

# The FLEXnav Precision Dead-reckoning System <sup>1</sup>

by

Lauro Ojeda, Giulio Reina, Daniel Cruz and Johann Borenstein

The University of Michigan

*Abstract*— Research at the University of Michigan’s Mobile Robotics Lab aims at the development of an accurate proprioceptive (i.e., without external references) position estimation (PPE) system for planetary rovers. Much like other PPE systems, ours uses an inertial measurement unit (IMU) comprising three fiber-optic gyroscopes and a two-axes accelerometer, as well as odometry based on wheel encoders.

Our PPE system, however, is unique in that it does not use the conventional Kalman Filter approach for fusing data from the different sensor modalities. Rather, our system combines data based on expert rules that implement our in-depth understanding of each sensor modality’s behavior under different driving and environmental conditions. Since our system also uses Fuzzy Logic operations in conjunction with the Expert Rules for finer gradation, we call it Fuzzy Logic Expert navigation (FLEXnav) PPE system.

The paper presents detailed experimental results obtained with our FLEXnav system integrated with our planetary rover clone “Fluffy” and operating in a Mars-like environment. In addition, we compare the results of our FLEXnav system with results obtained from a conventional Kalman Filter. The paper also introduces new methods for wheel slippage detection and correction, along with comprehensive experimental results.

## 1 INTRODUCTION

Proprioceptive position estimation (PPE), more commonly known as dead-reckoning, is widely used for measuring the relative displacement of mobile robots. Many conventional high-end dead-reckoning systems for ground-vehicles (typically, mobile robots) appear to be implemented according to a common approach: A 6-axes INS is fused with odometry using a Kalman Filter technique. Kalman Filters use statistical error models [Tonouchi, et al., 1994; Krantz and Gini, 1996; Fuke and Krotkov, 1996; Baumgartner et al, 2001] to predict the behavior of sensor components.

In this paper we investigated an alternative approach, based on in-depth physical understanding of sensors and their associated error sources. The philosophy behind our approach is that certain error mechanisms can be defined quite accurately by expert reasoning. The resulting PPE system is called Fuzzy Logic and Expert rule-based navigation system (FLEXnav) [Ojeda and Borenstein, 2002].

In addition to the FLEXnav method for data fusion, our system employs innovative measures for improving position estimation accuracy. The paper here describes measures that were optimized for 6-wheel-drive/6-wheel-steer vehicles with a rocker-bogie suspension system. For simplicity, we will refer to mobile platforms of this design as “planetary rovers” throughout this paper. Well-known planetary rovers are Rocky 8 and FIDO – both developed at the Jet

---

<sup>1</sup> Parts of this paper were presented at the 2002 IEEE International Conference on Robotics and Automation, Washington DC, May 11-15, 2002; some other parts were presented at the 2004 IEEE Aerospace Conference, Big Sky, MT, March 6-13, 2004.

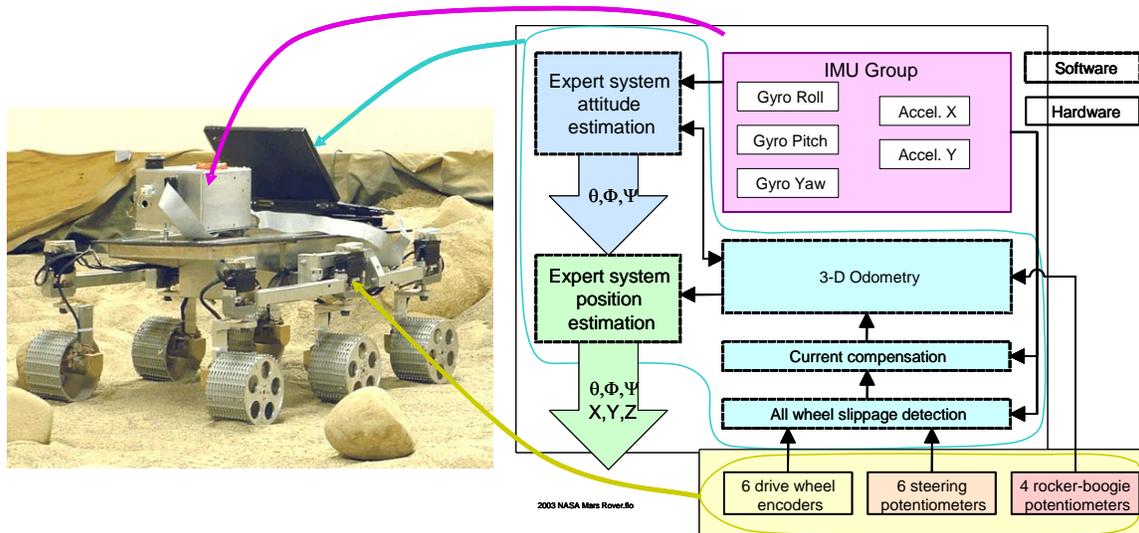
Propulsion Lab (JPL – see, for example, [Tunstel et al., 2002]). We investigated the following accuracy-enhancing measures:

*I. Redundant Encoders* – Odometric measurements typically require encoder data from one wheel on each side of the robot. However, if wheels are slipping, then only a single non-slipping wheel is needed to perform odometry computations – the needed encoder data of the wheel on the other side can be recovered from gyro data. Numerous expert rules in our system help determine which wheel is most likely to be the one that was slippage-free in a given sampling interval.

*II. All-wheel Slippage Detection* – This group of three individual measures compares data from different encoders with (1) each other, (2) gyro data, and (3) motor current data, to determine whether all wheels were slipping in a given sampling interval. If so, then odometry becomes ineffective and a flag is raised.

*III. Current-compensated Odometry* – This method aims at extracting quantitative information about the extent of wheel slippage from the all-wheel slippage indicators, and then uses that information to reduce odometry errors.

In order to develop our system and validate its performance, we built “Fluffy” a kinematic equivalent to JPL’s Rocky-8 and FIDO planetary rovers. Fluffy is about half the size of JPL’s FIDO rovers and it is shown in Figure 1.



**Figure 1:** The University of Michigan-built Fluffy and a block diagram of the FLEXnav dead-reckoning system.

The remainder of this paper is organized as follows. Section 2 describes our FLEXnav system in detail. Section 3 introduces further enhancements to the basic FLEXnav system, and presents related experimental results. Comprehensive experimental results from our FLEXnav system implemented on Fluffy and driving through a Mars-like, 3-D landscape are presented in Sections 4. Section 5 compares a subset of results obtained with the FLEXnav system to results obtained from a basic Kalman Filter applied to the exact same raw data set.

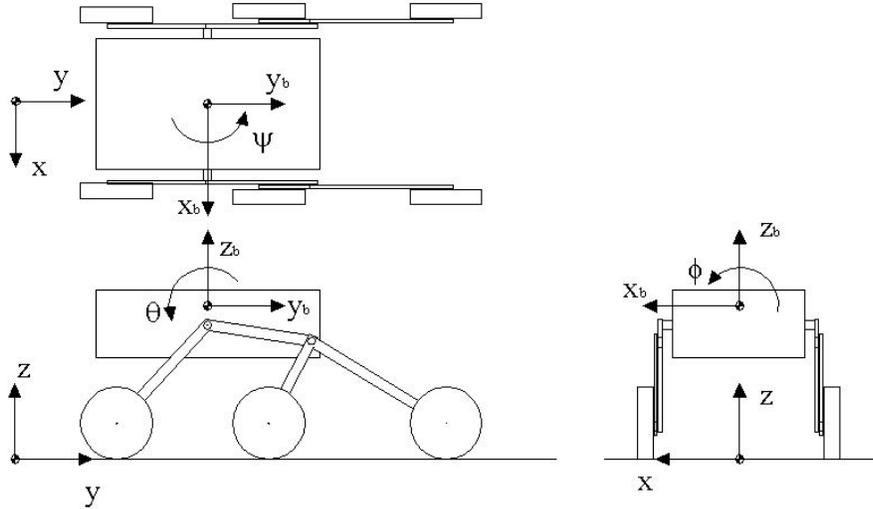
## 2 THE FLEXNAV SYSTEM

A block diagram of the FLEXnav sensor suite, as implemented on Fluffy, is shown in Figure 1. The main inertial sensors are three single-axis fiber optic gyroscopes (FOGs) that measure rates of yaw, roll, and pitch; the two accelerometers are of lesser importance. The inertial sensors are mounted in an aluminum enclosure on top of Fluffy.

Fluffy is also equipped with six wheel encoders, six steering potentiometers, and potentiometers for measuring all relevant angles within the rocker-bogie mechanism and between the rocker-bogie mechanism and the chassis.

### Attitude Estimation

The attitude of a robot is a set of three angles measured between the robot's body and the absolute world coordinate system. The term "navigation frame" is used for a world coordinate system, in which the x-axis points east, the y-axis points north, and the z-axis is parallel but opposite in sign to the local gravity vector. Another coordinate system, called "body frame," can be thought of as embedded in the robot body. The body frame follows a prevailing convention used in ground vehicles for assigning the axes of coordinate systems, so that its x-axis points to the right, the y-axis points forward, and the z-axis points upward. Body axes are labeled  $x_b$ ,  $y_b$ , and  $z_b$  [Kelly, 1995], and the accelerometers and gyros described in this paper were mounted in alignment with these axes. The vehicle attitude is usually expressed in terms of three special angles known as "Euler angles." In our system the RPY or ZXY convention is chosen. Figure 2 shows the three Euler angles  $\phi$ ,  $\theta$ , and  $\psi$ , which are usually referred to as *roll* (sometimes also called "bank angle"), *pitch* (also called "elevation"), and *yaw* (also called "heading" or "azimuth") respectively. Rates of rotation of the body frame relative to the navigation frame can be expressed in terms of the derivatives of the Euler angles, called "Euler rates."



**Figure 2:** Robot axes and Euler angles (adopted from [Kelly, 1995]).

Specifically, Euler rates  $\Omega_e$  and body rates of rotation  $\Omega_b$  are related by:

$$\Omega_e = [\dot{\theta} \quad \dot{\phi} \quad \dot{\psi}]^T = C_b^e \Omega_b \quad (1)$$

where

$$C_b^e = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ \tan \theta \cdot \sin \phi & 1 & -\tan \theta \cdot \cos \phi \\ -\frac{\sin \phi}{\cos \theta} & 0 & \frac{\cos \phi}{\cos \theta} \end{bmatrix}$$

$$\Omega_b = [\omega_x \quad \omega_y \quad \omega_z]^T$$

$\omega_x$ ,  $\omega_y$ , and  $\omega_z$  are the rates of rotation of the vehicle around the respective axes of the body frame.

Euler angles can be calculated from Euler rates by integrating  $\Omega_e$  over time:

$$\begin{bmatrix} \theta & \phi & \psi \end{bmatrix}^T = \int \Omega_e dt \quad (2)$$

For many indoor mobile robotics applications, where floors are typically level, it is an acceptable and widely used assumption that  $\phi$  and  $\theta$  can be considered to equal zero. With this assumption we can rewrite Eq. (1):

$$\dot{\psi} = \omega_z \quad (3)$$

According to Eq. (3) the heading angle  $\psi$  can be estimated by integrating only  $\omega_z$ . However, on rugged terrain Eq. (3) does not hold and all three attitude parameters must be considered.

### Attitude Estimation Sensors

Attitude relative to the horizontal plane  $xy$  (i.e., roll and pitch –  $\phi$  and  $\theta$ , respectively) is often referred to as “*tilt*.” Low-cost tilt sensors are commercially available, typically in the form of electrolytic fluid sensors. However, most of these sensors are suitable only for static or quasi-static conditions because of their slow time response. In other words, they are suitable only for operation on very gently rolling terrain or at constant speeds. For operation on rugged terrain, it is necessary to use redundant and complementary sensors such as gyroscopes and accelerometers, as will be explained below.

For airborne applications such as missiles or aircraft, all three attitude parameters are required. The only way to obtain these parameters is through integration of rate information from sets of three mutually perpendicular high-quality gyroscopes. In mobile robots, on the other hand, we can make use of several mitigating assumptions, that can’t be made in airborne applications, such as:

1. The mobile robot is horizontal, near horizontal or at constant tilt, *most* of the time (this is also true for most long-range flights).
2. Mobile robot velocities and accelerations are small – orders of magnitude lower than those of missiles or aircraft, for which most high-quality IMUs are designed.
3. Wheeled and even tracked vehicles allow for the use of odometry. This advantage is mostly absent in aircraft or watercraft, which can only measure velocities relative to the surrounding medium.

In our system we use gyroscopes of different performance levels. The higher-quality FOG is used for the  $z$ -axis. This is because, according to assumption (1) above,  $\omega_z$  has the greatest impact on the heading angle  $\psi$ , which is the most important angle for land vehicles. Two lower-quality FOGs are used to measure the less important angular rates  $\omega_x$  and  $\omega_y$ .

Under assumption (1) (flat or moderately rolling terrain) and according to Eq. (1),  $\omega_x$  and  $\omega_y$  have less impact on the heading angle than  $\omega_z$  does. In earlier work we developed a precision calibration system for gyroscopes that reduces the errors due to the non-linearity of the scale factor and temperature by about one order of magnitude compared to an off-the-shelf unit [Ojeda et al., 2000]. We applied this same calibration technique to the FOGs in our system here.

We also incorporated two accelerometers along the  $x_b$  and  $y_b$  axes to estimate tilt (i.e.,  $\phi$  and  $\theta$ ). This can be done when the robot is static or moving linearly at a constant speed (a similar

sensor configuration was used in [Rehbinder and Hu, 2001]). Under these conditions tilt can be calculated as:

$$\varphi = \sin^{-1}\left(\frac{g_x}{g \cos\theta}\right) \quad (4)$$

$$\theta = \sin^{-1}\left(\frac{g_y}{g}\right) \quad (5)$$

where

$g$  - Gravitational acceleration

$g_{x/y}$  -  $x/y$ -component of the gravitational acceleration

Similar to gyroscopes, accelerometers suffer from bias drift problems. It is well established that accelerometers are generally not suitable for measuring linear displacement in mobile robots [Barshan and Durrant-Whyte, 1995]. This is because accelerometer measurements must be integrated twice to yield position, and thus even small amounts of drift will grow substantially and without bound. However, when the vehicle is static or moving at constant speed, accelerometers can be used as tilt sensors since tilt information can be derived directly from the accelerometer readings, according to Eqs. (4) and (5). For the measurement of tilt with accelerometers, integration is not needed and therefore drift is not a dominant source of errors. Rather, other error sources become relatively more significant, such as inaccuracy, noise, non-linearity, and sensitivity to vibration. Nonetheless, accelerometers can be useful to bound and reset the tilt information calculated by the gyros.

Under dynamic conditions, i.e., when the robot accelerates, accelerometers will also measure the acceleration of the robot in addition to the robot's tilt. However, this undesirable ambiguity can be resolved using encoder readings.

### Static Bias Drift Compensation

One well known source of errors in gyroscopes is static bias drift. In this paper we are not focusing on this (significant) error source because a trivial short-term calibration method is effective in dealing with this problem: Prior to each mobile robot mission (i.e., while the robot is standing still) the gyro output is sampled for, say, 5-10 seconds, and the readings are averaged. Then, once the robot is moving, this averaged static bias value is subtracted from all subsequent gyro readings. Because of the relatively low static bias drift in FOGs, mission durations on the order of 10 minutes are feasible before a new static bias drift value should be determined by repeating the above short-term calibration.

One should also note that this short-term calibration procedure also compensates for the constant planetary rotation, provided the rover does not spend significant amounts of time on slopes. While the rover drives on slopes, planetary rotation affects all gyros differently from the way they were affected during the static bias drift calibration phase.

The static bias drift of the accelerometers can be accounted for in a similar fashion during periods of stand-still on a perfectly horizontal terrain patch. However, during a real mission one cannot assume that a given terrain patch, on which the robots stops, is perfectly horizontal. Therefore it is necessary to measure the actual tilt of the robot during the static bias drift compensation phase through the use of electrolyte-type tilt sensors. Yet, in the experiments at our lab we performed the static bias drift measurements only on a perfectly horizontal terrain patch, to minimize the complexity of our system.

## Fuzzy Data Fusion Using Expert Rules

Based on the specific physical shortcomings and strengths of each sensor modality, we defined the following basic expert rules:

**Rule 1:** If the vehicle is in the process of turning about any of its axes, our best attitude estimate is the one derived from the gyroscope outputs

$$[\theta \ \varphi \ \psi]^T \approx [\theta_g \ \varphi_g \ \psi_g]^T \quad (6)$$

The index ‘g’ indicates that the value was derived from gyro data, and  $\theta_g$ ,  $\varphi_g$ , and  $\psi_g$  are computed from gyro data according to Eqs. (1) and (2). We define the symbol “ $\approx$ ” as meaning “weighted toward.” The meaning of this term will become evident in the discussion on Fuzzy Logic, below.

**Rule 2:** If the robot is not turning around any axis and is not accelerating linearly, the accelerometers can directly measure the roll and pitch attitude parameters ( $\phi_a$  and  $\theta_a$ ). If the conditions of Rule 2 are met for several seconds uninterruptedly, then we can also measure and correct for the bias drift errors of the gyroscope (see [Ojeda et al., 2000]) and reset the tilt parameters of the robot to the tilt estimated by the accelerometers (see Eqs. 4 and 5):

$$\varphi \approx \varphi_a \text{ and } \theta \approx \theta_a \quad (7)$$

Even though the sensor integration conditions are well defined and sensible, it is not feasible to implement them as strictly binary rules. This is due to the natural imprecision of the sensors and because conditions like “robot not turning” or “constant speed” are not realistic when the vehicle is in motion on rugged terrain.

A fusion algorithm that takes into account the physical capabilities and limitations of each sensor is therefore necessary. We found that Fuzzy Logic is well suited for this task.

- Fuzzy Logic uses rules to map inputs and outputs. Using natural language, expert rules such as the ones described above can be translated easily into the IF-THEN statements used by Fuzzy Logic rules.
- Fuzzy Logic is specifically designed to deal with the imprecision associated with noisy sensors.
- Trying to use a deterministic approach to solve this kind of problem would require the development of a highly non-linear system model, which, in turn, would increase the complexity and development time. Fuzzy Logic, on the other hand, can handle nonlinear models of arbitrary complexity [Jang et al., 1997].

Our fuzzy data fusion uses four fuzzy membership function inputs and two outputs, as shown in Figure 3. The first input represents the state of rotation, that is, whether the platform is rotating about any axis. The parameter that determines this condition is calculated by:

$$\omega_t[k] = \left| \omega_x[k] \right| + \left| \omega_y[k] \right| + \left| \omega_z[k] \right| \quad (8)$$

$k$  – Discrete time index

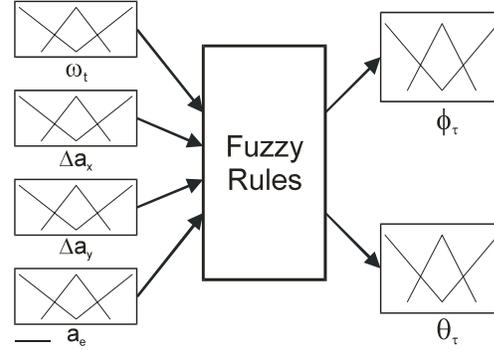
The second and third input use accelerometer data to determine if the acceleration of the robot is changing:

$$\Delta a_x[k] = a_x[k] - a_x[k-1] \quad (9)$$

$$\Delta a_y[k] = a_y[k] - a_y[k-1] \quad (10)$$

$\Delta a_x \approx 0$  and  $\Delta a_y \approx 0$  means that the robot is either standing or moving with constant acceleration, and that it is standing or moving on terrain that has a constant slope. The “standing-or-moving” ambiguity is resolved using encoder information, which is the fourth input to the system. The first derivative of the velocity, as measured by the encoders, represents the rover’s acceleration

$$a_e[k] = \frac{\Delta v_e[k]}{T_s} = \frac{v_e[k] - v_e[k-1]}{T_s} \quad (11)$$



**Figure 3:** Inputs and outputs of the FLEXnav system.

The outputs of the fuzzy fusion system,  $\phi_t$  and  $\theta_\tau$ , are dimensionless weighting factors that emphasize either the gyroscope readings (*GYRO*), the accelerometer readings (*ACCEL*), or weigh one relative to the other (*BOTH*). In practice the maximum range for these weighting factors are chosen by the system designer depending on the accuracy and noise specification of the sensors.

Based on the Fuzzy Logic rules above, we can now compute the attitude

$$\varphi[k] = \varphi_g[k] + (\varphi_a[k] - \varphi_g[k]) \varphi_\tau[k] \quad (12)$$

$$\theta[k] = \theta_g[k] + (\theta_a[k] - \theta_g[k]) \theta_\tau[k] \quad (13)$$

Using Eq. 2 it follows:

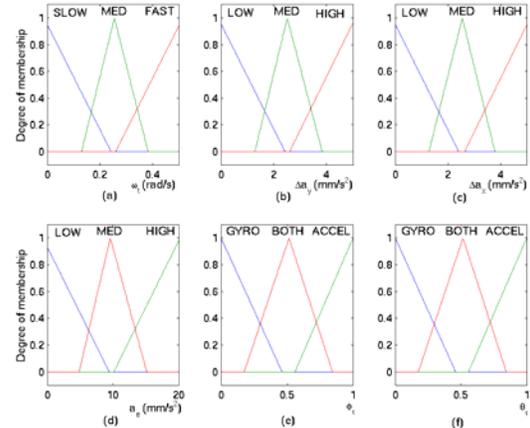
$$\varphi_g[k] = \varphi[k-1] + \dot{\varphi}_g[k] T_s \quad (14)$$

$$\theta_g[k] = \theta[k-1] + \dot{\theta}_g[k] T_s \quad (15)$$

where  $T_s$  is the sampling period, and with (from Eqs. 4 and 5):

$$\theta_a[k] = \sin^{-1} \left( \frac{g_y[k]}{g} \right) \quad (16)$$

$$\varphi_a[k] = \sin^{-1} \left( \frac{g_x[k]}{g \cos(\theta[k-1])} \right) \quad (17)$$



**Figure 4:** Membership functions of the fuzzy system.

Once the inputs and outputs are identified and defined, the relationship between them must be established. As mentioned above, Fuzzy Logic uses *if-then* rules to map inputs and outputs. For our Fuzzy Logic fusion system we translated our knowledge base (i.e., fusion rules) into the fuzzy rules shown in Table I. The fuzzy rules are derived directly from the two basic rules defined at the beginning of this section and they cover all possible combinations of the input variables. The membership functions used as input and output of our system are shown in Figure 4.

It should be noted that the FLEXnav system described up to this point only reduces errors in roll and pitch ( $\phi$  and  $\theta$ , respectively). It is important to reduce these errors, since on rugged terrain roll and pitch affect the computation of the heading angle  $\psi$ , according to Eq. (1).

**Table I:** Fuzzy Logic rules used in the FLEXnav system

- $\mathfrak{R}_1$  : *if*  $\omega_t$  is not SLOW *then*  $\phi_t$  is GYRO *and*  $\theta_t$  is GYRO
- $\mathfrak{R}_2$  : *if*  $\Delta a_x$  is HIGH *then*  $\theta_t$  is GYRO
- $\mathfrak{R}_3$  : *if*  $\Delta a_y$  is HIGH *then*  $\theta_t$  is GYRO
- $\mathfrak{R}_4$  : *if*  $\Delta a_e$  is HIGH *then*  $\phi_t$  is GYRO *and*  $\theta_t$  is GYRO
- $\mathfrak{R}_5$  : *if*  $\omega_t$  is SLOW *and*  $\Delta a_x$  is LOW *and*  $a_e$  is LOW *then*  $\phi_t$  is ACCEL
- $\mathfrak{R}_6$  : *if*  $\omega_t$  is SLOW *and*  $\Delta a_y$  is LOW *and*  $a_e$  is LOW *then*  $\theta_t$  is ACCEL
- $\mathfrak{R}_7$  : *if*  $\omega_t$  is SLOW *and*  $\Delta a_x$  is LOW *and*  $a_e$  is MED *then*  $\phi_t$  is BOTH
- $\mathfrak{R}_8$  : *if*  $\omega_t$  is SLOW *and*  $\Delta a_y$  is LOW *and*  $a_e$  is MED *then*  $\theta_t$  is BOTH
- $\mathfrak{R}_9$  : *if*  $\omega_t$  is SLOW *and*  $\Delta a_x$  is MED *and*  $a_e$  is LOW *then*  $\phi_t$  is BOTH
- $\mathfrak{R}_{10}$  : *if*  $\omega_t$  is SLOW *and*  $\Delta a_y$  is MED *and*  $a_e$  is LOW *then*  $\theta_t$  is BOTH
- $\mathfrak{R}_{11}$  : *if*  $\omega_t$  is MED *and*  $\Delta a_x$  is LOW *and*  $a_e$  is LOW *then*  $\phi_t$  is BOTH
- $\mathfrak{R}_{12}$  : *if*  $\omega_t$  is MED *and*  $\Delta a_y$  is LOW *and*  $a_e$  is LOW *then*  $\theta_t$  is BOTH
- $\mathfrak{R}_{13}$  : *if*  $\omega_t$  is MED *and*  $\Delta a_x$  is not LOW *then*  $\phi_t$  is GYRO
- $\mathfrak{R}_{14}$  : *if*  $\omega_t$  is MED *and*  $\Delta a_y$  is not LOW *then*  $\theta_t$  is GYRO
- $\mathfrak{R}_{15}$  : *if*  $\omega_t$  is MED *and*  $\Delta a_e$  is not LOW *then*  $\phi_t$  is GYRO *and*  $\theta_t$  is GYRO

### 3 MEASURES FOR IMPROVING ODOMETRY

In this section we present several innovative measures for improving odometry in mobile robots in general and in planetary rovers especially.

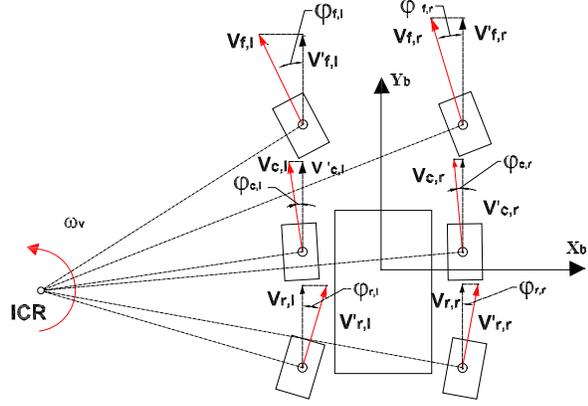
#### Redundant Encoders

Rocky or FIDO-type planetary rovers have six drive wheels equipped with encoders. Since usually only two or even just one (if operating in conjunction with a gyro) encoder(s) are needed for odometry, the six encoders on these vehicles are redundant. With redundant encoders the control system can select the single encoder or the encoder pair that produced the most accurate readings in a given sampling interval. Our expert rules-based technique for doing so is explained in detail in [Ojeda and Borenstein, 2004] and we omit the detailed explanation here.

Because of their six steerable drive wheels, planetary rovers are also over-constrained. In over-constrained vehicles the actual speeds and steering angles of the wheels have to match perfectly. Even small discrepancies will result in the wheels “fighting” each other. This phenomenon can be observed in many early Sports Utility Vehicles (SUVs) in 4-wheel drive mode on high-traction ground. However, when the SUV drives on slippery ground, such as sand, then this effect is less noticeable. This is because the wheels can slip easily to accommodate discrepancies in their speeds and steering angles. One method for reducing this effect and the associated wheel slippage in over-constrained mobile robots is to modify the control algorithm with the goal of reducing discrepancies in wheel speeds. We have developed such an algorithm, called “Cross-coupled Control.” This algorithm, originally developed by Borenstein and Koren [1987], was refined later and is described in detail in [Ojeda and Borenstein, 2004]. An alternative solution for the wheel synchronization problem was shown by Baumgartner et al. [2001], who used a voting scheme to synchronize the six wheels of JPL’s FIDO Rover.

## All-wheel Slippage Detection

The greatest enemy of odometric accuracy is wheel-slippage, and vehicles that travel on sandy surfaces are particularly at risk. If only some of the wheels are slipping while others are gripping, then the above-mentioned expert rule-based method for choosing data from multiple redundant encoders can help in selecting the most accurate source. Even if only one wheel is gripping and we manage to identify that wheel, odometry is “saved” because we can derive linear displacement from that encoder and rotation from the gyro. What is therefore of greatest concern is *all-wheel slippage* (AWS).



**Figure 5:** Nomenclature for wheel velocities in our planetary rover Fluffy.

In this section we discuss several methods for the detection of AWS conditions. The rationale is that it is generally beneficial to know that AWS has occurred (for a variety of obvious reasons), and, more interestingly, that we may be able to reduce odometry errors due to AWS, once we know that AWS has occurred.

In order to detect AWS, we developed a comprehensive set of what we call “*AWS indicators*.” The output of an AWS indicator is a binary flag that indicates that AWS has occurred. In the following sections we briefly describe the four AWS indicators that we tested. However, in our final implementation we used only the Current Indicators explained in Section 4.

### Encoder Indicator (EI)

Figure 5 shows a schematic diagram of the wheels in a planetary rover. Each wheel has a linear velocity vector  $V$  and a steering angle  $\phi$ , which is measured between the longitudinal direction of the vehicle and the steering direction of the wheel.

The projection of the speed vector  $V$  onto the  $y$ -axis is called “longitudinal velocity component,”  $V'$ .

On smooth terrain,  $V'$  of the three wheels on either side should be equal, and our hypothesis is that unequal speeds in the three wheels of a side suggest wheel slippage.

### Gyro Indicator (GI)

This method aims at detecting wheel slippage by comparing encoder data with gyro data. The motion of a rigid body in space can always be seen as pure rotation about a so-called “Instantaneous Center of Rotation” (ICR), as shown in Figure 5. The ICR may change from moment to moment. For straight-line motion the radius from the ICR to each wheel is of infinite length.

We can compute the rate-of-turn of the rover from the Front, Center, and Rear encoder pair, according to

$$\omega_e = \frac{d_l \cos \phi_l - d_r \cos \phi_r}{B T_s} \quad (18)$$

where

$d$  - Distance traveled by the left and right wheel of wheel pair

$B$  - Track width (distance between the left and right wheel of a wheel pair)

$\varphi$  - Left and right steering angle

We can now compare each of the three  $\omega_e$  with the rate-of-turn measured by the z-axis gyro, which we consider the ground truth in this approach. If no slippage occurred in a wheel pair, then one can expect good correspondence between the rates-of-turn derived from the encoders of that wheel pair and the gyro. Poor correspondence suggests wheel slippage.

#### *Accelerometer Indicator (AI)*

The AI compares encoder readings with readings from an accelerometer mounted in longitudinal direction. Since this indicator is not effective for slow-moving planetary rovers, where accelerations resulting from wheel slippage are extremely small and of poor signal-to-noise ratio, we omit further discussion of this indicator in this paper.

#### *Current Indicator (CI)*

The Current Indicator (CI) aims at detecting AWS by monitoring the electric currents in the drive motors, which are known to be roughly proportional to torques applied to the wheels. One widely accepted model for the prediction of wheel slippage on loose terrain is the so-called Coulomb-Mohr soil failure criteria [Bekker, 1956]. The Coulomb-Mohr soil failure criterion predicts that *total* wheel slippage on sandy terrain occurs when the shear stress applied to a given terrain exceeds the maximum shear stress that the terrain can bear

$$\tau_{\max} = c + \sigma_{\max} \tan \varphi \quad (19)$$

where:

$c$  – Cohesion of the soil

$\varphi$  – Internal friction angle of the soil

$\sigma_{\max}$  – Maximum normal component of the stress region at the wheel-terrain interface

Eq. (19) describes only the condition of total slippage, that is, the wheel rotates but the vehicle's forward motion is nil or negative [Reina, 2004]. On level terrain, the shear stress on loose sand for the full range of slippage may be described by an exponential function [Janosi and Hanamoto, 1961]:

$$\tau(\theta) = [c + \sigma(\theta) \tan \varphi] \left( 1 - e^{-\frac{j(\theta)}{K}} \right) \quad (20)$$

where

$j$  – Shear displacement

$K$  – Shear deformation modulus

The normal stress,  $\sigma$ , the wheel sinkage,  $z$ , and the angle  $\theta$  are related according to (see Figure 6) [Bekker, 1969; Wong, 2001]:

$$\sigma(z) = (k_c + k_\phi b) \left( \frac{z}{b} \right)^n \quad (21)$$

$$z(\theta) = r(\cos\theta - \cos\theta_1) \quad (22)$$

where:

$k_c$  – Cohesive modulus of terrain deformation

$k_\phi$  – Frictional modulus of terrain deformation

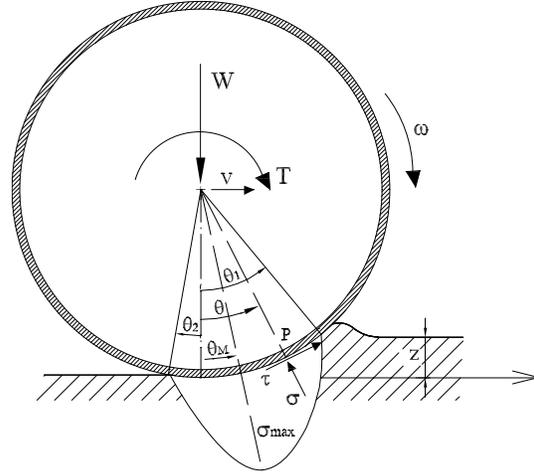
$n$  – Exponent of terrain deformation

$r$  – Wheel radius

$b$  – Wheel width

$\theta_1$  – Angle between vertical and leading edge of wheel contact patch

$\theta_2$  – Angle between vertical and trailing edge of wheel contact patch



**Figure 6:** Wheel-soil interaction model (adapted from [Bekker 1969]).

Shear displacement  $j$  is related to wheel slip  $i$  and to angle  $\theta$  according to:

$$j(\theta) = r[\theta_1 - \theta - (1-i)(\sin\theta_1 + \sin\theta)] \quad (23)$$

$$i = 1 - \frac{V}{r\omega} \quad (24)$$

$V$  – Linear speed of the wheel

$\omega$  – Angular rate of the wheel

The torque  $T$ , with which the soil resists the rotation of the wheel, can be computed as the integral of the shear stress over the contact patch with respect to  $\theta$  [Bekker, 1969]:

$$T = r^2 b \int_{\theta_2}^{\theta_1} \tau(\theta) d\theta \quad (25)$$

A method for estimating soil parameters during an actual mission was introduced by Iagnemma and Dubowsky [2000, 2004]. For the experiments described in our paper we used the soil parameters from [Wong, 2001].

One important phenomenon in frictional soil (such as dry sand) is the so-called “slip-sinkage effect” [Bekker, 1960; Soltynski, 1963]. In order to model wheel-terrain interaction accurately, this effect must be taken into account. Bekker [1960] describes the relation between slippage and sinkage as:

$$\frac{\tau_{\max}}{\sigma - p'_{crit}} = \frac{j}{z_j} \quad (\sigma > p'_{crit}) \quad (26)$$

where  $p'_{crit}$  is called the “Terzaghi bearing capacity” [Terzaghi, 1942] given by

$$p'_{crit} = cN_c + \gamma [N_q + (z_s + z_j) + 0.5bN_\gamma] \quad (27)$$

where  $N_c$ ,  $N_q$ , and  $N_\gamma$  are constants,  $\gamma$  is the density of the soil,  $z_s$  is the sinkage due to static load, and  $z_j$  is the sinkage due to slip. Total sinkage can be estimated by:

$$z = z_s + z_j \quad (28)$$

Eq. (25) correlates torque with slippage. We solved Eq. (25) numerically, based on the sand parameters provided by [Wong, 2001] and characteristics of our robot. The partially analytical

solution can be verified experimentally, by measuring momentary motor currents and correlating them with actually measured wheel slippage. In order to measure actual wheel slippage, we built an absolute position and speed sensing system, called ‘‘SAPS’’ (explained in more detail in Section 4). With the SAPS it is easy to measure the actual forward speed of the rover,  $v_{act}$ . We computed wheel slippage  $i$  by subtracting  $v_{act}$  from the nominal speed,  $v_{odo}$ , which is the angular speed of the wheel as measured by the encoders:

$$i = 1 - \frac{v_{act}}{v_{odo}}$$

The results of numerous measurements relating actual slip to motor currents as well as the numerical solution are plotted in Figure 7. Either approach shown in Figure 7 (numerical solution or experimental) can now be used to derive a slippage correction function,  $S_c(I)$ , that predicts the amount of slippage as a function of the measured motor current,  $I$ . For simplicity, we approximated  $S_c(I)$  by a straight line fitted to the experimental solution of Figure 7, as shown in Figure 8. The equation for this line is:

$$S_c = \begin{cases} \xi(I - I_{slip}) & I > I_{slip} \\ 0 & \text{otherwise} \end{cases}$$

where

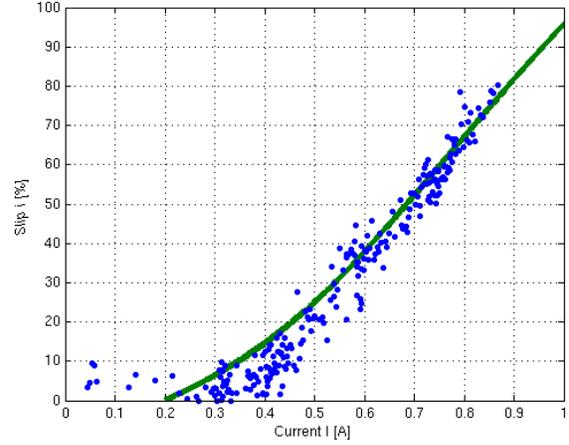
$\xi$  – Scale factor for slippage correction

The intersection of line  $S_c$  with the x-axis, denoted  $I_{slip}$ , is of great physical significance, since it represents the *onset* of wheel slippage.

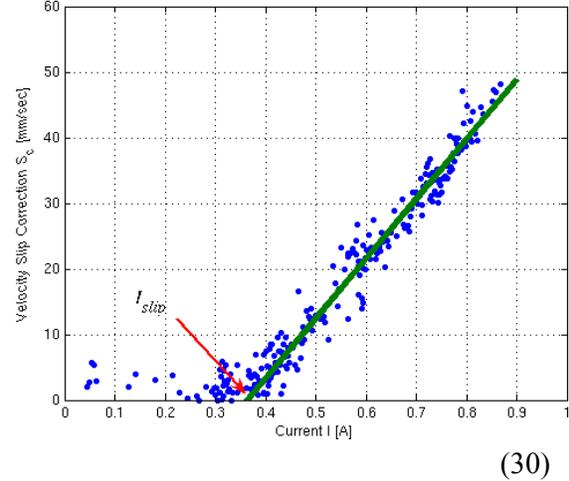
For the practical use of Eq. (30) with the Current Indicator (CI), we compute  $I$  as the average of the rear and center wheel pair motor currents (we disregard the front wheel pair entirely in our implementation of the CI because it carries only a disproportionately small load). We emphasize that, although  $I$  is computed as the average current from four drive motors, the correction (see below) is applied only if all four motor currents individually exceed  $I_{slip}$ .

As a further refinement of the CI we take into account that when the rover climbs up a slope, less of its weight acts as a force normal to the surface. As a result, slippage will occur at lower levels of torque. We account for this fact by lowering the current threshold that signifies the onset of slippage,  $I_{slip}$ , by an empirically determined factor  $C_{pitch}$ . To do so, we rewrite Eq. (30):

$$S_c = \begin{cases} \xi(I - I_{slip} - C_{pitch}\theta_r) & I > I_{slip} \\ 0 & \text{otherwise} \end{cases} \quad (31)$$



**Figure 7:** Relating wheel slip with torque and motor currents: (solid line) determined numerically, (dots) measured experimentally. (29)



**Figure 8:** Current-Slip relationship: (dots) determined experimentally and (solid line) optimally fitted linear slippage correction function  $S_c(I)$  using Least Squares. (30)

Note that in Eq. (31) the symbol that we use for the rover’s pitch angle,  $\theta_r$ , is different from the one used in Section 2 for the pitch angle. This is in order to avoid confusion with the symbol for the patch contact angle,  $\theta$ , which is widely used in the scientific literature on Terramechanics.

Once the amount of slippage is estimated according to Eq. (31), it is subtracted from the linear speed of the rover as measured by odometry during the same sampling interval:

$$v_c = v_{odo} - S_c \quad (32)$$

One should note that when a wheel climbs up a rock, the motor current may also exceed  $I_{slip}$ , even though the wheel is not slipping. However, the CI raises the AWS flag only if the currents of all four motors of the center and rear wheel pairs exceed  $I_{slip}$  in the same sampling interval.

### Discussion

All four AWS indicators discussed above were shown in our experiments to provide useable data for indicating all-wheel slippage. However, for the extremely slow-moving Mars Rover (a maximum speed of 60 mm/sec was used in our experiments because this is the envisaged speed for the 2009 mission), the benefits of the EI, GI, and AI are marginal. For this reason we omit the presentation of the somewhat inconsistent experimental data obtained with them in this paper. We focus, instead, on the CI, which, as it turned out, provides not only useful flagging of AWS, but indeed is also useful for *correcting* odometry errors incurred by AWS. We call the method described above for correcting odometry errors with values derived from momentary motor currents, “*iComp*” method.

## 4 EXPERIMENTAL RESULTS

In this section we provide comprehensive experimental results obtained with different navigation features on different terrains. To generate a ground truth reference for these experiments, we installed in our sandbox an absolute position measurement system that uses four ultrasonic receivers at the corners of the sandbox and a star-like formation of four ultrasonic transmitters mounted on our rover Fluffy. We call this system “Sonar Absolute Positioning System” (SAPS). Within the confined area of our sandbox this system provides absolute position information in real-time and with sub-centimeter accuracy. From the SAPS-derived ground truth data we can easily determine when the rover is experiencing AWS: the speed measured by the SAPS is no longer equal to the commanded speed of the rover.

For each set of experiments in this section we compute the Absolute Error. To do so, we first compute the absolute average of errors in X and Y direction, labeled  $X_e$  and  $Y_e$ , respectively. Specifically, if for a certain experiment we performed  $m$  runs and measured position errors  $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$  at the end of each run, then

$$X_e = \frac{1}{m} \sum_{i=1}^m |x_i| \text{ and } Y_e = \frac{1}{m} \sum_{i=1}^m |y_i| \quad (33)$$

The Absolute Error is then computed as

$$Error = \sqrt{X_e^2 + Y_e^2} \quad (34)$$

The same error is also expressed as a percentage of the total travel distance  $D$ :

$$Error[\%] = 100 \frac{Error}{D} \quad (35)$$

## Straight Line Experiments

The straight line experiments aimed at determining the effectiveness of CI-based slippage compensation (i.e., the iComp method). In order to perform these experiments, we set up two different terrains in our 6×4 m sandbox:

**Experiment #1a:** A single sand mound was created from loose sand only

**Experiment #1b:** Fist-sized rocks were embedded in the sand mound of Experiment #1a and at other places along the test path; all other conditions were the same. The rocks significantly reduced the amount of slippage, even for the higher mounds. The purpose of including the rocks was to create conditions, in which motor currents also surpassed  $I_{slip}$ , but without causing all-wheel slippage.

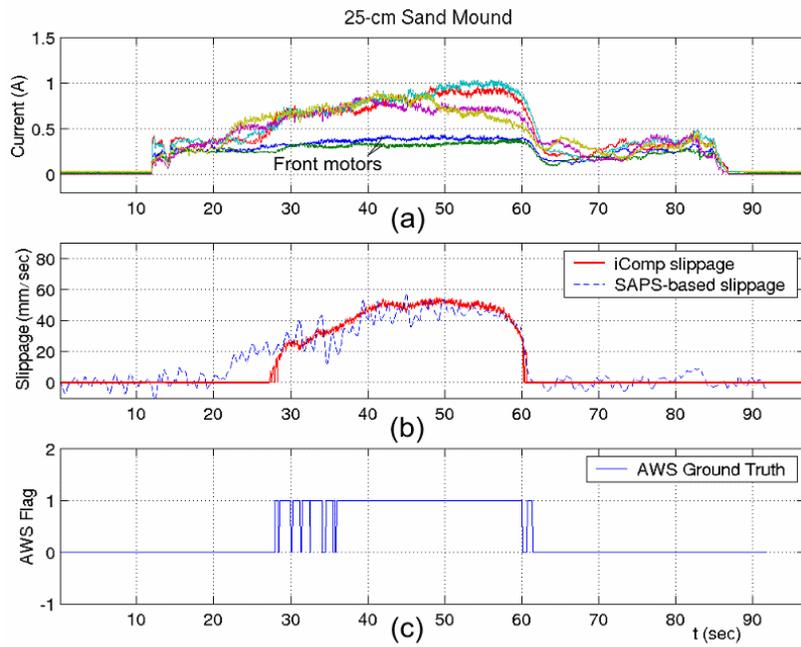
In both sets of experiments we ran the rover along a  $D = 4$  m straight path on sand at its nominal speed of 60 mm/sec. The sand mound had four different heights: 10 cm, 15 cm, 20 cm, and 30 cm. Depending on the height of the mound, this would create more or less slippage.

For the experiments on pure sand mounds, the results of a traverse is shown in Figure 9. As noted earlier, the currents of the two front drive motors (blue and green line) are always disregarded, because those wheels bear only a small portion of the rover weight. The results of the experiments are plotted in Figure 10a, and a numeric summary is shown in Table II. The circular markers in Figure 10 show runs with odometry only, i.e., without FLEXnav and without iComp. The triangular markers show runs with FLEXnav only, but without iComp, and the square markers show runs using FLEXnav with iComp. As is evident, slippage compensation yields a reduction in position errors of as much as one order of magnitude.

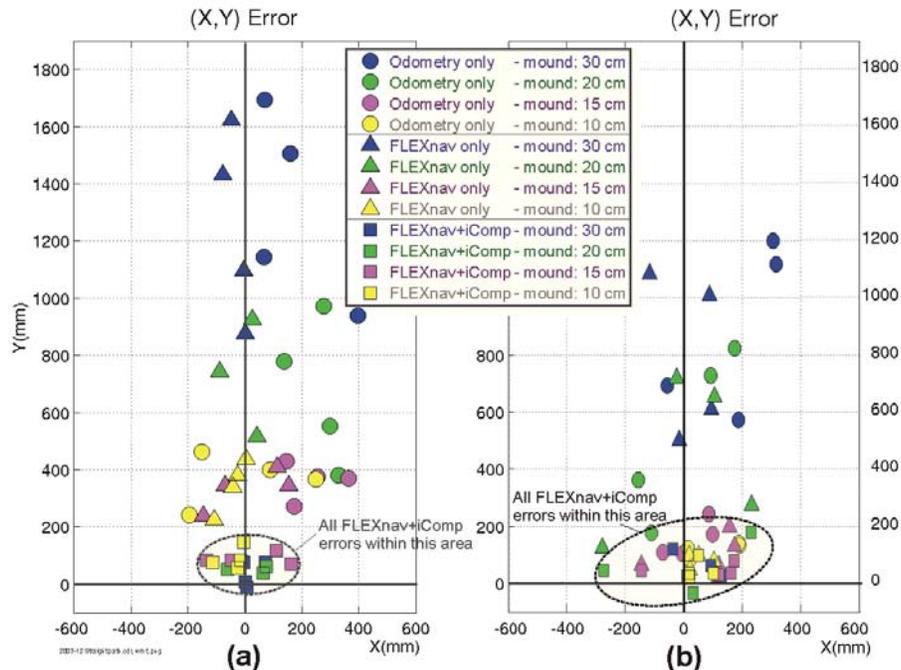
For the experiments on sand mounds with rocks, the iComp method assumes that any motor current in excess of  $I_{slip}$  is either the result of wheel slippage or of the wheel climbing up a rock. To distinguish between these two possibilities, the iComp method flags AWS only if all four weight-bearing wheels (center and rear wheel pair) simultaneously meet the condition  $I > I_{slip}$ . The results of Experiment #1b are shown graphically in Figure 10b, and a numeric summary is given in Table II. As is evident from both representations, compared with Experiment #1a, the errors for the iComp method are slightly larger, and the errors for odometry-only are smaller (because of the overall reduction in wheel slippage). Nonetheless, the iComp method still performed up to one order of magnitude better than odometry-only.

**Table II:** Summary of position errors at the end of Experiment #1, 4-m straight line path with mounds of different heights

Mound Height [mm]	Odometry only		FlexNav only		FlexNav + iComp	
	Error [mm]	Error [%]	Error [mm]	Error [%]	Error [mm]	Error [%]
<i>Experiment #1a: Pure sand mounds</i>						
300	1,322	33.0	1258	31.0	48	1.2
200	720	18.0	633	16.0	81	2.0
150	430	11.0	357	8.9	145	3.6
100	406	10.0	349	8.7	98	2.5
<i>Experiment #1b: Sand mounds with embedded, fist-sized rocks</i>						
300	715	18.0	565	14.0	111	2.8
200	921	23.0	805	20.0	88	2.2
150	539	13.0	470	12.0	174	4.4
100	169	4.2	188	4.7	156	3.9



**Figure 9:** Experiment with the iComp method on a 4-meter long straight path on loose sand. The result included the traversal of one single large sand mound 25 cm in height. The rover encountered the mound at about  $t = 20$  sec. (a) all six motor currents (the two low currents belong to front “axle” motors, which were disregarded); (b) wheel slippage estimated by the iComp method (solid line) and wheel slippage measured by the SAPS (dashed line); (c) ground truth AWS flag, as determined by the SAPS.



**Figure 10:** Experiment #1: Final positioning error for 4-meter straight runs across one sandy mound. The mound had different heights in different runs, and consisted of (a) sand-only, (b) sand with embedded rocks.

## Closed Path Experiments

We created in our sandbox three different terrains:

**Experiment #2:** Sand-covered but otherwise flat terrain with a few fist-sized rocks embedded in the sand.

**Experiment #3:** Sand-covered terrain with two large sand mounds and with fist-sized rocks embedded in the sand mounds, such that about half of each rock emerged from the sloped surface. On this terrain slippage was minimal because of the extra grip that the rocks provide.

**Experiment #4:** Sand-covered terrain with two large sand mounds and no rocks. This terrain produced a lot of slippage.

We refer to sets of multiple runs on each of the three terrains as one experiment. Each of the three experiments consisted of four runs in clockwise (cw) and four runs in counter-clockwise (ccw) direction. Running in both cw and ccw direction is very important: if a mobile robot is tuned and tested in only one direction,

then it is highly likely that the key parameters are tuned in such a way that dominant systematic error sources (such as those resulting from inaccuracies in measuring track width and wheel diameters) compensate for each other, and great accuracy is *seemingly* achieved. However, if the robot is run in the other direction, then those error sources no longer compensate for each other but rather add up, resulting in potentially large errors. A detailed analysis of this subject is given in [Borenstein and Feng, 1996]. That paper also explains that a third significant error, the error in determining the effective wheel radius, can be eliminated in a trivial fashion prior to the cw and ccw experiments. To do so, the robot is driven along a straight line. Then, when stopped, the experimenter compares the actual travel distance with the travel distance reported based on odometry. The difference between those two measurements shows the effective wheel radius error, which can then be corrected in software.

In each run Fluffy traveled autonomously along a pre-programmed, near-rectangular path. Each run consisted of three uninterrupted loops, resulting in total travel distances of  $D = 41\text{ m}$  or  $D = 43\text{ m}$  per run (depending on the terrain), and a total of 1,080 degrees of turning (see Figure 11 and Figure 12). The speed in all runs was 60 mm/sec and the rover did not stop before turning.

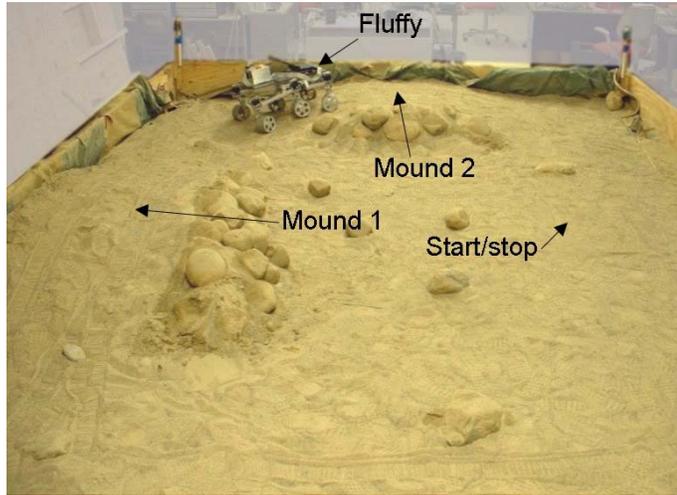


Figure 11: Fluffy driving up a steep slope (at point 'A' in Figure 12).

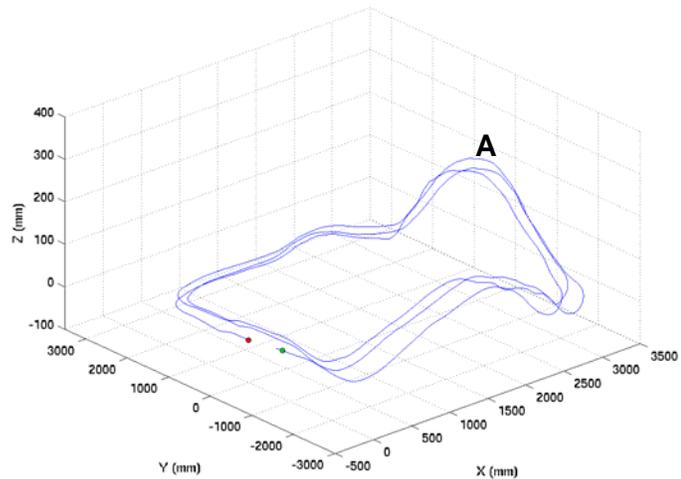


Figure 12: Actual 3-D trajectory of the rover during a typical 3-lap run under conditions of Experiment #4.

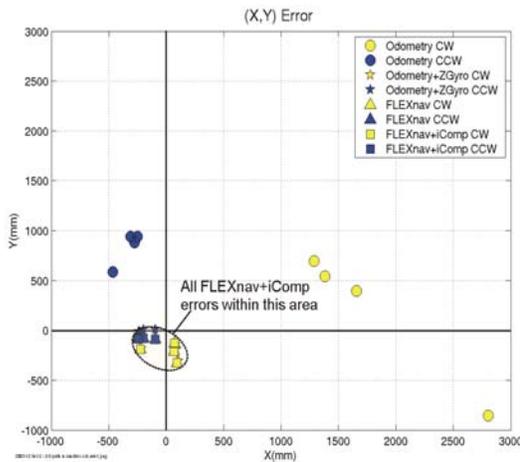
The robot started each run at a marked location (0, 0) and at the end of the run stopped at (what it “thought” was) the same position (0, 0). The discrepancy between the actual stopping position and the starting position, measured with a tape measure, is the so-called “return position error,” which is used as the final positioning error for the computation of the absolute error.

In each run we collected data from all onboard sensors. Then, in post-processing, we applied four different positioning methods to the data:

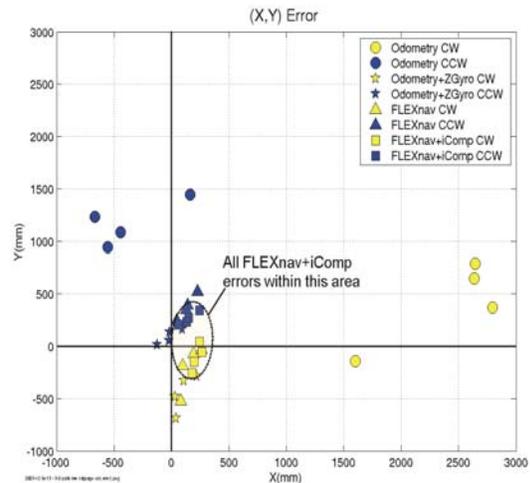
1. Odometry only
2. Odometry combined with FLEXnav
3. Odometry enhanced by a single-axis gyro (yaw)
4. Odometry combined with FLEXnav and iComp

Table III shows the results for the three experiments and the four tested positioning methods in matrix form, while Figure 13, Figure 14 and Figure 15 show the return position error for every run in graphical form. We note that in some cases odometry enhanced by a single-axis gyro reported small errors where we expected larger ones. These situations occurred when certain non-systematic (i.e., terrain-specific) errors compensated for each other.

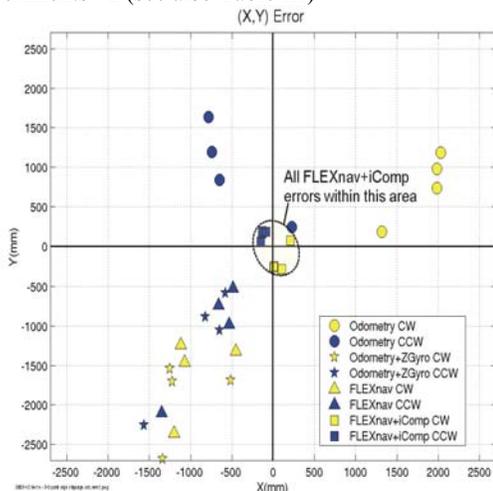
Similar experiments were conducted with a planetary rover by Baumgartner et al. [2001]. The error achieved in their system was 0.72% of total travel distance. However, they did not run the experiment in both cw and ccw direction, as we believe is necessary for conclusive results. Also, the experimental conditions described in their paper do not specifically mention any sig-



**Figure 13:** Return Position Errors plotted for Experiments #2 (see also Table III)



**Figure 14:** Return Position Errors plotted for Experiments #3 (see also Table III)



**Figure 15:** Return Position Errors plotted for Experiments #4 (see also Table III)

**Table III:** Experimental conditions and results for the three experiments described in Section 4. See Figure 13, Figure 14 and Figure 15 for detailed results.

Dir.	Odometry only		Odometry + Single-axis (yaw) gyro		FlexNav only		FlexNav + iComp	
	Error [mm]	Error [%]	Error [mm]	Error [%]	Error [mm]	Error [%]	Error [mm]	Error [%]
<i>Experiment #2: No slopes at all, flat horizontal sand. No observed wheel slippage, <math>D = 41.2\text{ m}</math></i>								
CW	897	2.2	195	0.47	212	0.51	212	0.51
CCW	1,890	4.6	242	0.59	243	0.59	243	0.59
<i>Experiment #3: Two moderate slopes with rocks embedded in the sand to minimize wheel slippage, <math>D = 43.1\text{ m}</math></i>								
CW	1,260	3.0	113	0.26	400	1.0	302	0.76
CCW	2,470	5.7	455	1.10	257	0.64	255	0.64
<i>Experiment #4: Two steep slopes made of sand only, resulting in substantial slippage, <math>D = 43.1\text{ m}</math></i>								
CW	1,150	2.7	1,500	3.50	1,323	3.0	196	0.45
CCW	1,980	4.6	2,190	5.00	186	4.3	230	0.53

nificant slopes or rocks and no photograph is provided. For this reason we assume that the conditions of Baumgartner’s experiment were not meant to promote significant wheel slippage, as is the case in our Experiment #4.

The results of odometry combined with FLEXnav and iComp on high slippage terrain are remarkable, since they show that our system performed equally well on high slippage terrain as on non-slippage terrain. This feat has not been demonstrated to date by any other dead-reckoning system.

## 5 COMPARISON OF FLEXNAV WITH KALMAN FILTERS

Since our FLEXnav method is fundamentally different from the widely used Kalman Filter (KF) method, we present in this section a comparison of the two methods. In order to compare the performance of both methods we applied them to recorded sets of raw data from the experiments described in Section 4.

We implemented the original KF algorithm, which requires a linear relationship of the state variables (i.e., linear dynamics). In this basic KF the system and measurement noise should be Gaussian. Of course there are many variations of the original KF algorithm; the two variations found most often in practical applications, especially with nonlinear dynamic systems, are the Linearized Kalman Filter (LKF) and the Extended Kalman Filter (EKF).

The Kalman filter estimates the state of the system using a linear combination of the measurement and the prior estimate of the state based on the system dynamics. This is done by using a blending factor or “gain” that is computed by the algorithm. After the Kalman Filter reaches steady state (i.e., has converged), the gain remains constant. This is the reason why in many real-time applications the gain as well as other parameters used by the algorithm can be pre-computed and stored in memory instead of being computed in real-time during operation.

Figure 16 shows the estimate of the pitch of the robot using (a) accelerometer data only; (b) gyro data only; or accelerometer data fused with gyro data using (c) a Kalman Filter or (d) FLEXnav. It is apparent that both the KF and FLEXnav produce very similar results and both eliminate the effects of drift in the gyros in a similar manner. The blending factor (gain) of the KF and the one used by FLEXnav are shown in Figure 17.

One shortcoming of the basic KF is that when noise conditions differ from those specified by the KF, the gains generated by the algorithm are not optimal and neither is the output of the KF. To illustrate this case we used a data set, in which we had observed significant wheel slippage. The slipping wheels induced vibration noise, evident in the noisy accelerometer signals in Figure 18. The corresponding gains are shown in Figure 19: the gain of the KF remains con-

stant while the gain of the FLEXnav system changes according to the instantaneous sensor readings. It is apparent from Figure 19 that FLEXnav reduces the weight of the accelerometer signals when this signal is affected by the vibrations caused by wheel-slippage.

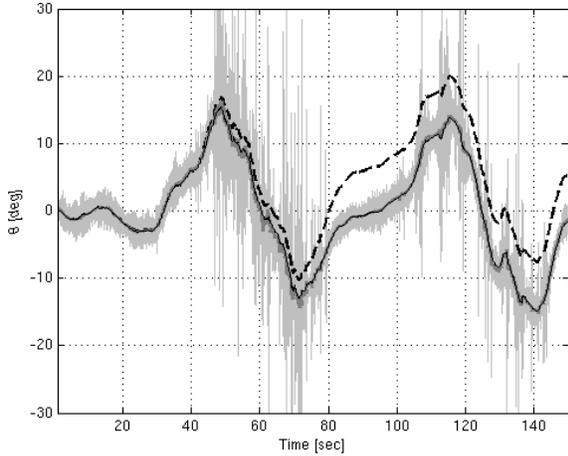
An important advantage of FLEXnav over the Kalman filter approach is that FLEXnav allows incorporating the designer’s experience into the system by means of Fuzzy Logic and Expert Rules. Fuzzy systems do not require a system model, which makes it especially useful when dealing with systems of high complexity. A side-by-side comparison of the KF and FLEXnav is shown in Table IV.

We should emphasize, though, that more sophisticated KF variants exist. Notably the EKF adjusts its gain continually, and so does, in some cases, the LKF. Indeed, the scientific literature offers other variations of the basic KF, which make those systems more robust to unexpected noise. However, it was not our intention to develop such sophisticated Kalman Filter approaches. Rather, we wanted to get a better understanding of how a conventional Kalman Filter approach handles position estimation data in comparison to our FLEXnav algorithm.

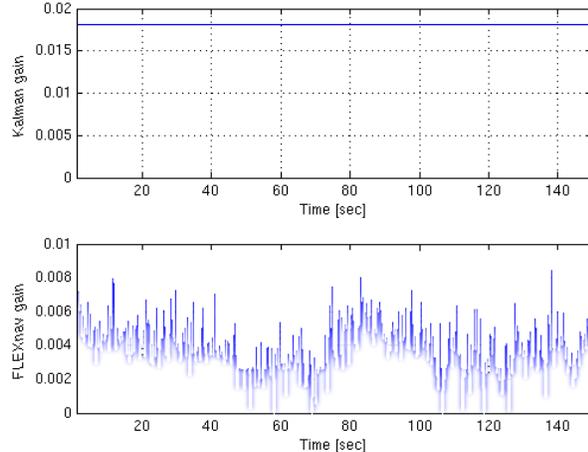
## 6 CONCLUSIONS

In this paper we introduced our Fuzzy Logic Expert Rule navigation (FLEXnav) dead-reckoning system, implemented on a planetary rover. We also presented results of extensive experimentation with the FLEXnav system and a comparison of our system with the more common Kalman Filter approach.

Besides the basic FLEXnav system, we introduced several measures aimed at improving dead-reckoning accuracy in planetary rovers. We described the function of four all-wheel slippage



**Figure 16:** Vehicle pitch during a 150-second traverse of the sand box. The two large peaks are the result of traversing two sand mounds. Noisy light gray line: raw accelerometer data; dashed black line: pitch derived from Gyro data only; dark gray line: fused data using Kalman filter; thin black line: fused data using FLEXnav.



**Figure 17:** Blending factor (gain) used by the Kalman Filter (upper plot) and by FLEXnav (lower plot). While the gain of the KF is constant, the FLEXnav gain changes continuously as a function of sensor readings.

**Table IV:** Comparison of the salient features of the basic Kalman Filter and FLEXnav.

	FLEXnav	Kalman Filter
<b>System Model</b>	Not Required	Linear System
<b>Noise Characteristics</b>	Any	Gaussian
<b>Computational requirements</b>	Medium	Low
<b>Robustness</b>	Very good	Good
<b>Ability to incorporate experience</b>	Yes	Not

(AWS) indicators that compare readings from different sensors in order to detect AWS. The most successful one, the Current Indicator, is not only able to detect AWS, but it can also correct odometry errors caused by AWS. This compensation method, called iComp, was shown to be effective on high-slippage, sandy slopes, without incurring undue errors when running on low-slippage terrain.

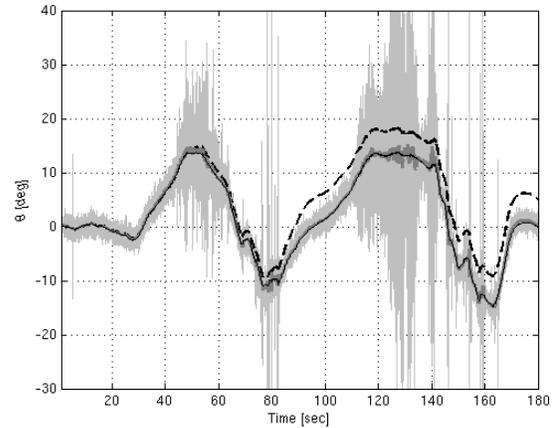
The extensive experimental results shown in this paper demonstrate that the FLEXnav system with iComp performs consistently with errors of well under one percent of distance traveled. Indeed, in the numerous runs on extremely challenging terrains and in both cw and ccw direction, the average errors were typically in the range of 0.5-0.7% of distance traveled.

#### ACKNOWLEDGEMENTS

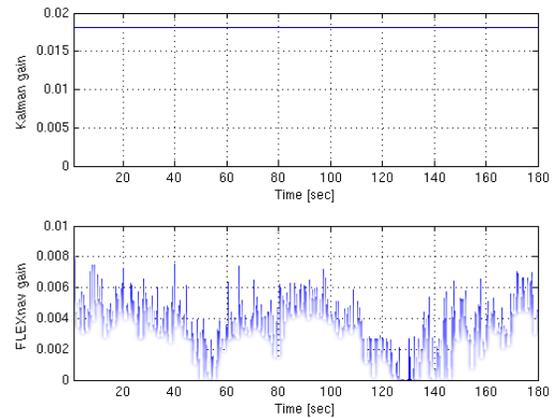
This work was funded by NASA/JPL Contract No. 1232300. The authors would also like to thank the 15 volunteering students and professionals who helped design and build our planetary rover Fluffy.

#### 7 REFERENCES

- Barshan, B.; and Durrant-Whyte; H. F., 1995, "Inertial Navigation Systems for Mobile Robots." *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, June, pp. 328-342.
- Bekker, G.; 1956, "Theory of Land Locomotion." *University of Michigan Press*, Ann Arbor, MI.
- Bekker, G.; 1960, "Off the Road Locomotion." *University of Michigan Press*, Ann Arbor, MI.
- Bekker, G.; 1969, "Introduction to Terrain-Vehicle Systems." *University of Michigan Press*, Ann Arbor, MI.
- Baumgartner, E.T.; Aghazarian, H.; and Trebi-Ollennu, A.; 2001, "Rover Localization Results for the FIDO Rover." *SPIE Photonics East Conference*, Orlando, FL, Oct., 28-29.
- Borenstein, J.; and Koren, Y.; 1987, "Motion Control Analysis of a Mobile Robot." *Transactions of ASME, Journal of Dynamics, Measurement and Control*, Vol. 109, No. 2, pp. 73-79.
- Borenstein, J.; and Feng. L.; 1996, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 6, Dec., pp. 869-880
- Iagnemma, K.; and Dubowsky, S.; 2000, "Mobile Robot Rough-Terrain Control (RTC) for Planetary Exploration." *Proc. 26th ASME Biennial Mechanisms and Robotics Conference*, Baltimore, MD, Sept. 10-13.



**Figure 18:** Vehicle pitch during a 180-second traverse of the sand box. Note the large amplitude of the noise in the second peak, caused by extensive wheel slippage. Noisy light gray line: raw accelerometer data; dashed black line: pitch derived from Gyro data only; dark gray line: fused data using Kalman filter; thin black line: fused data using FLEXnav.



**Figure 19:** Blending factor (gain) used by the Kalman Filter (upper plot) and by FLEXnav (lower plot).

- Iagnemma, K.; Kang, S.; Shibly, H.; and Dubowsky, S.; 2004, "On-Line Terrain Parameter Estimation for Planetary Rovers." *IEEE Transactions on Robotics*, October, Volume 20, Number 2, pp. 921-927
- Jang, J.-S.; Sung, C.-T. and Mizutani, E.; 1997, "Neuro-Fuzzy and Soft Computing: A computational Approach to Learning and Machine Intelligence." *Matlab Curriculum Series*.
- Janosi, B.; and Hanamoto, B.; 1961, "Analytical Determination of Drawbar Pull as a Function of Slip for Tracked Vehicles in Deformable Soils." *Proc. of the First Int. Conf. On Terrain-Vehicle Systems*, Edizioni Minerva Tecnica, Torino.
- Kelly, A.; 1995, "An Intelligent, Predictive Control Approach to High-Speed Cross-Country Autonomous Navigation Problem." *Ph.D. Thesis. The Robotics Institute, Carnegie Mellon University*, Pittsburgh, PA.
- Krantz, D.; and Gini, M.; 1996, "Non-Uniform Dead-Reckoning Position Estimate Updates." *Proc. of the IEEE Int. Conf. on Robotics and Automation Minneapolis*, MN, April 22-25, pp. 2061-2066.
- Fuke, Y.; Krotkov, E.; 1996, "Dead reckoning for a lunar rover on uneven terrain." *Robotics and Automation, Proc. IEEE Int. Conf.*, Vol. 1 , April 22-28 , Pages:411 - 416.
- Ojeda, L.; Chung, H.; and Borenstein, J.; 2000, "Precision-calibration of Fiber-optics Gyroscopes for Mobile Robot Navigation." *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, pp. 2064-2069.
- Ojeda, L.; and Borenstein, J.; 2002, "FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain." *Proc. IEEE Int. Conf. on Robotics and Automation*. Washington DC, USA, 11 - 15 May, pp. 317-322.
- Ojeda, L.; and Borenstein, J.; 2004, "Methods for the Reduction of Odometry Errors in Overconstrained Mobile Robots." *Autonomous Robots (Journal)*, Vol 16, pp. 273-286.
- Rehbinder, H.; Hu, X.; 2001, "Drift-free attitude estimation for accelerometer rigid bodies." *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, May 21-26, pp. 4244-4249.
- Reina, G.; 2004, "Rough Terrain Mobile Robot Localization and Traversability with Applications to Planetary Explorations." *PhD Thesis, Politecnico of Bari*, Italy.
- Terzaghi, K.; 1942, "Theoretical Soil Mechanics." *Wiley*, New York, NY
- Tunstel et al.; 2002, "FIDO Rover System Enhancements for High-Fidelity Mission Simulations." *Proc. 7th Int. Conf. on Intelligent Autonomous Systems (IAS-7)*, Marina del Rey, CA, Mar 25-27, pp. 349-356.
- Tonouchi, Y.; Tsubouchi, T.; and Arimoto, S.; 1994, "Fusion of Dead-reckoning Positions with a Workspace Model for a Mobile Robot by Bayesian Inference." *Int. Conf. on Intelligent Robots and Systems*, Munich, Germany, September 12-16, pp. 1347-1354.
- Wong, J.; 2001, "Theory of Ground Vehicles." *Wiley-Interscience*, New York, NY.