Autonomous Robots (Journal), Vol. 16, pp. 273-286.

Methods for the Reduction of Odometry Errors in Over-constrained Mobile Robots

by

Lauro Ojeda[1] and Johann Borenstein[2]

ABSTRACT

This paper presents an analysis of odometry errors in over-constrained mobile robots, that is, vehicles that have more independent motors than degrees of freedom. Examples of over-constrained vehicles are the various 6-wheeled Mars Rovers like Rocky-7, Rocky-8, or Fido.

Based on our analysis we developed two novel measures aimed at reducing odometry errors. We also developed a novel method that serves as a framework for the implementation of the two new measures, as well as for other, conventional error reducing measures.

One of the two new measures, called "Fewest Pulses Measure," makes use of the observation that most terrain irregularities, as well as wheel slip, result in an erroneous overcount of encoder pulses. The second new measure, called "Cross-coupled Control Measure," optimizes the motor control algorithm of the robot to reduce synchronization errors that would otherwise result in wheel slip with conventional controllers.

The novel method that serves as a framework for other measures is based on so-called "Expert Rules." In this paper we formulate three expert rules aimed at reducing dead-reckoning errors. Two of these expert rules are related to the foregoing discussion on error reducing measures. The third expert rule adds a gyroscope to the system and we re-examine the effectiveness of the odometry error-reducing measures in the context of this addition.

In the work described in this paper we modified a Pioneer AT skid-steer platform by providing it with four independent drive motors and encoders. We implemented our error-reducing measures and the expert rule method on this over-constrained platform and present experimental results.

[1,2]) Affiliation of Authors:

[1)] Lauro Ojeda
University of Michigan
Department of Mechanical Engineering
Advanced Technologies Lab
1101 Beal Avenue
Ann Arbor, MI 48109
Ph: (734) 647-1803
email: lojeda@umich.edu

[2)] Johann Borenstein
University of Michigan
Department of Mechanical Engineering
Advanced Technologies Lab
1101 Beal Avenue
Ann Arbor, MI 48109
Ph: 734-763-1560
email: johannb@umich.edu

[3]) A poster version of this paper was presented in a poster session at the 2003 SPIE AeroSense Symposium, Orlando, FL, April 21-25, 2003, and a shorter version of this paper is included in the proceedings of that conference.

1 INTRODUCTION AND STATEMENT OF PROBLEM

This paper discusses ways of improving dead-reckoning in mobile robots. Dead-reckoning is the estimation of changes in position and heading of a vehicle relative to a known starting position. Dead-reckoning differs from other methods of position estimation in that it does not use any external beacons (such as GPS), or any other natural or artificial landmarks. Instead, dead-reckoning uses only internal sensors, such as gyroscopes, accelerometers, and wheel encoders. Position estimation using *only* wheel encoders is called "odometry." With odometry one can measure both linear displacement as well as changes in heading. However, on less-than-ideal ground conditions, odometry alone quickly accumulates large position and heading errors.

A significant reduction in heading errors is possible by the addition of gyroscopes to the dead-reckoning system. Of course, gyroscopes do well only in measuring rotation; they do not measure linear displacement. Another potential source for linear displacement measurements are accelerometers. The well-known problem with accelerometers is that their output has to be integrated twice to yield linear displacement. As a result, even the relatively small bias drift of high-quality accelerometers quickly grows into large position errors [Barshan and Durrant-Whyte, 1995], making this sensor modality unsuitable for measuring linear displacements. Yet another approach, called "Visual Odometry" [Olson et al., 2000], is currently being developed at JPL. This method employs computer vision to detect and track natural terrain features and derives relative position estimates from the shift of these features. However, currently this promising method is mostly being used when the vehicle is stationary, or perhaps moving at an extremely slow speed. We therefore feel that the seemingly narrow focus of this paper on the reduction of odometry-derived linear displacement errors in wheeled vehicles is warranted, since there is currently no other viable way to measure linear displacement in fast moving robots.

Several interesting research efforts reported in the scientific literature aim at combining 6-DOF Inertial Measurement Units (IMUs) with odometry. One comprehensive system of this kind was described by Dissanayake et al. [2001]. These authors fused a 6-DOF IMU with data from a differential, real-time kinetic (RTK) GPS and odometry from a single wheel sensor. Their data fusion approach derives insights from odometry and applies them as constraints to the IMU system. Specifically, Dissanayake et al. say that a vehicle traveling on a straight road has zero-velocities in directions perpendicular to the rolling direction of the rear wheels. For the on-road applications envisaged by the authors this is, of course, correct. Our work, on the other hand, focuses on the analysis and reduction of odometry errors for vehicles traveling on rugged terrain, and especially for so-called "over-constrained" mobile robots. Land vehicles with more than three kinematically independent motors are defined[4] as over-constrained because there are only three degrees-of-freedom of motion for land vehicles. Many mobile robots have even more independent motors for driving and steering. For example, the Mars Rovers Rocky-8 and Fido, which were designed and built by the Jet Propulsion Labs [Volpe et al. 2000; Tunstel, et al., 2002], have six independently controlled drive motors and six independently controlled steering motors for their six wheels. Other examples of mobile robots with over-constrained drive systems are the six-wheeled SHRIMP made by Bluebotics [2003], the Nomad arctic traverse

[4] This is a simplified definition that is suitable for the purpose and scope of this paper. An in-depth treatment of vehicle kinematics, degrees of freedom, and constrains is presented in [Muir and Neuman, 1987].

robot developed at Carnegie Mellon University [Shamah et al., 1998], and wheeled snake-type robots like the ones developed by Hirose et al. [1991]. For all of these vehicles the designers accept the reduction of odometric accuracy in their over-constrained systems as a tradeoff for increased mobility.

This paper analyses some aspects of odometry errors in these over-constrained mobile robots and proposes improvements that can be applied without the need for mechanical redesign. To do so we turned our attention to the 4-wheel-drive/skid-steer (4WDSS) mobile robots of the Pioneer-series [Activemedia 2003] and the ATRV-series [iRobot 2003]. Interestingly the manufacturers of these robots chose to drive the wheels on each side of these platforms by two kinematically linked electric motors, thereby providing a total of four drive motors in each vehicle. However, since the two motors on each side are mechanically linked to each other by belts or chains, the two motors on each side act as one.

In order to create an over-constrained platform for our experiments we simply removed the drive belts from the motor pairs on each side of an off-the-shelf Pioneer AT platform. We also equipped the two originally encoder-less motors with incremental optical encoders. The result was the platform shown in Figure 1. Using this modified configuration of the Pioneer AT-based platform we investigated means for reducing odometry errors in over-constrained mobile robots, as explained in Section Measures for Reducing Odometry Errors. Also in Section 2 we introduce a method for combining these odometry error-reducing measures[5] with other measures for reducing linear displacement errors in over-constrained mobile robots, including the use of a gyro. Experimental results are provided in Section 3 and Conclusions are presented in Section 4.

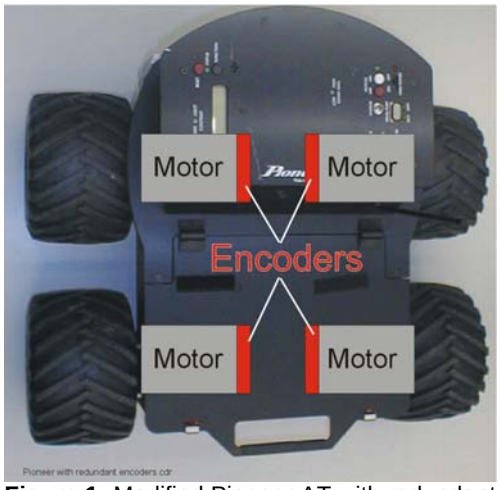

Figure 1: Modified Pioneer AT with redundant encoders.

2 MEASURES FOR REDUCING ODOMETRY ERRORS

In this section we present two candidate measures for reducing odometry errors. We called them: (1) The "Fewest Pulses Measure" and (2) the "Cross-coupled Control Measure." We also present a method for combining these two and other measures, called the "Expert Rules Method."

[5] For greater clarity we distinguish between a "method" and its components by calling the latter "measures." In this paper we present one novel *method*, which serves as a framework for several *measures*. We also present two new measures for the reduction of odometry errors.

2.1 The Fewest Pulses Measure

Since only two encoders are needed to compute basic odometry, four encoders, as used in our modified Pioneer AT, are redundant. One approach to arbitration among redundant encoders is to apply a simple rule:

> "*Use only the data from the encoder-pair*[6] *that provided the least number of pulses during the last sampling interval.*"

The rationale behind this approach is that for most error-causing conditions encoders "over-count." That is, they provide more pulses than what corresponds to the actual linear distance traveled, when they encounter wheel slip or bumps or cracks on the ground.

However, in our work we soon discovered that there is a technical caveat with this approach. The caveat is related to the rounding error inherent in measuring wheel rotation with incremental encoders. To explain this effect, consider a conventional differential-drive mobile robot with a total of just two wheel encoders, where the same two encoders are sampled for odometry during each sampling interval. Since encoder pulses are discrete events, sampling might produce a number like, say, 35 pulses for a given sampling interval, even though the actual rotation of the wheel might have corresponded to, say, 35.6 pulses. The truncated .6 pulses are not a problem, because they go into the next encoder sample, which would read 36 pulses, even though the wheel rotation during that next sampling interval might still have corresponded to only 35.6 pulses (at constant speed, etc.).

However, when the Fewest Pulses Measure is applied in our redundant encoder system, then it is not the same encoder that gets sampled every time, and which would therefore carry over a truncated fraction into the next sampling interval. Rather, since we always select the encoder with the smallest number of pulses in a sampling interval, we would always select the encoder that reads 35 pulses, while the other encoder might read 36 pulses (to stick with the above example). Furthermore, since this decision causes an unrecovered truncation error every time we sample, the overall error can be rather large.

It is possible to estimate the error resulting from this approach probabilistically. To do so, let us assume that there is no wheel slip and that the robot moves at a speed that will produce at least one encoder pulse per sampling period.

Next, consider Figure 2a, which shows the pulse trains from two encoders on one side of the robot. For the purpose of this explanation we illustrate the worst case in terms of errors introduced by the Fewest Pulses Measure, which occurs when the two encoders have a phase difference of $\varphi=180°$. Furthermore we illustrate in Figure 2a the special case, in which the duration of one encoder pulse is exactly equal to one sampling interval. In this special case there is a 50% probability of getting the same number of ticks from both encoders in any sampling interval, since both encoders have the same reading during exactly 50% of the interval. During the remaining 50% of the time, one encoder will produce one tick less than the other. The probabilistic error in this special case is $\varepsilon'_{max} = 50\%$. The best case, in terms of errors introduced

[6] An "encoder-pair" in our redundant system is comprised of one encoder from the right side and one from the left side. An encoder pair could thus comprise, for example, the left front encoder and the rear right encoder.

by the Fewest Pulses Measure, occurs when φ=0°. In this case, of course, the probabilistic error of switching from one encoder to the other is $\varepsilon'_{min} = 0\%$. We can thus assume that in the special case of Figure 2a, the average probabilistic error $\varepsilon' = (\varepsilon'_{max} + \varepsilon'_{min})/2 = 25\%$ encoder ticks per sampling interval *T*. Thus the average probabilistic error, E_p, measured in millimeters per second (of travel time) and incurred when using the Fewest Pulses Measure, can be estimated by:

$$E_p = \varepsilon' F \Delta L \qquad (1)$$

where:

F - sampling frequency, measured in sampling intervals per second [int/sec]

ΔL - linear distance traveled per encoder tick [mm/tick]

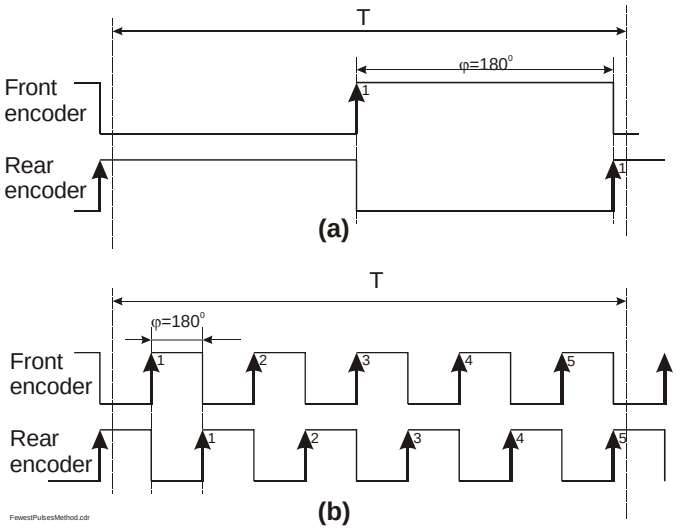

Figure 2: In the Fewest Pulses Measure the worst case occurs when pulse trains from the front and rear encoder are offset by 180°. (a) Special case of encoder pulse durations equaling exactly one sampling interval *T*. (b) General case of several encoder ticks being sampled in one sampling interval *T*.

Figure 2b shows the more realistic case in which multiple encoder ticks are produced during a sampling interval.

While Eq. 1 provides the designer with a probabilistic estimate for the amount of error incurred by the Fewest Pulses Measure, in practice it is better to correct for the truncation error in a deterministic way, as follows. Obviously, the truncation error is always an amount between 0.0 and 1.0; on average 0.5. Furthermore, the truncation error always results in an underestimation of distance traveled. We can thus reduce the effect of the truncation error by adding a correction factor of $c = 0.5$ to the reading of the encoder with the fewest pulses, whenever readings are switched from one encoder to the other on that side and a difference exist between both encoders.

Under most conditions this approach is quite effective. However, as we mentioned earlier, the Fewest Pulses Measure doesn't apply to all conditions, as discussed in the next section.

2.2 The Cross-coupled Control Measure

At low speeds (e.g., slower than 200 mm/sec) and while driving over a low traction surface like sand, we observed that there is actually rather significant wheel skid[7]. Skidding is different from slipping in that skidding wheels always produce *fewer* encoder pulses than tracking wheels,

[7] Because of ambiguity in the way the terms slip and skid are used in the scientific literature, we define these terms in the context of this paper as follows: "Slip" or "slipping" is the loss of traction that results when the motor applies too much power to the wheel, e.g., over-acceleration. "Skid" or "skidding" is the loss of traction that results when the vehicle is in motion but a wheel is blocked or partially blocked from rolling by excessive friction in the drive train or the effect of motor braking. We refer to slip and skid collectively as "Slippage."

while slipping wheels produce *more* encoder pulses than tracking wheels. Our analysis revealed that there are two causes for the excessive wheel skid in over-constrained robots:

1. In conventional motion control for differential-drive vehicles the software prescribes separate reference velocities to the left and right wheels. Under normal conditions there are short transient times, during which the motors try to get up to the commanded speed; and then, if the control loop is properly designed, the wheels settle at their commanded speed. However, external disturbances and slippage will interfere with the commanded speeds and result in additional transient times, during which the wheels don't follow the commanded speeds exactly. This situation is exacerbated when driving at very slow speeds, because then internal friction in gears and bearings causes significant disturbances all the time and oftentimes halts a wheel altogether.
2. In differential-drive mobile robots, such as the venerable LabMate platform, momentary deviations from commanded speeds only reduce the accuracy, at which the robot tracks a given path. This is not a big problem in most applications. However, in a skid-steer platform like the Pioneer AT the problem is much more severe. This is because the four-wheel drive Pioneer AT is mathematically severely over-constrained, that is, it has four controlled motors but only one independent degree-of-freedom: forward/backward motion. As a result, all wheels have to run at the exact same speed to avoid slippage. Yet, even when commanded to move straight forward, any momentary mismatch between wheel velocities will force the wheels to skid or slip unpredictably.

One way to reduce the occurrence of slippage in over-constrained vehicles is the use of the so-called "Cross-coupled" Control (CCC) Measure. This measure was originally developed by Borenstein and Koren [1987] and later refined by Feng, Koren, and Borenstein [1993]. 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 Mars Rover.

The CCC Measure continuously compares the actual encoder pulses from the left and right wheel of a robot and issues corrective commands to the motors to slow down the motor that is faster and speed up the motor that is slower than the other. The overall effect of this measure is that the velocities of the left and right wheels of a robot are matched more tightly even in the presence of internal and external disturbances. Figure 3 shows a block diagram with the implementation of cross-coupled control between the left-front and left-rear motor.

The actual implementation of CCC is not very complex, typically just a few lines of code when the sole purpose of the controller is to assure equal pulse counts from two motors, as is the case for the two motors on one side of the robot. Of course, a 4WDSS platform requires two identical CCC loops to be implemented, one to couple the two left-hand side motors and one to couple the two right-hand side motors. In addition, a third CCC loop must be implemented to couple average pulse counts from the left to average pulse counts from right side of the robot. This third CCC loop is slightly more difficult to implement because steering in a 4WDSS robot requires unequal speeds for the left and right-hand side motors. A detailed description of this more complex CCC implementation, along with the equations for implementing it, is given in [Feng et al., 1993]. Experimental results with the CCC Measure are provided in Section 3.1.

We should mention one other important benefit of the CCC Measure, although that benefit is not directly related to odometry. In suspensionless platforms like the Pioneer AT series or the ATRV series, it is often the case that one wheel is "in the air" while the vehicle travels over

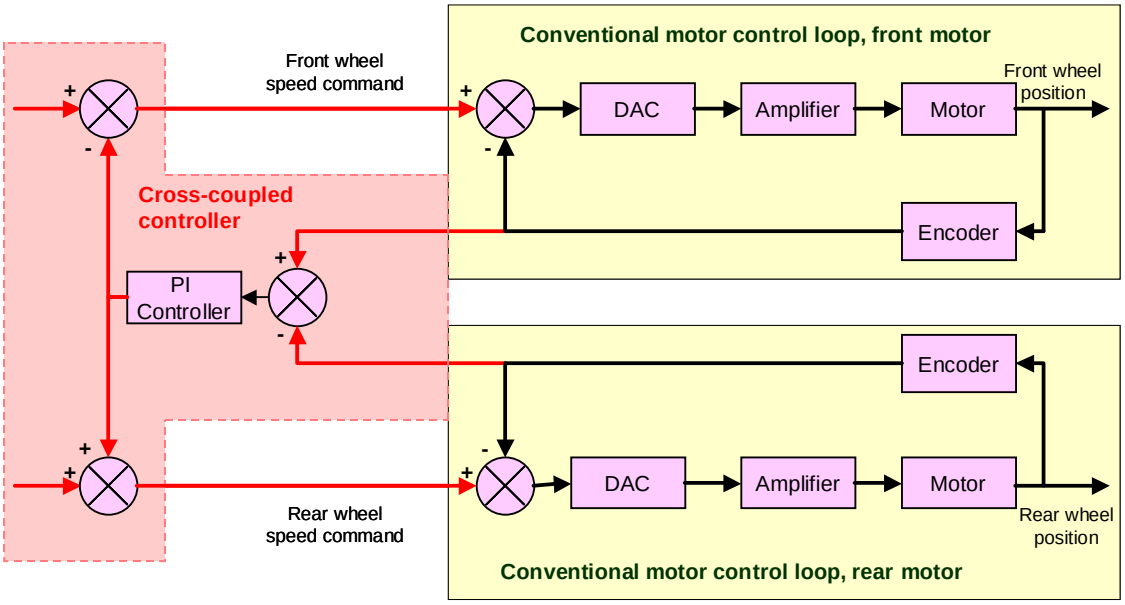

Figure 3: Block diagram of a Cross-coupled Control (CCC) loop, here shown for the two left-hand wheels.

irregular (that is: not flat) terrain. If each of the wheels is independently powered by a motor, as is the case in our modified Pioneer AT, then the free-spinning wheel and motor don't contribute torque to propel the vehicle forward. As a result, the vehicle often stalls on high-resistance surfaces, such as deep sand. The manufacturers of the above 4WDSS vehicles overcome this problem by physically linking the wheels and motors on each side by a drive belt. This assures that both wheels are powered by both motors and that the combined torque of the motors is available to the remaining wheel if the other wheel on that side is in the air. The CCC measure partially overcomes this problem by increasing the speed of the "stuck" wheel, as the measure attempts to equalize the speed of the two cross-coupled wheels on that side.

2.3 The Expert Rule Method

While the Cross-coupled Control Measure can reduce some degree of slippage, it is clear that slippage will also occur for other reasons than the ones described above, and that these additional slippage-induced odometry errors can't be corrected by improving the controller.

In order to reduce these odometry errors we developed a rule-based expert system that differentiates between different cases and offers a course of action for each one of them. We recognize that none of the expert rules provides an absolutely correct solution, but we hypothesize that the rules provide an equal or better solution than the "Baseline Rule" below:

Baseline Rule: Average Encoder Reading

This "rule" represents the trivial approach to using four encoders in an over-constrained 4WDSS vehicle: the readings of the two encoders on each side of the robot are averaged

$$C_L = (C_{LF}+C_{LB})/2$$
and (2)
$$C_R = (C_{RF}+C_{RB})/2$$

where we defined the following indices: L = Left; R = Right; F = Front; B = Back, and where

C – number of encoder pulses during the sampling interval (C can be a non-integer, due to averaging)

A special case of this approach is the unmodified, off-the-shelf Pioneer, where the wheels on each side are mechanically coupled. If there were encoders on all four wheels of the Pioneer, then C_L=C_{LF}=C_{LB} and C_R=C_{RF}=C_{RB}.

To improve performance beyond that provided by the Baseline Rule we define the following two sets of Expert Rules that are applied during each sampling interval:

Expert Rule #1 (Average + Smallest) – If the front and rear encoder on either side of the robot produce a similar number of encoder pulses in a sampling interval (i.e., the difference being one pulse or less), then the number of pulses reported by both encoders on that side is assumed to be correct. In this case the displacement of that side of the robot (Left or Right) is computed as the arithmetic average of the front and rear wheel encoder reading. This is identical to the baseline rule.

Otherwise, if the difference is more than one pulse, then the wheel with the greater pulse count is assumed to be slipping and we select the wheel with the lower count to provide the encoder reading for that side of the robot and for that sampling interval. Note that this is the Expert Rule formulation of the "Fewest Pulses Measure," and we are also adding the correction factor c = 0.5, as explained at the end of Section 2.1.

As we will show in the Experimental Results Section, Expert Rule #1 is very effective at reducing odometry errors. However we will define below Expert Rule #2 that reduces errors even further.

We must emphasize that up to this point our discussion centered strictly on odometry, without the involvement of inertial sensors, notably gyros. In the following Expert Rule #2 we will use a gyro. However, with regard to that Expert Rule #2 we use the gyro not to determine the relative change in heading (yaw) of the robot, as would be the conventional use of a gyro. Rather, in the context of this paper we are interested in the gyro only because of its unconventional ability to reduce linear displacement errors, in the way described below. Our existing fully functional "FLEXnav" 6-DOF IMU/odometry system, which makes full use of gyros, is described in [Ojeda and Borenstein, 2002). However, the FLEXnav system was not used in any of the experiments reported in this paper, in order to keep the focus strictly on odometry.

This second set of expert rules is derived from the comparison of odometry data with gyro data. Our hypothesis is that odometry errors due to slip and skid don't produce the same pulse count in the left and right wheel in general, and in particular if one wheel is slipping and the other is gripping. As a result, the heading computed from odometry under these circumstances suggest that the vehicle turned, even if the vehicle actually moved straight forward. We hypothesize that comparison of the odometry-computed heading with the heading computed by an accurate gyro may be able to detect this type of error. However, in order to not only detect the error but to also correct it, we need to have at least one wheel that did not slip or skid and we have to know which wheel it is. Expert Rule #2 can thus be defined as follows:

Expert Rule #2 (Average + Smallest + Angle) – If only one side of the robot had correct encoder readings (per the test in Expert Rule #1 above) but not the other, then the linear displacement on the correct side should be computed from the encoders, while the

displacement of the side suspected of slippage should be computed from the angular speed measured by the onboard gyro, according to the following equations:

$$\omega = \frac{D_R - D_L}{BT} \tag{3}$$

where either

$$D_L = D_R - \omega_Z T B$$

or (4)

$$D_R = D_L + \omega_Z T B$$

and:

ω_Z – angular speed measured by the gyroscope

B – track width (distance between contact points of left and right wheels with the floor)

D – distance traveled by the left (L) or right (R) side of the robot

T – sampling period

As the results in the next section show, this expert rule was very effective in reducing odometry errors. This is particularly true for errors caused by wheel slippage, which is the predominant source of error.

3 EXPERIMENTAL RESULTS

The experiments described in this section were performed on an artificial sand track inside our lab. The track was 12 meters long, 2 meters wide, and about 3 centimeters thick (see Figure 4). The elevation profile of the sand track was generally horizontal but there were small local irregularities caused by footsteps and prior traverses, as shown in Figure 5.

In a typical experiment the Pioneer was carefully placed at the Start Point (SP) at one end of the track, facing the End Point (EP) located at the other end of the track. Before each run the robot's initial heading was carefully aligned using a rigidly mounted laser pointer that beamed a light spot onto a marked wall beyond the End Point. We estimate that the initial heading error due to the limitations of this alignment method is less than ±0.05°.

Once properly aligned, we directed the robot manually toward the End Point using radio control. In this remotely controlled experiment we led the robot along two different trajectories. One was a straight path and the other a winding path roughly similar to the one shown in Figure 6. Following the trajectory of the winding path results in a total travel distance of ~11.2 m. During motion the onboard computer gathered encoder and gyroscope information for subsequent off-line analysis. Care was taken to stop the robot as close to the End Point as possible. With the human-operated radio control we managed consistently to stop the robot within a circle of 2 cm radius around the exact End Point. We estimate the total maximal error resulting from limitations of our experimental system to be 10 cm in lateral positions (X-axis in the results graphs below) and 2 cm in radial direction (Y-axis). The error is larger in X-direction because of the errors in the initial heading.

Figure 4: Pioneer on the indoor sand track.

Figure 5: Typical terrain detail of the indoor sand track.

Once the robot was stopped at the End Point, we stopped the onboard data collection and ran our various odometry computation measures on the collected data. The final result from running any of the odometry computation measures was a *computed* End Point $EP_c = (X_c, Y_c)$. Comparison of the computed End Point with the actual End Point EP_a = (0.00 m, 10,000 mm) yielded the final position error $\varepsilon = (\varepsilon_x, \varepsilon_y)$, where $\varepsilon_x = X_c - 0$ and $\varepsilon_y = Y_c - 10{,}000$ for the particular computation measure and for the particular experiment. Result graphs in the remainder of this paper show plots of ε for the different experiments.

The gyro used in Expert Rule #2 was a KVH E-Core RA2100 fiber optic gyro [KVH 2003]. We re-emphasize that for the experiments reported here we did not use the gyro directly to measure heading; we used it only to detect and correct odometry errors resulting from wheel slippage as explained in Expert Rule #2.

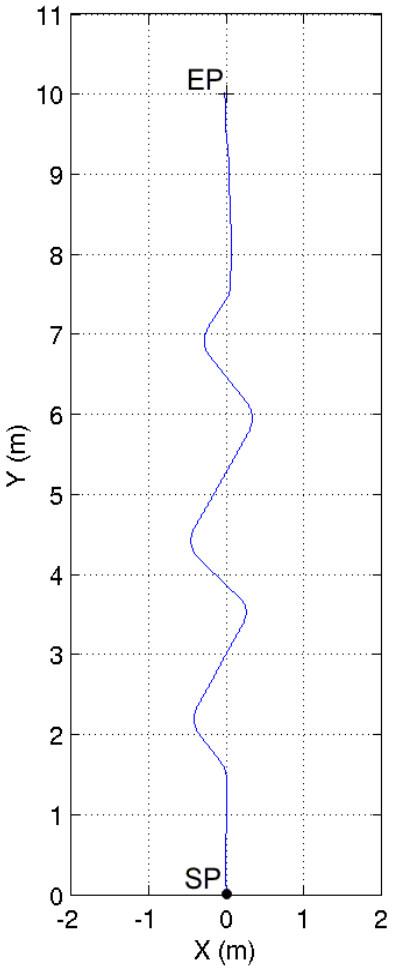

Figure 6: The winding path trajectory.

3.1 Experimental Results With the Cross-coupled Control Measure

We conducted a series of experiments that aimed at comparing the performance of Cross-coupled Control (CCC) to that of a conventional motion controller. The latter was implemented as a standard Proportional-Integral-Differential (PID) controller, as customary in many robotic platforms. We defined two performance metrics for the experiment: Effectiveness in avoiding slippage and effectiveness in improving odometric accuracy.

To test the effectiveness of the CCC Measure with regard to producing equal encoder counts with both the front and rear wheel on either side of the robot, we drove the Pioneer on our indoor sand track at fixed speeds of 100, 200, 400, and 600 mm/sec along the winding path of Figure 6. During the run the control software computed a measure of effectiveness, *S*. *S* is defined as the sum of all absolute differences between front and rear wheel encoders on either side, divided by the total number of encoder counts.

$$S_L = \frac{1}{T_L}\sum_{i=1}^{M}\Delta C_{i,L} \qquad (5a)$$

$$S_R = \frac{1}{T_R}\sum_{i=1}^{M}\Delta C_{i,R} \qquad (5b)$$

where

$$T_L = \frac{1}{2}\sum_{i=1}^{M}\left|C_{i,LF} + C_{i,LB}\right| \qquad (6a)$$

$$T_R = \frac{1}{2}\sum_{i=1}^{M}\left|C_{i,RF} + C_{i,RB}\right| \qquad (6b)$$

$$\Delta C_{i,L} = |C_{i,LF} - C_{i,LB}| \qquad (7a)$$

$$\Delta C_{i,R} = |C_{i,RF} - C_{i,RB}| \qquad (7b)$$

i – the i-th sampling interval

M – total number of sampling intervals

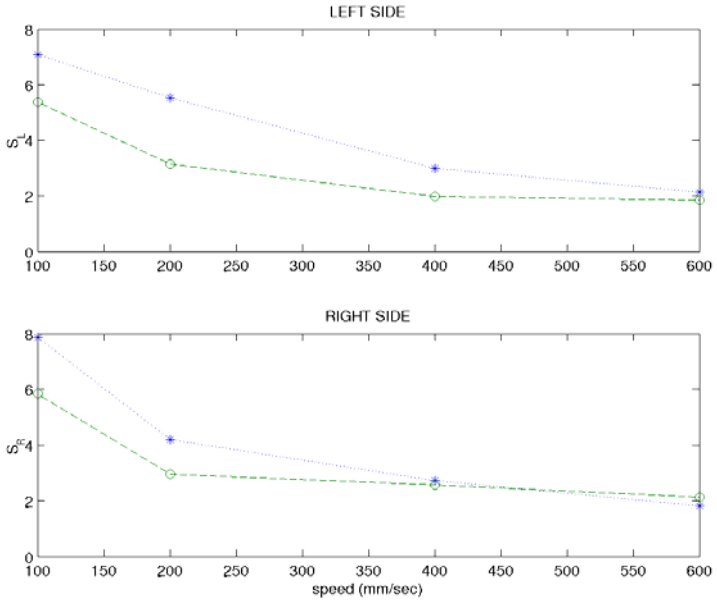

Figure 7: The (absolute) sum of the encoder count differences is a measure for the effectiveness of Cross-coupled Control in equalizing the speed of kinematically independent drive wheels. Legend: '*' = PID, 'O' = CCC.

The results of this experiment are shown in Figure 7. It is apparent from these results that the CCC Measure is indeed effective in minimizing the difference in wheel speeds.

In a different experiment, this one aimed at measuring the effectiveness of CCC in reducing odometry errors, we drove the robot again along the winding path of Figure 6. We also applied two different control algorithms: conventional PID control and CCC. For each speed the experiments were performed at least twice with each type of controller. Figure 8 shows the stopping positions of the robot as computed by conventional odometry, that is, by averaging the encoder counts of the front and rear wheel on each side of the robot. We recall that the actual stopping position of the robot was exactly at (0.0, 10.0 m). We performed eight runs with each control method. The Mean Square Error (MSE), *E*, for the eight runs with each controller runs was:

PID control: $E_{PID} = 365$ mm
CCC: $E_{CCC} = 266$ mm

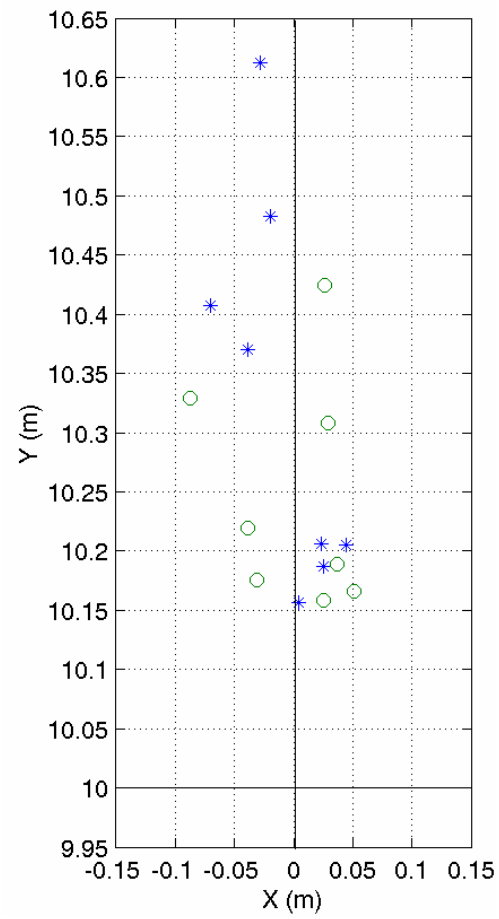

Figure 8: Results of the winding path experiment under PID and CCC control. Legend: '*' = PID, 'O' = CCC.

Table I: Experimental conditions and results for the Expert Rule experiments.

	Speed	Baseline Rule (Average)		Expert Rule #1 (Average +Smallest)		Expert Rule #2 (Avrg+Smallest +Heading)	
Path	(mm/s)	Xerror (mm)	Yerror (mm)	Xerror (mm)	Yerror (mm)	Xerror (mm)	Yerror (mm)
Straight	50	13	56	11	-129	13	29
Straight	50	50	55	45	-204	49	29
Straight	100	38	64	35	-85	39	49
Straight	100	95	47	92	-92	95	30
Straight	150	-147	64	-146	-15	-147	49
Straight	150	195	27	193	-51	195	10
Straight	200	-25	33	-25	-30	-25	20
Straight	200	226	34	224	-28	225	20
Straight	500	5	35	7	-52	6	6
Straight	500	-25	48	-25	-13	-25	30
Straight	1000	57	52	56	-30	56	15
Straight	1000	22	58	21	-5	21	26
Straight	1500	-19	95	-19	19	-19	43
Straight	1500	15	94	14	3	14	35
Winding	100	-127	974	-16	20	-17	91
Winding	200	52	908	97	48	110	85
Winding	200	918	1563	9	-65	21	-22
Winding	200	-217	714	-214	-46	-212	-16
Winding	300	-116	786	-18	-73	-18	-6
Winding	500	-10	734	137	40	154	74
Winding	500	211	1530	102	67	128	32

The results of this experiment show that CCC provides an improvement in odometric accuracy when travelling over sandy terrain, although this improvement is not substantial.

3.2 Experimental Results with the Expert Rule Method

In this section we compare the odometric accuracy obtained with the Baseline Rule (i.e., simple average of the encoder outputs of each side of the robot) with the odometric accuracy obtained with the two expert rules defined in Section 2.3. The experimental conditions for the experiments described here are essentially identical to the ones described for the CCC Measure in Section 3.1.

We performed at total of 22 runs under these conditions and list the results in Table I. As in previous results presented in this paper, the errors represent the difference between the actual stopping position of the robot and the position computed by our odometry system. Note that for each line item in Table I we performed the experiment only once, while recording the odometry data. Then, in post processing, we applied the Baseline Rule and the Expert Rule Method to the exact same set of data. Figure 9 shows the results of Table I graphically.

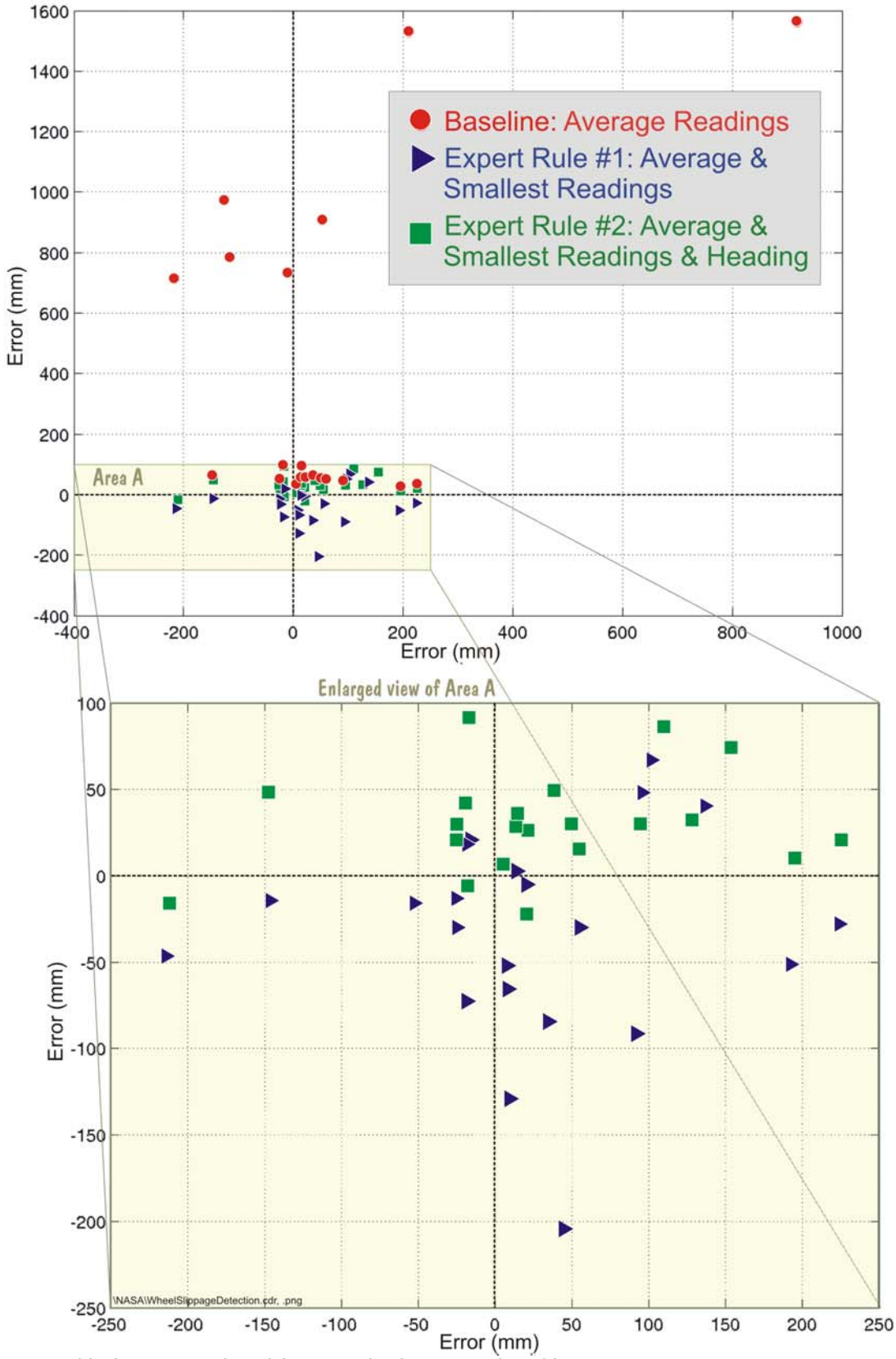

Figure 9: Graphical representation of the errors for the 22 runs in Table I.

These results show clearly the superiority of the expert rule-based approach over the Baseline Rule, i.e., the simple averaging of encoder readings. Between the two expert rules, Expert Rule #2 (Average & Smallest Readings & Heading) shows the best results (see enlarged view in Figure 9).

3.3 Comparison Between Cross-coupled Control an the Expert Rule Method

We applied Expert Rule #2 on both odometry results obtained in the experiments of Section 3.1, which, as we recall, used two different control methods: CCC and PID control. The results are shown in Figure 10. Comparing Figure 10 to Figure 8 shows that the Expert Rule Method results in significantly smaller odometry errors than the Baseline Rule.

Specifically, the Mean Square Error result for the Expert Rule method was E = 56 mm and E = 71 mm when combined with conventional PID control and CCC, respectively. These two results are almost identical (the small difference of 15 mm is insignificant since our measurement error is on the order of $\sqrt{100^2 + 20^2} \cong 100$ mm. The apparent reason for this is that the Expert Rule error correction is effective in correcting those errors that CCC was aimed to prevent. The most important results of this study are tabulated in Table II.

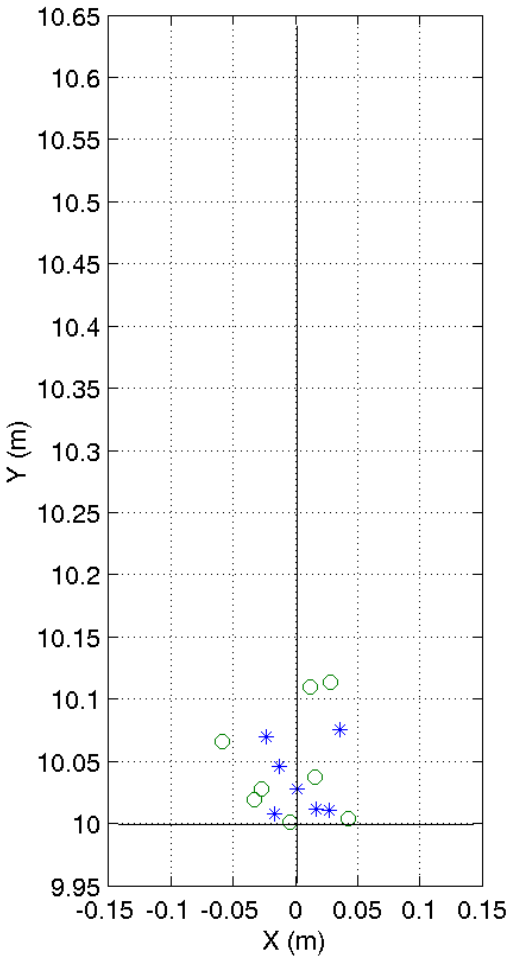

Figure 10: Odometry errors at the end of the winding path experiment· with the Expert Rules. Legend: '*' = Expert Rules with conventional PID controller, 'O' = Expert Rules with CCC.

4 CONCLUSIONS

In this paper we introduced two measures for the reduction of odometry errors in over-constrained mobile robots. We also present a method for combining these and other measures to reduce odometry errors. The results are applicable to virtually all over-constrained vehicles that have redundant encoders (i.e., encoders attached to more than two independent drive wheels).

Our findings were that in an over-constrained vehicle an inefficient control algorithm could contribute to the amount of wheel slippage. The Cross-coupled Control Measure explained in this paper is capable of reducing some of these errors by assuring that the motion controller-prescribed ratio between wheels rotations is maintained.

Table II: Summary of Mean Square Errors of the eight experimental runs obtained with the different methods described in this paper.

Odometry correction method	Control method	
	Conventional PID controller	Cross-coupled Control (CCC)
Baseline (simple average of each side)	E = 365 mm	E = 266 mm
Expert Rule #2 (Average & Smallest & Heading)	E = 56 mm	E = 71 mm

However, we also found that what we call the "Expert Rule Method," is significantly more effective in reducing odometry errors. Indeed, the Expert Rule Method resulted in a Mean Square Error of E = 71 mm, which is smaller than our measurement error of ~100 mm and represents a relative error of less than 0.7% over the total travel distance of ~11 meters (see Table 2). We consider this a remarkable result, given that our test vehicle was a skid-steer platform driving in a zigzag pattern over sand. Even though this result could not be improved by using the CCC Measure, we still find it beneficial to do so because the CCC Measure provides smoother motion.

We should emphasize, however, that under most circumstances it is best to run 4WDSS mobile robots like the popular Pioneer AT series or ATRV series in their off-the-shelf configuration, that is, with drive belts linking the two wheels on each side. In other experiments (not described in this paper) we ran our Pioneer AT with those drive belts in place, and obtained results comparable to those obtained with our Expert Rule Method. In the drive belt experiments we did apply the gyro correction as explained under Expert Rule #2, since gyro correction is equally applicable for configurations with and without redundant encoders. Nonetheless, the methods applied in this paper are useful in over-constrained vehicles such as Mars Rovers.

Acknowledgements

This work was funded by NASA/JPL Contract No. 1232300 and by the U.S. Department of Energy under Award No. DE-FG04-86NE3796. The authors also wish to express their gratitude to Daniel Cruz who conducted many of the experiments described in this paper.

5 REFERENCES

Activmedia 2003 – ActivMedia Robotics, 44 - 46 Concord Street, Peterborough, NH 03458, http://www.activrobots.com/ROBOTS/p2at.html.

Barshan, B. and Durrant-Whyte, H.F., 1995, "Inertial Navigation Systems Mobile Robots." *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, June 1995, pp. 328-342.

Baumgartner, E.T., Aghazarian, H., and Trebi-Ollennu, A, "Rover Localization Results for the FIDO Rover." *SPIE Photonics East Conference*, Orlando, FL, October, 28-29, 2001.

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.

BlueBotics 2003 – BlueBotics SA, Parc Scientifique PSE-C, 1015 Lausanne EPFL, Switzerland, http://www.bluebotics.com.

Dissanayake, G. et al., 2001, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications." *IEEE Transactions on Robotics and Automation*, Vol. 17 Issue 5, Oct. 2001, pp. 731-747.

Feng, L, Koren, Y., and Borenstein, J., 1993, "Cross-coupled Motion Controller for Mobile Robots." *IEEE Control Systems*, December, pp. 35-43.

Hirose, S. et al., 1991, "Design of Practical Snake Vehicle: Articulated Body Mobile Robot KR-II." *Proc. 5th Int. Conf. Advanced Robotics*, Pisa, Italy, pp. 833-838.

iRobot 2003 - Twin City Office Center, 22 McGrath Highway, Suite 6, Somerville, MA 02143, http://www.irobot.com/rwi/p01.asp.

KVH 2003 – KVH Industries, Inc., 8412 W. 185th St., Tinley Park, IL 60477, USA, http://www.kvh.com.

Muir, P.F. and Neuman., C.P., 1987, "Kinematic Modeling of Wheeled Mobile Robots." *Journal of Robotic Systems*. Vol. 4. No. 2. pp. 281-340.

Ojeda, L. and Borenstein, J., 2002, "FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain." *Proc. 2002 IEEE Int. Conf. on Robotics and Automation,* Washington DC, USA, 11-15 May, pp. 317-322.

Olson, C.F.; Matthies, L.H.; Schoppers, H.; Maimone, M.W., "Robust Stereo Ego-motion for Long Distance Navigation." Computer Vision and Pattern Recognition, Hilton Head, South Carolina, June 13-15, 2000, p. 453-458 vol. 2.

Shamah, B., Apostolopoulos, D., Rollins, E., and Whittaker, W.L., 1998, "Field Validation of Nomad's Robotic Locomotion." *Proc. SPIE - Mobile Robots XIII and Intelligent Transportation Systems, Vol. 3525,* Boston, MA, Nov. 3-4, pp. 214-222.

Tunstel, E., T. et al., "FIDO Rover Field Trials as Rehearsal for the 2003 Mars Exploration Rover Mission." *9th Intl. Symp. on Robotics & Applications, 5th World Automation Congress*, Orlando, FL, June 2002.

Volpe, R., Baumgartner, E. Schenker, P., and Hayati, S., "Technology Development and Testing for Enhanced Mars Rover Sample Return Operations." *Proceedings of the 2000 IEEE Aerospace Conference*, Big Sky, Montana, March 18-25, 2000, pp. 247 -257.