

# Online Calibration of Vehicle Powertrain and Pose Estimation Parameters using Integrated Dynamics

Neal Seegmiller, Forrest Rogers-Marcovitz, and Alonzo Kelly

**Abstract**—This paper presents an online approach to calibrating vehicle model parameters that uses the *integrated dynamics* of the system. Specifically, we describe the identification of the time constant and delay in a first-order model of the vehicle powertrain, as well as parameters required for pose estimation (including position offsets for the inertial measurement unit, steer angle sensor parameters, and wheel radius). Our approach does not require differentiation of state measurements like classical techniques; making it ideal when only low-frequency measurements are available. Experimental results on the LandTamer and Zoë rover platforms show online calibration using integrated dynamics to be fast and more accurate than both manual and classical calibration methods.

## I. INTRODUCTION

Accurate calibration is essential for high-performance control and position estimation on autonomous vehicles [1].

The required calibration includes a model of the powertrain dynamics. For a wheeled mobile robot (WMR), the powertrain is the group of components that generate power and transmit it to wheels. No powertrain is capable of instantaneously changing wheel speeds; often delays and transients in the powertrain response are significant and must be accounted for using model predictive control.

The calibration also includes intrinsic parameters required for position estimation. If the position estimation system includes an inertial measurement unit (IMU), the pose of the IMU relative to the vehicle body frame must be known. Wheel odometry requires accurate dimensional values such as wheelbase and wheel radius, as well as calibrated steer angle sensors for articulated wheels.

Unfortunately, manual calibration of all these parameters can be time-consuming and inaccurate. For example, the IMU is often mounted deep in the robot chassis, making position offsets difficult to measure. Nominal CAD dimensions often can not be trusted. Furthermore, calibration must be repeated as hardware degrades or shifts with use in the field. Ideally, vehicle model calibration should be automated, online, and accurate.

### A. Related Work

Many WMR controllers ignore powertrain dynamics and rely on feedback to correct errors that could be prevented with better predictive models. Others operate at low or constant speeds where the effects of powertrain transients

are negligible. Yu et al. developed a dynamic skid-steered vehicle model that accounts for motor saturation and power limitations, but do not attempt automated calibration [2]. Others have devised methods of identifying time delays in the context of sensor calibration, e.g. for LIDAR [3] and for IMU/camera systems via a registration technique [4].

Outside of robotics literature, there are methods of calibrating induction motor parameters [5] and engine models too complex for our needs. However, we are aware of no unified method of simultaneous online calibration of a time constant and delay for a WMR powertrain, much less with the robustness to noise of our presented approach.

Recent work on calibrating IMU position has focused on the relative pose between the IMU and other sensors [6] [7], not between the IMU and vehicle body frame. Some high-end GPS/INS systems such as the Novatel SPAN have proprietary calibration routines for the IMU to GPS antenna offset and angular offsets with respect to the vehicle, but these require high-frequency differential GPS updates<sup>1</sup>.

An early example of automated odometry calibration is the “UMBmark” test in which wheelbase and wheel diameter are identified as the WMR traverses a preprogrammed square trajectory [8]. Roy and Thrun presented an online, probabilistic method of calibrating odometry, but it requires frequent sensor scans and learns simplified translational and rotational error parameters [9]. Antonelli et al. present a calibration process similar to ours in that predicted trajectories are integrated, but theirs is an offline process that learns a redundant set of parameters [10]. We are aware of no method that simultaneously calibrates both IMU offsets and odometry parameters online using temporally sparse pose data.

### B. Integrated (Perturbative) Dynamics

This paper presents an *integrated equation error* approach to calibrating intrinsic vehicle model parameters that uses the *integrated dynamics* (ID) of the system. System models are commonly known in the form of a differential equation:

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{p}) \quad (1)$$

where  $\underline{x}$  is the state vector,  $\underline{u}$  is the input vector, and  $\underline{p}$  is the vector of parameters to be identified. The classical model identification approach uses the differential equation directly, which requires observations of  $\dot{\underline{x}}$  and  $\underline{x}$ . Often  $\dot{\underline{x}}$  can not be measured directly, so measurements of  $\underline{x}$  are

Manuscript received September 16, 2011.

N. Seegmiller is a Ph.D. candidate at the Carnegie Mellon University Robotics Institute, F. Rogers-Marcovitz is a robotics engineer at the National Robotics Engineering Center, and A. Kelly is a professor at the Robotics Institute {nseegmiller, forrest, alonzo}@cmu.edu

<sup>1</sup>SPAN-SE User Manual, <http://www.novatel.com/assets/Documents/Manuals/om-20000124.pdf>

numerically differentiated with respect to time. For example, the *Springer Handbook of Robotics* teaches the estimation of manipulator inertial parameters using this approach, which requires double differentiation of joint angle data [11].

In contrast, we identify  $\underline{p}$  using the integrated dynamics:

$$\underline{x}(t) = \underline{x}(t_0) + \int_{t_0}^t \underline{f}(\underline{x}(\tau), \underline{u}(\tau), \underline{p}) d\tau \quad (2)$$

In effect, we integrate the prediction rather than differentiate the measurement, which has several advantages. This approach avoids numerical derivatives which can be noisy and require accurate timestamps and high-frequency sensors. In addition, integrated predictions account for delayed effects of the parameters. For example, angular velocity error has no *instantaneous* effect on translational velocity, but does have an increasing effect on translational position error with distance traveled.

The cost of using integrated dynamics is added complexity in deriving the Jacobian (or gradient) required for parameter estimation.  $\frac{\partial \underline{x}(t)}{\partial \underline{p}}$  is harder to compute than  $\frac{\partial \dot{\underline{x}}}{\partial \underline{p}}$  because it requires at least a numerical solution to the (often non-linear) differential equation. To facilitate efficient computation of the Jacobian in our nonlinear WMR application we derive the *integrated perturbative dynamics* (IPD) of the system. These are obtained by solving the linearized differential equation:

$$\delta \dot{\underline{x}} = F \delta \underline{x} + G \delta \underline{u}, \quad F = \frac{\partial \underline{f}}{\partial \underline{x}}, \quad G = \frac{\partial \underline{f}}{\partial \underline{u}} \quad (3)$$

$$\delta \underline{x}(t) = \Phi(t, t_0) \delta \underline{x}(t_0) + \int_{t_0}^t \Phi(t, \tau) G(\tau) \delta \underline{u}(\tau, \Delta \underline{p}) d\tau \quad (4)$$

(3) and (4) are valid linear approximations of the system for small perturbations  $\delta \underline{x}$  and  $\delta \underline{u}$ . Here, we chose to limit the parameterization to the input vector  $\underline{u}$ ; perturbations  $\delta \underline{u}$  are attributed to errors in the parameter estimates  $\Delta \underline{p}$ . The transition matrix  $\Phi$  is explained in Section III-C. Previously, the authors demonstrated the use of IPD for the identification of slip parameters and stochastic dynamics [12]; here IPD is used to identify *intrinsic* odometry parameters.

For brevity, we will refer to our integrated equation error approach as the “ID/IPD approach” and the classical approach (which uses the differential equation) as the “DE approach.” In the following sections we explain how ID/IPD can be used with an extended Kalman filter (EKF) to estimate vehicle parameters online. Section II explains the identification of a time constant and delay in a powertrain model, and Section III explains the calibration of IMU position offsets, steer angle sensors, and wheel radius. Experimental results show improved convenience and accuracy over manual calibration. Note that no special trajectories are required; calibration can be performed during normal operation. Because identification is performed online, the calibration should automatically adjust to changing hardware conditions (e.g. a low battery or a flat tire).

## II. CALIBRATION OF THE VEHICLE POWERTRAIN

WMR powertrains can not change wheel speeds instantaneously; doing so would require infinite torque. The dynam-

ics of many vehicle powertrains can be modeled by a time delay and a first-order transient response:

$$\dot{\omega} = \frac{1}{\tau_c} (\tilde{\omega}(t - \tau_d) - \omega(t)) \quad (5)$$

where  $\omega$  denotes the angular velocity of the wheel. We present a model for an individual wheel, but models may also be learned for sets of wheels.  $\tilde{\omega}$  denotes the commanded velocity,  $\tau_c$  the time constant, and  $\tau_d$  the time delay. Given the angular velocity at some initial time  $t_0$  the velocity at the future time  $t$  is given by the integral:

$$\omega(t) = \omega(t_0) + \int_{t_0}^t \frac{1}{\tau_c} (\tilde{\omega}(\tau - \tau_d) - \omega(\tau)) d\tau \quad (6)$$

In practice this integral is computed numerically using the recursive discrete-time relation:

$$\omega[i + 1] = \omega[i] + \frac{1}{\tau_c} \left( \tilde{\omega} \left[ i - \frac{\tau_d}{\Delta t} \right] - \omega[i] \right) \Delta t \quad (7)$$

where the integer in brackets  $[\ ]$  denotes the time index and  $\Delta t$  denotes the time step size.

### A. Online Identification of Powertrain Parameters

The time constant  $\tau_c$  and time delay  $\tau_d$  can be estimated online using an extended Kalman filter. Here we identify all the necessary variables to perform an EKF measurement update: the state  $\underline{x}$ , measurement  $\underline{z}$ , predicted measurement  $\underline{h}(\underline{x})$ , measurement uncertainty  $R$ , and the measurement Jacobian  $H$ . In short, at time  $t_0$  we predict the wheel velocity at some future time  $t$ ; at time  $t$  we compare the measured wheel velocity to the prediction, and update our estimates of  $\tau_c$  and  $\tau_d$  accordingly.

The EKF state contains the parameters we are estimating:  $\underline{x} = [\tau_c \ \tau_d]^T$ . The measurement  $\underline{z}$  is the wheel velocity at time  $t$ , as measured by the wheel encoder. The predicted measurement  $\underline{h}(\underline{x})$  is the predicted wheel velocity at time  $t$  and is calculated using (6) and our current estimates of  $\tau_c$  and  $\tau_d$ . The measurement uncertainty  $R$  is sensor-dependent.

The measurement Jacobian is:  $H = \left[ \frac{\partial \omega(t)}{\partial \tau_c} \quad \frac{\partial \omega(t)}{\partial \tau_d} \right]$ . The partial derivative with respect to the time constant  $\tau_c$  is:

$$\frac{\partial \omega(t)}{\partial \tau_c} = \int_{t_0}^t \frac{-1}{\tau_c^2} (\tilde{\omega}(\tau - \tau_d) - \omega(\tau)) - \frac{1}{\tau_c} \frac{\partial \omega(\tau)}{\partial \tau_c} d\tau \quad (8)$$

This partial derivative is also calculated numerically using a recursive, discrete-time relation analogous to (7). No straightforward analytical solution for the derivative with respect to the time delay is known, so  $\frac{\partial \omega(t)}{\partial \tau_d}$  is calculated numerically (using the central-difference for example).

### B. Experimental Results

Experimental results were obtained on a LandTamer, a hydraulic-drive skid-steered autonomous vehicle (see Fig. 1). As the left and right wheels exhibited similar dynamics, a single model was learned for both. A time interval  $(t - t_0)$  of 0.5 seconds was chosen; the time interval should be on the order of the time constant  $\tau_c$ .

In the first test, the LandTamer autonomously drove in circular trajectories of varying curvature and speed. In Fig.



Fig. 1. The LandTamer, a hydraulic-drive skid-steered autonomous vehicle.

TABLE I  
POWERTRAIN MODEL MEAN SQUARED ERROR (DEG/S)<sup>2</sup>

Test	Calibration method		
	Command	DE	ID
Circles	70.97	11.78	9.92
Teleop	501.37	105.20	94.53

2 the vehicle is commanded to instantly decelerate while traversing an arc. The powertrain model, using the online *integrated dynamics* estimates of the time delays and time constants, accurately captures the transient response of the actual wheel speeds (as measured by encoders).

In the second test, the LandTamer is teleoperated and commands alternate between  $\sim 0.5$  and 4 rad/s (see Fig. 3). The commands change so quickly that the wheels rarely reach their target speed, which is accurately captured by the calibrated model. Note also that the parameters learned in the second test differ from those in the first; this suggests the powertrain dynamics depend on the operating mode, which justifies the need for online calibration. The rate of convergence depends on the variation in velocity commands. In this test, Fig. 4 shows convergence in one minute starting from poor initial estimates.

We also implemented an online identification filter that uses the differential equation directly. Every effort was made to improve performance, including careful tuning of a band pass filter for the noisy double derivatives of encoder ticks (i.e. the measurement  $\dot{\omega}$ ), but the classical DE approach still underperformed the ID approach, especially at identifying the time delay. This is because, using the DE approach,  $\frac{\partial \dot{\omega}}{\partial \tau_d}$  is observable (or nonzero) only at the instant when commands change but using the ID approach,  $\frac{\partial \omega(t)}{\partial \tau_d}$  is observable whenever a command change occurs in the interval  $t - t_0$ . Intentionally poor initial parameter estimates were chosen ( $\tau_c = 0.2$  s,  $\tau_d = 0.25$  s), but with sufficiently large initial uncertainty, to highlight the difference in performance.

Table I presents the mean squared error (MSE) of wheel velocity predictions for powertrain models calibrated using both the DE and ID approaches, and one that naïvely predicts the nominal commands. MSE is larger in the “Teleop” test because command changes are more frequent and aggressive.

### III. CALIBRATION OF THE POSE ESTIMATION SYSTEM

This section explains how parameters required for pose estimation can be calibrated online using *integrated per-*

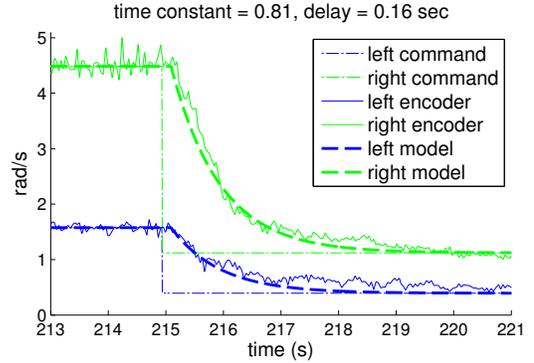


Fig. 2. Left and right wheel angular velocities vs. time as the LandTamer decelerates. The speeds predicted by the calibrated model closely match the actual (encoder) speeds.

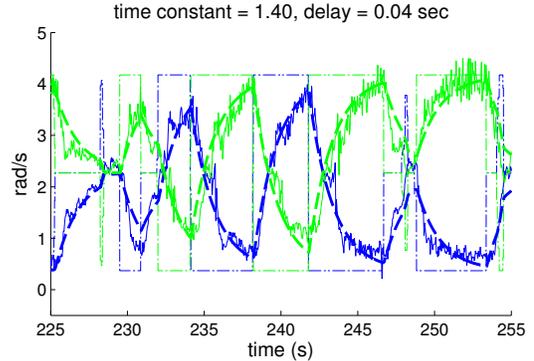


Fig. 3. Wheel velocities vs. time as the LandTamer is teleoperated. See legend in Fig. 2

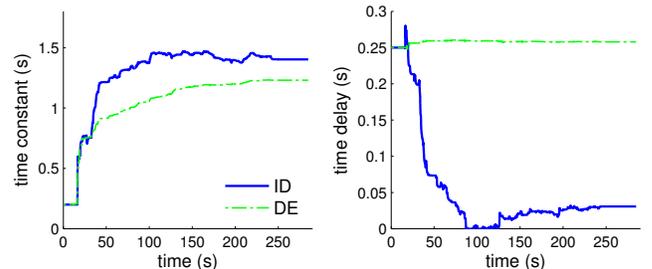


Fig. 4. Estimates of powertrain parameters vs. time during the teleoperated LandTamer test, using both the integrated dynamics (ID) and differential equation (DE) online calibration approaches. The DE approach fails to identify the time delay which affects the time constant estimate.

*turbative dynamics* (IPD). To demonstrate the process, we calibrate the position of the IMU, steer angle sensors, and the wheel radius on the Zoë rover. Zoë previously surveyed the distribution of biological life in Chile’s Atacama desert [13]. Zoë has four independently driven wheels and two passively articulated axles (see Fig. 5).

#### A. Odometry Equations

When performing dead reckoning in 3D (assuming a zyx Euler angle convention), the pose of the vehicle  $\rho$  in a ground-fixed frame is predicted by integrating the following

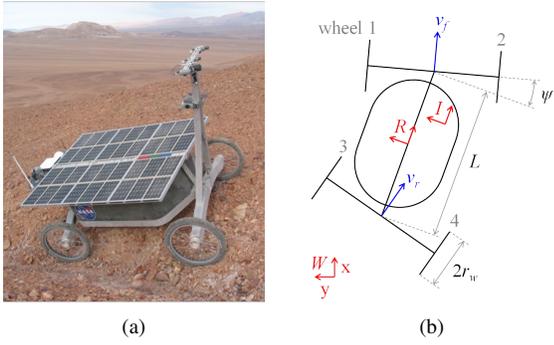


Fig. 5. (a) The Zoë rover in the Atacama desert. (b) A diagram of the rover. The (W)orld, (R)obot, and (I)MU coordinate systems are shown in red.

kinematic differential equation:

$$\dot{\underline{\rho}} = \underline{f}(\underline{\rho}, \underline{u})$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} c\theta c\beta & c\theta s\beta s\gamma - s\theta c\gamma & 0 \\ s\theta c\beta & s\theta s\beta s\gamma + c\theta c\gamma & 0 \\ 0 & 0 & \frac{c\gamma}{c\beta} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_\theta \end{bmatrix} \quad (9)$$

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

Where  $\underline{u} = [V_x \ V_y \ V_\theta]^T$  is the velocity of the vehicle in the body frame, and is computed as follows:

$$\underline{u}(\underline{p}) = H_v^+ \underline{v}$$

$$\begin{bmatrix} V_x \\ V_y \\ V_\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{L} & 0 & -\frac{1}{L} \end{bmatrix} \left( \frac{r_w}{2} \begin{bmatrix} (\omega_1 + \omega_2)c\psi_f \\ (\omega_1 + \omega_2)s\psi_f \\ (\omega_3 + \omega_4)c\psi_r \\ (\omega_3 + \omega_4)s\psi_r \end{bmatrix} \right) \quad (10)$$

In (10),  $L$  is the vehicle length,  $r_w$  the wheel radius, and  $\omega_1$  to  $\omega_4$  are wheel angular velocities (see Fig. 5(b)). The front and rear steer angles ( $\psi_f$ ,  $\psi_r$ ) are sensed using potentiometers; measured steer angle is a linear function of the potentiometer output voltage  $V$ :

$$\psi_f = m_f V_f + b_f, \quad \psi_r = m_r V_r + b_r \quad (11)$$

The pose of the robot at the end of the path segment is predicted by integrating the kinematic differential equation (9) as follows:

$$\underline{\rho}(t) = \underline{\rho}(t_0) + \int_{t_0}^t \underline{f}(\underline{\rho}(\tau), \underline{u}(\tau, \underline{p})) d\tau \quad (12)$$

### B. IMU Pose Equations

By default, the GPS/INS unit on Zoë returns the pose of the IMU with respect to the world frame (e.g. UTM coordinates). In order to accurately estimate the pose of the robot with respect to the world frame we must know the relative pose of the IMU/robot frames.

For a path segment from  $t_0$  to  $t$  the predicted final pose of the IMU is a composition of poses (calculated by multiplying the corresponding homogenous transforms):

$$\begin{aligned} \underline{\rho}_{I_f}^W &= \underline{\rho}_{I_i}^W * \underline{\rho}_{R_i}^{I_i} * \underline{\rho}_{R_f}^{R_i} * \underline{\rho}_{I_f}^{R_f} \\ \underline{\rho}_{I_f}^W &= \underline{\rho}_{I_i}^W * (\underline{\rho}_I^R)^{-1} * \underline{\rho}_{R_f}^{R_i} * \underline{\rho}_I^R \end{aligned} \quad (13)$$

In (13), the lowercase subscripts  $i$  and  $f$  denote initial and final poses (i.e. at the beginning and end of a path segment).  $\underline{\rho}_I^R$  is the current estimate of the pose of the IMU with respect to the robot frame.  $\underline{\rho}_{I_i}^W$  is the GPS/INS measurement of the initial pose of the IMU with respect to the world frame.  $\underline{\rho}_{R_f}^{R_i}$  is the final pose of the robot with respect to the initial robot frame, and is computed using the odometry integral (12).

### C. Online Identification of Pose Estimation Parameters

This section explains how the parameters required for pose estimation are calibrated online using an extended Kalman filter. Just as in the section on powertrain model identification, we identify all the necessary variables to perform a measurement update.

The state vector contains the parameters we wish to estimate:

$$\underline{x} = \begin{bmatrix} \underline{\rho}_I^R \\ \underline{p}_{odom} \end{bmatrix}, \quad \underline{p}_{odom} = [m_f \ b_f \ m_r \ b_r \ r_w]^T \quad (14)$$

$\underline{p}_{odom}$  is a vector of odometry parameters. The measurement  $\underline{z}$  is the final pose of the IMU in the world frame  $\underline{\rho}_{I_f}^W$ , as measured by the GPS/INS. The predicted measurement  $\underline{h}(\underline{x})$  is the predicted final pose of the IMU computed using (13)

The measurement uncertainty  $R$  depends on the accuracy of the pose measurement device. The measurement Jacobian  $H$  can be considered in two parts:

$$H = \begin{bmatrix} \frac{\partial \underline{\rho}_{I_f}^W}{\partial \underline{\rho}_I^R} & \left( \frac{\partial \underline{\rho}_{I_f}^W}{\partial \underline{\rho}_{R_f}^{R_i}} \frac{\partial \underline{\rho}_{R_f}^{R_i}}{\partial \underline{p}_{odom}} \right) \end{bmatrix} \quad (15)$$

$\frac{\partial \underline{\rho}_{I_f}^W}{\partial \underline{\rho}_I^R}$  and  $\frac{\partial \underline{\rho}_{I_f}^W}{\partial \underline{\rho}_{R_f}^{R_i}}$  are Jacobians of the pose composition equation (13) with respect to intermediate poses; they can be solved for in closed form using symbolic mathematics software.  $\frac{\partial \underline{\rho}_{R_f}^{R_i}}{\partial \underline{p}_{odom}}$  is the Jacobian of the odometry integral (12) with respect to the steer angle and wheel radius parameters.

The linearized error dynamics of the system help to compute this Jacobian. The following convolution integral relates input velocity perturbations to a change in terminal pose:

$$\delta \underline{\rho}(t) = \Phi(t, t_0) \delta \underline{\rho}(t_0) + \int_{t_0}^t \Gamma(t, \tau) \delta \underline{u}(\tau, \Delta \underline{p}_{odom}) d\tau \quad (16)$$

In short, (16) attributes errors in our terminal pose prediction to errors in the initial pose measurement and velocity errors  $\delta \underline{u}$ , caused by errors in our odometry parameter estimates.  $\Phi$  and  $\Gamma$  are the *transition* and *input transition* matrices:

$$\Phi(t, \tau) = \begin{bmatrix} 1 & 0 & -(y(t) - y(\tau)) \\ 0 & 1 & x(t) - x(\tau) \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$\Gamma(t, \tau) = \Phi(t, \tau) \frac{\partial \underline{f}}{\partial \underline{u}} \quad (18)$$

The transition matrix  $\Phi(t, \tau)$  accounts for the nonlinear effect that heading errors at time  $\tau$  have on (x,y) position at the future time  $t$ . Due to space limitations the derivation of

(17) and (18) is not included here, but can be found in [14]. Applying (16), the rightmost term in  $H$  is:

$$\frac{\partial \rho_{Rf}^{Ri}}{\partial p_{odom}} = \int_{t_0}^t \Gamma(t, \tau) U_p d\tau \quad (19)$$

where  $U_p$  is the Jacobian of  $\underline{u}$  in (10) with respect to  $p_{odom}$ . The Leibniz rule is used to move the derivative inside the integral.

#### D. Experimental Results

Experimental data was collected while driving the Zoë rover by joystick at Flagstaff hill in Pittsburgh, Pennsylvania. Zoë is equipped with a high-end differential GPS/INS unit that produced independent pose measurements (without odometry aiding). Note however that GPS is not necessary; ground truth need only be accurate for *relative* pose measurements a few seconds apart so other low-cost techniques could work such as visual registration of a few carefully surveyed landmarks.

In a short traverse of just 163 m, the online IPD filter produced accurate values for IMU position offsets and odometry parameters. The final parameter estimates are shown in Table II along with manually calibrated values. In the “Manual” column the IMU offsets and wheel radius were measured with a tape measure, and the steer angle potentiometer parameters were calibrated manually using precision-machined angle set blocks.

TABLE II  
CALIBRATED PARAMETER VALUES

	Manual	IPD
$x_I^R$ (cm)	38.0	46.02
$y_I^R$ (cm)	-15.0	-19.86
$m_f$ (rad/V)	-0.1128	-0.1192
$b_f$ (rad)	-0.0054	0.0019
$m_r$ (rad/V)	-0.1080	-0.1181
$b_r$ (rad)	-0.0105	0.0021
$r_w$ (cm)	32.5	31.62

Although the IPD parameter values may seem to differ little from the manual values, they produce a significant improvement in pose prediction accuracy. Fig. 6 shows scatter plots of along-track and cross-track pose prediction error for 6 second path segments. Notice the mean prediction error using the IPD parameter estimates is reduced from 21 cm to near zero, and the standard deviations of error are reduced by 20-40%. Quantitative results including heading error are shown in Table III. The improvement is even more pronounced when integrating the *entire* predicted path (see Fig. 7). Note that, integrated odometry pose predictions used IMU attitude measurements ( $\gamma$ ,  $\beta$ ) but *not* heading ( $\theta$ ).

Fig. 8 compares IPD performance to the classical (DE) approach which uses the differential equation (9) directly. In the classical approach, pose measurements are numerically differentiated to obtain velocity measurements:

$$\dot{\underline{\rho}}[k] = \frac{\underline{\rho}[k+1] - \underline{\rho}[k-1]}{t[k+1] - t[k-1]} \quad (20)$$

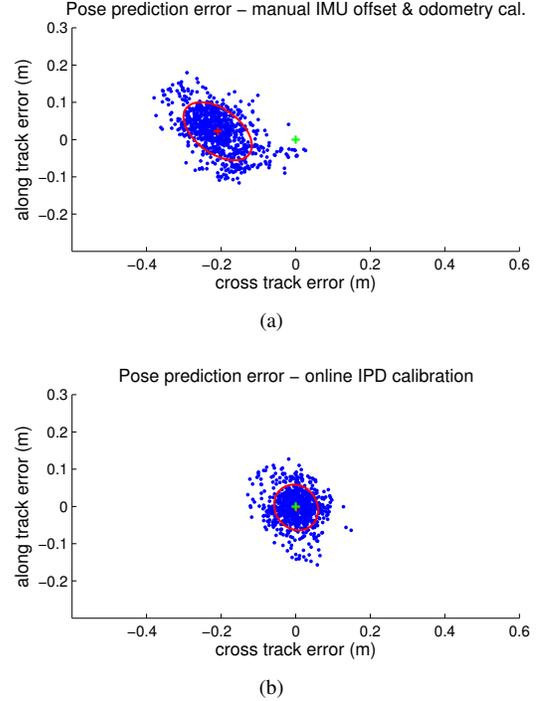


Fig. 6. Scatter plots of pose prediction error for overlapping 6 second path segments. (a) using manual calibration parameter estimates. (b) using online IPD estimates, computed using 20-fold cross-validation.

TABLE III  
ODOMETRY POSE PREDICTION ERROR<sup>1</sup>

Axis	Manual		Online IPD <sup>2</sup>	
	$\mu$	$\sigma$	$\mu$	$\sigma$
Along-track (cm)	2.23	5.23	-0.28	4.12
Cross-track (cm)	-20.91	6.15	0.13	3.96
Heading (deg)	-1.48	0.85	-0.01	0.49

<sup>1</sup> for 6 second path segments

<sup>2</sup> computed using 20-fold cross-validation

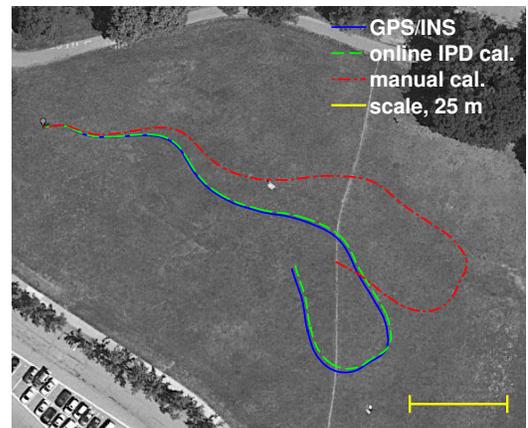


Fig. 7. GPS/INS path of the Zoë rover at Flagstaff hill. The integrated wheel odometry path using the online IPD parameter estimates is much more accurate than wheel odometry using manual calibration values.

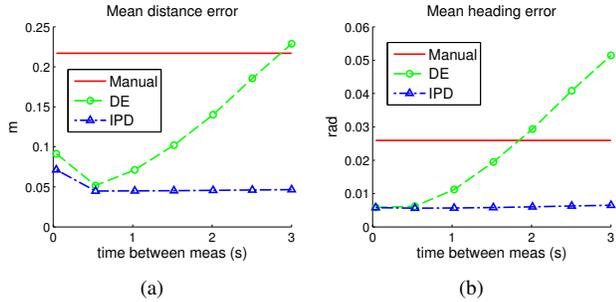


Fig. 8. Comparison of IPD and DE (differential equation) approaches. (a) Mean (x,y) Euclidean distance error of pose predictions vs. the time between pose measurements ( $t-t_0$ ) during training. (b) Mean of the *absolute value* of heading errors. Pose prediction errors are computed for 6 second path segments. When using the classical DE approach, pose prediction errors increase with the delay between measurements.

Accuracy of the numerical derivatives and performance of the DE approach degrade as the delay between pose measurements increases. Pose prediction errors in Fig. 8 are computed for 6 second path segments. This explains the spike in error in Fig. 8(a) when calibrating to very short path segments; IPD calibration results are best when the model is trained and evaluated using paths of comparable length.

In this test, wheel slip was kept to a minimum by Zoë’s passive steering design, high traction with the grassy terrain, and a speed limit of 1 m/s. Minimizing wheel slip is important when calibrating odometry parameters because of observability issues. For example, wheel radius error can not be distinguished from longitudinal wheel slip that is proportional to forward velocity  $V_x$ . Intrinsic parameters may also be indistinguishable; for example,  $\theta_f^R$  can not be concurrently calibrated with  $b_f$  and  $b_r$ . Parameters are indistinguishable if they produce linearly dependent columns in the measurement Jacobian,  $H$ .

#### IV. CONCLUSIONS

This paper demonstrated how integrated dynamics (ID) and integrated perturbative dynamics (IPD) can be used to automate the tedious but essential aspects of vehicle calibration. The ID/IPD approach has advantages over classically derived calibration techniques which use the system differential equation directly. Specifically, ID/IPD enables direct use of state measurements instead of their numerical derivatives which can be extremely noisy. We used ID/IPD in combination with an extended Kalman filter to identify intrinsic parameters of the vehicle powertrain and pose estimation system online.

Using integrated dynamics, we presented a method to identify both the time constant and delay of a first-order powertrain model, without requiring double differentiation and bandpass filtering of encoder ticks. In addition, using an online IPD algorithm we simultaneously calibrated IMU position offsets and odometry parameters (steer angle sensors, wheel radius) on the Zoë rover in one short traverse. Pose predictions using the IPD approach were significantly more accurate than using the manual or classical (differential

equation) calibration approaches, especially when delays between pose measurements were large.

In conclusion, ID/IPD is an effective tool for the identification of intrinsic robot parameters, especially when only low-frequency state measurements are available. In future work we will apply active learning techniques (e.g. [15]) to command trajectories that will most reduce the uncertainty of the parameter estimates.

#### ACKNOWLEDGMENT

This research was made with U.S. Government support under and awarded by the Army Research Office (W911NF-09-1-0557) and by the DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. The authors would also like to thank David Wettergreen and Dominic Jonak for making Zoë available for testing.

#### REFERENCES

- [1] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner, “Toward reliable off road autonomous vehicles operating in challenging environments,” *International Journal of Robotics Research*, vol. 25, no. 5/6, 2006.
- [2] W. Yu, O. Chuy, E. Collins, and P. Hollis, “Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle,” *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 340–353, april 2010.
- [3] M. Sheehan, A. Harrison, and P. Newman, “Automatic self-calibration of a full field-of-view 3d n-laser scanner,” in *Proc. International Symposium on Experimental Robotics*, New Delhi, India, December 2010.
- [4] J. Kelly and G. S. Sukhatme, “A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors,” in *Proc. International Symposium on Experimental Robotics*, New Delhi, India, December 2010.
- [5] S. Shaw, “Numerical methods for identification of induction motor parameters,” Master’s thesis, MIT, 1995.
- [6] J. Kelly and G. S. Sukhatme, “Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration,” *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.
- [7] F. Mirzaei and S. Roumeliotis, “A kalman filter-based algorithm for imu-camera calibration: observability analysis and performance evaluation,” *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1143–1156, Oct 2008.
- [8] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 869–880, 1996.
- [9] N. Roy and S. Thrun, “Online self-calibration for mobile robots,” in *Proc. IEEE International Conference on Robotics and Automation*, 1998.
- [10] G. Antonelli, S. Chiaverini, and G. Fusco, “A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation,” *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 994–1004, oct. 2005.
- [11] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008, ch. 14.
- [12] 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.
- [13] D. Wettergreen *et al.*, “Second experiments in the robotic investigation of life in the Atacama desert of Chile,” in *Proc. 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005.
- [14] A. Kelly, “Linearized error propagation in odometry,” *International Journal of Robotics Research*, vol. 23, no. 1, pp. 179–218, Feb 2004.
- [15] J. K. Kruschke, “Bayesian approaches to associative learning: From passive to active learning,” *Learning & Behavior*, vol. 36, no. 3, pp. 210–226, Aug. 2008.