A Dynamic-Model-Based Wheel Slip Detector for Mobile Robots on Outdoor Terrain

Chris C. Ward and Karl Iagnemma

Abstract—This paper introduces a model-based approach to estimating longitudinal wheel slip and detecting immobilized conditions of autonomous mobile robots operating on outdoor terrain. A novel tire traction/braking model is presented and used to calculate vehicle dynamic forces in an extended Kalman filter framework. Estimates of external forces and robot velocity are derived using measurements from wheel encoders, inertial measurement unit, and GPS. Weak constraints are used to constrain the evolution of the resistive force estimate based upon physical reasoning. Experimental results show the technique accurately and rapidly detects robot immobilization conditions while providing estimates of the robot's velocity during normal driving. Immobilization detection is shown to be robust to uncertainty in tire model parameters. Accurate immobilization detection is demonstrated in the absence of GPS, indicating the algorithm is applicable for both terrestrial applications and space robotics.

Index Terms—Extended Kalman filter (EKF), mobile robots, robot–terrain interaction, wheel slip.

I. INTRODUCTION

MOBILE ROBOT position estimation systems typically rely (in part) on wheel odometry as a direct estimate of displacement and velocity [1], [6]. 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 traction control. 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 [2]; however, this does not apply for all-wheel drive vehicles or those without redundant encoders. Ojeda et al. [3] 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; however, this technique does not estimate the degree of wheel slip (i.e., whether the robot is fully immobilized). Ojeda et al. have also proposed a motor-current-based slip estimator [4]; 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. Visual odometry (VO) is used in [29] to estimate robot velocity and slip for a slip prediction algorithm. Although VO can be accurate on average over time, the authors report VO errors of ~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 antilock braking systems (ABSs); however, such work generally applies at significantly higher speeds than is typical for autonomous robots.

A large amount of work has utilized Kalman filters with inertial and absolute measurements to enhance dead reckoning and estimate lateral slip. A navigation system is proposed in [7] that uses inertial measurements combined with a sensor error model in a Kalman filter to increase measurement accuracy. Traditional dead reckoning accuracy is improved in [6] and [8] by including inertial measurements. Absolute position updates from GPS are fused in [5] and [9] with a model-based Kalman filter to estimate vehicle sideslip and improve position estimation accuracy, and in [10], 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 [11]. None of these works, however, explicitly considers the effects of longitudinal wheel slip or vehicle 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 [12], [14]. Additionally, GPS provides low-frequency updates (e.g., typically near 1 Hz [13]) making GPS alone too slow for immobilization detection, where as close to instantaneous detection as possible is desired to avoid excessive position errors.
Another potentially simple approach could rely on comparison of wheel velocities to a robot body velocity estimate derived from integration of a linear acceleration measurement in the direction of travel. As shown in [28], however, such an approach is not robust at low speeds during travel on rough, outdoor terrain. Dissanayake et al. [30] proposed a method of aiding the inertial estimate using vehicle constraints; however, the method is not appropriate on uneven, low-traction terrain. Ward and Iagnemma [28] also demonstrated a signal-recognition-based approach to the immobilization detection problem.

Here, a method is presented for detecting robot wheel slip and immobilization that does not require redundant wheel encoders or motor current measurements. The proposed approach uses a dynamic vehicle model fused with wheel encoder, inertial measurement unit (IMU), and (optional) GPS measurements in an extended Kalman filter (EKF) to create an estimate of the robot’s longitudinal velocity. An insight of this approach is the realization that a robot becomes immobilized due to an external force resisting motion, be it a gravitational force resisting movement on an incline or an impact force exerted during a collision. The proposed algorithm utilizes a novel tire traction/braking model in combination with sensor data to estimate external resistive forces acting upon the robot and calculate the robot’s acceleration and velocity. Weak constraints are used to constrain the evolution of the resistive force estimate based upon physical reasoning. The algorithm has been shown to accurately detect immobilized conditions on a variety of terrain types and provide an estimate of the robot’s velocity during “normal” driving. The algorithm has been run in real time onboard a mobile robot and is shown to be robust to periods of GPS dropout. The proposed approach captures all relevant dynamics using one well-defined, continuous model as opposed to approaches that seek to capture the complete dynamics by combining multiple limited models such as in [31] and [32], or that learn a generalized motion model [34].

This paper is organized as follows. In Section II, the vehicle dynamic model and tire model are presented. In Section III, the slip detection algorithm, including the process and measurement models used in the EKF, is described. In Section IV, experimental results showing the algorithm accurately detecting immobilized conditions as a robot traverses a variety of natural terrains at a range of speeds are presented. An analysis of the algorithm’s sensitivity to tire model parameters is also presented. Additionally, accurate immobilization detection is demonstrated in the absence of GPS updates. In Section V, conclusions are drawn from this paper and future work is suggested.

II. DYNAMIC MODELS

A. Robot Configuration

The robot configuration considered in this paper is shown in Fig. 1. The robot has four rubber pneumatic tires and is a front-wheel differential-drive configuration with undriven rear wheels that are freely rotating castors mounted to a rear pivot joint suspension. The robot body-fixed coordinate system and kinematic parameters are shown in Fig. 2, and the robot body and tire forces are shown in Fig. 3. The dynamic models presented later are specific to this robot configuration; however, the modeling process is adaptable to other wheeled vehicle configurations. This robot was designed for the Defense Applied Research Projects Agency (DARPA) Learning Applied to Ground Robots (LAGR) research program [15].

B. Vehicle Dynamics

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. 3). 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 paper, 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 vehicle acceleration along the body x-axis is

$$\dot{v}_{bx} = \frac{1}{m} \left( \sum_{i=1}^{2} F_{i, \text{tract}} + \sum_{i=1}^{4} F_{i, \text{roll res}} + F_{\text{disturb}} - mg \sin(\phi) \right)$$

$$= f_{\text{tire}} + a_{\text{dist, bx}} - g \sin(\phi) \quad (1)$$

where \( m \) is the total vehicle mass, \( g \) is the acceleration due to gravity, and \( f_{\text{tire}} \) and \( a_{\text{dist, bx}} \) are the equivalent x-axis body accelerations due to tire forces and the disturbance force. Assuming the vehicle’s axis of yaw rotation is approximately the z-axis body axis, the vehicle’s yaw angular acceleration is

$$\psi = \frac{e}{2I} \left( F_{1, \text{tract}} + F_{1, \text{roll res}} - F_{2, \text{tract}} - F_{2, \text{roll res}} \right)$$

$$= g_{\text{tire}} \quad (2)$$

where \( J \) is the vehicle’s moment of inertia about the body z-axis and \( e \) is the distance between front wheel centers. In general, if a robot has nonnegligible lateral forces that do not act through the yaw axis, they must be estimated [21] and included in (2).

C. Normal Forces

Calculation of the robot’s normal forces with arbitrary body roll \((\theta)\) and pitch \((\phi)\) is in general an underconstrained problem. Methods proposed in the literature [16, 17] typically consider a simplified two-wheeled “bicycle” model, which can be applied for two- or four-wheeled vehicles when roll effects are ignored. In [18], it is suggested that normal forces be estimated by considering the elasticity of the terrain using tire–soil contact models presented in [19]. A rigid body solution can also be found (utilizing the Moore–Penrose generalized inverse), assuming point tire–soil contact [20].

For the robot configuration considered in this paper, assuming 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

$$N_1 = \frac{W}{2} \cos(\phi) \left( \frac{1}{a + b} (b \cos(\theta) - h \tan(\phi)) - \frac{2h}{c} \sin(\theta) \right) \quad (3)$$

$$N_2 = \frac{W}{2} \cos(\phi) \left( \frac{1}{a + b} (b \cos(\theta) - h \tan(\phi)) + \frac{2h}{c} \sin(\theta) \right) \quad (4)$$

$$N_3 = N_4 = \frac{W}{2(a + b)} \left( h \sin(\phi) + a \cos(\phi) \cos(\theta) \right). \quad (5)$$

As a notational convenience, we define the “normal accelerations” as

$$n_{f,i} = \frac{N_1}{m}, \quad n_{f,r} = \frac{N_2}{m}, \quad n_r = \frac{N_3}{m} = \frac{N_4}{m}. \quad (6–8)$$

D. Traction/Braking Model

A large body of research has been performed on modeling tire forces on rigid and deformable terrain. Most models are semiempirical and express tire traction/braking forces as a function of wheel slip \( i \) and wheel skid \( i_s \), where [21]

$$i = 1 - \frac{v}{\hat{v}} \quad (9)$$

and

$$i_s = 1 - \frac{r \hat{\omega}}{v} \quad (10)$$

where \( r \) is the tire radius and \( \hat{\omega} \) is the wheel angular velocity. For example, in [21], the traction force of a pneumatic tire on rigid terrain is formulated as

$$F_{\text{traction}} = \left\{ \begin{array}{ll} K_i \varepsilon \approx C_i i, & \varepsilon \leq i_{\text{critical}} \quad (11) \\ \mu_p N - \frac{\lambda (\mu_p N - K_i)}{2l_i K_i}, & i > i_{\text{critical}} \end{array} \right.$$
Fig. 4. Representative traction coefficient versus relative velocity curve indicating effect of the three traction parameters. The traction coefficient is $F_{\text{traction}}/N$.

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. 4).

Fig. 5 shows a plot of traction versus wheel slip for lines of constant wheel velocity using the proposed traction/braking model, as well as a representative traction force curve generated using (11). Whereas the traction–slip curve does not vary with wheel velocity using the slip-based model, the proposed model does. Assuming the robot typically operates near a nominal velocity, the proposed model can be interpreted as a pseudolinearization around the nominal operating velocity. Fig. 6 compares displacement estimates calculated using the two traction models of Fig. 5. With wheel velocities ranging from 0 to 2 m/s, the difference in displacement between the two estimates is on the order of centimeters. When the range of operating velocities is within an order of magnitude of the nominal velocity, a single tire model should be sufficient for most applications; however, multiple models using different constants at multiple operating points can be employed if needed.

E. Rolling Resistance Model

Rolling resistance is generally modeled as a combination of static- and velocity-dependant forces [17], [21]. Here, a function with form similar to (12) is proposed as a continuously differentiable formulation of the rolling resistance with the static force smoothed at zero velocity to avoid a singularity. The rolling resistance is

$$F_{\text{roll res}} = -\text{sign}(v_{\text{wd}})N(R_1(1 - e^{-A_{\text{roll}}|v_{\text{wd}}|}) + R_2|v_{\text{wd}}|)$$

(15)

where $R_1$, $A_{\text{roll}}$, and $R_2$ are positive constants. Fig. 7 shows a representative rolling resistance versus wheel translational velocity curve.

F. Combined Tire Dynamics

Combining (1), (6)–(8), (12), and (15), the vehicle acceleration due to tire traction/braking forces can be calculated as

$$f_{\text{tire}} = n_{f,l} \left( \frac{\text{sign}(v_1)C_1(1 - e^{-A_i|v_1|}) + C_2v_1}{-R_2,\text{front}} \right)$$

$$-R_2,\text{front} v_3$$

$$+ n_{f,r} \left( \frac{\text{sign}(v_4)C_1(1 - e^{-A_i|v_4|}) + C_2v_4}{-R_2,\text{front}} \right)$$

$$-R_2,\text{front} v_4$$

$$- 2n_r (\text{sign}(v_5)R_1,\text{rear}(1 - e^{-A_i|v_5|}) - R_2,\text{rear} v_5)$$

(16)

where

$$v_1 = v_{\text{rel,front,left}}, \quad v_2 = v_{\text{rel,front,right}}, \quad v_3 = v_{\text{wd,front,left}}$$

$$v_4 = v_{\text{wd,front,right}}, \quad v_5 = v_{bz}.$$
Combining (2), (12), and (15), the yaw acceleration due to tire forces is

\[
g_{\text{tire}} = \frac{c}{2J} N_{f,t} \left( \begin{align*}
\text{sign}(v_1)C_1 \left( 1 - e^{-A_1 |v_1|} \right) + C_2 v_1 \\
-\text{sign}(v_3)R_{1,\text{front}} \left( 1 - e^{-A_{1,\text{roll},\text{front}} |v_3|} \right) \\
-R_{2,\text{front}} v_3
\end{align*} \right)
\]

\[
- \frac{c}{2J} N_{f,r} \left( \begin{align*}
\text{sign}(v_2)C_1 \left( 1 - e^{-A_1 |v_1|} \right) + C_2 v_2 \\
+\text{sign}(v_4)R_{1,\text{front}} \left( 1 - e^{-A_{1,\text{roll},\text{front}} |v_4|} \right) \\
-R_{2,\text{front}} v_4
\end{align*} \right).
\]

The models in (16) and (17) will be utilized in the slip estimation algorithm presented in the following section.

III. SLIP DETECTOR ALGORITHM

A. Extended Kalman Filter

The slip detector algorithm utilizes an EKF to integrate sensor measurements with the nonlinear vehicle 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 \( \nu \) such that

\[
z_k = h(\hat{x}_k, \nu_k).
\]

The standard EKF time update equations using the notation of [23] are

\[
\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0)
\]

\[
P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T
\]

and the EKF measurement update equations using Joseph’s form of the covariance update equation [24] are

\[
K_k = P_k^- H_k^T \left( H_k P_k^- H_k^T + V_k R_k V_k^T \right)^{-1}
\]

\[
\hat{x}_k = \hat{x}_k^- + K_k \left( z_k - h(\hat{x}_k^-) \right)
\]

\[
P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T.
\]

The relations \( f(\hat{x}_{k-1}, u_k, 0) \) and \( h(\hat{x}_k^-) \) 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, \( P \) is the state error covariance matrix, and \( A_k, W_k, H_k, \) and \( V_k \) are process and measurement Jacobian matrices, where

\[
A_{i,j} = \frac{\partial f_i}{\partial x_j}(\hat{x}_{k-1}, u_k, 0)
\]

\[
W_{i,j} = \frac{\partial f_i}{\partial u_j}(\hat{x}_{k-1}, u_k, 0)
\]

\[
H_{i,j} = \frac{\partial h_i}{\partial z_j}(\hat{x}_k^-, 0).
\]

To apply the EKF to real-world sensors with unique, inconsistent sampling rates, a modified form of the EKF update process is required. The time update equations (20), (21) are calculated at a constant time step \( \Delta t \), such that in the absence of measurements, the state estimate is updated based upon the dynamic model. When a new measurement from sensor \( \sigma \) becomes available, the measurement update equations (22)–(24) are computed for that measurement only, using \( H_{k,\sigma}, V_{k,\sigma}, R_{k,\sigma}, \) and \( h_{\sigma} \), which are the portions of the measurement Jacobians, measurement error covariance, and measurement function corresponding to sensor \( \sigma \) [8], [25]. These equations are repeated for each additional measurement available at a given time step \( k \). If no measurement is available at time step \( k \), then (22)–(24) are not used; instead, \( \hat{x}_k = \hat{x}_k^- \) and \( P_k = P_k^- \).

B. State–Space Model Formulation

The vehicle and sensor dynamics are formulated as a state–space model using the following state vector:

\[
x = [v_{bx}, \dot{b}_{ax}, a_{\text{dist},bz}, \omega_l, \omega_r, \dot{\psi}, b_{gyo}]^T
\]

where \( \omega_l \) and \( \omega_r \) are the angular velocities of the left and right front wheels and \( b_{ax} \) and \( b_{gyo} \) are the accelerometer \( x \)-axis and yaw gyro walking biases, respectively, which are part of the IMU error model suggested in [22].

Typical errors found in accelerometers and rate gyros of low-cost IMU’s are due to constant offsets \( c \), walking biases \( b \), and...
sensor noise $\nu$, such that [22]

$$z_{\text{meas}} = z_{\text{actual}} + c + b + \nu$$  

(29)

with

$$b = \frac{1}{\tau} b + \sqrt{\frac{2fs \sigma^2}{\tau}} w$$  

(30)

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

Using the previous state vector, the vehicle dynamics can be written as

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

(31)

where $w_i$ are zero-mean white noise and $f_{\text{controller}}(u)$ is the wheel acceleration that 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 $f_{\text{controller},l}(u) = -w_5$ and $f_{\text{controller},r}(u) = -w_6$.

In practice, the high frequency and accuracy of wheel velocity measurements commonly allow an accurate estimate of $\omega_l$ and $\omega_r$ without modeling $f_{\text{controller}}$. When estimating the vehicle dynamics, we therefore neglect $f_{\text{controller}}$, assuming that the desired velocity is approximately constant between sensor updates. This assumption has the effect of smoothing the wheel speed measurements (helpful for removing pulses that can occur in velocity measurements derived from encoder pulses). Discretizing the state equations and neglecting the zero-mean process noise $w_i$, the a priori estimate of the robot state at time step $k$ is

$$\hat{x}_k = \begin{bmatrix}
    \hat{v}_{bx,k-1} + f_{\text{tire}}(\hat{x}, \theta, \varphi) \Delta t + \hat{a}_{\text{dist},bx,k-1} \Delta t - g \sin(\varphi) \Delta t \\
    \hat{\dot{\omega}}_{ax,k-1} \left(1 - \frac{\Delta t}{\tau_{ax}}\right) \\
    \hat{\dot{\omega}}_{\text{dist},bx,k-1} \\
    \hat{\dot{\omega}}_{\text{dist},bl,k-1} \\
    \hat{\psi}_{k-1} + g_{\text{tire}}(\hat{x}, \theta, \varphi) \Delta t \\
    \hat{\dot{\psi}}_{g\psi,k-1} \left(1 - \frac{\Delta t}{\tau_{g\psi}}\right)
\end{bmatrix}.$$  

(32)

### C. Measurement Model

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

$$z = [x_{\text{IMU}}, \dot{\psi}_{\text{IMU}}, \dot{x}_{\text{GPS}}, \omega_{l,\text{enc}}, \omega_{r,\text{enc}}]^T$$

where $x_{\text{IMU}}$ and $\dot{\psi}_{\text{IMU}}$ are IMU measurements of $x$-axis acceleration and yaw rate, $\dot{x}_{\text{GPS}}$ is the component of the GPS velocity measurement along the body $x$-axis, and $\omega_{l,\text{enc}}$ and $\omega_{r,\text{enc}}$ are the left and right front wheel encoder angular velocity measurements. For the IMU measurements, the sensor model given by (29) is used. Note that for $x_{\text{IMU}}$, $z_{\text{actual}} = \hat{v}_{bx} + g \sin(\varphi)$, as the accelerometer measures gravity even if the robot is stopped. Simplifying, the measurement vector can be modeled as

$$z = h(x, \nu) = \begin{bmatrix}
    f_{\text{tire}}(x, \theta, \varphi) + c_{ax} + b_{ax} + a_{\text{dist},bx} \\
    \dot{\psi} + c_{g\psi} + b_{g\psi} \\
    \dot{v}_{bx} \\
    \omega_l \\
    \omega_r \\
    \nu_1 \\
    \nu_2 \\
    \nu_3 \\
    \nu_4 \\
    \nu_5
\end{bmatrix} + \begin{bmatrix}
    \nu_1 \\
    \nu_2 \\
    \nu_3 \\
    \nu_4 \\
    \nu_5
\end{bmatrix}.$$

(33)

where $c_{ax}$ and $c_{g\psi}$ are constant offsets of the $x$-axis accelerometer and yaw gyro, respectively, and $\nu_i$ are zero-mean white noise. To approximate the constant offsets, they are initialized to the average of the first $n$ IMU measurements, subtracting out the acceleration due to gravity from the measurement. When the robot is at rest, the constant offsets are updated with new measurements using the exponential moving average (EMA) [26]:

$$E\text{M}A_{\text{current}} = (\text{measurement}_{\text{current}} - E\text{M}A_{\text{prev}}) \left(\frac{2}{1 + p}\right) + E\text{M}A_{\text{prev}}$$  

(34)
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. The EMA is not guaranteed to converge to the true value of the constant offsets, but instead converges to a locally constant offset over a window determined by the forgetting factor \( \lambda \), which is sufficient in practice. The EMA is easily and recursively calculated making it suitable for online implementation.

The estimated measurement vector is

\[
\hat{z} = h(\hat{x}_k, 0) = \begin{bmatrix} f_{\text{tire}}(\hat{x}_k, \theta, \varphi) + c_{ax} + \hat{b}_{ax,k} + \hat{a}_{\text{dist}, bx,k} \\ \hat{\psi}_k + c_{gz} + \hat{b}_{gz,k} \\ \hat{\omega}_l,k \\ \hat{\omega}_r,k \end{bmatrix}.
\]  

(35)

D. Weak Constraints

The disturbance \( a_{\text{dist}, bx} \) and accelerometer walking bias \( b_{ax} \) have both been modeled as random walks. Practically, the only difference between these variables in the model are that \( a_{\text{dist}, bx} \) appears in the calculation of \( \dot{b}_{bx} \) while \( b_{ax} \) does not, and that \( a_{\text{dist}, bx} \) is assigned a larger covariance in the matrix \( Q \) 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 upon insight into the physical nature of the disturbance. These rules are implemented using weak constraints described in [27] and implemented in a vehicle model in [11]. Unlike \textit{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.

The linear weak constraints considered here are treated the same as physical measurements in the EKF framework. If some user-defined conditions are met (i.e., the physics-based rules), (22)–(24) are used to update the state vector and system covariance matrix with an associated noise covariance matrix \( R_{\text{K}, \sigma} \), for each weak constraint. This is in contrast to \textit{ad hoc} techniques that may not propagate state changes through the system covariance matrix. Each weak constraint also has an associated noise covariance matrix \( R_{\delta \sigma} \). If the constraint is precisely known, then the covariance is zero and the constraint is considered a \textit{strong} constraint. All of the constraints applied here have nonzero covariance.

The following pseudo code outlines the EKF update process including the weak constraints:

```plaintext
while (vehicle operational) {
  increment EKFTime by constant dt
  EKF time update (20), (21)
  if (IMU measurement available) {
    Do EKF measurement update (22)–(24) using: \( H_{\text{IMU}}, V_{\text{IMU}}, R_{\text{IMU}}, z_{\text{IMU}}, h_{\text{IMU}} \)
  } if (GPS measurement available) {
    Do EKF measurement update (22)–(24) using: \( H_{\text{GPS}}, V_{\text{GPS}}, R_{\text{GPS}}, z_{\text{GPS}}, h_{\text{GPS}} \)
  }
  if (encoder measurement available) {
    Do EKF measurement update (22)–(24) using: \( H_{\text{encoder}}, V_{\text{encoder}}, R_{\text{encoder}}, z_{\text{encoder}}, h_{\text{encoder}} \)
  }
  for \( i = 1 : (\text{number of weak constraints}) \) {
    if (Weak Constraint \( i \) condition satisfied) {
      Do EKF measurement update (22)–(24) using: \( H_{WCi}, V_{WCi}, R_{WCi}, z_{WCi}, h_{WCi} \)
    }
  }
  end while
```

Table I summarizes the weak constraints employed in this paper. The middle column presents the condition that must be met for the constraint to be applied and the third column gives the measurement innovation to be used in (23), which becomes \( \hat{x}_k = x_k + K_k (z_{WCi} - h_{WCi} (\hat{x}_k, 0)) \) for each weak constraint \( i \). Constraints 1–3 constrain the nature of the disturbance based on physical reasoning, to maintain observability of the state, similar to the implementation in [11]. Constraint 4 allows for calibration of the IMU biases when the robot is stopped. The forth column lists the \( R \) values used for each weak constraint, normalized by the average of the \( R \) values for the real measurements (a smaller value indicates higher weighting). In practice, as a precaution to prevent the filter from diverging during fault conditions such as malfunctioning sensors, additional constraints could be applied using this framework to limit the magnitude and rate of change of some of the states based on known physical characteristics of the robot (i.e., the robot may have a known top speed). For some of the conditions, the variable VelDir is used, defined as

\[
\text{VelDir} = \text{sign}(\omega_l + \omega_r)
\]  

(36)

such that VelDir equals 1 if the wheels are driving forward, 0 if the wheels are stopped, and \(-1\) if the wheels are driving in reverse. \( i_{\text{EMA}} \) is the EMA (34) of the average of the left and right front wheel slip.

E. Slip and Immobilization Detection

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 (9). 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 EMA (34) 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
### IV. EXPERIMENTAL RESULTS

#### A. Robot Description

An autonomous mobile robot developed for the DARPA LAGR program has been used to experimentally validate the algorithm (see Fig. 8). The robot is 1.2 m long \( \times \) 0.7 m wide \( \times \) 0.5 m and has the kinematic configuration discussed in Section II-A. The robot is equipped with 4096 count per revolution front wheel encoders, an Xsens MT9 IMU, a Garmin GPS 16 differential GPS, and two stereo pairs of video cameras (not used in this paper). The IMU provides the acceleration and angular rate measurements for the measurement update and a filtered estimate of \( \theta \) and \( \phi \) used directly in the normal force calculation. The robot has been used to collect data to process offline using a Matlab implementation of the slip detector, as well as to run an online C++ implementation on one of the robot’s 2.0 GHz Pentium M computers.

#### B. Determination of Model Parameters

The algorithm requires knowledge or estimates of a number of constant parameters. The robot mass and center of gravity location were directly measured. The measurement noise and walking bias process noise covariances were drawn from sensor data sheets and sensor measurements. The process noise and weak constraint covariances were initially set to values estimated using physical reasoning, before manually tuning the values to achieve improved filter performance.

A series of simple experiments were performed on multiple terrain types to estimate the tire parameter values. \( C_1 \) was estimated by measuring the force produced by spinning the robot’s wheels while it was restrained with a spring scale. \( C_2 \) was zero, as no nominal terrain-independent value was indicated by the test data. \( R_1 \) was estimated by measuring the force required to pull the robot forward with the wheels freely spinning. \( A_t \) and \( R_2 \) were chosen based upon tire force curves in the literature [10] and upon experimentation with the algorithm. \( A_{roll} \) was chosen to be large to approximate a static rolling resistance force. From these experiments, a set of nominal parameters was extracted that yielded good slip detection performance over many terrain types. Table II summarizes the tire constants used for this paper.

<table>
<thead>
<tr>
<th>Description</th>
<th>Condition</th>
<th>“Measurement” Innovation</th>
<th>Normalized ( R ) values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) The modeled disturbance force should only oppose motion.</td>
<td>if ((\text{sign}(\hat{\alpha}<em>{\text{dist},b,k}) = \text{VelDir})) AND ((\text{sign}(\hat{\alpha}</em>{\text{dist},b,k}) \neq 0))</td>
<td>((z_{k,wc} - h_{wc}(\hat{\xi}<em>r,0)) ) = ((0 - \hat{\alpha}</em>{\text{dist},b,k}))</td>
<td>[0.68]</td>
</tr>
<tr>
<td>2) The disturbance should act quickly. The disturbance should not gradually increase such that the EMA of the wheel slip slowly increases. Only applies when the average wheel slip is small. ( \text{min}_{t_0} ) ( \text{thresh}_t ), and ( a ) are user-defined constants</td>
<td>if ((0 &lt; \frac{\Delta t_{\text{EMA}}}{\Delta t} &lt; \text{min}<em>{t_0})) AND ((\frac{t</em>{\text{EMA}}}{t_{\text{thresh}}_t} &lt; \alpha ) ( &lt; 1)</td>
<td>((z_{k,wc} - h_{wc}(\hat{\xi}<em>r,0)) = (a \hat{\alpha}</em>{\text{dist},b,k} - \hat{\alpha}_{\text{dist},b,k}))</td>
<td>[68]</td>
</tr>
<tr>
<td>3) The disturbance can stop the robot, but should not pull the robot backwards. If the robot is moving backwards, then either it is sliding down a hill and the disturbance should be zero, or the estimated disturbance is too high and should be reduced.</td>
<td>if ((\text{sign}(\hat{\nu}_{\text{dist},b,k}) = -\text{VelDir}))</td>
<td>((z_{k,wc} - h_{wc}(\hat{\xi}<em>r,0)) = a(0 - \hat{\alpha}</em>{\text{dist},b,k}))</td>
<td>[0.14]</td>
</tr>
<tr>
<td>4) When robot is fully stopped, the disturbance force and walking biases should tend to zero for calibration of the IMU constant biases. ( t_{\text{stop}} ) is a constant. ( T ) is the length of time the condition has been met.</td>
<td>if ((\omega_y = 0) ) AND ((\omega_x = 0))</td>
<td>([0,0,0] + [-\hat{\beta}<em>{\text{dist},b,k} \hat{\alpha}</em>{\text{dist},b,k} \hat{\beta}_{\phi,b,k}] )</td>
<td>[0.07 0 0 1.4 0]</td>
</tr>
</tbody>
</table>
C. Algorithm Performance

The algorithm was applied to 21 experimental test runs. During these tests, the robot traveled approximately 120 m over a range of terrain types including loose mulch, loose gravel over hard dry soil, mud, and various grasses. The robot was driven at speeds ranging from 0.1 to 1 m/s. The test runs include 20 instances of the robot coming to a complete stop with the wheels still spinning, which were initiated by holding the robot back using a spring scale. The tests were performed on nominally level terrain with the robot commanded to drive in a straight line. Preliminary tests show equivalent results on nonlevel terrain and while the robot autonomously navigates arbitrary paths.

The slip detector correctly identified each of these 20 instances as immobilized with an average detection time of 0.4 s. All data with the robot driving freely or sitting at rest were correctly labeled as normal driving, with the exception of two false positives. In total, less than 0.2% of the data points were falsely labeled as immobilized.

Fig. 9 shows a plot of the robot driving unconstrained over grass at 1 m/s. In the top plot, it can be seen that the filter’s estimated robot velocity follows the measured wheel velocity. At \( t \sim 14 \) s, there is a spike in the calculated wheel velocity; however, the estimated robot velocity correctly smooths this quantization error. The second plot shows the estimated disturbance, which remains small while the robot is driving. Just after the robot stops, suspension displacement creates a small spike in the disturbance. The third plot shows the EMA of the wheel slip. While driving, the wheel slip is estimated at approximately 3%, which is physically reasonable. The increased slip while accelerating and braking is also expected. The detector correctly labeled the entire dataset as driving normally (i.e., “immobilized?” = 0). The fourth plot shows the error covariance from the Kalman filter \( P \) matrix for \( v_{bx} \) and \( a_{\text{dist},bx} \), normalized by the process noise covariance. In all results, the normalized error remains well bounded, below unity, indicating that the filter is consistent.

Fig. 10 shows a plot of the robot attempting to drive forward at 1 m/s on grass while restrained with a spring scale to produce 100% wheel slip. The velocity estimate shows that the robot accelerates against the spring, but quickly becomes immobilized. The disturbance estimate approaches a near-constant resistive value ranging from \(-2.8\) to \(-3.1 \, \text{m/s}^2\) while the robot is immobilized, before returning to zero when the wheels stop spinning. During this test, the spring scale measured a 325 N force holding the robot back. The equivalent body acceleration for the 117 kg robot is \(2.8 \, \text{m/s}^2\), which closely agrees with the estimated disturbance. The slip EMA quickly approaches 100% and the detector identifies the robot as immobilized at \( t = 1.85 \) s.

Note that the GPS velocity estimate for these two tests (conducted in open terrain with few trees or tall buildings) was very accurate, and thus, GPS-based slip detection is possible. However, GPS measurements were available at 1 Hz, slower than desired for detection. Additionally, GPS returns the average velocity over the previous time step, and thus, the measurement is truly accurate for 0.5 s prior to the reported measurement time (in Figs. 9 and 10, the GPS velocity plots are time shifted by 0.5 s to account for this). In the example shown in Fig. 10, immobilization could not be detected by GPS until \( t \sim 2.8 \) s, nearly 1 s slower than the proposed algorithm.

### Table II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Nominal Value Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_f )</td>
<td>0.52</td>
</tr>
<tr>
<td>( A_f )</td>
<td>20 , \text{s/m}</td>
</tr>
<tr>
<td>( C_s )</td>
<td>0</td>
</tr>
<tr>
<td>( R_{1,\text{front}} )</td>
<td>0.08</td>
</tr>
<tr>
<td>( R_{2,\text{front}} )</td>
<td>0.05</td>
</tr>
<tr>
<td>( A_{\text{roll}} )</td>
<td>50 , \text{s/m}</td>
</tr>
<tr>
<td>( R_{1,\text{rear}} )</td>
<td>0.0075</td>
</tr>
<tr>
<td>( R_{2,\text{rear}} )</td>
<td>0.02</td>
</tr>
</tbody>
</table>
D. Sensitivity to Tire Model Parameters

To study the algorithm’s sensitivity to tire model parameter values, the 21 experimental datasets were reprocessed, individually varying one of the five tire constants by ±20%. In all 210 tests, the algorithm correctly identified all 20 immobilizations. The number of false positives for each case is summarized in Table III. It was observed that the algorithm performance was most sensitive to changes in $C_1$. Increasing $C_1$ increases the maximum modeled traction, making the model less likely to estimate that traction has been lost and the wheels are slipping. Conversely, decreasing $C_1$ reduces the modeled available traction, increasing the likelihood of wheel slip in the model and causing an increase in the number of false immobilization detections. Even in the worst case, only 0.3% of the data points were falsely labeled immobilized.

Two additional cases were evaluated as a limited study of second-order sensitivities. The first should be the worst case for false positives, with both traction parameters −20% and all resistance parameters +20%. In this case, all immobilizations were detected and there were six false detections. The second should be the worst case for correct detections, with both traction parameters +20% and all resistance parameters −20%. In this case, there was only one false detection; however, one immobilization was not detected. In summary, the algorithm appears to be quite robust to errors in the estimated tire model parameters. It should be noted that the algorithm’s velocity estimate accuracy will depend on the accuracy of the tire model for the current terrain.

E. Algorithm Performance Without GPS

The 21 experimental datasets were reprocessed without including GPS velocity measurements (i.e., using wheel encoder and IMU measurements only). The algorithm again correctly identified all 20 immobilizations. Without GPS, the false immobilization detections increased from two to four (0.35% of all data points). Fig. 11 shows the results of processing the dataset of Fig. 10 without using GPS measurements. In this case, the two plots are nearly indistinguishable. These results suggest that the algorithm can be applied on systems lacking reliable GPS, such as mobile robots in urban surroundings, underwater, or where GPS is not available such as for Mars rovers. However, GPS can increase accuracy and improve performance when available.

V. CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

A dynamic model-based slip detector has been proposed that has proven effective at detecting robot immobilization over a variety of terrains. The detector utilizes a novel tire traction/braking model and weak constraints to estimate external forces acting on the robot. The algorithm can be applied to any vehicle with an IMU, wheel encoders, and (optionally) GPS. Sensitivity analysis has indicated that accurate immobilization detection is possible with relatively coarse engineering estimates of tire model parameters. The algorithm also yields qualitatively accurate estimates of the robot’s velocity and could potentially be implemented in a position estimation system that is robust to wheel slip. Future experiments should include ground-truth measurements to assess the velocity estimation accuracy. Inclusion of a lateral tire force model could potentially allow estimation of sideslip within the presented framework.

This paper focused on techniques for autonomously adapting the tire model parameters that would allow the algorithm to provide highly accurate velocity estimates as well as improve the slip detection time and reliability over variable terrain. This paper also explored fusing the output of multiple slip detection algorithms to increase detection speed and accuracy.

ACKNOWLEDGMENT

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

REFERENCES


Chris C. Ward received the B.S. degree in mechanical engineering from the University of California, Berkeley, in 2005, and the M.S. degree in mechanical engineering from Massachusetts Institute of Technology (MIT), Cambridge, in 2007. He was with the Field and Space Robotics Laboratory, MIT, where he was engaged in research on dynamic outdoor mobile robots. He is currently a Mechatronic Design Engineer at ASML Lithography Systems, Inc., Wilton, CT, where he is engaged in research on control systems for high-speed precision stages for semiconductor lithography systems.

Karl Iagnemma received the B.S. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 1994, and the M.S. and Ph.D. degrees in mechanical engineering from Massachusetts Institute of Technology (MIT), Cambridge, in 1997 and 2001, respectively. He is currently a Principal Research Scientist in the Department of Mechanical Engineering, MIT, where he was also a National Science Foundation Graduate Fellow. He has been a Visiting Researcher at the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory and the National Technical University of Athens (Greece). His current research interests include 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 the author of the monograph Mobile Robots in Rough Terrain: Estimation, Planning and Control With Application to Planetary Rovers (Springer, 2004). Dr. Iagnemma is currently an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS.