

Odometry and Calibration Methods for Multi-Castor Vehicles

James Doebbler, Jeremy J. Davis, John L. Junkins, and John Valasek
Aerospace Engineering Department, Texas A&M University
College Station, TX 77843, USA

{james.doebbler, jeremy.davis, junkins, valasek}@tamu.edu

Abstract— We are developing a mobile robot capable of emulating general 6-degree-of-freedom spacecraft relative motion. The omni-directional base uses a trio of active split offset castor drive modules to provide smooth, holonomic, precise control of its motion. Encoders measure the rotations of the six wheels and the three castor pivots. We present a generic odometric algorithm using a least squares framework which is applicable to vehicles with two or more castors and apply it to our unique vehicle configuration. As the accuracy of odometry algorithms depends on the accuracy to which the model parameters are known, a method to perform calibration on the physical robot is needed. We present a geometric calibration method based solely on internal sensor measurements. We present a range of simulation results comparing our odometry results to other algorithms under various systematic and non-systematic errors. We evaluate the ability of our calibration method to accurately determine the true values of our system parameters. The odometry algorithm was also implemented and tested in hardware on our robotic platform. The results presented in the paper validate the calibration and odometry algorithms in both simulation and hardware.

I. INTRODUCTION

Multi-vehicle proximity operations of spacecraft, from formation flying to automated rendezvous and docking, represent an extremely active area of current research as industry moves toward smaller, cheaper satellites. Ground-based testing of the autonomous control algorithms for multiple vehicles with sensors or docking hardware in-the-loop provides significant risk reduction for such missions. We are developing a relative motion emulator based on multiple mobile platforms, termed Relative Motion Vehicles (RMVs), consisting of an omni-directional base with a Stewart platform mounted atop [1]. The base provides large motions in 3-degrees-of-freedom (DOF), whereas the Stewart platform provides smaller motion in all 6-DOF to a high degree of precision, so it can be used to “clean-up” the less precise motion of the base.

The mobile platform approach provides several distinct advantages over existing facilities like NRL’s Proximity Operations Testbed [2] and NASA’s Flight Robotics Facility [3]:

- 1) Allows for un-tethered circumnavigation of two or more vehicles.
- 2) Mobile nature enables testing or demonstration at any location with a large enough workspace.
- 3) Provides a low-cost alternative to larger installations while maintaining high fidelity.
- 4) Supports testing of non-spacecraft multi-vehicle systems, such as autonomous aerial refueling.

While Stewart platforms meeting the needs of this project are commercially available, the omni-directional base must meet motion requirements exceeding typical mobile robot designs. For instance, conventional robots have undesirable coupling between translation and rotation due to steering dynamics. We have designed an omni-directional base utilizing a trio of active split offset castors (ASOCs) to achieve the desired motion.

Current methods of odometry focus primarily on a standard two-wheeled differentially-steered vehicle. The odometry equations for these drive types are well known and widely used. Borenstein presents a method called Internal Position Error Correction (IPEC), which uses a vehicle with two castors to achieve an order of magnitude improvement in odometry performance over a single-castored vehicle [4]. In this paper, we generalize and extend the ideas of IPEC to vehicles with two or more castors, with specific application to our three-castored ASOC vehicle.

Critical to the odometry process is precise knowledge of many internal geometric parameters. Whereas multiple calibration methods exist for the standard two-wheeled robot, the process is much more difficult for multi-castored robots. We address the problem by providing a geometric solution to parameter determination using the robot’s internal sensors.

This paper presents details of algorithms to perform calibration and odometry on an omni-directional mobile base for use in emulating on-orbit proximity operation dynamics. Drive train selection and design is discussed in Section II. Algorithmic details of both the odometry and calibration methods necessary to achieve the high precision positional knowledge through internal angle encoders are discussed in Section III, with simulation results being presented in Section IV. Results from testing these algorithms using a one-third scale prototype are discussed in Section V. Section VI presents conclusions and recommendations for future work.

II. DESIGN CONFIGURATION

This section overviews key aspects of the physical design of the RMVs and the associated design decisions.

A. Defined Requirements

True holonomic omni-directional motion of the base must be achieved with strict motion and sensing requirements as discussed in [1]. Since the errors in the base position can be corrected using the Stewart platform, the base control accuracy is an order of magnitude lower than the desired

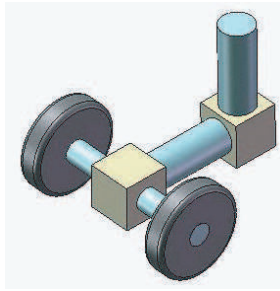


Fig. 1. Active Split Offset Castor (ASOC)

position knowledge. We set a target mass of 300 kg for the mobile base - an order of magnitude heavier than the Stewart platform - so that the motion of the Stewart platform and payload can be treated as a minor disturbance when controlling the RMV. Developing an omni-directional robot of such proportions with such stringent tracking and knowledge requirements results in a unique design configuration.

B. Drive Mechanism

The key design consideration of the mobile base is the drive configuration. In order to accurately model the dynamics of a spacecraft docking maneuver, the overall RMV must be able to accurately follow the desired trajectory. Several different drive mechanisms were considered for the mobile base, including individually steerable wheels, Mecanum wheels, and steerable offset castors [1]. We have adopted an attractive alternate drive method that uses drive modules consisting of a configuration known as an active split offset castor [5][6].

The ASOC design consists of two independently driven motors mounted along a common axis on a castor that is attached to the mobile base by a freely rotating vertical pivot. The axis of the wheels is offset in the horizontal plane from the pivot point in the direction perpendicular to the wheel axis. This creates a physical design such that any planar velocity of the pivot point can be achieved by driving the two independent wheels. The general concept is depicted in Fig. 1. A vehicle driven with at least two of these castors can achieve true holonomic motion with no steering dynamics, thus achieving the same advantages as the single offset castor while reducing the scrubbing torque. Though there are six motorized wheels, the total number of motors is the same as the steerable wheels considered previously. Unlike steerable wheels, however, all of the motors can be exactly the same, an advantage in terms of motor characterization, maintenance, and supportability.

C. Drive Module Layout

A minimum of three wheel modules are required for stability, and at least two of the castors must be powered. The third wheel module can be either passive or driven. We selected the option of having all three castor modules being driven and controlled, as it allows a more even distribution of motor torques between the motors throughout the envelope of

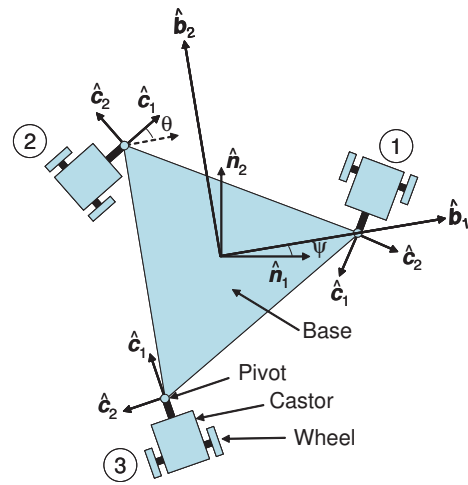


Fig. 2. Relative Motion Vehicle

allowable trajectories. The distribution of the three powered castors can be seen in Fig. 2.

We define the locations of the pivots using polar coordinates (R_i, ϕ_i) with respect to the body-fixed B -frame. Note that the choice of the location and orientation of the body-fixed frame is completely arbitrary. For simplicity, we designate the center of the three pivots as the origin of the B -frame and select castor 1 to lie along the \hat{b}_1 axis ($\phi_1 = 0$).

D. Kinematics and Control

We have developed the kinematics using a body-fixed B frame, that is fixed to the base of the robot and three castor-fixed C_i frames as seen in Fig. 2. The angle the B frame makes with the inertial N frame is denoted ψ , whereas the angle each castor makes with respect to the B -frame is denoted θ_i . By assuming no wheel slippage and by measuring the castor angles, the body-axis linear and angular velocities uniquely map into the six wheel velocities. Control is accomplished through a kinematics-based controller using feedback of wheel velocities and measurements of θ_i from angular encoders mounted on each pivot [1][7].

III. ALGORITHMS

The overall system will incorporate a combination of “dead-reckoning”, or internal position feedback, and a periodic external inertial update. This section describes a method for highly accurate calibration as well as a multi-castor odometry algorithm for internal position measurement.

A. Calibration

Since odometry methods inherently rely on geometric parameters, pose estimates from odometry can only be as accurate as the calibration allows. Borenstein has developed a calibration method using the results of the University of Michigan Benchmark (UMBmark) test for a two-wheeled robot [8], and other researchers have put forth a variety of other methods improving on the UMBmark calibration for a differentially driven robot [9][10]. For any multi-castored vehicle, however, calibration methods are non-existent in

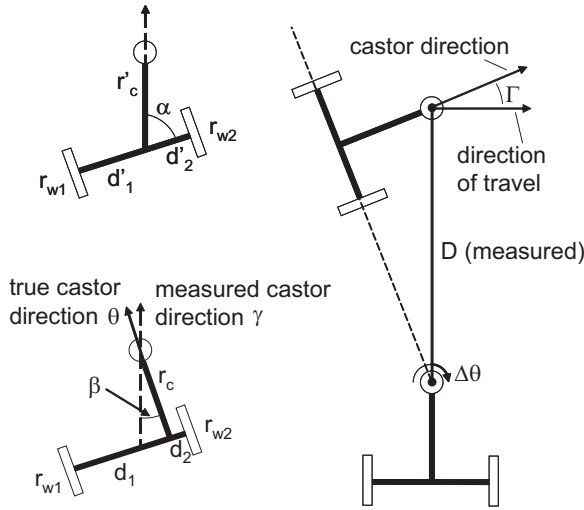


Fig. 3. Calibration set-up and definition of parameters

the literature, and the calibration problem is an order of magnitude more challenging.

As described in the previous section, each motor/wheel assembly is equipped with an incremental encoder to record wheel rotations, and each vertical pivot is equipped with an absolute encoder to measure castor angles. We have developed a geometric approach for calibrating our three castor mobile robot that relies entirely on these internal sensor measurements, though this method could be applied to any vehicle with more than one powered castor.

We make only two assumptions about the configuration for this approach. First, the distance between any two castor pivots, D , is assumed to be known. This parameter is easily measured prior to attaching castors to the pivot shafts. Second, we assume that the two wheel axles on a given castor are colinear, something that can be assured if the wheel bearings on a castor are mounted with a single shaft passing through all of them. This second assumption is one implicitly assumed in all two-wheeled robot calibration methods.

The parameters to be estimated are

- r_c - the length of the line perpendicular to the wheel axles passing through the pivot,
- r_{wi} - the wheel radii,
- d_i - the distance from the wheels to the pivot along the axle, and
- β - the difference between the true castor angle θ , and the measured castor angle γ

as shown in Fig. 3. Note from the left side of the figure that an angular misalignment of the wheel axis of $90^\circ - \alpha$ can be transformed into a pivot angle misalignment β and a new set of r_c and d_i parameters. The β term not only handles this angular misalignment of the wheels, but also compensates for any pivot encoder angle bias.

The reference motion for calibration is simply to hold one of the castors stationary while applying a small constant voltage to the remaining wheels. By aligning the active castors in one direction or another, the robot will revolve

about the pivot of the stationary castor in either a clockwise (CW) or counter-clockwise (CCW) manner. The angular distance traveled by each wheel, c_i , is recorded, as is the angular distance through which the stationary castor pivot rotates, $\Delta\theta$.

The path angle Γ can be calculated from the angles reported by the pivot encoders, γ , from the runs in both CW and CCW directions as follows:

$$\Gamma = \frac{180^\circ - (\gamma_{ccw} - \gamma_{cw})}{2} \quad (1)$$

We define θ_\perp to be the angle that should be measured by the castor encoder when the castor is aligned with the direction of travel in the CW maneuver. Note that θ_\perp is defined geometrically by the choice of the B -frame with respect to the pivots. Combined with the measured castor angles, γ , the pivot encoder bias, β , is found using:

$$\beta = \frac{\gamma_{ccw} + \gamma_{cw} - 180^\circ}{2} - \theta_\perp \quad (2)$$

Additionally, a simple geometric relationship provides r_c :

$$r_c = D \sin \Gamma \quad (3)$$

Both the CW and CCW runs provide information in the following form regarding the relationship between wheel radius, wheel base, and the measured quantities.

$$\Delta\theta_{cw} (D \cos \Gamma + d_1) = c_{cw1} r_{w1} \quad (4)$$

$$\Delta\theta_{cw} (D \cos \Gamma - d_2) = c_{cw2} r_{w2} \quad (5)$$

$$\Delta\theta_{ccw} (D \cos \Gamma - d_1) = c_{ccw1} r_{w1} \quad (6)$$

$$\Delta\theta_{ccw} (D \cos \Gamma + d_2) = c_{ccw2} r_{w2} \quad (7)$$

Solving these equations for r_w produces:

$$r_{wi} = \frac{2D \cos \Gamma (\Delta\theta_{ccw} c_{cw i} - \Delta\theta_{cw} c_{ccw i})}{\Delta\theta_{ccw} c_{cw i} + \Delta\theta_{cw} c_{ccw i}} \quad (8)$$

And solving for d leaves the equation:

$$d_i = \frac{s_i D \cos \Gamma (\Delta\theta_{ccw} c_{cw i} - \Delta\theta_{cw} c_{ccw i})}{\Delta\theta_{ccw} c_{cw i} + \Delta\theta_{cw} c_{ccw i}}, \quad s_1 = 1, s_2 = -1 \quad (9)$$

Thus, all parameters of a given castor can be estimated from a single run CW and CCW about a stationary castor. In our case, with three castors, executing rotations about all three castors provides two estimates for each castor's parameters that we can average to provide improved accuracy. It is important to note that this procedure merely requires the active wheels to be commanded a voltage low enough that no wheel slippage occurs - no additional rigs or measuring apparatus are required.

B. Odometry

The internal encoders can be used to integrate the motion of the wheels to form an inertial position estimate, a process referred to as odometry. As with all dead-reckoning schemes, odometry is subject to accumulation of error, and estimates based on encoders must be updated with an external measurement or landmark recognition. Odometry is subject to both systematic errors, such as errors in wheel radii estimates,

and non-systematic errors, such as floor irregularities, wheel slippage, or typical random measurement error. It is important to note that in odometry, unlike IMU integration, error accumulates as a function of distance traveled rather than as a function of time. Borenstein provides a detailed discussion of odometry methods and sources of error [11][12].

Of particular interest is the OmniMate Mobile Robot - an omni-directional platform that uses two powered split castors with additional free castors for stability. Using the Internal Position Error Correction (IPEC) algorithm he developed, Borenstein was able to reduce the dead-reckoning uncertainty of OmniMate by an order of magnitude compared to conventional wheeled robots [4]. This is intuitive, since each castor could provide its own inertial position estimate and with the relationship - angles and distance - between the two castors measured, the position of the vehicle is known with improved accuracy. This result applies to the mobile platform developed here with three independent castors, and should allow for further improvement on the 0.1% average accumulation error seen with OmniMate [8].

The IPEC algorithm is built on the premise that errors in heading angle lead to pose estimate errors that are orders of magnitude larger than the small translational errors introduced at a given update. IPEC utilizes each castor's position estimate to compute the heading angle of the vehicle through the geometric relationship

$$\psi = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (10)$$

where ψ is the heading angle of the vehicle and (x_i, y_i) are the coordinates of the center of each castor. Whereas the heading angle of each castor could be computed independently, this use of each castors translational information - which suffers minimally from systematic and non-systematic errors over a single step - reduces heading estimate errors. This vehicle heading estimate is combined with the measured castor angles to update the heading estimate of each castor, and the position of the second castor is updated to be consistent with the pose of the first castor.

The IPEC method can be applied to any two castors on our platform by using the positions of the pivot points rather than those of the castors in the above equations - a distinction unique to having an offset castor. We have developed a method based on the ideas of IPEC that can be used for more than two castors to take advantage of all information available.

First, we note that the inertial position of a pivot (x_i^+, y_i^+) at time t^+ relates to the inertial position of the center of the robot (x_c^+, y_c^+) by the following relationship

$$x_i^+ = x_c^+ + R_i \cos(\psi^+ + \phi_i), \quad y_i^+ = y_c^+ + R_i \sin(\psi^+ + \phi_i) \quad (11)$$

$$x_i^- = x_c^- + R_i \cos(\psi^- + \phi_i), \quad y_i^- = y_c^- + R_i \sin(\psi^- + \phi_i) \quad (12)$$

where R_i and ϕ_i describe the position of the pivot with respect to the center of the robot. The corresponding relationship at t^- is also shown. Subtracting these two equations and making a small angle assumption about the change in heading angle,

$\Delta\psi$, results in the following equations which are linear in the unknowns $(\Delta x_c, \Delta y_c, \Delta\psi)$:

$$\Delta x_i = \Delta x_c - \Delta\psi R_i \sin(\psi^- + \phi_i) \quad (13)$$

$$\Delta y_i = \Delta y_c + \Delta\psi R_i \cos(\psi^- + \phi_i) \quad (14)$$

Standard odometry equations as given by Borenstein [11] are adapted to compensate for the offset of the pivot from the wheel axis to generate the $(\Delta x_i, \Delta y_i)$ from the wheel encoder readings. Thus, as in IPEC, we utilize only the information regarding translational motion of each castor to inform our heading estimate rather than the heading information from each castor. Equations 13 and 14 can be used in a linear least squares algorithm employing n castors.

The validity of the small angle assumption depends on the angular rate of the vehicle and position update frequency. In practice, the low complexity of this algorithm allows high update frequencies to ensure the validity of this assumption. Even a robot rotating at $90^\circ/sec$ with position updates at 20 Hz would see a $\Delta\psi$ of less than 5° at each update step.

When applying such algorithms to hardware, algorithmic complexity must be considered. Given a measurement model of $Y = HX$ with all measurements weighted equally, the linear least squares solution is given by

$$X = (H^T H)^{-1} H^T Y \quad (15)$$

where

$$X = \begin{bmatrix} \Delta x_c \\ \Delta y_c \\ \Delta\psi \end{bmatrix}, Y = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta y_1 \\ \Delta y_2 \\ \vdots \end{bmatrix} \quad (16)$$

and,

$$H = \begin{bmatrix} 1 & 0 & -R_1 \sin(\psi^- + \phi_1) \\ 1 & 0 & -R_2 \sin(\psi^- + \phi_2) \\ \vdots & \vdots & \vdots \\ 0 & 1 & R_1 \cos(\psi^- + \phi_1) \\ 0 & 1 & R_2 \cos(\psi^- + \phi_2) \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (17)$$

Significant computational savings can be achieved by pre-computing the required matrix inverse which takes the form

$$(H^T H)^{-1} = \frac{1}{n(b^2 + c^2 - nd)} \begin{bmatrix} c^2 - nd & -bc & nb \\ -bc & b^2 - nd & nc \\ nb & nc & -n^2 \end{bmatrix} \quad (18)$$

where

$$b = -n(m_y \cos \psi^- + m_x \sin \psi^-) \quad (19)$$

$$c = n(m_x \cos \psi^- - m_y \sin \psi^-) \quad (20)$$

$$d = \sum_{i=1}^n R_i^2 \quad (21)$$

TABLE I
MEAN PARAMETRIC ERRORS OVER 1000 SIMULATIONS

Parameter	Uncalibrated	Calibrated
r_c	0.21%	0.09%
r_w	2.07%	0.04%
d	0.89%	0.04%
β	7.98°	0.01°

and m_x and m_y are the $x-y$ coordinates in the body frame describing the “center of mass” of the pivots with respect to the center of the vehicle. Note that defining the center of the robot to be the center of the three pivots results in $b = c = 0$, and Eq. 15 simplifies to

$$X = \begin{bmatrix} \Delta x_c \\ \Delta y_c \\ \Delta \psi \end{bmatrix} = \frac{1}{n} \sum_{i=1}^n \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \frac{1}{R} (\Delta y_i \cos(\psi^- + \phi_i) - \Delta x_i \sin(\psi^- + \phi_i)) \end{bmatrix} \quad (22)$$

In this special case, the translation of the center of the base is simply the average of the motion measured by each castor.

Just as in IPEC, the final, critical, step is to update the heading angle of each castor with the newly computed vehicle heading and the measured castor angles.

It should be noted that in the case with more than two castors, the redundant information could be used to detect when a castor has been subjected to large non-systematic errors. Once detected, this invalid information could be disregarded from the final odometric position estimate in order to increase robustness to these errors. This extension is not the focus of this paper and is left for future study.

IV. SIMULATION

In order to test the validity of both algorithms, we performed simulations using 1000 different sets of randomly varied parameters. The mean errors of each parameter from the true value are listed as the uncalibrated values in Table I.

A. Calibration

As discussed earlier, accurate calibration is required for the odometry to achieve the accuracies required for our system. To evaluate our approach, we ran 1000 different sets of parameters through the calibration algorithm. We compare the mean absolute error of the calibrated numbers over all 1000 runs to the mean parametric errors prior to calibration. Orders of magnitude improvement in the parameter estimates can be seen in Table I.

B. Odometry

The UMBmark test consists of the robot traversing a square both clockwise and counterclockwise five times and measuring the position estimate error at the end of each circuit [8]. The magnitude of the average position error for both clockwise (r_{cw}) and counterclockwise (r_{ccw}) is

TABLE II
UMBMARK TEST SIMULATION RESULTS

Algorithm	Uncalibrated E_{max}	Calibrated E_{max}
IPEC-1	394.6 mm 6.89%	71.7 mm 1.25%
CAST-1	394.1 mm 6.88%	71.6 mm 1.25%
CAST-0	311.0 mm 5.43%	39.0 mm 0.68%

computed, and the maximum of these two values provides a measure of the accuracy of the odometry, E_{max} .

We simulated a robot based on the dimensions of our prototype traversing a 1.5 meter square with random parametric errors consistent with the measurement uncertainties of our uncalibrated prototype shown in Table I, with the exception of β , which we assumed to be zero. All encoder readings were produced with quantization errors and an update rate of 20 Hz. We simulated the UMBmark test on 1000 different sets of parameters and computed the average value of E_{max} over those runs for several different algorithms.

We first implemented IPEC, considering only two castors, and then implemented our least squares solution using the same two castors as well as all three castors. We denote these combinations as IPEC- i and CAST- i , where i represents the castor not used. Following this notation, CAST-0 represents the solution with no castor excluded: the least squares solution using all three castors. Table II shows the results of the UMBmark test simulation using the uncalibrated parameters, including the error as a percent of distance traveled.

The large numbers shown for the uncalibrated cases in Table II are a result of large parametric uncertainties - up to 4% in some cases - but this has no effect on the comparison of one algorithm to another. The two-castor least squares provides nearly the same solution as IPEC, whereas the three-castor solution shows a 20% improvement.

After calibration, the odometry simulation was rerun with the calibrated parameters, and the pose estimation errors reduced by a factor of 5-8 as seen in the calibrated results of Table II. The three-castor least squares improves the estimate by almost a factor of 2 over either two-castor estimate.

V. HARDWARE-BASED IMPLEMENTATION

In this section, we discuss a one-third scale prototype that was used to demonstrate design feasibility and for use in testing and development of data fusion techniques, control laws, and a dynamics model of the full-scale platform. The actuator and sensor configurations of the full-scale platform were implemented on the prototype to accurately evaluate state estimation and control methods. The prototype presented here possesses all of the required sensors, but at lower precisions than the full-size mobile platform.

The prototype has optical encoders on the wheels and on the pivots, so that data can be used in feedback control laws just like the full scale implementation. The wheel encoders can measure in increments of 0.3°, whereas the pivot encoders can measure in increments of 0.07°. Fig. 4 shows a picture of the one-third scale prototype.

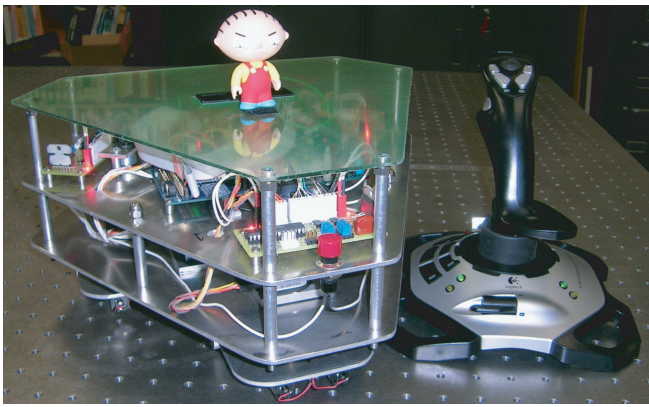


Fig. 4. Prototype mobile base with joystick controller

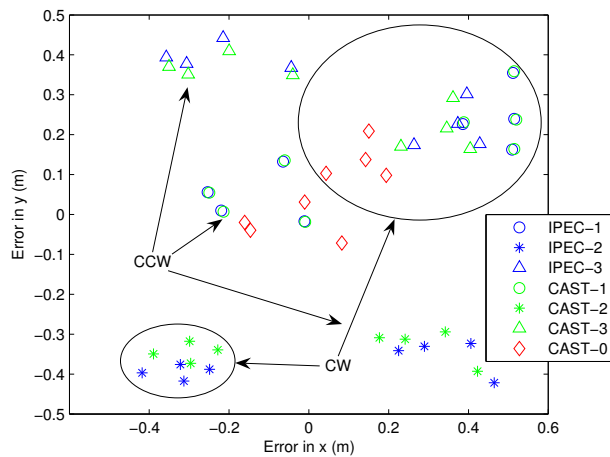


Fig. 5. Results of UMBMark test using prototype. IPEC- i and CAST- i refer to the i^{th} castor being excluded. CAST-0 uses all 3 castors

The one-third scale hardware platform has been completed and can be controlled using either a joystick or pre-defined trajectories which it follows open-loop. We present here the results of a slightly modified UMBmark test using the prototype to evaluate the wheel odometry position estimation algorithms discussed earlier. To complete this test, the data was collected for four runs in each direction along a 1.5 m square and then run through each of the algorithms offline. Fig. 5 shows the final errors for each run for each algorithm, and Table III shows the average error, E_{max} . We show the results for IPEC by using only two of the three castors at a time, in all three combinations since parametric errors lead to different results for different combinations of castors. We also present our corresponding least squares solutions. As in the simulation, the two-castor least squares results very nearly match the IPEC results, and the three castor result (CAST-0) provides the best solution. As in the odometry simulation, these errors are large overall and have meaning only for comparisons of algorithms until the calibration methods have been employed.

TABLE III
PROTOTYPE TEST RESULTS

Algorithm	E_{max}	
IPEC-1	540 mm	9.5%
IPEC-2	511 mm	9.0%
IPEC-3	458 mm	8.0%
CAST-1	544 mm	9.5%
CAST-2	459 mm	8.0%
CAST-3	432 mm	7.6%
CAST-0	191 mm	3.3%

VI. CONCLUSIONS AND FUTURE WORK

Algorithms for multi-castor calibration and odometry have been presented. Simulation results verify both approaches when compared to the true parameters and IPEC, and the tests in hardware using our prototype further validate the algorithms. The next step is to test these algorithms on another platform and implement the least squares odometry algorithm as realtime feedback for control purposes. We are currently designing the full-scale version of our prototype to complete this testing.

REFERENCES

- [1] J. Davis, J. Doebbler, K. Daugherty, J. Junkins, and J. Valasek, "Aerospace Vehicle Motion Emulation Using Omni-directional Mobile Platform," in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, Aug 2007.
- [2] G. Creamer, F. Pipitone, C. Gilbreath, D. Bird, and S. Hollander, "NRL Technologies for Autonomous Inter-Spacecraft Rendezvous and Proximity Operations," in *Proceedings of the John L. Junkins Astrodynamics Symposium, AAS/AIAA Space Flight Mechanics Meeting*, College Station, Texas, May 2003, paper AAS 03-272.
- [3] F. D. Roe, D. W. Mitchell, B. M. Linner, and D. L. Kelley, "Simulation Techniques for Avionics Systems - An Introduction to a World Class Facility," in *Proceedings of the AIAA Flight Simulation Technologies Conference*, San Diego, California, July 1996, paper 96-3535.
- [4] J. Borenstein, "Experimental Results from Internal Odometry Error Correction with the OmniMate Mobile Robot," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, Dec. 1998.
- [5] H. Yu, M. Spenko, and S. Dubowsky, "Omni-Directional Mobility Using Active Split Offset Castors," *ASME Journal of Mechanical Design*, vol. 126, no. 5, pp. 822–829, Sept. 2004.
- [6] M. Spenko, H. Yu, and S. Dubowsky, "Analysis and Design of an Omnidirectional Platform for Operation on Non-Ideal Floors," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, Washington DC, May 2002, pp. 726–731.
- [7] X. Bai, J. J. Davis, J. Doebbler, J. Turner, and J. L. Junkins, "Dynamics, Control and Simulation of a Mobile Robotics System for 6-DOF Motion Emulation," in *World Congress on Engineering and Computer Science*, San Francisco, CA, Oct 2007.
- [8] J. Borenstein and L. Feng, "UMBmark: A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots," University of Michigan, Ann Arbor, Michigan, Tech. Rep. UM-MEAM-94-22, Dec. 1994.
- [9] T. Abbas, M. Arif, and W. Ahmed, "Measurement and Correction of Systematic Odometry Errors Caused by Kinematics Imperfections in Mobile Robots," in *SICS-ICASE International Joint Conference*, Bexco, Busan, Korea, October 2006.
- [10] P. R. Kedrowski, C. F. Reinholtz, and D. C. Conner, "Optimized parametric calibration of autonomous vehicles," in *Proceedings of the SPIE The International Society for Optical Engineering*, Newton, MA, Oct 2001.
- [11] J. Borenstein, H. Everett, and L. Feng, *Navigating Mobile Robots: Sensors and Techniques*. Wellesley, MA: AK Peters Ltd, 1996.
- [12] J. Borenstein, H. Everett, L. Feng, and D. Wehe, "Mobile Robot Positioning: Sensors and Techniques," *Journal of Robotic Systems*, vol. 14, no. 4, pp. 231–249, 1997.