texture, surface geometry, and other features (see, for example, Rasmussen 2001; Manduchi 2004; Karlsen and Witus 2007). The authors are aware of no previous work utilizing classification techniques for robot immobilization detection.

In the approach presented in this paper, a support vector machine (SVM) classifier is trained offline to recognize immobilized conditions within a feature space formed using inertial measurement unit (IMU) and optional wheel speed measurements. The trained SVM can then be used to quickly detect immobilization with little computation. Experimental results show the algorithm to quickly and accurately detect immobilization in various scenarios.

In addition to the classification-based approach to immobilization detection, a method for immobilization detection based on a dynamic model of robot-terrain interaction is also presented. This algorithm was initially proposed in Ward and Iagnemma (2007). This algorithm uses a dynamic robot model fused with wheel encoder, IMU, and (optional) GPS measurements in an extended Kalman filter (EKF) to create an estimate of the robot's longitudinal velocity and wheel slip. This method is presented here to allow investigation of improving immobilization detection algorithm accuracy and robustness via fusion of multiple immobilization detectors.

An initial fusion technique is proposed to minimize false immobilization detections. A technique second is proposed to increase overall detection accuracy while maintaining rapid detector response. The two fusion techniques are demonstrated with experimental data using the classification-based algorithm proposed in this work and the dynamic model-based algorithm initially proposed in Ward and Iagnemma (2007).

This paper is organized as follows. In Sect. 2 the classifier-based immobilization detection algorithm is presented. In Sect. 3 a complementary dynamic model-based immobilization detection algorithm is summarized. The performance of the classification-based algorithm is experimentally demonstrated in Sect. 4. In Sect. 4 two techniques are also presented for fusing the outputs of the classification-based and model-based immobilization detection algorithms. The effectiveness of the fusion techniques is analyzed experimentally. In Sect. 5 conclusions are drawn from this work and future work is suggested.

2 Classification-based mobile robot immobilization detection

2.1 Classification algorithm overview

The algorithm proposed here was inspired by the observation that a human in a vehicle with eyes closed can quickly and robustly distinguish whether the vehicle is:

1) completely stopped with wheels stopped,
2) driving normally over outdoor terrain, or
3) immobilized, with the wheels rotating but slipping.

Even in the absence of training for this task and without visual feedback, a human can interpret cues such as vehicle heave/jounce and motor/engine sound signature to discriminate between cases 1–3 with reasonable accuracy.

The proposed algorithm uses a signal-recognition approach to detect mobile robot immobilization (case 3 above) based on inertial and wheel speed measurements. The measurements are used to form n features that can be used to distinguish between the two classes "immobilized" and "normal driving." A support vector machine (SVM) is used to determine class boundaries within the n-dimensional feature space (Chang and Lin 2001).

The SVM is trained using a hand-labeled data set of l instance-label pairs $(\mathbf{q}_1, r_1), \ldots, (\mathbf{q}_i, r_i), \ldots, (\mathbf{q}_l, r_l)$ with $\mathbf{q}_i \in \Re^n$ and $r_i \in \{-1, 1\}$ (Hastie et al. 2001; Hsu et al. 2006). In this work, "normal" is labeled as $r_i = -1$ and "immobilized" as $r_i = 1$. The l training instance feature vectors, $\mathbf{q}_i$, are combined to form the $l \times n$ feature matrix, $\mathbf{Q} = [\,\mathbf{q}_1 \cdots \mathbf{q}_l\,]^T$, and the labels form the $l \times 1$ training label vector, $\mathbf{r} = [\,r_1 \cdots r_l\,]^T$.

Classification accuracy is improved by scaling each feature type to have similar magnitudes (Hsu et al. 2006). To scale each feature to the range $[-1, 1]$, an $n \times n$ scale factor matrix, $\mathbf{S}$, is formed such that:

$$\mathbf{S}_{i,j} = \begin{cases} \frac{1}{\max(|\text{column } j \text{ of } \mathbf{Q}|)} & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and the scaled training feature matrix, $\tilde{\mathbf{Q}}$, is then:

$$\tilde{\mathbf{Q}} = \mathbf{Q} \cdot \mathbf{S}. \tag{2}$$

$\tilde{\mathbf{Q}}$ and $\mathbf{r}$ are used to train a SVM using a radial basis function kernel, since this kernel performs well with both non-linear and linear class relations and requires few kernel parameters (Hsu et al. 2006). SVM parameters are found using a brute force search to find a parameter set that minimizes the average classification error and error standard deviation of a v-fold cross-validation (Hsu et al. 2006). The final SVM model is trained using the best SVM parameter set and the entire training data set.

The parameter search and SVM training can be computationally expensive. However training is performed only once, offline, producing an SVM model suitable for computationally inexpensive online classification. Note that during online classification, each measured feature vector, $\mathbf{q}$, is first multiplied by the scale factor matrix, $\mathbf{S}$, before classification by the trained SVM.

During online classification, the output of the SVM's decision function is a scalar decision value, $\hat{f} \in (-\infty, \infty)$,

where the value of $\hat{f}$ is a measure of the distance of the instance from the class boundary in the n-dimensional feature space. Typically an instance is labeled as:

$$label = \begin{cases} \text{immobilized}(1) & \text{if } \hat{f} > 0, \\ \text{normal}(-1) & \text{if } \hat{f} < 0, \\ \text{unknown}(0) & \text{if } \hat{f} = 0. \end{cases} \tag{3}$$

However, increased accuracy can usually be achieved at the expense of lowered labeling completeness (i.e. labeling some instances "unknown") using the following:

$$label = \begin{cases} \text{immobilized}(1) & \text{if } \hat{f} > threshold, \\ \text{normal}(-1) & \text{if } \hat{f} < -threshold, \\ \text{unknown}(0) & \text{if } -threshold \leq \hat{f} \leq threshold, \end{cases} \tag{4}$$

where $threshold \geq 0$. In this work (3) has been used unless otherwise specified, meaning that all data has been classified.

2.2 Feature vector selection

In this work four features have been chosen to form the feature vector $\mathbf{q}_i = \lfloor q_{i,1}, q_{i,2}, q_{i,3}, q_{i,4} \rfloor$. Each feature is a numerical representation of sensor data that attempts to mimic the sensory cues a human operator would exploit when attempting to detect immobilized conditions. Data is sampled at a rate f_s and a numerical transform is calculated on a set of N data points for each feature instance. Figure 2 illustrates the coordinate system used in feature definitions.

The first two features were chosen as the variance of the N element groupings i of roll rate, $\dot{\boldsymbol{\theta}}_{i,N}$, and pitch rate, $\dot{\boldsymbol{\varphi}}_{i,N}$, such that:

$$q_{i,1} = \text{var}\left(\dot{\boldsymbol{\theta}}_{i,N}\right) = E\left(\left(\dot{\boldsymbol{\theta}}_{i,N} - E\left(\dot{\boldsymbol{\theta}}_{i,N}\right)\right)^2\right), \tag{5}$$

$$q_{i,2} = \text{var}\left(\dot{\boldsymbol{\varphi}}_{i,N}\right) = E\left(\left(\dot{\boldsymbol{\varphi}}_{i,N} - E\left(\dot{\boldsymbol{\varphi}}_{i,N}\right)\right)^2\right). \tag{6}$$

These two features are a measure of the degree of roll and pitch experienced by a robot during travel over uneven outdoor terrain. During normal outdoor driving, terrain unevenness leads to variations in roll and pitch. This variation is minimized when the robot is immobilized.

The third feature was chosen as a measure of the variation in the z-axis (vertical) acceleration. The variance is a measure of the total variation from the mean over all frequencies; however empirical results have shown that only high frequency z-axis acceleration signal variation effectively distinguishes immobilized conditions. For feature three, $\mathbf{P}_{\mathbf{a}_z,i}$, the p element vector of the power spectrum coefficients of grouping i of z-axis acceleration is calculated using a discrete Fourier transform, where:

$$p = \left\lceil \frac{N+1}{2} \right\rceil, \tag{7}$$

where $\lceil\ \rceil$ is the ceiling function. Then feature three is calculated as:

$$q_{i,3} = \sum_{k=\lceil p/2 \rceil}^{p} P_{a_z,i,k}. \tag{8}$$

Again, during normal outdoor driving, terrain unevenness leads to variations in vertical acceleration, which is minimized when the robot is immobilized. For this work $N = 50$ and $f_s = 100$ Hz, resulting in a sum of the frequency content from 25 to 50 Hz. This frequency range was empirically determined to perform well for the robot system used in the experimental trials described later in this paper.

Feature four was chosen as the mean of the magnitude of the wheel angular accelerations:

$$q_{i,4} = \text{mean}\left(\left|\dot{\boldsymbol{\omega}}_{l,i,N} + \dot{\boldsymbol{\omega}}_{r,i,N}\right|\right) = E\left(\left|\dot{\boldsymbol{\omega}}_{l,i,N} + \dot{\boldsymbol{\omega}}_{r,i,N}\right|\right), \tag{9}$$

where $\dot{\boldsymbol{\omega}}_{l,i,N}$ and $\dot{\boldsymbol{\omega}}_{r,i,N}$ are the N element groupings i of the left and right wheel angular accelerations, respectively. As above, during normal outdoor driving, terrain unevenness leads to variations in wheel torque, leading to variations in wheel angular acceleration. This variation is minimized when the robot is immobilized.

3 Dynamic model-based mobile robot immobilization detection

An alternate wheel slip detection algorithm has been developed based on a model-based estimation framework (Ward

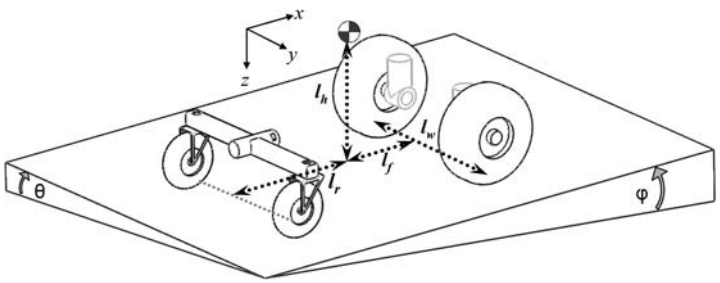

Fig. 2 Robot kinematic parameters

and Iagnemma 2007, 2008). This algorithm uses a dynamic robot model fused with wheel encoder, IMU, and (optional) GPS measurements in an extended Kalman filter to create an estimate of the robot's longitudinal velocity and wheel slip. Here the model-based slip detection algorithm is briefly summarized for completeness. Further details are presented in Ward and Iagnemma (2008). The purpose of this presentation is to allow investigation (in Sect. 4) of methods for fusing the outputs of the classification-based immobilization detection algorithm (described in Sect. 2) and the model-based immobilization detection algorithm.

3.1 Robot description

The robot configuration considered in this work is modeled on an autonomous mobile robot developed for the DARPA LAGR (Learning Applied to Ground Robots) research program (Learning Applied to Ground Robots 2006) (see Fig. 3). The robot is 1.2 m long × 0.7 m wide × 0.5 m, has four rubber pneumatic tires, and is a front-wheel differential-drive configuration. The robot body-fixed coordinate system and kinematic parameters are shown in Fig. 2. The robot is equipped with 4096 count per revolution front wheel encoders, an Xsens MT9 IMU, and a Garmin GPS 16 differential GPS (not used in this algorithm). IMU and wheel encoders are sampled at 100 Hz. The robot has been used to collect data to process offline using a Matlab implementation of the slip detector. A dynamic model of this robot is derived in Ward and Iagnemma (2008) and presented in Appendix A for completeness.

3.2 State space robot model

The robot and sensor dynamics for the robot described in Sect. 3.1 are formulated as a state space model using the following state vector:

$$\mathbf{x} = \lfloor v_{bx}, b_{ax}, a_{dist,bx}, \omega_l, \omega_r, \dot{\psi}, b_{g\psi} \rfloor^T,$$

where v_{bx} is the robot's velocity in the longitudinal direction, $a_{dist,bx}$ is the equivalent x-axis body acceleration due to any disturbance force(s), ω_l and ω_r are the angular velocities of the left and right front wheels, $\dot{\psi}$ is the yaw rate, and b_{ax} and $b_{g\psi}$ are the accelerometer x-axis and yaw gyro walking biases, respectively, which are part of the IMU error model suggested in Flenniken et al. (2005).

Typical errors found in accelerometers and rate gyros of low-cost IMUs are due to constant offsets, c, walking biases, b, and sensor noise, ν, such that (Flenniken et al. 2005):

$$z_{meas} = z_{actual} + c + b + \nu, \tag{10}$$

with

$$\dot{b} = -\frac{1}{\tau} b + \sqrt{\frac{2 f_s \sigma^2}{\tau}} w, \tag{11}$$

where z_{meas} is the measured acceleration or angular rate, z_{actual} is the true value of the measured variable, ν is assumed to be zero mean white noise, τ is a time constant, $\sigma^2 = E[b^2]$, and w is zero mean white noise with $E[w^2] = 1$.

Using the above state vector, the robot dynamics can be written as:

$$\dot{\mathbf{x}} = \begin{bmatrix} f_{tire}(\mathbf{x}, \theta, \varphi) + a_{dist,bx} - g \sin(\varphi) \\ -\frac{1}{\tau_{ax}} b_{ax} \\ 0 \\ f_{controller,l}(u) \\ f_{controller,r}(u) \\ g_{tire}(\mathbf{x}, \theta, \varphi) \\ -\frac{1}{\tau_{gz}} b_{g\psi} \end{bmatrix} + \begin{bmatrix} w_1 \\ \sqrt{\frac{2 f_s \sigma_{abx}^2}{\tau_{ax}}} w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ \sqrt{\frac{2 f_s \sigma_{gbz}^2}{\tau_{gz}}} w_7 \end{bmatrix}, \tag{12}$$

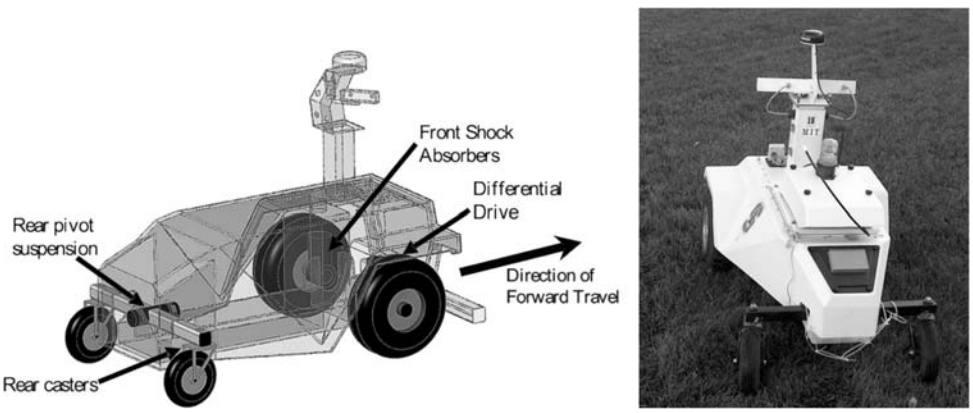

Fig. 3 Rendering (*left*) and image (*right*) of the mobile robot considered in this work

where f_{tire} and g_{tire} are the equivalent x-axis and yaw-axis body accelerations due to tire forces (see Appendix A), and $f_{controller}(u)$ is the wheel acceleration, which is a function of the robot's onboard velocity controller and the desired velocity. Note that with constant desired velocity, an ideal wheel speed controller would achieve $f_{controller,l}(u) = w_5$ and $f_{controller,r}(u) = w_6$. In practice, the high frequency and accuracy of wheel velocity measurements commonly allows an accurate estimate of ω_l and ω_r without modeling $f_{controller}$. When estimating the robot dynamics, we therefore neglect $f_{controller}$, assuming that the desired velocity is approximately constant between sensor updates.

Discretizing the state equations and neglecting the process noise (w_i), the a priori estimate of the robot state at time step k is:

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} \hat{v}_{bx,k-1} + f_{tire}(\hat{\mathbf{x}}, \theta, \varphi)\Delta t + \hat{a}_{dist,bx,k-1}\Delta t - g\sin(\varphi)\Delta t \\ \hat{b}_{ax,k-1}(1 - \frac{\Delta t}{\tau_{ax}}) \\ \hat{a}_{dist,bx,k-1} \\ \hat{\omega}_{l,k-1} \\ \hat{\omega}_{r,k-1} \\ \hat{\dot{\psi}}_{k-1} + g_{tire}(\hat{\mathbf{x}}, \theta, \varphi)\Delta t \\ \hat{b}_{g\psi,k-1}(1 - \frac{\Delta t}{\tau_{gz}}) \end{bmatrix}. \tag{13}$$

3.3 Measurement model

The model-based slip detector algorithm utilizes measurements from the IMU, GPS, and front wheel encoders. The measurement vector is:

$$\mathbf{z} = \lfloor \ddot{x}_{IMU}, \dot{\psi}_{IMU}, \dot{x}_{GPS}, \omega_{l,enc}, \omega_{r,enc} \rfloor^T,$$

where $\ddot{x}_{IMU}$ and $\dot{\psi}_{IMU}$ are IMU measurements of x-axis acceleration and yaw rate, $\dot{x}_{GPS}$ is the component of the GPS velocity measurement along the body x-axis, and $\omega_{l,enc}$ and $\omega_{r,enc}$ are the left and right front wheel encoder angular velocity measurements. For the IMU measurements, the sensor model given by (10) is used. Note that for $\ddot{x}_{IMU}$, $z_{actual} = \dot{v}_{bx} + g\sin(\varphi)$, as the accelerometer measures gravity even if the robot is stopped. Simplifying, the measurement vector can be modeled as:

$$\begin{aligned} \mathbf{z} &= h(\mathbf{x}, v) \\ &= \begin{bmatrix} f_{tire}(\mathbf{x}, \theta, \varphi) + c_{ax} + b_{ax} + a_{dist,bx} \\ \dot{\psi} + c_{gz} + b_{g\psi} \\ v_{bx} \\ \omega_l \\ \omega_r \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}, \end{aligned} \tag{14}$$

where c_{ax} and c_{gz} are constant offsets of the x-axis accelerometer and yaw gyro respectively and v_i are zero mean white noise.

The estimated measurement vector is:

$$\hat{\mathbf{z}} = h(\hat{\mathbf{x}}_k^-, 0) = \begin{bmatrix} f_{tire}(\hat{\mathbf{x}}_k^-, \theta, \varphi) + c_{ax} + \hat{b}_{ax,k}^- + \hat{a}_{dist,bx,k}^- \\ \hat{\dot{\psi}}_k^- + c_{gz} + \hat{b}_{g\psi,k}^- \\ \hat{v}_{bx,k}^- \\ \hat{\omega}_{l,k}^- \\ \hat{\omega}_{r,k}^- \end{bmatrix}. \tag{15}$$

The disturbance, $a_{dist,bx}$, and accelerometer walking bias, b_{ax}, are both modeled as random walks. Practically, the only differences between these variables in the model are that $a_{dist,bx}$ appears in the calculation of $\dot{v}_{bx}$ while b_{ax} does not (see Appendix A), and that $a_{dist,bx}$ is assigned a larger covariance in the EKF process noise matrix so that it can evolve more quickly than b_{ax}.

Although a direct measure of the disturbance force is generally not available, rules governing its evolution can be developed based on insight into the physical nature of the disturbance. These rules are implemented using weak constraints described in Geeter et al. (1997) and implemented in a robot model in Julier and Durrant-Whyte (2003). Unlike ad hoc solutions, weak constraints are a principled method for integrating rules and constraints into the Kalman filter framework. Weak constraints can be viewed as virtual measurements or observations. Details on the weak constraints employed in the model-based slip detection algorithm are presented in Ward and Iagnemma (2007, 2008).

3.4 EKF implementation

The model-based slip detection algorithm utilizes an EKF to integrate sensor measurements with the nonlinear robot model. The EKF structure requires that the discrete, nonlinear process model be written in the form $\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, w_{k-1})$, where $\hat{x}_k^-$ is the a priori estimate of the state vector, x, at time step k and f is a nonlinear function of the previous state estimate, $\hat{x}_{k-1}$, the current input vector, u_k, and process noise, w_{k-1}. The measurement vector, z, is a nonlinear function, h, of the true, current state vector and sensor noise v such that $z_k = h(x_k, v_k)$.

The standard EKF time update equations using the notation of Welch and Bishop (2001) and Joseph's form of the covariance update equation (Grewal and Andrews 1993) are used. The relations $f(\hat{x}_{k-1}, u_k, 0)$ and $h(\hat{x}_k^-, 0)$ express the estimated state and measurement vectors, $\hat{x}_k^-$ and $\hat{z}$, by evaluating the nonlinear process and measurement equations, assuming zero noise. Q and R are process and measurement noise covariance matrices and A_k, W_k, H_y, and V_k are process and measurement Jacobians.

The EKF provides an estimate of the robot's forward velocity and the front wheels' angular velocities. Using these

estimates, a criterion for detecting when the robot is immobilized is desired. A natural choice for an "immobilized" metric is the wheel slip i (defined as $i = 1 - (v/r\omega)$, where r is the tire radius). In practice, the calculated wheel slip can be noisy. For example, when the robot is stopped, an incremental wheel motion will yield a calculated slip of 100%, even though the robot is not immobilized. To improve robustness, the exponential moving average (EMA) (Moving Averages 2006) is employed:

$$EMA_{current} = \left(measurement_{current} - EMA_{prev}\right)\left(\frac{2}{1+p}\right) + EMA_{prev} \tag{16}$$

which is an approximation of the time average of the measurement over the last p samples, with a higher weight given to the most recent measurements. Here an EMA of the average of the left and right wheel slips is calculated and immobilization is detected if the EMA is larger than a threshold value. The threshold value is chosen empirically. A low value allows the detector to react quickly, however can be prone to falsely detecting immobilized conditions. In practice, since measurement noise can cause large variations in calculated slip at low speeds, the threshold can be chosen to vary with speed. The above technique represents one possible criterion for detecting immobilization that has worked well in practice, however other criteria are possible, such as learning of failure models (Plagemann et al. 2007).

4 Experimental results

4.1 Classification algorithm performance

The SVM classifier was trained on experimental data gathered by the robot shown in Fig. 3 during traversal of mud, loose mulch, and various grasses at speeds ranging from 0 to 1.0 m/s. The training data included 14 instances of the robot coming to a complete stop with the wheels still spinning, which were initiated by retarding robot motion with a spring scale. Using $N = 50$, the classifier was trained with 408 instance-label pairs, 18% of which were labeled as immobilized.

The classifier was tested using two distinct data sets. In the first set, the robot was driven once again over grass, however immobilization was initiated when the robot experienced significant wheel slip while attempting to surmount a hill. In the second set, the robot was driven over loose gravel mixed with dry, brittle soil, and immobilization was initiated by retarding robot motion with a spring scale. Note that this terrain type was not present in the training data set.

Test results using all four features described in Sect. 2.2 are shown in Fig. 4. Total classification accuracy was 94.7%. The figure shows that all incorrectly labeled points were near an actual immobilized period, with 98.1% of normal points correctly classified. The 1.9% of normal points incorrectly classified as immobilized were all near the start or end of an immobilized period, which could indicate small errors in hand labeling of these extremal points. 75% of immobilized instances were classified correctly; however all immobilized periods were recognized as immobilized in at least some of the data instances comprising that occurrence.

Using only the first three features so that only IMU measurements were required, total classification accuracy was 92.0%, with 97.7% of normal instances and 59.1% of immobilized instances correctly classified. With only three features, classification accuracy was reduced, however false immobilized detections remained low and all immobilized occurrences were again detected.

Figure 5 shows a receiver operating characteristic (ROC) curve for classification of the test data set using all four features. The vertical axis shows the percentage of total

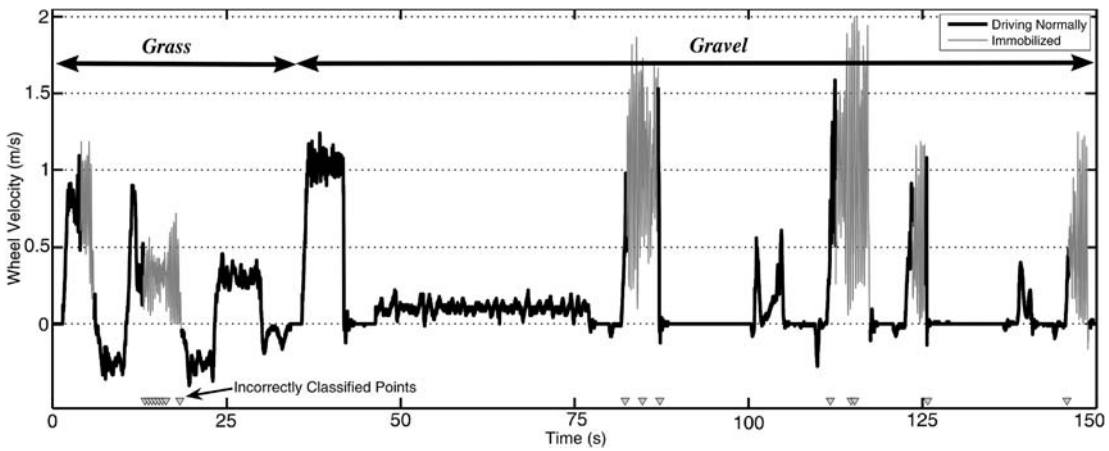

Fig. 4 Experimental results of classifier-based immobilization detection. Each incorrectly classified point is a 0.5 second instance. Wheel velocity is the effective linear velocity at the wheel radius

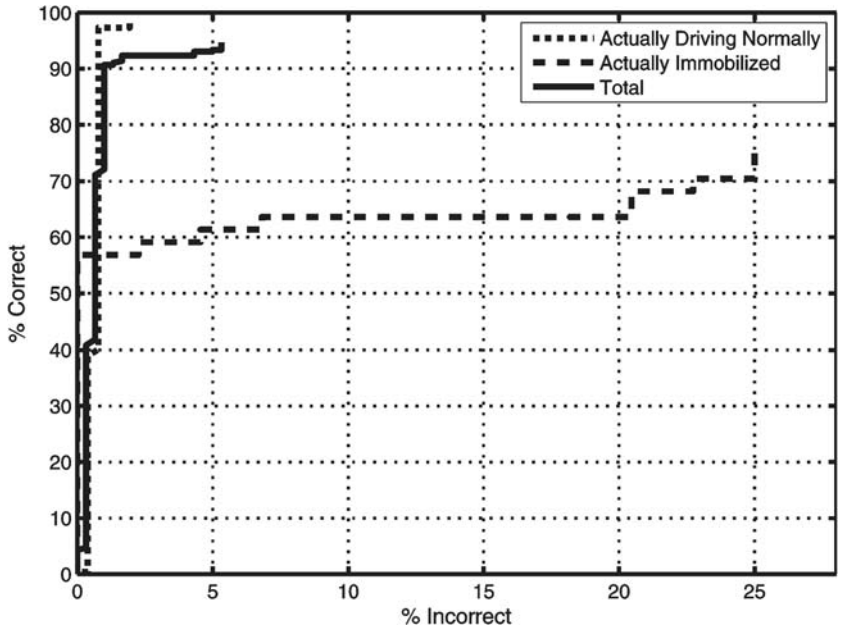

Fig. 5 ROC curve for immobilization detection experimental results

instances that are classified correctly while the horizontal axis shows the percentage classified incorrectly. The curves are generating by progressively increasing *threshold* in (4), causing fewer points to be classified and more points to be "unknown." Thus, increasing *threshold* results in a more conservative classifier. The upper-right endpoint of each line is the classifier accuracy with all instances classified (*threshold* = 0).

It can be seen that as *threshold* is increased, the percentage of incorrect classification initially deceases rapidly, while the percent correct remains nearly constant, meaning in this region the majority of correctly labeled points were further than *threshold* from the class boundary. This curve shows the possible tradeoffs between number of instances labeled and labeling accuracy and can be a useful design tool for practical algorithm implementation.

4.2 Detector fusion

4.2.1 Fusion techniques

To increase immobilization detection accuracy two techniques have been explored to fuse multiple detector outputs. The first technique (termed Fusion 1) is designed to minimize false immobilization detections at the expense of increasing the number of immobilized instances incorrectly classified as normal. For d detectors, D_i, each with output 1 for "immobilized" and -1 for "normal":

$$\text{Fusion 1} = \begin{cases} 1 & \text{if } (D_1 = 1) \text{ AND } (D_2 = 1) \ldots \\ & \quad \text{AND } (D_i = 1) \ldots \text{ AND } (D_d = 1), \\ -1 & \text{otherwise.} \end{cases} \tag{17}$$

Thus Fusion 1 detects immobilization only if all detectors agree that the robot is immobilized.

The second technique (termed Fusion 2) is designed to increase total detection accuracy and yield faster immobilization detection than Fusion 1. For Fusion 2, each detector output, D_i, is expressed as a continuous variable on the interval $[-1, 1]$, with an output of 1 meaning the detector is completely confident that the robot is immobilized, -1 meaning the detector is completely confident the robot is driving normally, and 0 meaning there is an equal probability of the robot being immobilized or driving normally. Fusion 2 is a weighted average of the detector outputs:

$$\text{Fusion 2} = \begin{cases} 1 & \text{if } \sum_{i=1}^{d} w_i D_i > a, \\ -1 & \sum_{i=1}^{d} w_i D_i < -a, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where a is a threshold value and w_i are weights with:

$$\sum_{i=1}^{d} w_i = 1. \tag{19}$$

The values of w_i can be chosen based on a user's overall level of confidence in the individual detector outputs.

4.2.2 Fusion results

The performance of the fusion techniques described in Sect. 4.2.1 was studied using the classification-based detector proposed in Sect. 2 (i.e. the SVM method) and the model-based detector summarized in Sect. 3 (i.e. the EKF method). The output of the SVM method was scaled for Fusion 2 by

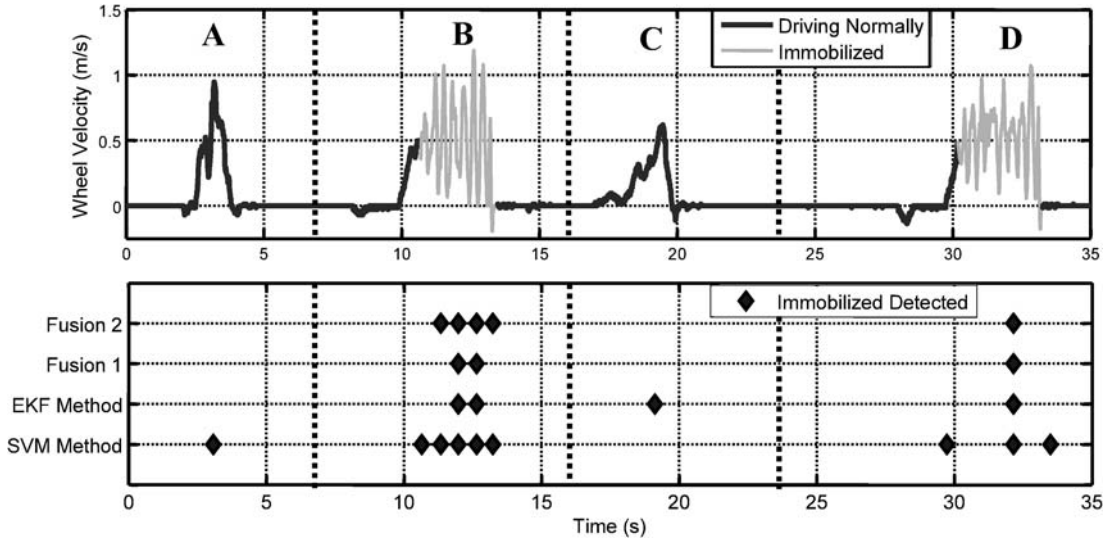

Fig. 6 Detector fusion results. Wheel velocity is effective linear velocity at the wheel radius

first determining the smallest *threshold* for which all classified training points are classified correctly, $threshold_{100\%}$. Then:

$$D_{SVM} = \text{sat}\left(\frac{\hat{f}}{threshold_{100\%}}, 1\right), \tag{20}$$

where the saturation function, $\text{sat}(x, y)$, is defined here as:

$$\text{sat}(x, y) = \begin{cases} x & \text{if } |x| < |y|, \\ \text{sign}(x) \cdot |y| & \text{otherwise.} \end{cases} \tag{21}$$

The EKF method outputs a detected class for each of the N data points that make up instance i of the SVM method, but does not output a confidence value. D_{EKF} is therefore taken as the mean of the N data points, providing an estimate of the detector's confidence. If half of the N points are classified as immobilized, then there is approximately a 50% chance the robot was immobilized during those data points and $D_{EKF} = 0$. This estimate has the drawback of assigning low confidence when immobilization begins near the end of the N points, possibly leading to sub-optimal detection time; however it provides a computationally simple method to test the fusion technique performance. For Fusion 2, $w_1 = w_2 = 0.5$ and $a = 0$ were used.

Figure 6 shows a dataset of the robot driving over loose mulch that demonstrates the relative performance of the fusion techniques. In sections A and C of the dataset the robot was driven normally under remote control, and in sections B and D the robot was commanded to drive forward at 0.5 m/s but was restrained with a spring scale, causing immobilization. The bottom plot indicates the moments when immobilization was detected by the two detectors and two fusion techniques.

It can be seen that in section A the SVM method falsely detected immobilization, likely due to the rapid wheel speed oscillation. The EKF method, however, correctly labeled this instance as normal driving, allowing both fusion techniques to correctly label this section as normal. Similarly, in section C the EKF method misclassified an instance as immobilized, but the SVM method and both fusion methods correctly classified this section.

In section B, the SVM method detected immobilization very rapidly, while the EKF method's detection time was approximately 1.0 second slower. In this case, the SVM method detected immobilization immediately after the robot began to decelerate, while the EKF method detected immobilization when the robot came to a stop. As expected, Fusion 1 only detected immobilization when both detectors agreed. Fusion 2 was able to detect immobilization approximately 0.5 seconds sooner than Fusion 1 because the SVM method expressed high confidence in its output while the EKF method expressed an uncertain output (i.e. an output near 0). In section D, the SVM method expressed a low confidence in its early immobilization detection and neither fusion technique detected immobilization until the EKF method was in agreement.

Table 1 compares detection accuracy of the four methods when run on the experimental dataset of Sect. 4.1, which included 301 half second instances. All four techniques detected the 6 immobilized periods. The SVM method detected immobilization the fastest, followed by Fusion 2; however in some cases the SVM method detected immobilization before the robot was stopped, accounting for the 3 false positives. Both fusion techniques eliminated these false positives, with Fusion 2 demonstrating the highest total accuracy.

Table 1 Comparison of accuracy of detection and fusion techniques on the experimental dataset

	SVM method	EKF method	Fusion 1	Fusion 2
Total accuracy:	94.7%	95.7%	91.7%	98.0%
# False positives:	3	0	0	0
# False negatives:	13	13	25	6

Although not shown in Fig. 6 or Table 1, it is possible that a detector could falsely label an instance with high enough confidence for the point to be mislabeled by Fusion 2 but not Fusion 1. Fusion 1 should therefore be more robust to false positives. Note that if both detectors mislabel an instance, it will be mislabeled under both fusion techniques.

5 Conclusions

A signal recognition based approach to detecting mobile robot immobilization has been proposed and experimentally validated. Four distinguishing features have been proposed for the algorithm requiring an IMU and (optionally) wheel encoders or tachometers, both common sensors on outdoor mobile robots. Future work will explore the effects of SVM kernel selection and robot speed and configuration on algorithm performance and test the algorithm on alternate terrain types and scenarios.

Two simple detector fusion techniques have been proposed to combine the output of the classifier-based immobilization detector and a dynamic model-based detector. Fusion 1 resulted in a conservative approach to minimize false detections, while Fusion 2 provided faster performance while potentially allowing more false detections. Both fusion techniques were shown to eliminate false immobilization detections on the experimental dataset and increase overall accuracy compared to each individual detector. Future work will explore using various fusion techniques to combine more than two detectors, and for alternative applications including terrain classification.

Acknowledgement The authors would like to thank Andrew Lookingbill at Stanford University for his help in collecting experimental data.

Appendix A: Robot dynamic model

A.1 Robot dynamics

Here a dynamic model of the robot shown in Fig. 3 is presented. Modeled forces acting on the robot include gravity, a lumped external disturbance force, and tire forces acting at the four tire-terrain contact patches (see Fig. 7). The disturbance force can represent a variety of external forces such as wind resistance or the force caused by collision with an obstacle. In this work we limit the disturbance force to forces resisting vehicle motion.

Tire forces are composed of a normal component, traction/braking component, rolling resistance component, and lateral force component. The traction/braking forces are negligible for any undriven, freely rolling wheels, as is the case for the rear wheels of the robot considered here. The rear lateral forces can also be neglected because the rear castors spin freely and thus usually align with their velocity vectors.

The robot acceleration along the body x-axis is:

$$\dot{v}_{bx} = \frac{1}{m}\left(\sum_{i=1}^{2} F_{i,tract} + \sum_{i=1}^{4} F_{i,roll\ res} + F_{disturb} - mg\sin(\varphi)\right) = f_{tire} + a_{dist,bx} - g\sin(\varphi), \tag{22}$$

where m is the total vehicle mass and g is the acceleration due to gravity. Assuming the vehicle's axis of yaw rotation is approximately the point midway between the front tires and neglecting any yaw moment due to gravity (which is small for moderate vehicle roll), the vehicle's yaw angular acceleration is:

$$\ddot{\psi} = \frac{l_w}{2J}(F_{1,tract} + F_{1,roll\ res} - F_{2,tract} - F_{2,roll\ res}) = g_{tire}, \tag{23}$$

where J is the vehicle's moment of inertia about the body z-axis. In general, if a robot has non-negligible lateral forces which do not act through the yaw axis, they must be estimated and included in (23).

Calculation of the robot's normal forces with arbitrary body roll and pitch is in general an underconstrained problem. Methods proposed in the literature (Gillespie 1992; Bekker 1956) typically consider a simplified 2-wheeled "bicycle" model, which can be applied for 2 or 4 wheeled vehicles when roll effects are ignored. In Hung and Orin (2001) it is suggested that normal forces be estimated by considering the elasticity of the terrain using tire-soil contact models presented in Bekker (1969). A rigid body solution can also be found (utilizing the Moore-Penrose generalized inverse), assuming point tire-soil contact (Kumar and Waldron 1988).

For the robot configuration considered here, the assumption of zero moment about the passive rear suspension pivot joint allows the rear left and right normal forces to be assumed equal. With this assumption the normal force calculation is no longer underconstrained and an explicit solution exists. For normal force calculations it is also assumed that the vehicle longitudinal acceleration is negligibly small, which is generally valid for slow-moving robots. The normal forces are:

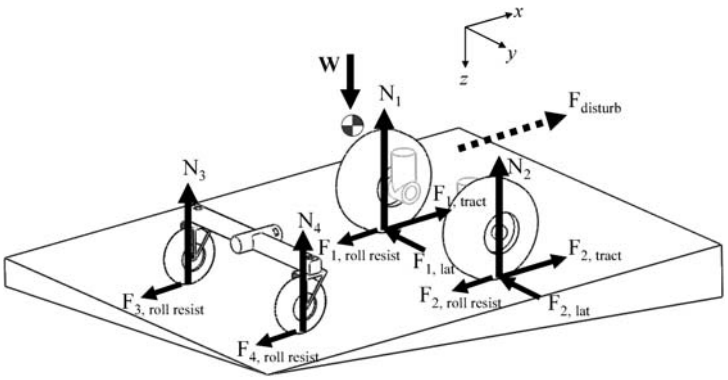

Fig. 7 Diagram showing vehicle and tire forces

$$N_1 = \frac{W}{2}\cos(\varphi)\left(\frac{1}{l_f + l_r}(l_r\cos(\theta) - l_h\tan(\varphi)) - \frac{2l_h}{l_w}\sin(\theta)\right), \tag{24}$$

$$N_2 = \frac{W}{2}\cos(\varphi)\left(\frac{1}{l_f + l_r}(l_r\cos(\theta) - l_h\tan(\varphi)) + \frac{2l_h}{l_w}\sin(\theta)\right), \tag{25}$$

$$N_3 = N_4 = \frac{W}{2(l_f + l_r)}(l_h\sin(\varphi) + l_f\cos(\varphi)\cos(\theta)). \tag{26}$$

As a notational convenience we define the "normal accelerations" as:

$$n_{f,l} = \frac{N_1}{m}, \qquad n_{f,r} = \frac{N_2}{m}, \qquad n_r = \frac{N_3}{m} = \frac{N_4}{m}.$$

A.2 Traction/braking model

A large body of research has been performed on modeling tire forces on rigid and deformable terrain. Most models are semi-empirical and express tire traction/braking forces as a function of wheel slip i and wheel skid i_s (where $i_s = 1 - (r\omega/v)$). Implementation of a slip-based tire traction model has many practical difficulties, including the need to distinguish the cases of traction and braking and driving forward and reverse to correctly calculate slip or skid. Another difficulty is introduced by the fact that most formulations are undefined at zero slip (i.e. when $\omega = 0$ while slipping or $v = 0$ while skidding). Additionally, some models require separate formulations for the low and high slip regimes. In Ward and Iagnemma (2007) a unified, explicitly differentiable traction/braking model was proposed that captures the critical elements of the models proposed in the literature. This model is summarized here for completeness.

In the proposed model the traction/braking force is expressed as a function of the wheel's relative velocity, rather than slip. Slip is a normalized version of relative velocity. A relative velocity-based formulation does not introduce the singularities found in slip-based formulations and is consequentially convenient to apply within an extended Kalman filter framework. The simplified model is:

$$F_{Traction} = N(\text{sign}(v_{rel})C_1(1 - e^{-A_t|v_{rel}|}) + C_2 v_{rel}), \tag{27}$$

where v_{rel} is the velocity of the tire relative to the ground:

$$v_{rel} = r\omega - v_{fwd}, \tag{28}$$

and v_{fwd} is the tire's forward velocity, computed as:

$$v_{fwd,left} = v_{bx} + \frac{l_w\dot{\psi}}{2}, \qquad v_{fwd,right} = v_{bx} - \frac{l_w\dot{\psi}}{2}, \tag{29}$$

where C_1, A_t, and C_2 are constants.

The simplified model is continuously differentiable and can predict both traction and braking forces, without a need to distinguish the two cases. Additionally, this model requires three terrain/tire dependant parameters (C_1, A_t, and C_2). By comparison, the popular "Magic Formula" empirical tire model requires six parameters (Wong 2001). C_1 is a positive constant which can be viewed as the maximum tire-terrain traction coefficient. A_t is also a positive constant and is the slope of the traction curve in the low relative velocity region. C_2 is the slope in the high relative velocity range and can be positive or negative depending on the terrain (see Fig. 8).

Whereas the traction-slip curve does not vary with wheel velocity using a slip-based model, the proposed model does. Assuming the robot typically operates near a nominal velocity, the proposed model can be interpreted as a pseudo-linearization around the nominal operating velocity. In Ward and Iagnemma (2008) it is shown that when the range of operating velocities is within an order of magnitude of the nominal velocity, the proposed tire model provides reasonably accurate traction force calculations (i.e. within 10%) compared to slip-based models. Note that multiple models using different constants at multiple operating points can be employed if needed.

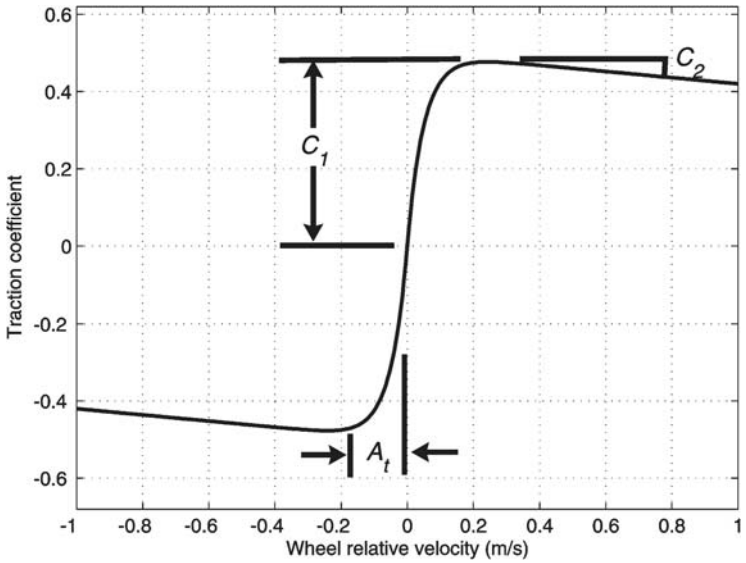

Fig. 8 Representative traction coefficient vs. relative velocity curve indicating effect of the three traction parameters. The traction coefficient is $F_{traction}/N$

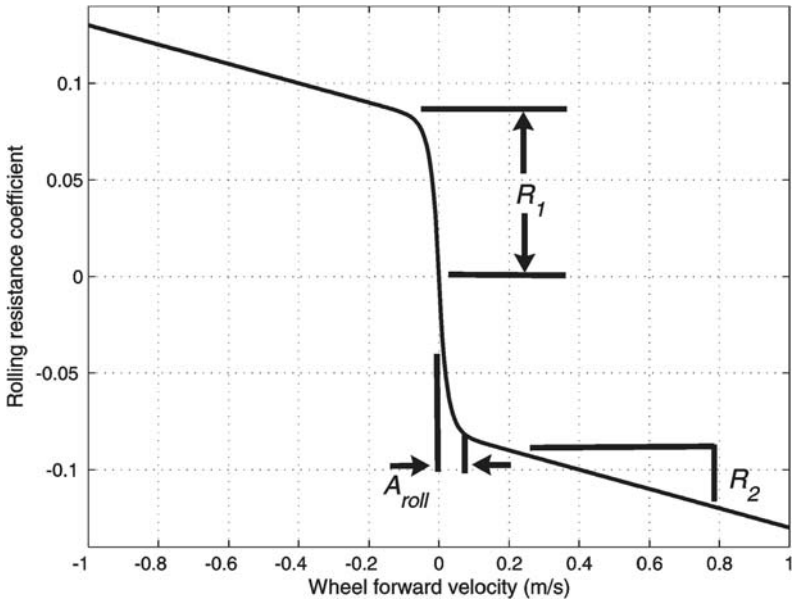

Fig. 9 Representative rolling resistance coefficient vs. velocity curve indicating the effect of varying the three resistance parameters. The rolling resistance coefficient is $F_{roll\ resist}/N$

A.3 Rolling resistance model

Rolling resistance is generally modeled as a combination of static and velocity dependant forces (Bekker 1956; Wong 2001). Here a rolling resistance formulation with form similar to (27) is presented as a continuously differentiable function with the static force smoothed at zero velocity to avoid a singularity. The rolling resistance is:

$$F_{roll\ res} = -\operatorname{sign}(\upsilon_{fwd}) \times N(R_1(1 - e^{-A_{roll}|\upsilon_{fwd}|}) + R_2|\upsilon_{fwd}|), \tag{30}$$

where R_1, A_{roll}, and R_2 are positive constants. Figure 9 shows a representative rolling resistance versus wheel translational velocity curve.

A.4 Combined tire dynamics

Combining (22), (24)–(27), and (30), the robot acceleration due to tire traction/braking forces can be calculated as:

$$\begin{aligned} f_{tire} = {} & n_{f,l}(\operatorname{sign}(\upsilon_1)C_1(1 - e^{-A_t|\upsilon_1|}) + C_2\upsilon_1 \\ & - \operatorname{sign}(\upsilon_3)R_{1,front}(1 - e^{-A_{roll}|\upsilon_3|}) - R_{2,front}\upsilon_3) \\ & + n_{f,r}(\operatorname{sign}(\upsilon_2)C_1(1 - e^{-A_t|\upsilon_2|}) + C_2\upsilon_2 \\ & - \operatorname{sign}(\upsilon_4)R_{1,front}(1 - e^{-A_{roll}|\upsilon_4|}) - R_{2,front}\upsilon_4) \\ & - 2n_r(\operatorname{sign}(\upsilon_5)R_{1,rear}(1 - e^{-A_{roll}|\upsilon_5|}) - R_{2,rear}\upsilon_5) \end{aligned} \tag{31}$$

where $\upsilon_1 = \upsilon_{rel,front,left}$, $\upsilon_2 = \upsilon_{rel,front,right}$, $\upsilon_3 = \upsilon_{fwd,front,left}$, $\upsilon_4 = \upsilon_{fwd,front,right}$, $\upsilon_5 = \upsilon_{bx}$.

Combining (23), (27), and (30), the yaw acceleration due to tire forces is:

$$\begin{aligned} g_{tire} = {} & \frac{l_w}{2J} N_{f,l} (\text{sign}(\upsilon_1) C_1 (1 - e^{-A_t |\upsilon_1|}) + C_2 \upsilon_1 \\ & - \text{sign}(\upsilon_3) R_{1,front} (1 - e^{-A_{roll,front} |\upsilon_3|}) - R_{2,front} \upsilon_3) \\ & - \frac{l_w}{2J} N_{f,r} (\text{sign}(\upsilon_2) C_1 (1 - e^{-A |\upsilon_2|}) + C_2 \upsilon_2 \\ & + \text{sign}(\upsilon_4) R_{1,front} (1 - e^{-A_{roll,front} |\upsilon_4|}) - R_{2,front} \upsilon_4). \end{aligned} \tag{32}$$

References

Anderson, R., & Bevly, D. (2004). Estimation of slip angles using a model based estimator and GPS. In *Proceedings of the 2004 American control conference* (Vol. 3, pp. 2122–2127).

Angelova, A., Matthies, L., Helmick, D., Sibley, G., & Perona, P. (2006). Learning to predict slip for ground robots. In *Proceedings of the IEEE international conference on robotics and automation*.

Barshan, B., & Durrant-Whyte, H. (1995). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation, 11*(3).

Bekker, M. (1956). *Theory of land locomotion*. Ann Arbor: Univ. of Michigan Press.

Bekker, M. (1969). *Introduction to terrain-vehicle systems*. Ann Arbor: Univ. of Michigan Press.

Borenstein, J., Everett, H., & Feng, L. (1996). *Where am I? Sensors and methods for mobile robot positioning*. Univ. of Michigan. Available: http://www-personal.umich.edu/~johannb/shared/pos96rep.pdf.

Brooks, C., & Iagnemma, K. (2005). Vibration-based terrain classification for planetary exploration rovers. *IEEE Transactions on Robotics, 21*(6).

Chang, C., & Lin, C. (2001). LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm. Accessed 2001.

Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., & Bradski, G. (2006). Self-supervised monocular road detection in desert terrain. In *Proceedings of the robotics science and systems conference*.

Dissanayake, G., Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (2001). The aiding of a low-cost strapdown measurement unit using vehicle model constraints for land vehicle applications. *IEEE Transactions on Robotics and Automation, 17*(5).

Flenniken, W., Wall, J., & Bevly, D. (2005). Characterization of various IMU error sources and the effect on navigation performance. In *Proceedings of the Institute of Navigation GNSS conference*.

Fuke, Y., & Krotkov, E. (1996). Dead reckoning for a lunar rover on uneven terrain. In *Proceedings of the IEEE international conference on robotics and automation*.

Ganapathiraju, A., Hamaker, J., & Picone, J. (2004). Applications of support vector machines to speech recognition. *IEEE Transactions on Signal Processing, 52*(8).

Geeter, J., Brussel, H., & Schutter, J. (1997). A smoothly constrained Kalman filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(10).

Gillespie, T. (1992). *Fundamentals of vehicle dynamics*. Warrendale: Society of Automotive Engineers.

GPS 16/17 Series Technical Specifications (2005). Revision A, Garmin International, Inc. Olathe, KS.

Grewal, M., & Andrews, A. (1993). *Kalman filtering: theory and practice*. Englewood Cliffs: Prentice-Hall.

Gustafsson, F. (1997). Slip-based tire-road friction estimation. *Automatica, 33*(6), 1087–1099.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer.

Hofmann-Wellenhof, B., Lichtenegger, H., & Collins, J. (2001). *Global positioning system: theory and practice* (5th ed.). New York: Springer.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2006). A practical guide to support vector classification. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf. Accessed 2006.

Hung, M., & Orin, D. (2001). Dynamic simulation of actively-coordinated wheeled vehicle systems on uneven terrain. In *Proceedings of the 2001 IEEE international conference on robotics and automation*.

Julier, S., & Durrant-Whyte, H. (2003). On the role of process models in autonomous land vehicle navigation systems. *IEEE Transactions on Robotics and Automation, 19*(1).

Karlsen, R.E., Witus, G. (2007). Terrain understanding for robot navigation. In *Proceedings of the IEEE/RSJ conference on intelligent robots and systems*.

Kelly, A. (1994). *A 3D state space formulation of a navigation Kalman filter for autonomous vehicles* (CMU Technical Report CMU-RI-TR-94-19).

Kumar, V., & Waldron, K. (1998). Force distribution in closed kinematic chains. *IEEE Transactions on Robotics and Automation, 4*(6).

Learning Applied to Ground Robots (2006). http://www.darpa.mil/ipto/Programs/lagr/vision.htm. Accessed 2006.

Manduchi, R. (2004). Learning outdoor color classification from just one training image. *Proceedings of European Conference on Computer Vision (ECCV), 28*(11).

Moving Averages (2006). http://www.stockcharts.com/education/IndicatorAnalysis/indic_movingAvg.html. Accessed July 2006.

Ojeda, L., Reina, G., & Borenstein, J. (2004). Experimental results from FLEXnav: an expert rule-based dead-reckoning system for Mars rovers. In *Proceedings of the IEEE aerospace conference*.

Ojeda, L., Cruz, D., Reina, G., & Borenstein, J. (2006). Current-based slippage detection and odometry correction for mobile robots and planetary rovers. *IEEE Transactions on Robotics, 22*(2).

Plagemann, C., Fox, D., & Burgard, W. (2007). Efficient failure detection on mobile robots using particle filters with Gaussian process proposals. In *Proceedings of the international joint conferences on artificial intelligence*.

Rasmussen, C. (2001). Laser range-, color-, and texture-based classifiers for segmenting marginal roads. In *Proceedings of conference on computer vision & pattern recognition technical sketches*.

Ryu, J., Rossetter, E., & Gerdes, J. (2002). Vehicle sideslip and roll parameter estimation using GPS. In *Proceedings of the advanced vehicle control international symposium*.

Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation, 15*(3).

Wadda, M., Yoon, K., & Hashimoto, H. (2000). High accuracy road vehicle state estimation using extended Kalman filter. In *Proceedings of 2000 IEEE intelligent transportation systems*.

Ward, C., & Iagnemma, K. (2007). Model-based wheel slip detection for outdoor mobile robots. In *Proceedings of the IEEE international conference on robotics and automation*.

Ward, C., & Iagnemma, K. (2008). A dynamic model-based wheel slip detector for mobile robots on outdoor terrain. *IEEE Transactions on Robotics*, *24*(4), 821–831.

Welch, G., & Bishop, G. (2001). An introduction to the Kalman filter. In *Proceedings of SIGGRAPH*.
Wong, J. (2001). *Theory of ground vehicles* (3rd ed.). New York: Wiley.

Karl Iagnemma is a principal research scientist in the Mechanical Engineering department at the Massachusetts Institute of Technology. He holds a B.S. from the University of Michigan, and an M.S. and Ph.D. from MIT, where he was a National Science Foundation Graduate Fellow. He has performed postdoctoral research at MIT, and has been a visiting researcher at the NASA Jet Propulsion Laboratory and the National Technical University of Athens. Dr. Iagnemma's primary research interests are in the areas of design, sensing, motion planning, and control of mobile robots in outdoor terrain, including modeling and analysis of robot-terrain interaction. He is author of the monograph *Mobile Robots in Rough Terrain: Estimation, Planning and Control with Application to Planetary Rovers* (Springer, 2004). He currently serves as an associate editor of the *IEEE Transactions on Robotics*.

Chris C. Ward received his M.S. in mechanical engineering from the Massachusetts Institute of Technology in Cambridge, MA in 2007. At MIT he worked in the Field and Space Robotics Laboratory where his research focused on dynamic outdoor mobile robots. He received his B.S. degree from the University of California, Berkeley. He is currently a mechatronic design engineer for ASML in Wilton, CT where he works on control systems for high-speed precision stages for semiconductor lithography systems.