IEEE Aerospace Conference 2004, Big Sky, MT, March 6-13, 2004, Paper ID: 1126

Experimental Results from FLEXnav: An Expert Rule-based Dead-reckoning System for Mars Rovers[1,2]

Lauro Ojeda, Giulio Reina*, and Johann Borenstein
University of Michigan
Advanced Technologies Lab
1101 Beal Ave, Ann Arbor, MI 48108
Ph.: 734-763-1560
lojeda@umich.edu; giulio.reina@unile.it; johannb@umich.edu

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 Mars 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 Mars Rover clone Fluffy and operating in a Mars-like environment. The paper also introduces new methods for wheel slippage detection and correction, along with preliminary experimental results.

TABLE OF CONTENTS

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 [1][2][3] to predict the behavior of sensor components. We believe that this approach is not ideal for odometry, because the statistical models can't represent well single "catastrophic" events – and the encounter of a large bump or rock is indeed catastrophic for an odometry-based system.

For many years UM's Mobile Robotics Lab has been developing an approach that favors in-depth physical understanding of sensors and their associated error sources over the statistics-based Kalman Filter methods. Our basic philosophy is that many error mechanisms can be defined more specifically and accurately by expert reasoning. The result of our effort is a proprioceptive position estimation system called Fuzzy Logic and Expert rule-based navigation system (FLEXnav).

In addition to our unique 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 vehicle with a rocker-boogie suspension system. For simplicity, we will refer to mobile platforms of this design as "Mars Rovers" throughout this paper. Well-known Mars Rovers are Rocky 8 and FIDO – all developed at the Jet Propulsion Lab (JPL – see, for example, [4]). The accuracy-enhancing measures currently employed in our system are listed below:

I. Redundant Encoders – Odometric measurements typically require encoder data from one wheel on each side of the

[1] 0-7803-8155-6/04/$17.00 ©2004 IEEE

[2] IEEEAC paper #1126, Version 3, Updated November 25, 2003

* Giulio Reina is a Ph.D. student at the Politecnico of Bari, Italy. He contributed to this work during a yearlong stay at the University of Michigan as a visiting scholar.

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 (AWSD) – 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 an AWSD flag is raised.

III. Fuzzy Logic Encoder Compensation (FLEC) – The FLEC methods aims at extracting quantitative information about the extent of wheel slippage from the All-wheel Slippage (AWS) indicators, and then uses that information to reduce odometry errors that were caused by AWS.

In order to develop our system and validate its performance, we built “Fluffy” – a kinematic equivalent to JPL’s Rocky-8 and FIDO Mars Rovers. Experimental results from our FLEXnav system implemented on Fluffy and driving through a Mars-like, 3-D landscape are presented in Sections 3. Section 4 introduces further enhancements to the basic FLEXnav system, and presents experimental results.

2. The FLEXnav System

A block diagram of the FLEXnav sensor suite, as implemented on Fluffy, is shown in Figure 1. The system comprise of these components:

1× KVH E-core RA2100 fiber optic gyro with analog output. This gyro is used for measuring yaw rate.

2× KVH E-core RA1100 fiber optic gyro, with analog output. These gyros are used to measure roll and pitch rates.

2× ICSensors model 3140-002 accelerometers, used for measuring tilt angles.

The sensor system is mounted in an aluminum case on top of Fluffy. Fluffy is equipped with six wheel encoders, six steering potentiometers, and potentiometers for measuring all relevant angles within the rocker-boogie mechanism and between the rocker-boogie mechanism and the chassis.

A detailed description of our method for attitude and position estimation within the FLEXnav system is presented in [5]. That paper also explains in detail how data from the different sensor modalities of the basic FLEXnav system is fused using expert rules and fuzzy logic. In order to avoid duplication, we refer the reader to that paper rather than repeating this material here.

Instead, we focus in this paper on novel methods for improving the dead-reckoning accuracy of planetary rovers, and especially Mars Rovers. We also present comprehensive experimental results from the basic FLEXnav system as well as from FLEXnav combined with some of the dead-reckoning improvements discussed in Sections 3 and 4.

Experimental results with the basic FLEXnav system

In order to evaluate the performance of the basic FLEXnav system installed on Fluffy, we conducted three sets of experiments. Table I summarizes the features of each set.

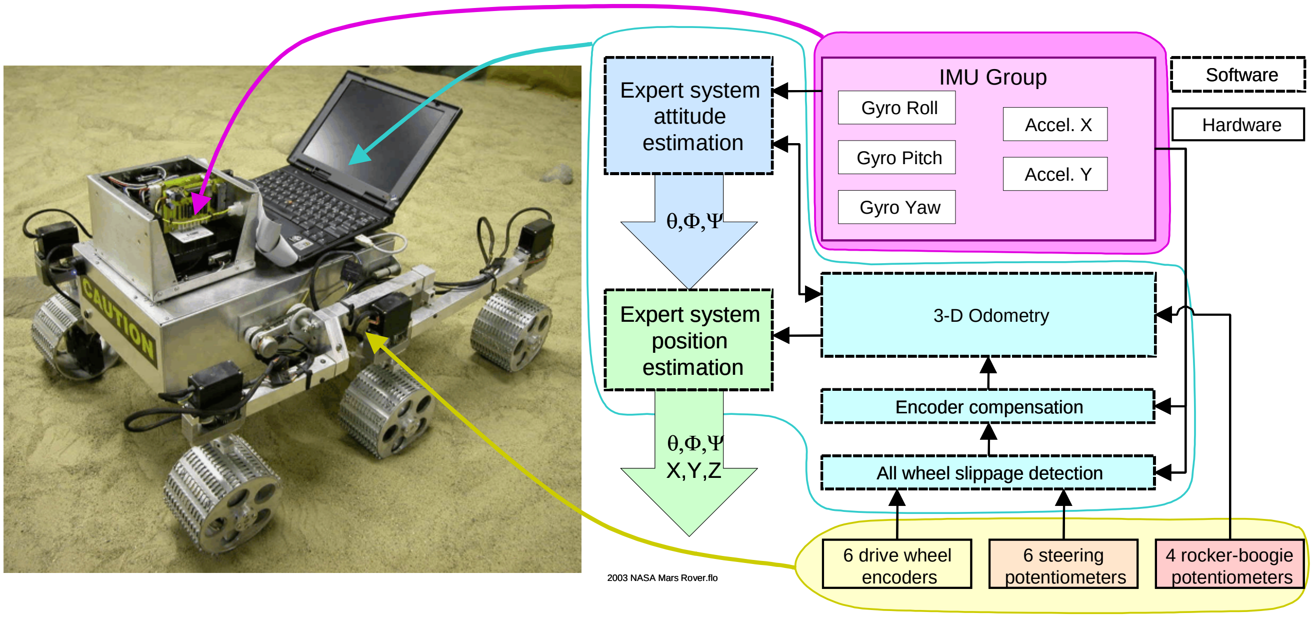

Figure 1: The University of Michigan-built Fluffy and a block diagram of the FLEXnav dead-reckoning system.

Table I: Experimental conditions and results for the three sets of experiments described in this section. See Figure 3 for detailed results. Note: “cw” stands for “clockwise” and “ccw” stands for “counter-clockwise.”

Experiment		**Set #0**	**Set #1**	**Set #2**
Terrain features		No slopes at all, flat horizontal sand. A few fist-size rocks. No observed wheel slippage	One elevation of ~30 cm, some observed wheel slippage, a few rocks.	Two steep slopes 15 and 35 cm height, a few fist-sized rocks. Substantial slippage. See Figure 2and Figure 3.
Number of runs		4×cw, 4×ccw	6×cw, 6×ccw	6×cw, 6×ccw
Total travel per run, *D*		43 m	45 m	40 m
Average of absolute errors. *E* and *E*[%]	ccw	(X_e, Y_e) = (14.4 cm, 7.2 cm) *E* = 16.1 cm; ; *E*[%] = 0.37%	(X_e, Y_e) = (44.2 cm, 17.6 cm) *E* = 47.6 cm; *E*[%] = 1.06%	(X_e, Y_e) = (19.6 cm, 38.6 cm) *E* = 43.3 cm; *E*[%] = 1.08%
	cw	(X_e, Y_e) = (23.1 cm, 13.3 cm) *E* = 26.7 cm; *E*[%]= 0.63%	(X_e, Y_e) = (26.6 cm, 40.9 cm) *E* = 48.8 cm; *E*[%]= 1.08%	(X_e, Y_e) = (9.0cm , 72.2 cm) *E* = 72.8 cm; *E*[%] = 1.82%

In all three experiments Fluffy was driven at 6 cm/sec under human remote control along a near-rectangular path in our indoor sandbox. Each run consisted of three uninterrupted loops, resulting in a total travel distances of $D = 40$ to 45 meters per run, and a total of 1,080 degrees of turning. Running in both clockwise (cw) and counter-clockwise (ccw) direction is extremely 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 error sources 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 huge errors. A detailed analysis of this subject is given in [6].

In each run the robot started at a marked location (0, 0) and, after three laps, was returned exactly to that location. Upon return to the starting position we measured the discrepancy between the actual robot position and the position reported by the FLEXnav system. The resulting discrepancies are the so-called “return position errors.” For each experiment Figure 4 shows a plot of the *x*- and *y*-values of the return position error for each individual run.

Table I also lists for each experiment and direction (cw and ccw) a set of *average of absolute errors,* X_e and Y_e. For example, if for a certain experiment we performed *n* runs and measured return position errors $(x_1, y_1), (x_2, y_2)\ldots (x_n, y_n)$ at the end of each run, then,

$$X_e = \frac{1}{n}\sum_{i=1}^{n} |\, x_i \,| \text{ and } Y_e = \frac{1}{n}\sum_{i=1}^{n} |\, y_i \,| \qquad (1)$$

Lastly, Table I also shows the absolute error, $E = \sqrt{X_e^2 + Y_e^2}$, for each of the three experiments and the same error expressed as a percentage of total travel distance *D*, *E*[%] = 100 *E/D*.

Similar experiments were conducted with a Mars Rover by Baumgartner et al. [3]. 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 significant slopes or rocks and no photograph is provided. For this reason we assume that the conditions were similar to those of our Set #0, or perhaps somewhere between our Set #0 and Set #1.

In Section 4 we describe a new experiment under the exact same conditions as Experiment #2, but this time with our method for slippage correction. We will compare the results of that experiment with the results of Experiment #2 as reported here.

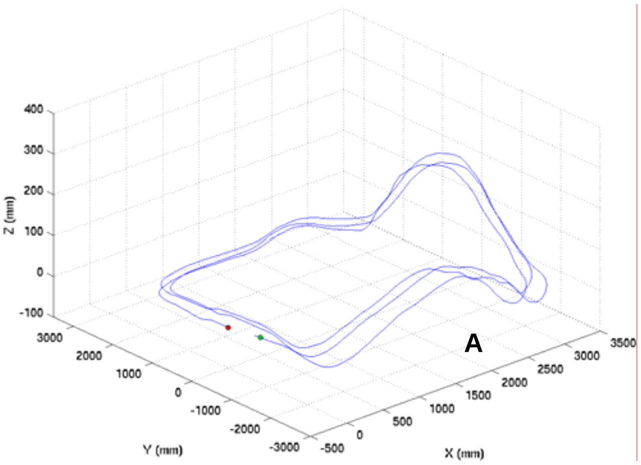

Figure 2: Actual 3-D trajectory of the robot during a typical 3-lap run under conditions of Experiment #2.

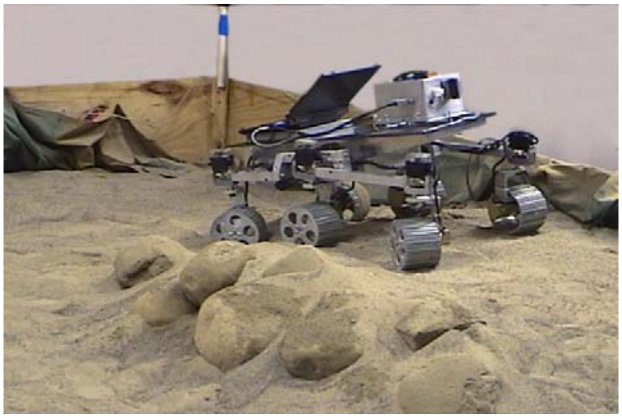

Figure 3: Fluffy driving up a steep slope (at point ‘A’ in Figure 2).

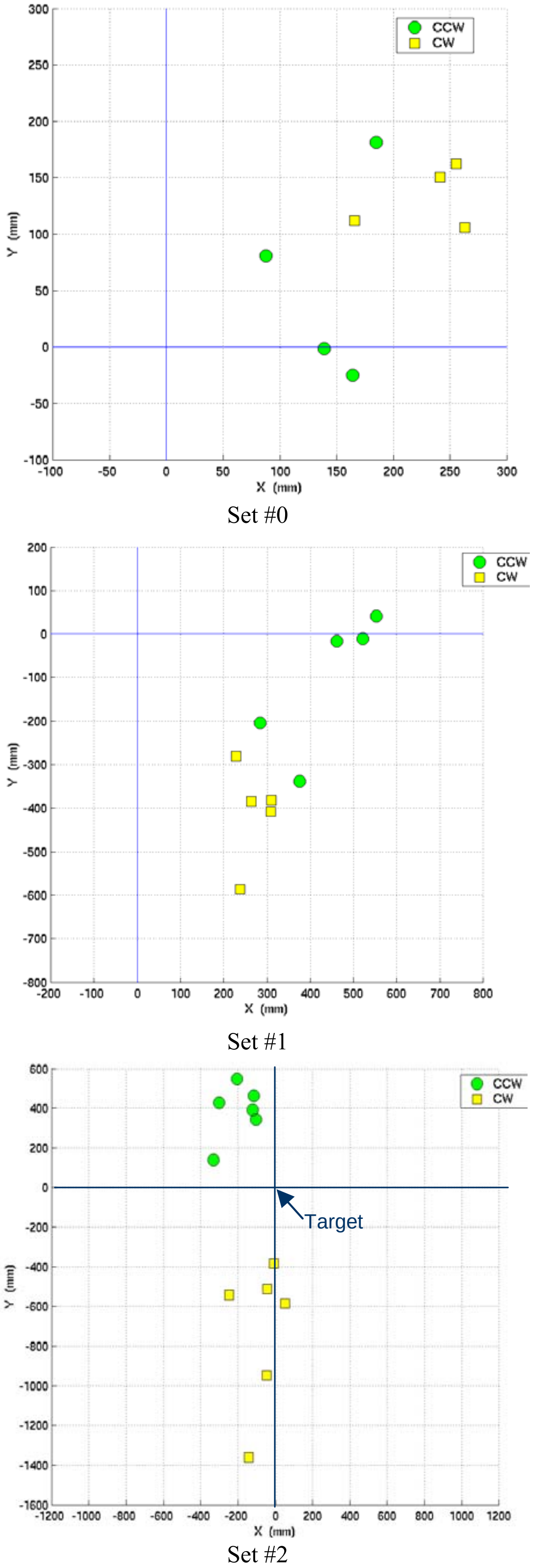

Figure 4: Return Position Errors plotted for the three sets of experiments.

3. Measures for improving odometry

In this section we present several innovative measured for improving odometry in mobile robots in general, and in Mars Rovers especially.

Redundant encoders and over-constrained drive systems

Rocky or FIDO-type Mars 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 Mars Rovers are redundant. With redundant encoders the control program 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 [7] and we omit the detailed explanation here.

Because of their six steerable drive wheels, Mars 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 [8], was refined later on and is also described in detail in [7]. An alternative solution for the wheel synchronization problem was shown by Baumgartner et al. [3], 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).

In this section we discuss several methods for the detection of AWS conditions. The rational is that it 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 can be a binary flag that indicates that AWS has occurred, or it can be a fuzzy quantity that expresses our certainty that AWS occurred. Fuzzified outputs from multiple indicators can be combined through fuzzy logic.

The most effective AWS indicators we found are:

Encoder Indicator (EI) – compares encoder readings with each other.

Gyro Indicator (GI) – compares encoder readings with those of the gyro that measures rate-of-turn around the z-axis.

Current Indicator (CI) – monitors motor currents, which are roughly proportional to the external torque applied to each wheel.

Acceleration Indicator (AI) – compares encoder readings with readings from an accelerometer mounted in longitudinal direction. Since this indicator is not effective for slow-moving Mars 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 the remainder of this section we discuss each indicator in some detail and offer experimental results.

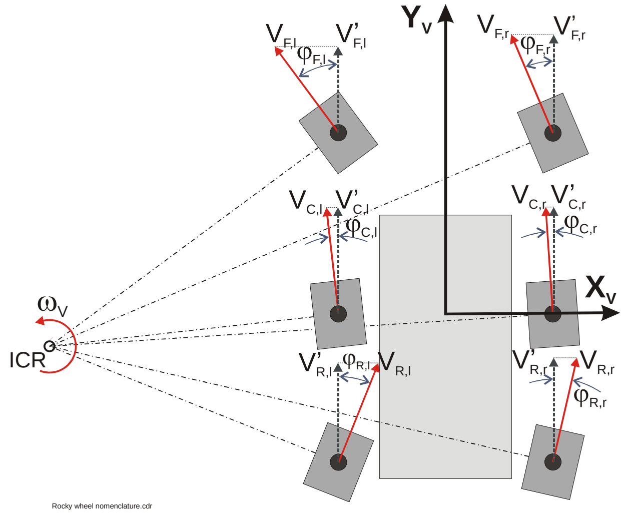

Figure 5: Nomenclature for wheel velocities in Mars Rovers

Encoder Indicator (EI) – Figure 5 shows a schematic diagram of the wheels in a Mars Rover. A coordinate system is attached to the vehicle so that the Y-axis of the vehicle coordinate system is aligned with the vehicle's longitudinal direction.

Each wheel *i,j* has a linear velocity vector $V_{i,j}$ and a steering angle $\varphi_{i,j}$ (index i = Front, Center, or Rear, and index j = left or right). $\varphi_{i,j}$ is measured between the longitudinal direction of the vehicle Y_V and the steering direction of the wheel. The projection of the speed vector $V_{i,j}$ onto the Y-axis is called "longitudinal velocity component" $V'_{i,j}$.

On smooth terrain the speed vector $V'_{i,j}$ 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. In order to express this hypothesis mathematically, we set up a fuzzy logic framework that computes the degree of confidence we have in the wheel encoders to be slippage free. Table II shows the Fuzzy Logic rules used for this purpose; this table must be applied for both the right and the

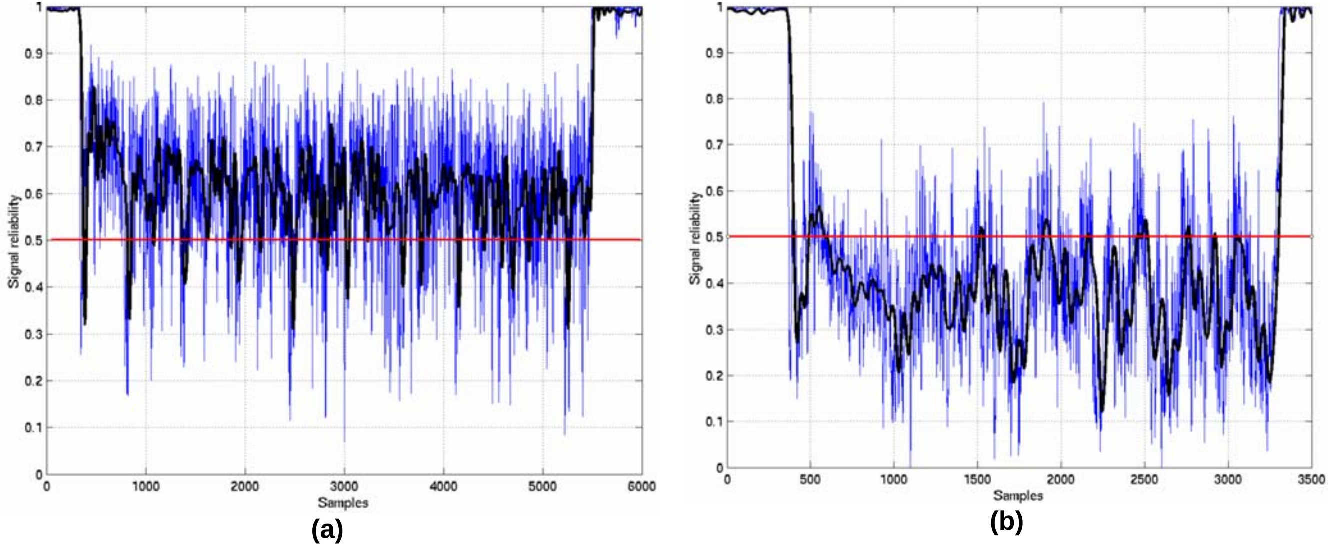

Figure 6: Output of the fuzzified Encoder Indicator for different terrains (blue lines). The black lines show the smoothed output. a. High traction terrain, no slippage. The AWS flag was raised 23% of the time. b. Sloped, sandy terrain causing lots of slippage. The AWS flag was raised 82% of the time. For binary output, the AWS flag is raised for output below 0.5.

Table II: Fuzzy Logic rules for the Encoder Indicator

Rule	**Input:** Difference in longitudinal speeds between			**Output:** Confidence in encoder reading (low = wheel slipping; high = wheel gripping)		
#	**Front-Center**	**Front-Rear**	**Center-Rear**	**Front**	**Center**	**Rear**
1	Small	Small	Small	High	High	High
2	Small	Small	Large	High	Med.	Med.
3	Small	Large	Small	Med.	High	Med.
4	Small	Large	Large	Med.	Med.	Low
5	Large	Small	Small	Med.	Med.	High
6	Large	Small	Large	Med.	Low	Med.
7	Large	Large	Small	Low	Med.	Med.
8	Large	Large	Large	Low	Low	Low

Table III: Fuzzy Logic rules for Gyro Indicator

Rule	**Input:** $\left|\omega_{\mathrm{Enc,i}} - \omega_{\mathrm{Gyro}}\right|$			**Output:** Confidence in wheel-pair encoder reading (low = wheel slipping; high = wheel gripping)		
#	**Front**	**Center**	**Rear**	**Front**	**Center**	**Rear**
1	Small	Small	Small	High	High	High
2	Small	Small	Large	High	High	Low
3	Small	Large	Small	Med.	Low	Med.
4	Small	Large	Large	Low	Low	Low
5	Large	Small	Small	Low	High	High
6	Large	Small	Large	Low	Med.	Low
7	Large	Large	Small	Low	Low	Low
8	Large	Large	Large	Low	Low	Low

left side of the rover. Figure 5 shows the output of the fuzzy logic system of the Encoder Indicator for (a) a high-traction and (b) a high-slippage terrain.

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 a 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 vehicle from the Front, Center, and Rear encoder pair, according to

$$\omega_{Enc,i} = 2\pi \frac{d_{i,j} \cdot cos\varphi_{i,j} - d_{i,j} \cdot cos\varphi_{i,j}}{bT} \qquad (2)$$

Where

$d_{i,j}$ - distance traveled by the left and right wheel of wheel pair *i*

b - distance between the left and right wheel of a wheel pair

T - sampling interval

We can now compare each of the three $\omega_{Enc,i}$ 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.

Table III shows the output of the GI fuzzy system. We have not yet begun developing fuzzy logic rules that allow combining the output of the GI with that of the EI. For now we are only looking at a binary output in the form of what we call the "AWS flag": when the fuzzy logic output is below a certain threshold (thus indicating a high likelihood for AWS), the AWS flag is raised.

Results can be further improved by taking into account the momentary pitch of the robot. When the pitch angle is large, then there is a greater likelihood that wheels will be slipping. We use this simple expert rule to modify the threshold that determined when the AWS flag is raised.

We defer the presentation of experimental results for the GI and the "GI + expert" indicator to Figure 9 (further below), where these results will be shown both separately and in combination with those of the Current Indicator, which is discussed next.

Current Indicator (CI) – The Current Indicator (CI) aims at detecting AWS through the use of the well-established physical model of rover-terrain interaction mechanics. Wheel-terrain interaction has been shown to play a critical role in rough terrain mobility (Bekker, 1956; Wong, 1993).

Wheel slippage on sandy terrain occurs when the shear applied to a given terrain exceeds the maximum amount of shear that terrain can bear, defined according to the Coulomb-Mohr soil failure criteria (see Figure 7) as:

$$F_{Max} = A \cdot \left(c + \sigma_{Max} \cdot tan\varphi\right) \qquad (3)$$

where:

c, φ – characterize the behavior of the terrain and are defined as cohesion and internal friction angle, respectively

A – wheel contact patch, which is a function of wheel geometry and of the weight acting on the wheel

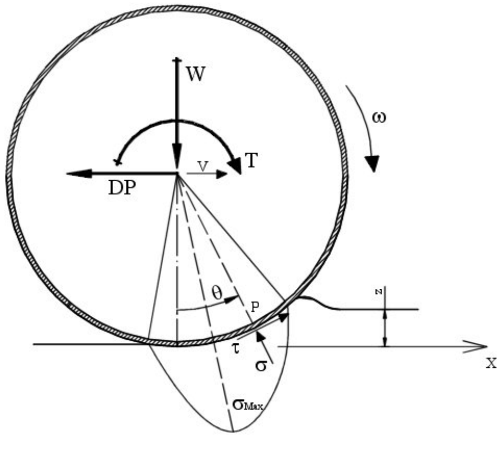

Figure 7: Wheel-soil interaction model (adapted from [9]).

σ_{Max} – is the maximum normal component of the stress region at the wheel-terrain interface (Figure 7).

For Mars-like sand, the values for the soil parameters can be found in [10]: $c = 1$ kPa and $\varphi = 30°$.

Equation (3) defines the maximum force (and thus torque) that the sandy ground can apply to the wheel while the wheel is still gripping. Since the current consumption of a motor is proportional to the applied torque, it follows that there is a maximum current, I_{slip}, that signifies the onset of wheel slippage.

We determined $I_{slip,i}$ empirically for each wheel pair i. To account for the relatively large variations in the relationship between wheels slippage and motor current, we defined a "red zone" for motor currents. The red zone has a width of ±$\Delta I = 20\%$ and is centered around $I_{slip,i}$. Thus, the condition

$$I_{slip,i} - \Delta I < I_{i,j} < I_{slip,i} + \Delta I \quad (4)$$

where $I_{i,j}$ is the current absorbed by the motor of wheel i,j *s*uggests possible wheel slippage. If and only if all six wheels meet condition (4) is the AWS flag raised. We should note that motor currents higher than $I_{slip,i} + \Delta I$ *are possible, for example when a wheel tries to rolls up a rock.*

The approach above is quite simplistic, while the true relationship between motor currents and wheel slippage is complex and not well researched. However, the purpose of our work was to develop a framework for different AWS indicators, and for that our less accurate empiric approach is sufficient. As in the case of the GI, we have not yet developed a set of fuzzy logic rules that allow combining the output of the CI with either that of the GI or that of the EI.

Experimental results with the Gyro and Current Indicators

In order to test the effectiveness of the GI and CI we conducted the experiment shown in Figure 8. Fluffy was commanded to follow a 4-meter straight path on sand at a speed of 6 cm/sec. On that path we had created two sand mounds, each about 20 cm high. The original sensor signals and the output of the indicators for one such run are plotted in Figure 9.

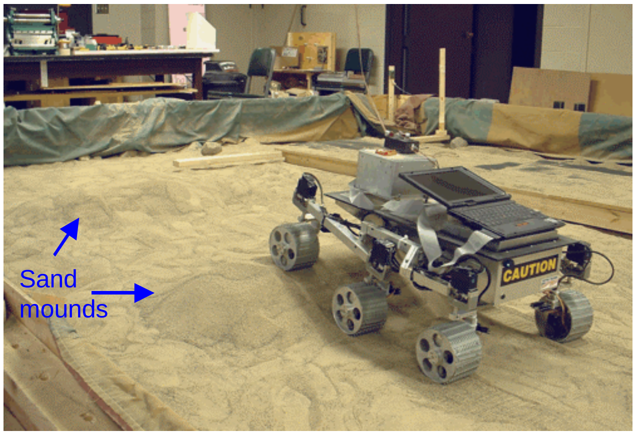

Figure 8: A 4-meter path with two sand mounds was used to test the Gyro and the Current Indicators.

Figure 9a shows the sensor signals used in the GI. The black line is the ground truth, provided by the gyro. The blue, green, and cyan lines are the rates-of-turn computed by the front, center, and rear encoder pairs, respectively. When the output of the fuzzy logic engine of Table III goes below a certain threshold, then the gyro-based AWS flag is raised. This is shown by a magenta dot in Figure 9c. Because many of the dots are very close to each other, they may look like a solid line.

Figure 9b shows the currents measured in the left motor of the center axis. Whenever that current is in or above the shaded area, the condition of Eq. (4) is met and slippage of that wheel is likely. When Eq. (4) is met for all six motors, then the current-based AWS flag is raised. This flag is shown as the blue dots in the bottom graph.

Installed in our sandbox is an absolute position measurement device that uses four ultrasonic receivers at the corners of the sandbox and a star-like formation of four ultrasonic transmitters mounted on Fluffy. Within the confined area of our sandbox this system provides absolute position information in real-time and with sub-centimeter accuracy. From this ground truth data we can easily determine when the rover was experiencing AWS: the speed measured by the absolute position sensor is no longer equal to the nominal speed of the rover. When this condition is detected, the ground truth AWS flag is raised, as indicated by the continuous red line in Figure 9c.

We can thus compare the accuracy of the GI and the CI AWS flags to the ground truth flag. For this experiment we found that the GI flagged AWS correctly 38% of the time. The CI flagged AWS correctly only 18% of the time. When the two flags were logically OR-ed, the indicators were correct 51% of the time. The percentage of false positives (warning of AWS when there actually was no AWS) was 10%.

As mentioned above, we did not yet combine the Encoder Indicator with the GI and CI. We believe that this combination and some tuning of the parameters will produce significantly better results than those presented above.

4. FUZZY LOGIC ENCODER COMPENSATION

If AWS Indicators can tell us that there is all-wheel slippage, then perhaps they can also provide some indication about the extent of slippage. Our philosophy behind this question is this: If a rover detects all-wheel slippage according to the methods described above, then the control software has three options:

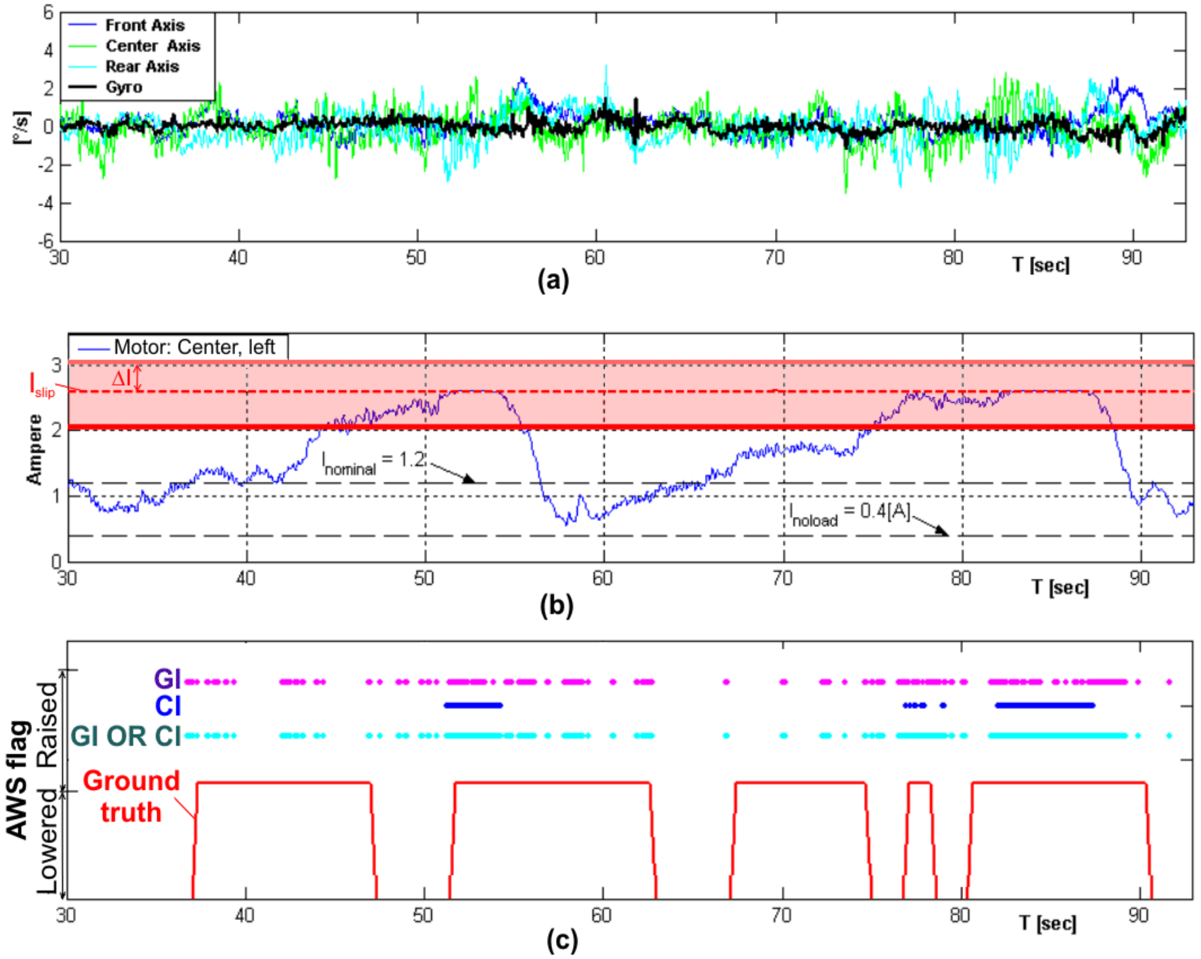

Figure 9: Effectiveness of the Gyro and Current indicators during the 4-meter traverse of the sand mounds in Figure 8. a. Rover rotation computed from front (blue), center (green), and rear (cyan) axis encoder pairs, as well as from the gyro (black); b. status of the AWS flag based on gyro indicator (magenta), current indicator (blue), and the Gyro OR current indicator (cyan). The red line shows ground truth.

1. Discontinue odometry, stop motion, and void its proprioceptive position estimation, because it is rendered useless through the slippage.

2. Continue odometry as usually, disregarding the fact that wheels were slipping and that the odometry data was faulty.

3. Continue odometry but make adjustments using all available knowledge.

We believe Option 3 is the best choice, because we think that knowledge of vehicle tilt, together with the quantitative analysis of some of the AWS indicators, will produce better results than Options 1 and 2. For example, it stands to reason that if AWS is flagged while going uphill, then actual vehicle speed will be less than nominal. That is, even though encoder pulses come in at their nominal rate, each pulse now represents less forward motion than under nominal conditions. Thus, a more accurate estimate can be produced if slippage-induced encoder inaccuracies are compensated for by software, based on knowledge of tilt and estimates of slippage.

In order to test this approach we used the fuzzy logic inference engine for the Encoder Indicator discussed in Section 3, according to Table II and Figure 5. We call the quantified output of the Encoder Indicator "Encoder Trustworthiness" (ET). The ET value is normalized between 0 and 1 and higher values suggest that we trust the output to be free of AWS.

The ET can then be applied to the encoder data that was corrupted by all-wheel slippage. In practice, this is done by defining a compensation function:

$$D^* = D_e C_F \tag{5a}$$

With

$$C_F = \begin{cases} 1 & for\ \ ET > 0.5 \\ 1 + SIGN \cdot ET \cdot K & for\ ET \leq 0.5 \end{cases} \tag{5b}$$

where

*D** – corrected travel distance

De – travel distance reported by encoder

CF – correction factor

SIGN – determined by the pitch angle of the robot: if pitch is positive, then SIGN is negative

K – empirically determined constant

Figure 10 illustrates this process in its most general form, that is, with all indicators contributing to the encoder compensation. Currently, however, only the EI is contributing.

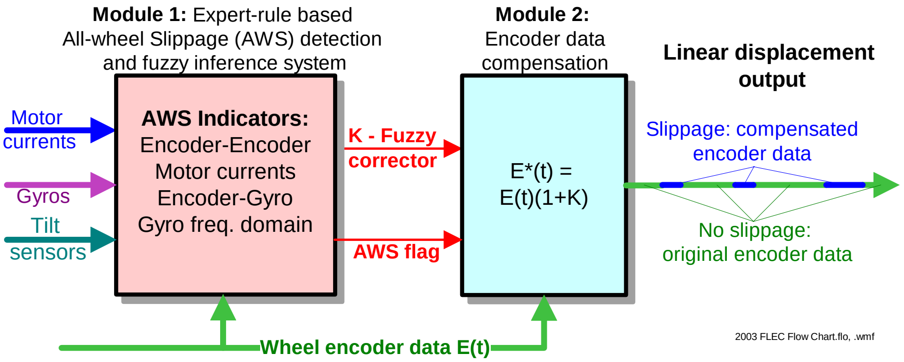

Figure 10: Ultimately envisaged implementation of the Fuzzy Logic Encoder Compensation (FLEC) method. Currently only the EI is being used for encoder compensation.

In order to test the utility of this approach, we re-ran Experiment #2 as described in Section 2, under the exact same conditions as described in Table II.

A plot of the return position errors is shown in Figure 11. Table V shows the quantitative results of this experiment, labeled "Set #3 (with FLEC)" compared against the results of the original Experiment #2 (without FLEC). A is evident from these results, FLEC reduced errors by about 40% in the high-slippage experiment.

5. Conclusions

In this paper we presented position estimation results from our Fuzzy Logic Expert Rule navigation (FLEXnav) dead-reckoning system, implemented on a Mars Rover. According to theses results our system is at least equal in performance to a comparable, Kalman Filter-based navigation system described in the scientific literature. It is impossible to make a more accurate comparison because the experimental conditions are insufficiently documented in the literature.

Besides the basic FLEXnav system, we introduced several measures aimed at improving dead-reckoning accuracy in Mars Rovers. We described in detail the function of three all-wheel slippage (AWS) Indicators that compare readings from different sensors in order to detect AWS. Although we have not yet completed the merging of these indicators, the results obtained to date and presented here clearly show the feasibility of this approach.

We also presented a novel method called "Fuzzy Logic Encoder Compensation" (FLEC). FLEC compensates for AWS-related odometry errors by adjusting encoder output in proportion to quantified AWS indicator output. The single large-scale experiment that we performed to date shows a reduction in errors of 30-50% over conventional position

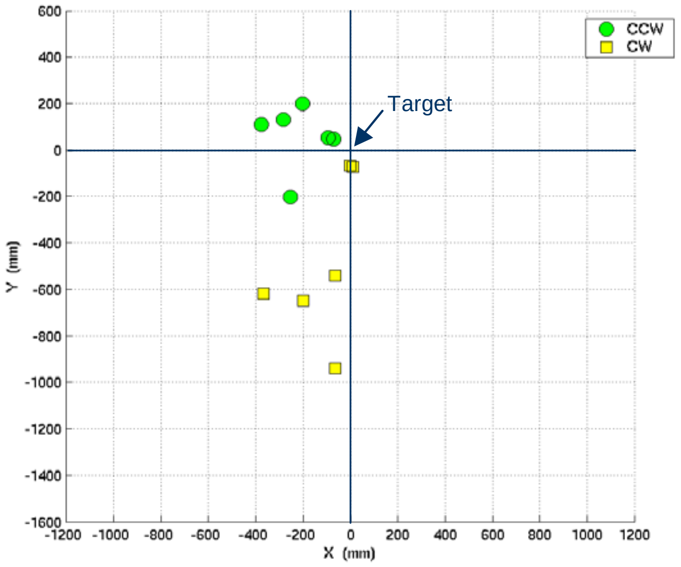

Figure 11: Return position errors for Experiment #3. This experiment is identical to Experiment #2, but this time it was run with AWS detection and Fuzzy Logic Encoder Compensation (FLEC).

Table V: Conditions and results for the high-slippage Experiment #2, without (Set #2) and with (Set #3) Fuzzy Logic Encoder Compensation.

Experiment		**Set #2** (without FLEC)	**Set #3** (with FLEC)
Terrain features		Two steep slopes 15 and 35 cm height, a few fist-sized rocks. Substantial slippage. See Figure 2 and Figure 3.	
Number of runs		6×cw, 6×ccw	
Total distance per run		40	
Average of absolute errors. E and E[%]	ccw	(X_e, Y_e) = (19.6 cm, 38.6 cm) E = 43.3 cm; E[%] = 1.08%	(X_e, Y_e) = (21.0 cm, 10.5 cm) E = 23.5 cm; E[%] = 0.59%
	cw	(X_e, Y_e) = (9.0 cm, 72.2 cm) E = 72.8 cm; E[%] = 1.82%	(X_e, Y_e) = (12.4 cm, 43.1 cm) E = 44.9 cm; E[%] = 1.12%

estimation.

ACKNOWLEDGEMENTS

This work was funded by NASA/JPL Contract No. 1232300. The authors would like to thank Daniel Cruz for guiding the undergraduate students who built Fluffy and for subsequently perfecting the platform.

6. References

[1] 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 (IROS '94).* Munich, Germany, September 12-16, pp. 1347-1354.

[2] Krantz, D. and Gini, M., 1996, “Non-Uniform Dead-Reckoning Position Estimate Updates.” *Proc. of the 1996 IEEE Int. Conference on Robotics and Automation Minneapolis*, MN, April 22-25, pp. 2061-2066.

[3] 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.

[4] Tunstel et al, 2002, “FIDO Rover System Enhancements for High-Fidelity Mission Simulations.” *Proc. 7th Int. Conference on Intelligent Autonomous Systems (IAS-7)*, Marina del Rey, CA, Mar 25-27, 2002, pp. 349-356.

[5] Ojeda, L. and Borenstein, J., 2002, “FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain.” *Proc. of the 2002 IEEE Int. Conference on Robotics and Automation*. Washington DC, USA, 11 - 15 May, pp. 317-322.

[6] 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

[7] Ojeda, L. and Borenstein, J., 2004, “Methods for the Reduction of Odometry Errors in Over-constrained Mobile Robots.” Accepted for publication in *Autonomous Robots*.

[8] 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.

[9] Bekker, G., “*Introduction to Terrain-Vehicle Systems*.” University of Michigan Press, 1969.

[10] Iagnemma, K. and Dubowsky, S., 2000, “Mobile Robot Rough-Terrain Control (RTC) for Planetary Exploration.” *Proc. 26th ASME Biennial Mechanisms and Robotics Conference*, DETC 2000, Baltimore, MD, Sept. 10-13.

Biography

***Lauro Ojeda** has a Bachelor Degree in Electronic Engineering and a Master of Science Degree in Electronics from the Army Polytechnic School - Ecuador. He is currently a research investigator at the Mobile Robotics Laboratory of the University of Michigan. His area of expertise is the development of accurate positioning systems for mobile robots. He has worked on several projects sponsored by DARPA, DOE and NASA. He is the key developer of the algorithms and techniques described in this paper.*

***Giulio Reina** graduated in Mechanical Engineering at the Politecnico of Bari, Italy in 2000. He worked at the University of Michigan Mobile Robotics Lab as a visiting scholar in 2002/03. He is currently a PhD student at the Politecnico of Bari. His research interests include ground autonomous vehicles, rough terrain mobile robot localization and kinematic design of mobile robots.*

***Johann Borenstein** (M'88) received the B.Sc., M.Sc., and D.Sc. in Mechanical Engineering in 1981, 1983, and 1987, respectively, from the Technion – Israel Institute of Technology. He is a Senior Research Scientist and Head of the Mobile Robotics Lab at the University of Michigan since 1987. His research interests include mobile robot position estimation and obstacle avoidance, and the design of novel robotic platforms. He has six patents, over 100 publications on mobile robotics, and won the 1998 Discover Magazine Award for Technological Innovation in Robotics.*