Continuous Vehicle Slip Model Identification on Changing Terrains

Forrest Rogers-Marcovitz, Neal Seegmiller, and Alonzo Kelly
Carnegie Mellon University, Robotics Institute
Pittsburgh, PA, USA
Email: {forrest, nseegmiller, alonzo}@cmu.edu

***Abstract*—Ground robotics rely on accurate dynamic models for high performance control and estimation systems. To use odometry or predict the robots motion, an accurate model is needed for the vehicle's slip. For mobile robots, the mapping between inputs and resultant behavior depends critically on terrain conditions which vary significantly over time and space so cannot be pre-programmed. *Integrated Perturbative Dynamics* is used to successively identify systematic and stochastic models of vehicle slip. This is a real-time algorithm which works over arbitrary trajectories, with fast convergence along terrain boundaries, which allows for reliable operation of ground vehicles over extended periods of time in changing environments. Results are shown on a tracked surveillance robot as it drives over four distinct terrain types over 35 minutes. Fast convergence of the slip mode parameters is observed when the robot crosses terrain boundaries.**

I. Introduction

Accurate dynamic models are essential for high performance control and estimation systems. For example, in autonomous vehicle applications, model predictive control can be implemented to perform path following, obstacle avoidance, and lane change maneuvers. In state estimation, the velocity kinematics of the platform are typically the basis for dead reckoning, or the system model in the estimator. In both control and estimation, the model is a differential equation containing unknown parameters, and the performance of the system depends critically on its accuracy.

In both predictive control and estimation contexts, it is equally important to understand the stochastic error that remains after the model is well calibrated. Such information can be a basis for assessing risk in cognition, for computing expectation in stochastic control, or for modeling system disturbances in optimal estimation. *Integrated perturbative dynamics* (IPD) has been used to model mobile robots and calibrate the model online using an extended Kalman filter. Past work has used IPD to identify systematic and stochastic models of wheel slip [13], powertrain dynamics, and odometry parameters [14].

A. Vehicle System Modeling

The aspects of wheel-terrain interaction that are needed for accurate ground robot models are neither well known nor easily measurable in realistic situations. Some have developed terramechanics-based models for slip estimation that require knowledge of tire constants and soil parameters [7] [15]. [3] constructed an artificial neural network that was trained offline and then used the network in a model predictive control context. These are offline techniques which cannot learn changes in the environment or vehicle.

Online published methods are mostly concerned with robust path-following (e.g. [11]), or the estimation of instantaneous wheel slip for feedback controllers. Some methods lump all unknown tire and soil parameters into slip ratios and angles; extended Kalman filters have been developed for real-time estimation of slip ratios and angles using velocity measurements [17] [10]. These online methods only care about the current instantaneous wheel slip and cannot be used for trajectory prediction.

Our approach relies on compensating a 3D wheel odometry solution for wheel slip. We deal with the case of arbitrary terrain. Our slip models try to account for the effects of all of the forces on the vehicle, rather than just those directly related to the controls. Notably, the composite effects of gravity and lateral acceleration are modeled to estimate instantaneous slip based on the control, or odometry, inputs and IMU-derived attitude.

The calibration of our slip model is based on the *Integrated Perturbative Dynamics* approach described in [13]. This approach uses an integral of the perturbative dynamics of 3D odometry in order to increase the sensitivity, and improve the conditioning, of the mapping between the history of wheel slip along an arbitrary trajectory and the observed pose error that it causes a few seconds later. This method captures the underlying disturbance dynamics as a function of all of input space and is calibrated online based on trajectories executed under normal operation.

B. Terrain Classification

A vehicle driving in real world off-road conditions will encounter a large variety of terrains. The online slip identification algorithm can handle mixed terrains in one of two ways. First, we can allow the model to continuously change across the multiple terrains. Since the natural environment does not normally have well defined boundaries between terrain types, this may allow for better seamless transitions and is not reliant on perception for terrain classification. This is the method we present in the paper and show, in the results section, that the slip model parameters can quickly converge after driving over new terrain

The second method for mixed terrains, uses perception for terrain classification. We will leave the evaluation between the two methods for future work but present some prior work which look at how perception can be used for terrain classification and self-supervised learning.

Offline leaning of terrain classification of hand-labeled terrain has combined vision- and laser-based classification [12]. Visual features include color and visual texture. Laser range features normally include statistics such as spread and standard deviation of ground height or surface normals. The different modes of sensing are often combined via Bayesian fusion or other meta-classifier fusion. Some terrains are better separated visually, e.g. hard surfaces with different slip, while others are better separated via laser range, e.g. asphalt and pavement. It should be noted that there are some terrains which can not be separated by either, such as snow that is hard or soft-packed. In these cases, it may be necessary to learn pre-immobilization detection using proprioception sensors that indirectly 'feel' the terrain underfoot [16].

Recent work has been done on *Self-Supervised Classification*, also called *Near-to-Far Learning* [5] [6] [9]. These approaches learn the visual appearance of terrain classes by relying on vibration-based sensing of wheel-terrain interaction to identify these terrain classes. The remote detection uses vision or laser features to separate terrain classes using either *Support Vector Machines* or *Mixtures of Gaussians*. The vibration features – which are sensed via accelerometers, gyros, or microphones – only give an idea of how rough the terrain is and not how the vehicle would slip on that terrain.

Self-Supervision allow the robot to classify terrains based on its needs and according to how they affect its behavior and not necessarily based on human-defined classes. Another essential point is that this classification is not fixed to a rigid number of classes but rather results from the robot's experience.

Much of the work on self-supervision for terrain classification was done by Anelia Angelova [2]. Her visual terrain classification used texture elements called "textons" which are clustered with the *k-means* algorithm. For each terrain, a slip behavior model was learned off-line which is dependent on the vehicle's roll and pitch but not velocity since the rover was driven at a constant speed. The slip behavior model used a type of receptive field regression algorithm that applies locally linear fits to the data to approximate a globally nonlinear function.

Angelova's latest published work focuses on supervised dimensionality reduction to recognize different terrains from color imagery in a fully automatic fashion, using the robot's mechanical sensors as supervision [1]. The intuition is that two visually similar terrains which are not normally discriminated in the visual space, and are mapped to the same cluster in the lower dimensional space, might be discriminated properly after introducing the supervision. It involves an *EM*-algorithm which clusters the inputs and selects the best features to represents each cluster.

II. Vehicle System Modeling

For a vehicle moving in contact with a surface, there are three degrees of freedom of motion as long as the vehicle remains in contact with the local tangent plane.

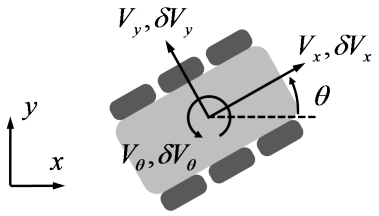

Fig. 1. Vehicle inputs and perturbations. Three degrees of freedom remain in the general case after terrain contact is enforced. Velocity inputs and disturbances are expressed in the body frame.

It is most natural to express actuation in the body frame. Given the vehicle's commanded linear and angular velocities, we have the following unconstrained kinematic differential equation for the time derivatives of 2D position and yaw with respect to a ground-fixed reference frame:

$$\dot{\underline{\rho}} = B(\gamma, \beta, \theta) \cdot \underline{u}_n \tag{1}$$

or, expressed in full:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} c\theta c\beta & c\theta s\beta s\gamma - s\theta c\gamma & 0 \\ s\theta c\beta & s\theta s\beta s\gamma + c\theta c\gamma & 0 \\ 0 & 0 & \frac{c\gamma}{c\beta} \end{bmatrix} \begin{bmatrix} V_{n,x} \\ V_{n,y} \\ V_{n,\theta} \end{bmatrix} \tag{2}$$

$$c = \cos(), \quad s = \sin(), \quad \gamma = roll, \quad \beta = pitch, \quad \theta = yaw$$

where the nominal velocity, $\underline{u}_n$, includes the forward velocity, $V_{n,x}$, the lateral velocity in the body-left direction, $V_{n,y}$, and the yaw rate with positive rotation in the counter-clockwise direction,$V_{n,\theta}$ - see Figure 1. This system is nonlinear because the heading angle appears in the coefficient matrix. It should be noted that we chose a velocity-driven model instead of a comprehensive physics-based model for simplicity and vehicle generality.

A. Linearized Error Dynamics

We will briefly cover the mathematics of linearizing the error dynamics of the systematic vehicle model - see [13] for a full development and explanation of the equations. Pose error of a path segment is attributed to the initial pose measurement error and input velocity perturbations due to slip, $\underline{u}_s$. Including these velocity perturbations, the kinematic differential equation becomes:

$$\dot{\underline{\rho}} = B(\gamma, \beta, \theta) \cdot (\underline{u}_n + \underline{u}_s), \quad \underline{u}_s = \begin{bmatrix} V_{s,x} \\ V_{s,y} \\ V_{s,\theta} \end{bmatrix} \tag{3}$$

In general form, the kinematic differential equation (2) can be written:

$$\dot{\underline{\rho}} = \underline{f}(\underline{\rho}(t), \underline{u}_n(t) + \underline{u}_s(t)) \tag{4}$$

Recall that the state ($\underline{\rho}$) is the pose of the vehicle in the ground-fixed reference frame and the inputs ($\underline{u}$) are the linear and

angular velocities in the body frame. By differentiating (4), we obtain the linearized error dynamics for deterministic or *systematic* error:

$$\delta\dot{\underline{\rho}} = F(t)\delta\underline{\rho}(t) + G(t)\underline{u}_s(t) \tag{5}$$

where F and G are Jacobian matrices:

$$F = \frac{\partial \underline{f}}{\partial \underline{\rho}} = \begin{bmatrix} 0 & 0 & -\dot{y} \\ 0 & 0 & \dot{x} \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$G = \frac{\partial \underline{f}}{\partial \underline{u}_s} = B(\gamma, \beta, \theta) \tag{7}$$

This allows us to define the *transition matrix*, Φ, and the *input transition matrix*, Γ:

$$\Phi(t,\tau) = e^{\int_\tau^t F(\zeta)\, d\zeta} = \begin{bmatrix} 1 & 0 & -(y(t) - y(\tau)) \\ 0 & 1 & (x(t) - x(\tau)) \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$\Gamma(t,\tau) = \Phi(t,\tau)B(\gamma,\beta,\theta) \tag{9}$$

Using the transition matrices, the solution to the first-order differential equation (5) is the following vector convolution integral:

$$\delta\underline{\rho}(t) = \Phi(t,t_0)\delta\underline{\rho}(t_0) + \int_{t_0}^{t} \Gamma(t,\tau)\underline{u}_s(\tau)d\tau \tag{10}$$

At the end of each path segment, the difference between current state at the last key frame is used for slip model identification. To minimize uncertainty, the two key frames should occur right after GPS, or other position measurements are made. Starting at the last key frame, we propagate the pose forward using the past odometry measurements, along with measured roll and pitch history, and the current perturbation model coefficients, $\underline{\alpha}$.

$$\underline{\rho}_{k,ipd} = \int_{t_{k-1}}^{t_k} B(\gamma(\tau),\beta(\tau),\theta(\tau)) \cdot \left(\underline{u}_n(\tau) + \underline{u}_s(\tau,\underline{\alpha})\right) d\tau \tag{11}$$

The Kalman filter measurement is the difference between final measured and final predicted poses (x, y, θ):

$$\underline{z} = \underline{\rho}_{k-1,ins}^{-1} \otimes \underline{\rho}_{k,ins} \tag{12}$$

$$h(\underline{x}) = \underline{\rho}_{k,ipd} + \underline{n}_{ipd} \tag{13}$$

Here, $\otimes$ refers to the pose addition found by multiplying the corresponding Homogeneous Transforms. The measurement residual is assumed to have a zero-mean Gaussian norm distribution $\underline{n}_{ipd} \sim \mathcal{N}(0, R_{sys})$.

The velocity perturbations have both a systematic and stochastic component. They are drawn from a distribution:

$$\delta\underline{u}(\delta\underline{\alpha},\tau) \sim \mathcal{N}\left(\underline{\mu}_{\delta u}(\delta\underline{\alpha}),\ Q\right) \tag{14}$$

The systematic component of $\delta\underline{u}$ (i.e. the mean of the distribution from which it is drawn, $\underline{\mu}_{\delta u}$) is attributed solely to errors in the parameter estimates, $\delta\underline{\alpha}$. The stochastic component is characterized by the covariance matrix Q which we will address in section II-C.

Because both state measurements and the input perturbations have a random component, our observations of the innovation has additional uncertainty. This measurement uncertainty is computed using the matrix convolution integral:

$$R_{sys} = \Phi(t,t_0)\Sigma_{x,meas}(t_0)\Phi(t,t_0)^T + \int_{t_0}^{t} \Gamma(t,\tau)Q(\tau)\Gamma(t,\tau)^T d\tau + \Sigma_{x,meas}(t) \tag{15}$$

B. Parameterization

Of course, wheel slip is not constant but depends on the commanded trajectory and terrain geometry. Accordingly, we parameterize the systematic component of $\underline{u}_s$ over velocities and accelerations predicted from the encoder measurements, and components of the gravity vector, which are available from the pose filter:

$$\underline{u}_s = \begin{bmatrix} V_{s,x} \\ V_{s,y} \\ V_{s,\theta} \end{bmatrix} = C\underline{\alpha} \tag{16}$$

$$C = \begin{bmatrix} \underline{c}_x & & \\ & \underline{c}_y & \\ & & \underline{c}_\theta \end{bmatrix} \tag{17}$$

$$\underline{c}_x = \begin{bmatrix} V_{n,x} & |V_{n,\theta}| & (V_{n,x}|V_{n,\theta}|) & g_x \end{bmatrix}$$

$$\underline{c}_y = \begin{bmatrix} V_{n,x} & V_{n,\theta} & (V_{n,x}V_{n,\theta}) & g_y \end{bmatrix}$$

$$\underline{c}_\theta = \begin{bmatrix} V_{n.x} & V_{n,\theta} & (V_{n,x}V_{n,\theta}) & g_x & g_y \end{bmatrix}$$

C is a 3×13 matrix in which all off-diagonal element blocks are zero. The use of absolute values in the parameterization of $V_{s,x}$ makes forward slip an even function of the measured angular velocity (which we observed experimentally). The slip rate parameter vector, $\underline{\alpha}$, includes the coefficients that are learned by the online filter.

This parameterization works well in practice but also makes intuitive sense. Wheel slip is fundamentally caused by forces acting on the vehicle. The velocity terms $V_{n,x}$ and $V_{n,\theta}$ are included because frictional contact forces are proportional to them. Centripetal acceleration $(V_{n,x}V_{n,\theta})$ and the components of the gravity vector $(g_x,\ g_y)$ represent the net applied non-contact forces.

Including forward acceleration $(\dot{V}_{n,x})$ in the parameterization of forward slip is intuitively reasonable; however, the results on our datasets were not affected by when included.

C. Stochastic System Modeling

Ideally, our predictions of state would be perfect after solving for the optimal parameter values $\underline{p}$. In reality this is never possible due to our use of simplified models (for fast computation) and random disturbances that can not be predicted. A model of stochastic dynamics enables us to, at least, bound the error on our predictions.

One standard technique to estimate uncertainty for a stochastic process is to repeat the process numerous times, then compute a sample covariance from the outcomes. However, the integrated error dynamics in (15) contains the trajectory dependent matrix $\Gamma(t,\tau)$. When calibrating online, we

don't have the luxury of controlling the reference trajectory such that the same trajectory is repeated multiple times. Furthermore, even when we can repeat the trajectory, it can only be repeated imperfectly.

Predicted variance is computed using the matrix convolution integral (15) and our current estimate of the stochastic parameters $\underline{q}$.

$$\underline{h}(\underline{q}) = R_{sys} \tag{18}$$

We must accept at the outset that online stochastic calibration requires some assumptions. Note first that the initial residuals computed during on-line calibration will contain both systematic and random error. In order to correctly calibrate variance we need to know the mean of the distribution, which requires accurate estimates of the systematic parameters. We therefore must accept that the Q matrix that we are calibrating is not characterizing only random error until the systematic calibration has converged on the mean. Of course, the mean itself is not necessarily constant either. We therefore, assume that the systematic parameters are changing relatively slowly compared to the time period required to estimate variance. Finally, the true process of uncertainty propagation for the system may not be linear; however, a linear approximation can be evaluated efficiently, and may be the only practical option online.

Let us define $\underline{r}(\underline{\rho})$ as the observed residual from the systematic calibration:

$$\underline{r}(\underline{\rho}) = \underline{\rho}(t)_{meas} - \underline{\rho}(t)_{pred} \tag{19}$$

We choose to formulate our observations of the variance Q as a scatter matrix of these residuals. Certainly these scatter matrices depend on Q and on the trajectory. If trajectories are not repeated during online calibration, we must somehow predict the scatter matrix observations for a set of trajectories before presenting the observation to the calibration system. That problem is tractable using the total probability theorem as presented in [8]. For now, we will formulate our measurement as a scatter matrix computed from a single residual; this is simply the outer product:

$$\underline{z}_{stoch} = \underline{r}(\underline{\rho})\underline{r}(\underline{\rho})^T \tag{20}$$

Even though these observations are noisy (and each corresponds to a unique trajectory), every observation improves our estimate of the stochastic parameters because we are calibrating trajectory-independent parameters of a "generative" model (i.e. the Q matrix).

1) Jacobian of the Stochastic Prediction: Here we compute the Jacobian of the stochastic prediction with respect to the stochastic parameters. In the simplest case, the Q matrix is constant (meaning not state or input dependent) and symmetric. The stochastic parameters $\underline{q}$ are simply the six independent elements of Q. In this case, the Jacobian with respect to the element q_{ij} is:

$$J_{q_{ij}} = \int_{t_0}^{t} \underline{\gamma}_{*i}(t,\tau)\underline{\gamma}_{*j}(t,\tau)^T \, d\tau \tag{21}$$

where $\underline{\gamma}_{*i}(t,\tau)$ is the i^{th} column of $\Gamma(t,\tau)$.

Note that the prediction and observation in the stochastic dynamics are both matrices. To avoid tensors when computing the Jacobian, it is convenient to reshape the unique elements of the prediction and observation matrices into vectors.

2) Stochastic Measurement Uncertainty: Next we derive the measurement uncertainty R_{stoch}. In practice, we found that correct calculation of the measurement uncertainty is critical to performance.

Recall that the stochastic measurement $\underline{z}_{stoch}$ is just the outer product of the residual $\underline{r}(\underline{\rho}) = \underline{\rho}(t)_{meas} - \underline{\rho}(t)_{pred}$, reshaped into a vector. Assuming the mean is known, $\underline{z}_{stoch}$ can be viewed as a sample covariance for a single sample, in which case, the measurement uncertainty R_{stoch} would be the *covariance of a sample covariance*.

To clarify, let us express each term in full. Assume, for example that the residual contains three elements: x, y, and θ.

$$\underline{r}(\underline{\rho}) = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \tag{22}$$

$$R_{sys} = \begin{bmatrix} V(x) & C(x,y) & C(x,\theta) \\ & V(y) & C(y,\theta) \\ & & V(\theta) \end{bmatrix} \tag{23}$$

In (23), V and C are abbreviations for variance and covariance respectively. Because $\underline{z}_{stoch}$ is a function of a random vector, it too is a random vector with its covariance R_{stoch}:

$$\underline{z}_{stoch} = \begin{bmatrix} xx & xy & x\theta & yy & y\theta & \theta\theta \end{bmatrix}^T \tag{24}$$

$$R_{stoch} = \begin{bmatrix} V(xx) & C(xx,xy) & C(xx,x\theta) & \dots \\ & V(xy) & C(xy,x\theta) & \dots \\ & & V(x\theta) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{25}$$

Each element of $\underline{z}_{stoch}$ is a product of random variables. The variance and covariance of products of random variables (i.e. the elements of R_{stoch}) can be calculated using the following two rules (from [4]):

$$V(ab) = V(a)V(b) + C^2(a,b) \tag{26}$$

$$C(ab,cd) = C(a,c)C(b,d) + C(a,d)C(b,c) \tag{27}$$

The rules (26) and (27) assume all random variables (a,b,c,d) have an expected value of zero. Assuming that the mean residual is zero is reasonable once the systematic parameters $\underline{p}$ are correctly calibrated. The variance and covariance terms required by the rules are obtained from R_{sys}.

D. Extended Kalman Filter

Kalman filters are an excellent means of online calibration using the IPD formulation. They make utilizing the measurement uncertainties, derived above, straightforward. Furthermore, the Q matrix in Kalman filters provides explicit control over the relative weight of history and the present measurement. The state $\underline{\chi}$ is simply the vector of parameters to be identified ($\underline{\alpha}$ or $\underline{q}$). The process update adds uncertainty

without changing the parameter estimates:

$$\underline{\mathcal{X}}_{k|k-1} = \underline{\mathcal{X}}_{k-1|k-1} \tag{28}$$

$$\mathcal{P}_{k|k-1} = \mathcal{P}_{k-1|k-1} + \mathcal{Q}_k \tag{29}$$

The measurement update equations in an extended Kalman filter are:

$$\underline{r}_k = \underline{z}_k - \underline{h}(\underline{\mathcal{X}}_{k|k-1}) \tag{30}$$

$$S_k = H_k \mathcal{P}_{k|k-1} H_k^T + R_k \tag{31}$$

$$K_k = \mathcal{P}_{k|k-1} H_k^T S_k^{-1} \tag{32}$$

$$\underline{\mathcal{X}}_{k|k} = \underline{\mathcal{X}}_{k|k-1} + K_k \underline{r}_k \tag{33}$$

$$\mathcal{P}_{k|k} = (I - K_k H_k)\mathcal{P}_{k|k-1} \tag{34}$$

where S is the covariance of the innovation $\underline{r}$, K is the Kalman gain, and $\mathcal{P}$ is the state estimate covariance.

Note that $\mathcal{X}$, $\mathcal{P}$, and $\mathcal{Q}$ in script font refer to variables in a calibration Kalman filter, which are not to be confused with the analogous variables in the system model such as the vehicle pose.

III. Experimental Validation

Tracked vehicles are possibly the worst case for slip since they must slip by design during any turn. Data was collected on a custom surveillance robot similiar to the Foster-Miller TALON. Even though the IPD system identification algorithm was originally developed for skid-steered and Ackermann vehicles, the vehicle motion model is general enough to use directly on a tracked robot with no changes. The robot was instrumented with encoders on each track, GPS, and an Honeywell HG1930 IMU including accelerometers and gyros. The GPS data was post-processed with a nearby base station for high accuracy positioning. The presented techniques should work for other pose measurements system which provide good accuracy for short periods such as *visual odometry* or high performance *inertial navigation*. The robot was driven on soft dirt, asphalt, loose rocks, and grass at relatively high speeds, for the robot's size, of up to 1.7 m/s and 3 rad/s. The identification algorithm ran continuously across all the terrains for a total time of 35 minutes.

Scatter plots of pose prediction errors (i.e. the residuals $\rho(t)_{meas} - \rho(t)_{pred}$) for the test are shown in Figure 2. At each timestep, the pose of the vehicle is predicted two seconds in the *future* using the latest estimate of the slip parameters. Two seconds later, the final pose is measured and parameter estimates are updated using the pose residual. This process is repeated for overlapping path segments. Each dot in Figure 2 represents the along-track and cross-track error of a single pose prediction; all data is processed but (for legibility) only 1000 data points are plotted. These data points are equally spaced in time and span the entire 35 minute experiment.

In Figure 2(a), pose predictions are made assuming no slip; i.e. $\underline{u}_s = 0$ and so $\underline{u} = \underline{u}_n$. This is the default skid-steer motion model commonly used when nothing is known about wheel slip. These kinematic predictions are quite poor under conditions of steep slopes, high speeds, and persistent understeer.

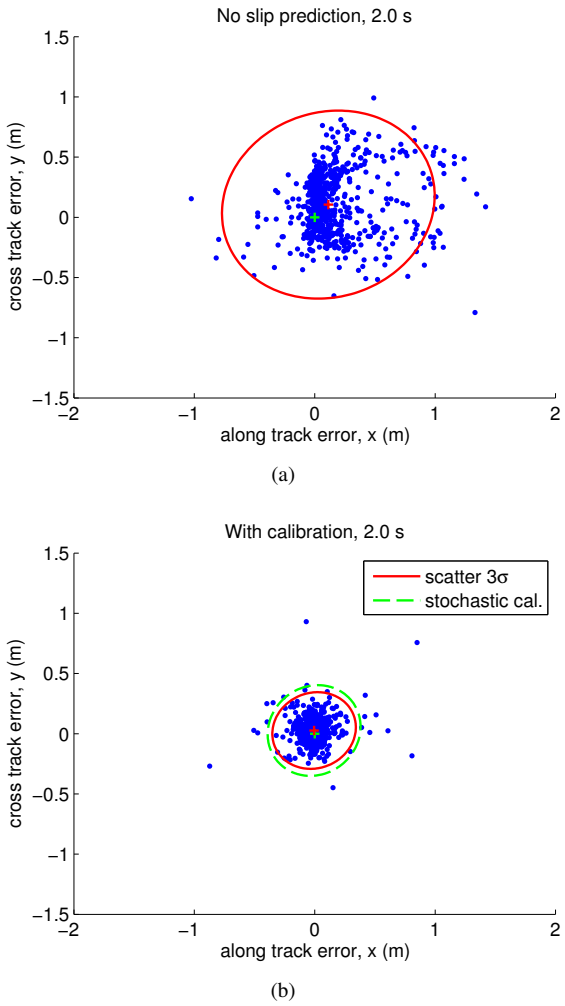

Fig. 2. Scatter plots of along track and cross track error for the robot on various surfaces. Each point represents predicted pose error at the end of a two-second path segment. Figure (a) figure shows predicted pose error assuming no slip; the figure (b) shows *online* prediction error during calibration. The three standard deviation error ellipse of the points is also shown.

Figure 2(b) shows prediction error *during online calibration* with an initial estimate of zero for all slip parameters.

Pose predictions during online calibration are significantly more accurate than no-slip predictions. Note that the mean error is reduced from 26.7 cm to 8.5 cm. Furthermore, the standard deviation of along track error and cross track error are both reduced by 60% respectively. The mean of predicted heading error is reduced by 83% with a standard deviation reduction of 71%.

The error that is not removed by the systematic calibration is accurately characterized by the calibrated stochastic model. Each dot represents a path segment of unique shape for which a unique pose error covariance $P'(t)$ is predicted by the calibrated stochastic model according to (15). As described in [8], the average of these predicted covariances (denoted by the dashed ellipse) is precisely the calibrated estimate of the sample covariance of the *set* of trajectories. The sample

covariance (denoted by the solid ellipse) is computed offline for comparison.

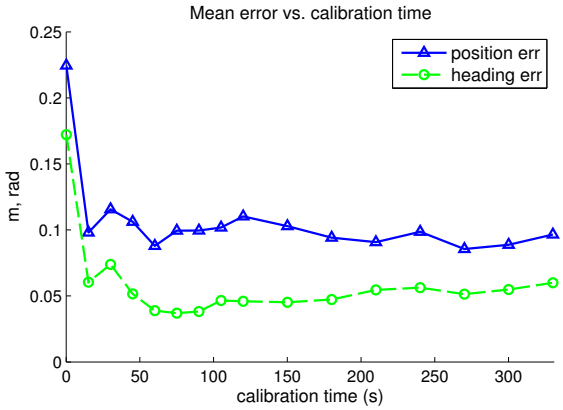

Fig. 3. Plots of the mean pose prediction error vs. calibration time as the robot drives over grass. The model is calibrated for 0-330 seconds of driving then evaluated on holdout data (the remainder of the 10.5 minute grass data). The solid (blue) line denotes the mean Euclidean distance error of the predicted (x,y) position for a 2 second path segment. The dashed (green) line denotes the mean of the absolute value of predicted heading error.

The time to converge depends primarily on the initial parameter estimates and the diversity of input trajectories. Figure 3 shows the pose prediction performance on holdout grass data after calibrating for limited periods ranging from 0 to 330 seconds. Based on pose prediction error, the filter nearly converges within seconds of driving when starting from the uncalibrated case (i.e. all slip parameters initialized to zero) given a sufficient set of trajectories. In practice, one could use the parameters calibrated for the same vehicle on different terrain as a better initial guess, thereby decreasing the convergence time.

IV. Conclusion and Future Work

The tracked surveillance robot drove over four distinct terrains while continuously updating it's systematic and stochastic slip model of the robots motion. The robot had no perception to perform terrain classification. Instead, we relied on the fast convergence of the slip parameters when the robot crossed a terrain boundary. Future work will look at how perception, with self-supervised learning, can classify terrain to predict the robot's motion across terrain boundaries.

The calibration system runs continuously during vehicle operation, as opposed to learning the parameters suitable for various terrain during a special calibration run. It should be emphasized that the presented method is a real-time algorithm which works over arbitrary trajectories which allows for continuously adjusting of the robot slip model over changes in both the terrain and the robot's handling for extended periods of time.

Acknowledgments

This work was supported by the U. S. Army Research Office under contract Real-Time Identification of Wheel Terrain Interaction Models for Enhanced Autonomous Vehicle Mobility (contract number W911NF-09-1-0557)

References

[1] Anelia Angelova, Larry Matthies, Daniel Helmick, and Pietro Perona. Dimensionality reduction using automatic supervision for vision-based terrain learning. In *Robotics: Science and Systems Conference*, 2007.

[2] Anelia Angelova, Larry Matthies, Daniel Helmick, and Pietro Perona. Learning and prediction of slip from visual information. *Journal of Field Robotics*, 24(3):205–231, 2007.

[3] Michael Bode. Learning the forward predictive model for an off-road skid-steer vehicle. Technical Report CMU-RI-TR-07-32, Robotics Institute, Pittsburgh, PA, September 2007.

[4] G. Bohrnstedt and A. Goldberger. On the exact covariance of products of random variables. *Journal of the American Statistical Association*, 64(328):1439–1442, December 2005.

[5] C.A. Brooks and K.D. Iagnemma. Self-supervised classification for planetary rover terrain sensing. In *Aerospace Conference, 2007 IEEE*, March 2007. doi: 10.1109/AERO.2007.352693.

[6] I. Halatci, C.A. Brooks, and K. Iagnemma. Terrain classification and classifier fusion for planetary exploration rovers. In *Aerospace Conference, 2007 IEEE*, March 2007. doi: 10.1109/AERO.2007.352692.

[7] Genya Ishigami, Akiko Miwa, Keiji Nagatani, and Kazuya Yoshida. Terramechanics-based model for steering maneuver of planetary exploration rovers on loose soil. *J. Field Robotics*, 24(3):233–250, 2007.

[8] Alonzo Kelly. Fast and easy systematic and stochastic odometry calibration. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, 2004.

[9] Ambroise Krebs, Cédric Pradalier, and Roland Siegwart. Adaptive rover behavior based on online empirical evaluation: Roverterrain interaction and near-to-far learning. *Journal of Field Robotics*, 27(2):158–180, 2010. ISSN 1556-4967. doi: 10.1002/rob.20332.

[10] Chang Boon Low and Danwei Wang. Integrated estimation for wheeled mobile robot posture, velocities, and wheel skidding perturbations. In *Proc. IEEE Intertnational Conference on Robotics and Automation*, pages 2355–2360, 2007.

[11] E. Lucet, C. Grand, D. Sallé, and P. Bidaud. Dynamic sliding mode control of a four-wheel skid-steering vehicle in presence of sliding. In *Proc. RoManSy*, Tokyo, Japan, July 2008.

[12] Wei Mou and A. Kleiner. Online learning terrain classification for adaptive velocity control. In *Safety Security and Rescue Robotics (SSRR), 2010 IEEE International Workshop on*, pages 1 –7, july 2010. doi: 10.1109/SSRR.2010.5981563.

[13] N. Seegmiller, F. Rogers-Marcovitz, G. Miller, and A. Kelly. A unified perturbative dynamics approach to vehicle model identification. In *Proc. International Symposium on Robotics Research*, Aug 2011.

[14] N. Seegmiller, F. Rogers-Marcovitz, and A. Kelly. Online calibration of vehicle powertrain and pose estimation parameters using integrated perturbative dynamics. In *Proc. International Conference on Robotics and Automation*, May 2012.

[15] Lakmal Seneviratne, Yahya Zweiri, Suksun Hutangkabodee, Zibin Song, Xiaojing Song, Savan Chhaniyara, Said Al-Milli, and Kaspar Althoefer. The modeling and estimation of driving forces for unmanned ground vehicles in outdoor terrain. *International Journal of Modeling, Identification, and Control*, 6(1): 40–50, 2009.

[16] Eric Trautmann and Laura E. Ray. Mobility characterization for autonomous mobile robots using machine learning. *Auton. Robots*, 30(4):369–383, 2011.

[17] Jingang Yi, Hongpeng Wang, Junjie Zhang, Dezhen Song, Suhada Jayasuriya, and Jingtai Liu. Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation. *IEEE Transactions on Robotics*, 25:1087–1097, October 2009.