Technical Report #94-22

UMBmark:

A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots

by

Johann Borenstein[1] and Liqiang Feng[2]

December 1994

Prepared by the University of Michigan
For the Oak Ridge National Lab (ORNL) D&D Program
and the
United States Department of Energy's
Robotics Technology Development Program
Within the Environmental Restoration, Decontamination and Dismantlement Project

1) Dr. Johann Borenstein
The University of Michigan
Department of Mechanical
Engineering and Applied Mechanics
Mobile Robotics Laboratory
1101 Beal Avenue
Ann Arbor, MI 48109
Ph.: (313) 763-1560
Fax: (313) 944-1113
Email: Johann_Borenstein@um.cc.umich.edu

2) Dr. Liqiang Feng
The University of Michigan
Department of Mechanical
Engineering and Applied Mechanics
Mobile Robotics Laboratory
1101 Beal Avenue
Ann Arbor, MI 48109
Ph.: (313) 936-9362
Fax: (313) 763-1260
Email: Feng@engin.umich.edu

Please direct all inquiries to Johann Borenstein

Executive Summary

Dead-reckoning is the most widely used method for determining the momentary position of a mobile robot. In most practical applications dead-reckoning provides easily accessible real-time positioning information in-between periodic absolute position measurements. The frequency at which the (usually costly and/or time-consuming) absolute measurements must be performed depends to a large degree on the accuracy of the dead-reckoning system.

This report introduces a method for measuring dead-reckoning errors in mobile robots, and for expressing these errors quantitatively. When measuring dead-reckoning errors, one must distinguish between (1) *systematic* errors, which are caused by kinematic imperfections of the mobile robot (for example, unequal wheel-diameters), and (2) *non-systematic* errors, which may be caused by wheel-slippage or irregularities of the floor. *Systematic* errors are a property of the robot itself, and they stay almost constant over prolonged periods of time, while non-systematic errors are a function of the properties of the floor.

Our method, called the *University of Michigan Benchmark test* (UMBmark), is especially designed to uncover certain *systematic* errors that are likely to compensate for each other (and thus, remain undetected) in less rigorous tests. This report explains the rationale for the carefully designed UMBmark procedure and explains the procedure in detail. Experimental test results from different mobile robots are presented and discussed. Our report also proposes a method called *extended UMBmark* for measuring non-systematic errors. Although the measurement of non-systematic errors is less useful because it depends strongly on the floor characteristics, one can use the *extended UMBmark* test for comparison of different robots under similar conditions. This report presents experimental results from six different vehicles tested for their susceptibility to *non-systematic* error by means of the *extended UMBmark* test. With the quantitative benchmark test proposed here, researchers will be able to compare the dead-reckoning accuracy of different robots, or they can measure and tune the performance of a single robot.

Perhaps the foremost contribution of the work described here is a unique and innovative method for the calibration of mobile robots. This method is called *UMBmark calibration* because it is based on measurements from the UMBmark test. Performing an occasional calibration as proposed here will increase the robot's dead-reckoning accuracy and reduce operation cost because an accurate mobile robot requires fewer absolute positioning updates. Many manufacturers or end-users calibrate their robots, usually in a time-consuming and non-systematic trial and error approach. By contrast, the *UMBmark calibration* is systematic and provides near-optimal results. Our procedure for measuring and correcting systematic dead-reckoning errors can be performed easily and without complicated equipment. Furthermore, UMBmark lends itself readily for adaptation as an automated self-calibration procedure. Experimental results are presented that show a consistent improvement of at least one order of magnitude in dead-reckoning accuracy (with respect to systematic errors) for a mobile robot calibrated with the UMBmark calibration.

Table of Content

1. INTRODUCTION

This report introduces a method for measuring the dead-reckoning accuracy in mobile robots. Our method, called the *University of Michigan Benchmark* (UMBmark) test, is especially designed and optimized for differential drive robots (like the TRC LabMate), but the method can be used for all robots. In the report we present the results from several different experiments in which different aspects of the dead-reckoning performance of six mobile robot (or robot configurations) were tested and compared. The six robots are:

1. TRC LabMate
2. Cybermotion K2A
3. CLAPPER (4-DOF platform)
4. Remotec Andros
5. Remotec Andros with "encoder trailer"
6. TRC LabMate with Smart Encoder Trailer (simulation only)

Before we describe these platforms and the experiments in greater detail, we will discuss the motivation for trying to improve dead-reckoning accuracy. In Section 2 we define some important concepts in dead-reckoning and in Section 3 we introduce the UMBmark procedure. Section 4 and 5 presents the experimental results from testing the six robot platforms with the UMBmark test. Finally, in Section 6 we present a method for calibrating mobile robots using the UMBmark test.

1.1 Absolute Positioning Methods

In most mobile robot applications two basic position-estimation methods are employed together: *absolute* and *relative* positioning [Borenstein and Koren, 1987; Hollingum, 1991; Byrne et al., 1992; Chenavier and Crowley, 1992; Evans, 1994]. Relative positioning is usually based on dead-reckoning (i.e., monitoring the wheel revolutions to compute the offset from a known starting position). Dead-reckoning is simple, inexpensive, and easy to accomplish in real-time. The disadvantage of dead-reckoning is its unbounded accumulation of errors.

Absolute positioning methods usually rely on (a) navigation beacons, (b) active or passive landmarks, (c) map matching, or (d) satellite-based navigation signals. Each of these absolute positioning approaches can be implemented by a variety of methods and sensors. Yet, none of the currently existing systems is particularly elegant. Navigation beacons and landmarks usually require costly installations and maintenance, while map-matching methods are either very slow or inaccurate [Cox, 1991], or even unreliable [Congdon et al, 1993]. With any one of these measurements it is necessary that the work environment either be prepared or be known and mapped with great precision. Satellite-based navigation (GPS) can be used only outdoors and has poor accuracy, on the order of 10-30 meters [Byrne, 1993]. Radio frequency-based systems are very expensive and are susceptible to reflections from metal objects [Byrne et al., 1992].

Another approach to the position determination of mobile robots is based on inertial navigation with gyros and/or accelerometers. Our own experimental results with this approach, as well as the

results published by Barshan and Durrant-Whyte [1993; 1994], indicate that this approach is not advantageous. Accelerometer data must be integrated twice to yield position, thereby making these sensors exceedingly sensitive to *drift*. Another problem is that accelerations under typical operating conditions can be very small, on the order of 0.01 g. Yet, fluctuation of this magnitude already occur if the sensor tilts relative to a perfectly horizontal position by only 0.5°, for example when the vehicle drives over uneven floors. Gyros can be more accurate (and costly) but they provide information only on the rate of rotation of a vehicle, so their data must be integrated once. This problem does not exist with electronic compasses that measure the orientation of the robot relative to the earth's magnetic field. However, electronic compasses are not recommended for indoor applications, because of the large distortions of the earth's magnetic field near power lines or steel structures [Byrne et al., 1992].

1.2 The Importance of Dead-reckoning

Improved dead-reckoning can dramatically reduce the cost for installations of mobile robot systems because it simplifies the fundamental problem of position determination.

Evidence for the importance of dead-reckoning accuracy was provided at the 1992 AAAI Mobile Robot Competition in San Jose, California. Mobile robots from 10 leading universities and research institutes took part in this competition, which comprised of three different "typical mobile robot tasks." The two teams with the first and second place winning entries, "CARMEL" and "FLAKEY," (U of Michigan and Stanford Research Institute, respectively) summarized their experience and insights from this competition in [Congdon et al., 1993]. Comparing the performance of U of M's CARMEL and SRI's FLAKEY the teams concluded:

> *"The use of precise dead-reckoning and long-range sensing gave CARMEL a marked advantage... [over FLAKEY's landmark navigation]."*

In spite of the apparent importance of accurate dead-reckoning, little research is directly aimed at improving the dead-reckoning accuracy of mobile robots. We attribute this observation to the fact that a large portion of research in mobile robotics is being done by the Artificial Intelligence (AI) community. AI researchers are traditionally concerned with the higher-level aspects of robotics. For this reason, AI researchers appear to focus on methods of feature extraction and map matching [Skewis et al., 1991; Kortenkamp et al. 1992; Rencken, 1994]. These research issues are of great importance for the future development of mobile robots, but at this time they are either too slow or too inaccurate to replace dead-reckoning altogether. Even Cox [1991], a proponent of map-matching, says about the virtues of dead-reckoning accuracy:

> *"There also appears to be a self sustaining property to this configuration [map-matching combined with dead-reckoning]: Accurate knowledge of position allows for fast robust matching, which leads to accurate knowledge of position."*

2. Properties of Dead-Reckoning Errors

Figure 2.1 shows a typical *differential drive* mobile robot, the *LabMate* platform manufactured by [TRC]. In this design incremental encoders are mounted onto the two drive motors to count the wheel revolutions. Using simple geometric equations, it is straight-forward to compute the momentary position of the vehicle relative to a known starting position. This computation is called dead-reckoning. For completeness, we rewrite the well known equations for dead-reckoning below.

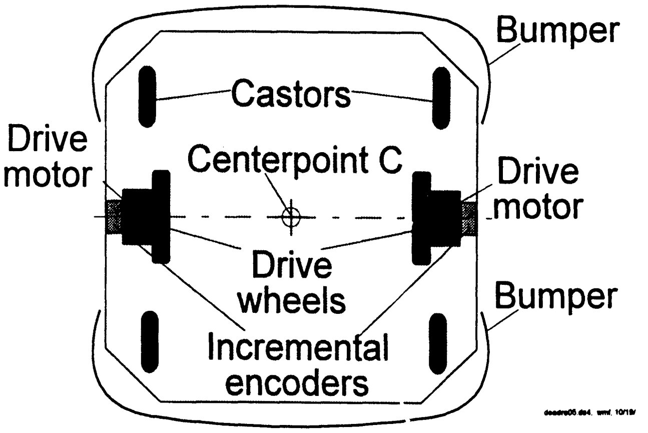

Figure 2.1: A typical differential-drive mobile robot (bottom view).

Suppose that at sampling interval I the left and right wheel encoders show a pulse increment of N_{L} and N_{R}, respectively. Suppose further that

$$c_{\mathrm{m}} = \pi D_{\mathrm{n}}/nC_{\mathrm{e}} \tag{2.1}$$

where

c_{m} - Conversion factor that translates encoder pulses into linear wheel displacement.

D_{n} - Nominal wheel diameter (in mm).

C_{e} - Encoder resolution (in pulses per revolution).

n - Gear ratio of the reduction gear between the motor (where the encoder is attached) and the drive wheel.

We can compute the incremental travel distance for the left and right wheel, $\Delta U_{\mathrm{L,I}}$ and $\Delta U_{\mathrm{R,I}}$, according to

$$\Delta U_{\mathrm{L/R,I}} = c_{\mathrm{m}} N_{\mathrm{L/R,I}} \tag{2.2}$$

and the incremental linear displacement of the robot's centerpoint C, denoted ΔU_i, according to

$$\Delta U_i = (\Delta U_{\mathrm{R,i}} + \Delta U_{\mathrm{L,i}})/2 \tag{2.3}$$

Next, we compute the robot's incremental change of orientation

$$\Delta\theta_i \;= (\Delta U_{\mathrm{R,i}} - \Delta U_{\mathrm{L,i}})/b \tag{2.4}$$

where b is the wheelbase of the vehicle, ideally measured as the distance between the two contact points between the wheels and the floor.

The robot's new relative orientation θ_i can be computed from

$$\theta_i = \theta_{i-1} + \Delta\theta_i \tag{2.5}$$

and the relative position of the centerpoint is

$$x_i = x_{i-1} + \Delta U_i \cos\theta_i \tag{2.6a}$$
$$y_i = y_{i-1} + \Delta U_i \sin\theta_i \tag{2.6b}$$

where
x_i, y_i - relative position of the robot's centerpoint c at instant I.

As one can se: from Eqs. (2.1) - (2.6), dead-reckoning is based on simple equations that are easily implement, and that utilize data from inexpensive incremental wheel encoders. However, dead-reckoning is based on the assumption that wheel revolutions can be translated into linear displacement relative to the floor. This assumption is only of limited validity. One extreme example is wheel slippage: If one wheel was to slip on, say, an oil spill, then the associated encoder would register wheel revolutions even though these revolutions would not correspond to a linear displacement of the wheel.

Besides this extreme case of total slippage, there are several other, more subtle reasons for inaccuracies in the translation of wheel encoder readings into linear motion. All of these error sources fit into one of two categories: (1) systematic errors and (2) non-systematic errors.

1. Systematic errors
a. Unequal wheel diameters
b. Average of both wheel diameters differs from nominal diameter
c. Misalignment of wheels
d. Uncertainty about the effective wheelbase (due to non-point wheel contact with the floor)
e. Limited encoder resolution
f. Limited encoder sampling rate

2. Non-systematic errors
a. Travel over uneven floors
b. Travel over unexpected objects on the floor
c. Wheel-slippage due to:

- slippery floors
- over-acceleration
- fast turning (skidding)
- external forces (interaction with external bodies)
- internal forces (e.g., castor wheels)
- non-point wheel contact with the floor

Systematic errors are particularly grave, because they accumulate constantly. On most smooth indoor surfaces systematic errors contribute much more to dead-reckoning errors than non-systematic errors. However, on rough surfaces with significant irregularities, non-systematic errors may be dominant.

2.1 Non-Systematic Dead-reckoning Errors

Non systematic dead-reckoning errors are those errors that are caused by interaction of the robot with unpredictable features of the environment. For example, irregularities of the floor surface, such as bumps, cracks, or debris, will cause a wheel to rotate more than predicted by Eq. (2.2), because the affected wheel travels up or down the irregularity, in addition to the — expected — horizontal amount of travel. Non-systematic errors are a great problem for actual applications, because it is impossible to predict an upper bound for the dead-reckoning error. Recent work at the University of Michigan [Borenstein, 1994a; 1994b; 1995] showed that by using redundant encoder data, non-systematic errors can be reduced by orders of magnitude.

2.2 Systematic dead-reckoning errors

Systematic errors are usually caused by imperfections in the design and mechanical implementation of a mobile robot. In the course of over 12 years of experimental work with differential-drive mobile robots we observed that the two most notorious systematic error sources are *unequal wheel diameters* and the *uncertainty about the effective wheelbase*. This opinion is reflected in the literature, where these two error sources are named most often [Borenstein and Koren, 1985; 1987; Crowley, 1989; Komoriya and Oyama, 1994; Everett, 1995].

a) Unequal wheel diameters. Most mobile robots use rubber tires to improve traction. These tires are difficult to manufacture to exactly the same diameter. Furthermore, rubber tires compress differently under asymmetric load distribution. Either one of these effects can cause substantial dead-reckoning errors.

b) Uncertainty about the wheelbase. The wheelbase is defined as the distance between the contact points of the two drive wheels of a differential-drive robot and the floor. The wheelbase must be known in order to compute the number of differential encoder pulses that correspond to a certain amount of rotation of the vehicle. Uncertainty in the effective wheelbase is caused by the fact that rubber tires contact the floor not in one point, but rather in a contact area. The resulting uncertainty about the effective wheelbase can be on the order of 1% in some commercially available robots.

In conventional mobile robots systematic errors can be reduced to some degree by careful mechanical design of the vehicle and by vehicle-specific calibration. In this paper we introduce new methods for finding and implementing such calibration factors.

2.3 Definition of systematic dead-reckoning errors

Systematic errors, as explained above, are vehicle-specific and don't usually change during a run (although different load distributions can change some systematic errors quantitatively). Thus, dead-reckoning can be improved generally (and in our experience, significantly) by measuring the individual contribution of the most dominant errors sources, and then counter-acting their effect in software.

As mentioned before, the two dominant systematic dead-reckoning error sources are *unequal wheel diameters* and the *uncertainty about the wheelbase*. We will denote these errors E_d and E_b, respectively. It is important to note that E_b has an effect only when turning, while E_d affects only straight line motion. E_d and E_b are dimensionless values, expressed as fractions of the nominal value.

Specifically, we define

$$E_d = D_R/D_L \tag{2.7}$$

where D_R and D_L are the *actual* wheel diameters.

The *nominal* ratio between the wheel diameters is of course 1.00. We also define

$$E_b = b_{actual}/b_{nominal} \tag{2.8}$$

where b is the wheelbase of the vehicle.

At this time we have defined only the wheelbase error, E_b, and the *ratio* between actual wheel diameters, E_d, as relevant factors. However, if the *average* of the two actual wheel diameters, denoted D_a, differs from the nominal wheel diameter, denoted D_n, then the vehicle will experience an additional dead-reckoning error, which we call the *scaling error* E_s. E_s affects straight-line motion and, according to Appendix A, pure turning motion. However, even though E_s *can* be a significant error, E_s is exceedingly easy to measure with just an ordinary tape measure. For this reason we will assume that E_s has been measured and corrected in software before any of the procedures described in this paper is performed. Consequently we don't consider E_s a dominant error, because even with a cheap tape measure E_s can be measured and corrected with an accuracy of 0.3-0.5% of full scale. For completeness, Appendix B describes a simple procedure for doing so and explains why the accuracy of a tape measure is sufficient for the task.

3. MEASURING DEAD-RECKONING ERRORS

In this section we investigate two methods for isolating and measuring systematic dead-reckoning errors. We discuss two test sequences (benchmark tests), which allow the experimenter to draw conclusions about the overall dead-reckoning accuracy of the robot, and to compare the performance of different mobile robots from different manufacturers.

The first benchmark test is called the "uni-directional square path" test. This test, or some variations of this test, have been mentioned in the literature [Cybermotion, 1987; Komoriya and Oyama, 1994], but we will show that this test is unsuitable for differential drive vehicles. To overccme the shortcomings of the uni-directional square path test, we introduce in Section 3.2 the "bi-directional square path test," called "UMBmark." In Section 3.3 we discuss the (limited) applicability of the UMBmark test to the measurement of *non-systematic* errors. Section 3.4 summarizes the UMBmark procedure.

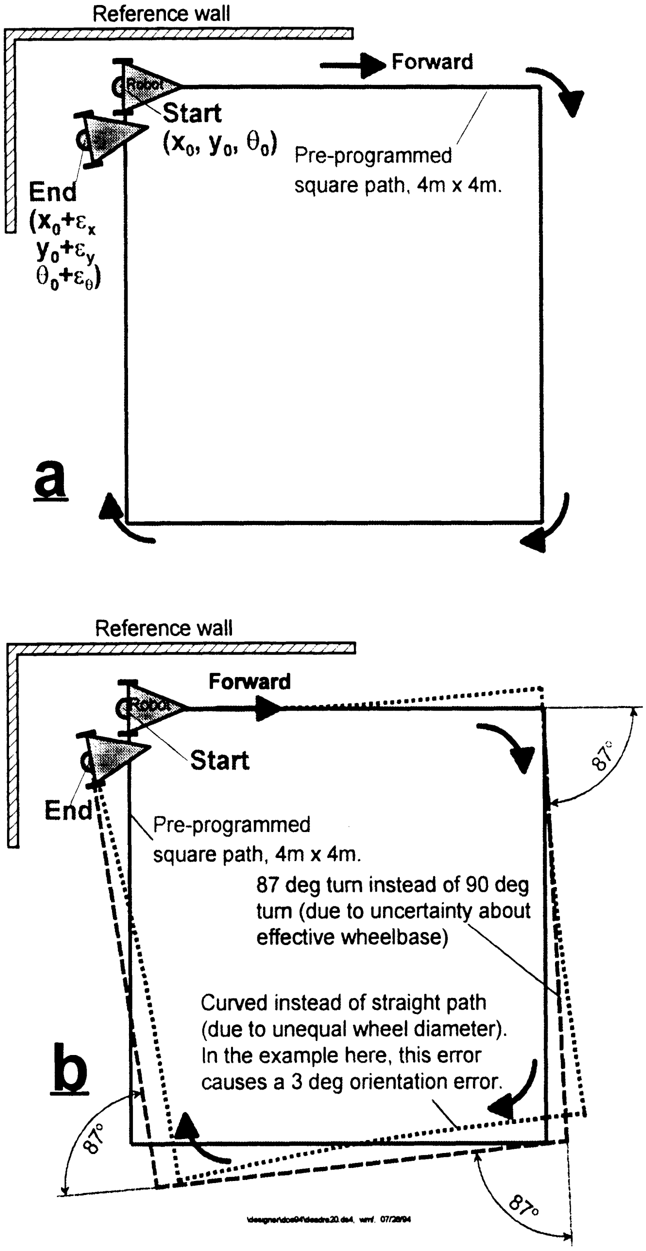
Reference wall
Forward
Robot
Start
(x_0, y_0, θ_0)
Pre-programmed square path, 4m x 4m.
End
$(x_0+\varepsilon_x$
$y_0+\varepsilon_y$
$\theta_0+\varepsilon_\theta)$
a
Reference wall
Forward
Robot
Start
End
Pre-programmed square path, 4m x 4m.
87 deg turn instead of 90 deg turn (due to uncertainty about effective wheelbase)
Curved instead of straight path (due to unequal wheel diameter). In the example here, this error causes a 3 deg orientation error.
87°
87°
87°
b

Figure 3.1: The unidirectional square path experiment.
a. The nominal path.
b. Either one of the two significant errors E_b or E_d can cause the same final position error.

3.1 The uni-directional square path as a benchmark test

Figure 3.1a shows a 4×4 m uni-directional square path. The robot starts out at a position x_0, y_0, θ_0, which is labeled START. The starting area should be located near the corner of two perpendicular walls. The walls serve as a fixed reference before and after the run: measuring the distance between three specific points on the robot and the walls allows accurate determination of the robot's absolute position and orientation.

The robot is programmed to traverse the four legs of the square path. The path will return the vehicle to the starting area, but, because of dead-reckoning and controller errors, not precisely to the starting position. Since this test aims at determining dead-reckoning errors and not controller errors, the vehicle does not need to be programmed to return to its starting position precisely — returning approximately

to the starting area is sufficient. Upon completion of the square path, the experimenter again measures the absolute position of the vehicle, using the fixed walls as a reference. These absolute measurements are then compared to the position and orientation of the vehicle as computed from dead-reckoning data. The result is a set of *return position errors* caused by dead-reckoning and denoted εx, εy, and $\varepsilon\theta$.

$$\begin{aligned}\varepsilon x &= x_{abs} - x_{calc} \\ \varepsilon y &= y_{abs} - y_{calc} \\ \varepsilon\theta &= \theta_{abs} - \theta_{calc}\end{aligned} \tag{3.1}$$

where

εx, εy, $\varepsilon\theta$ — Position and orientation errors due to dead-reckoning.
x_{abs}, y_{abs}, θ_{abs} — Absolute position and orientation of the robot.
x_{calc}, y_{calc}, θ_{calc} — Position and orientation of the robot as computed from dead-reckoning.

The path shown in Fig. 3.1a comprises of four straight line segments and four pure rotations about the robot's centerpoint, at the corners of the square. The robot's end position shown in Fig. 3.1a visualizes the dead-reckoning error.

While analyzing the results of this experiment, the experimenter may draw two different conclusions: (1) The dead-reckoning error is the result of unequal wheel diameters, E_d, as shown by the slightly curved trajectory in Fig. 3.1b (dotted line); or, (2) the dead-reckoning error is the result of uncertainty about the wheelbase, E_b. In the example of Fig. 3,1b, E_b caused the robot to turn 87° instead of the desired 90° (dashed trajectory in Fig. 3.1b).

As one can see in Fig. 3.1b, either one of these two cases *could* yield approximately the same position error. The fact that two different error-mechanisms might result in the same overall error may lead an experimenter toward a serious mistake: correcting only one of the two error sources in software. This mistake is so serious because it will yield apparently "excellent" results, as shown in the example in Fig. 3.2. In this example, we assume that the experimenter began "improving" performance by adjusting the wheelbase *b* in the control software. According to Eq. (2.4), the experimenter needs only to increase the value of *b* to make the robot turn more in each nominal 90° turn. In doing so, the experimenter will soon have adjusted *b* to the "ideal" value that will cause the robot to turn 93°, thereby effectively compensating for the 3° orientation error introduced by each slightly curved (but nominally straight) leg of the square path.

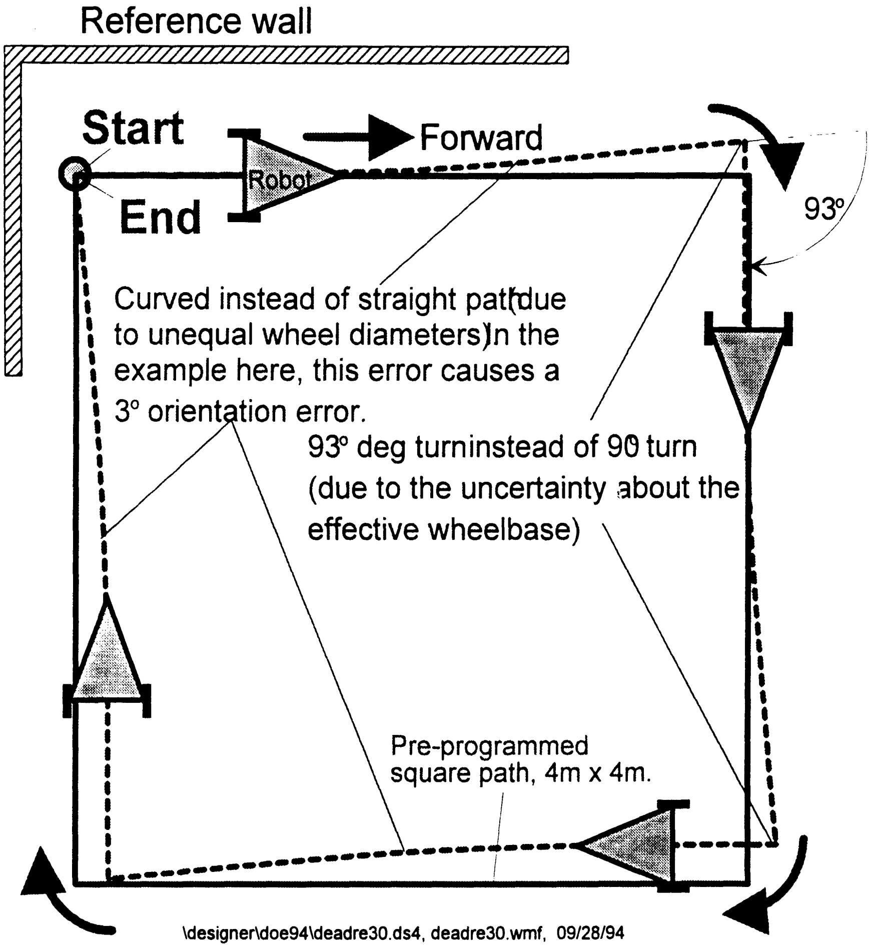
Reference wall
Start
Forward
Robot
End
93°
Curved instead of straight pat(due to unequal wheel diameters)In the example here, this error causes a 3° orientation error.
93° deg turninstead of 90 turn (due to the uncertainty about the effective wheelbase)
Pre-programmed square path, 4m x 4m.
\designer\doe94\deadre30.ds4, deadre30.wmf, 09/28/94

Figure 3.2: The effect of the two dominant systematic dead-reckoning errors E_b and E_d. Note how both errors may cancel each other out when the test is performed in only one direction.

We should note that another popular test path, the "figure-8" path [Tsumura et al., 1981; Borenstein and Koren, 1985; Cox 1991] can be shown to have the same shortcomings as the uni-directional square path.

3.2 The bi-directional square path experiment: "UMBmark"

The detailed example of the preceding section illustrates that the uni-directional square path experiment is unsuitable for testing dead-reckoning performance, because it can easily conceal two mutually compensating dead-reckoning errors. To overcome this problem, we introduce the *Bi-directional Square Path* experiment, called *University of Michigan Benchmark* (UMBmark). UMBmark requires that the square path experiment is performed in both clockwise and counter-clockwise direction. Figure 3.3 shows that the concealed dual-error from the example in Fig. 3.2 becomes clearly visible when the square path is performed in the opposite direction. This is so because the two dominant systematic errors, which may compensate for each other when run in only one direction, add up to each other and increase the overall error when run in the opposite direction.

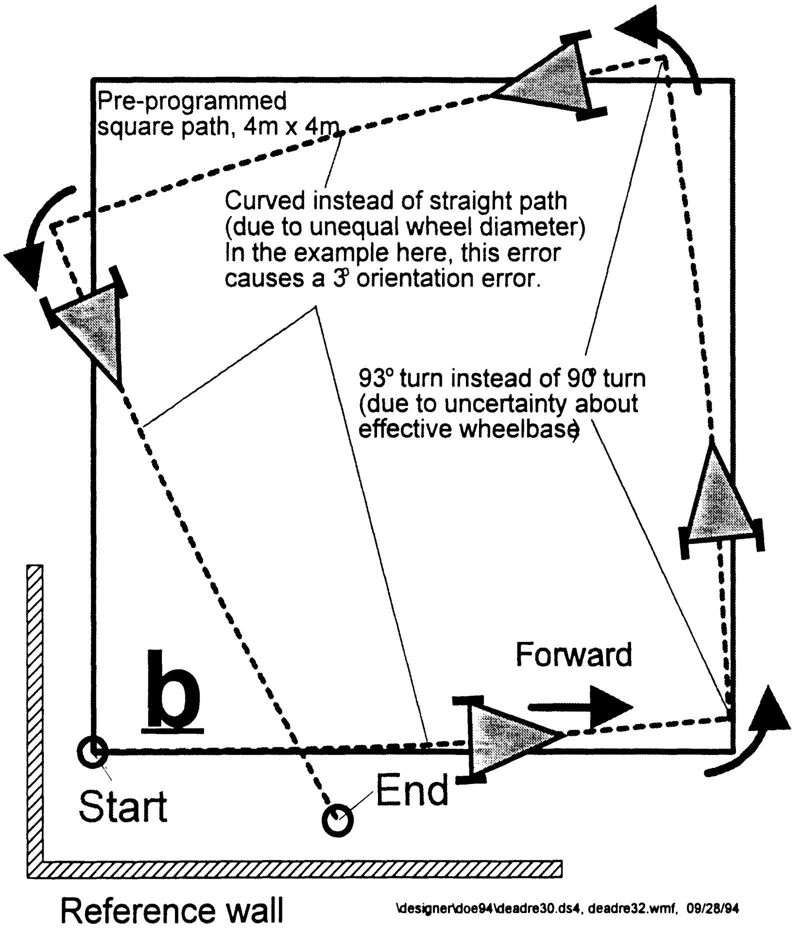
Pre-programmed
square path, 4m x 4m
Curved instead of straight path
(due to unequal wheel diameter)
In the example here, this error
causes a 3° orientation error.
93° turn instead of 90° turn
(due to uncertainty about
effective wheelbase)
b
Forward
Start
End
Reference wall
\designer\doe94\deadre30.ds4, deadre32.wmf, 09/28/94

Figure 3.3: The effect of the two dominant systematic dead-reckoning errors E_b and E_d : When the square path is performed in the opposite direction one may find that the errors add up.

The result of the *Bi-directional Square Path* experiment might look similar to the one shown in Fig. 3.4, which shows actual results with an off-the-shelf LabMate robot carrying an evenly distributed load. In this experiment the robot was programmed to follow a 4×4 m square path, starting at (0,0). The stopping positions for five runs each in clockwise (cw) and counter-clockwise (ccw) directions are shown in Fig. 3.4. Note that Fig. 3.4 is an *enlarged view* of the target area. The results of Fig. 3.4 can be interpreted as follows:

a. The stopping positions after cw and ccw runs are *clustered* in two distinct areas.

b. The distribution within the cw and ccw clusters are the result of *non-systematic* errors, as mentioned in Section 2.2. However, Fig. 3.4 shows that in an uncalibrated vehicle, traveling over a reasonably smooth concrete floor, the contribution of *systematic* errors to the total dead-reckoning error is notably larger[1] than the contribution of non-systematic errors.

[1] In informal tests with two other LabMate robots at our lab we have observed (but not methodically noted) greater systematic dead-reckoning errors than those in Fig. 3.4. These may be due to less balanced load-distributions in those earlier tests.

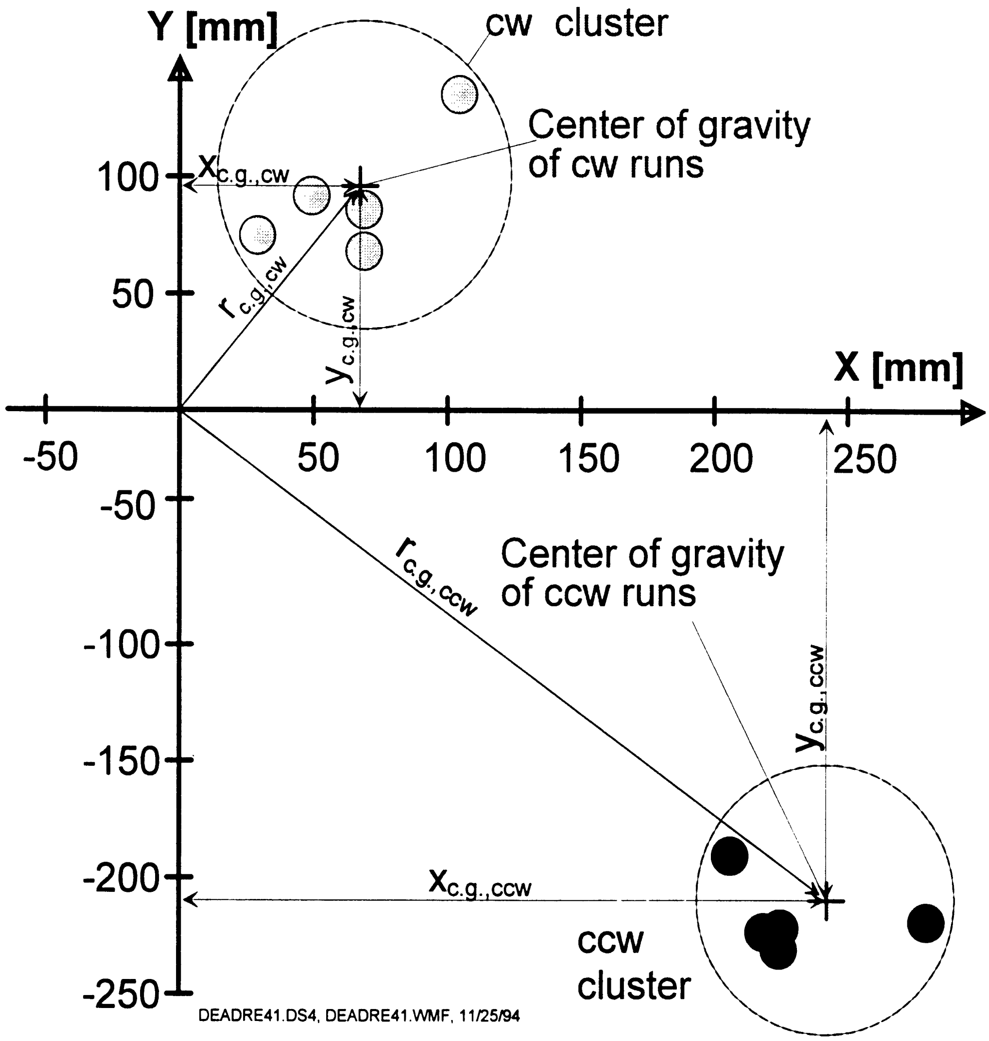

Figure 3.4: Typical results from running UMBmark (a square path run in both cw and ccw directions) with an uncalibrated vehicle.

After conducting the UMBmark experiment, one may wish to derive a single numeric value that expresses the dead-reckoning accuracy (with respect to systematic errors) of the tested vehicle. In order to minimize the effect of non-systematic errors, we suggest to consider the *center of gravity* of each cluster as representative for the dead-reckoning errors in cw and ccw directions.

The coordinates of the two centers of gravity are computed from the results of (3.1) as

$$x_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} \epsilon x_{i,cw/ccw}$$

$$y_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} \epsilon y_{i,cw/ccw} \qquad (3.2)$$

where $n = 5$ is the number of runs in each direction.

The absolute offsets of the two centers of gravity from the origin are denoted $r_{\text{c.g., cw}}$ and $r_{\text{c.g., ccw}}$ (see Fig. 3.4) and are given by

$$r_{c.g.,cw} = \sqrt{(x_{c.g.,cw})^2 + (y_{c.g.,cw})^2}$$

$$\text{and} \qquad (3.3)$$

$$r_{c.g.,ccw} = \sqrt{(x_{c.g.,ccw})^2 + (y_{c.g.,ccw})^2}$$

Finally, we define the larger value among $r_{\text{c.g., cw}}$ and $r_{\text{c.g., cw}}$ as the *measure of dead-reckoning accuracy for systematic errors*

$$E_{\text{max,syst}} = \max(r_{\text{c.g.,cw}}\, ; r_{\text{c.g.,ccw}}) \qquad (3.4)$$

The reason for not using the *average* of the two centers of gravity $r_{\text{c.g.,cw}}$ and $r_{\text{c.g.,ccw}}$ is that for practical applications, one needs to worry about the *largest* possible dead-reckoning error. Note that the final orientation error $\varepsilon\theta$ is not considered explicitly in the expression for $E_{\text{max,syst}}$. This is so because all systematic orientation errors are *implied* by the final position errors. In other words, since the square path has fixed-length sides, systematic orientation errors translate directly into position errors (as will be shown by Eq. (6.10) and (6.18) in Section 6).

3.4 Measuring Non-systematic Errors

Some limited information about a vehicle's susceptibility to non-systematic errors can be derived from the *spread* of the *return position errors* that was shown in Fig. 3.4, above. When running the UMBmark procedure on smooth floors (e.g., a concrete floor without noticeable bumps or cracks), an *indication* of the magnitude of the non-systematic errors can be obtained from computing the *estimated standard deviation*, σ. We will list σ in Section 4 (Experiments) along with other experimental results, but only with the disclaimer that all runs were performed on the same floor, which was fairly smooth and guaranteed free of large irregularities.

We caution that there is only limited value to knowing σ, since σ reflects only on the interaction between the vehicle and a certain floor. Furthermore, it can be shown that from comparing the σ from two different robots (even if they traveled on the same floor), one cannot necessarily conclude that the robots with the larger σ showed higher susceptibility to non-systematic errors.

In real applications it is imperative that the *largest possible disturbance* be determined and used in testing. For example, the σ of the test in Fig. 3.4 gives no indication at all as to what error one should expect if one wheel of the robot inadvertently traversed a large bump or crack in the floor.

For the above reasons it is difficult (perhaps impossible) to design a generally applicable quantitative test procedure for *non-systematic* errors. However, we would like to propose an easily

reproducible test that would allow to compare the susceptibility to non-systematic errors between different vehicles. This test, here called the *extended UMBmark*, uses the same bi-directional square path as UMBmark, but, in addition, introduces artificial bumps. Artificial bumps are introduced by means of a common, round, electrical household-type cable (such as the ones used with 15 Amp. 6-outlet power strips). Such a cable has a diameter of about 9-10 mm. It's rounded shape and plastic coating allow even smaller robots to traverse it without too much physical impact. In the proposed *extended UMBmark* test the cable is placed 10 times under one of the robot's wheels, during motion. In order to provide better repeatability for this test, and to avoid mutually compensating errors, we suggest that these 10 bumps be introduced as evenly as possible. The bumps should also be introduced during the first straight segment of the square path, and always under the wheel that faces the inside of the square. It can be shown [Borenstein, 1994b] that the most noticeable effect of each bump is a fixed orientation error in the direction of the wheel that encountered the bump. In the TRC LabMate, for example, the orientation error resulting from a bump of height h= 10 mm is roughly $\Delta\theta = 0.44°$ [Borenstein, 1994b].

In November 1994 Brad Holt from the University of Michigan's Mobile Robotics Lab talked to John Kennerly at ORNL about the expected floor characteristics at potential D&D application sites. The following are Brad Holt's notes from this conversation:

The floors are cement with an epoxy-coated surface that is "like an old car finish."

There are expansion joints along the column lines, (approximately 30×30 feet) and can be as deep as 1/4 inch. Nonstandard cracks are rare and would be "hairline."

The buildings are uninhabited, but are regularly surveyed and maintained. The odds of encountering chunks of debris such as wrenches or wires or even papers would be very rare if at all. In heavy rains, however, the roofs can leak, leaving puddles that are cleaned up after the storm passes. The floors are no longer swept regularly.

In general, the floors are "very clean", and "look good," and were I to view them, I would not know that they were unused, or not swept daily.

The specific buildings referred to were K31 and K33. K25 is "dirty," K27 and 29 are better, but are smaller and have a higher frequency of stairs, columns, and other obstacles.

Sidebar 1: Notes on expected floor characteristics at potential application sites.

Next, we need to discuss which measurable parameter would be the most useful one for expressing the vehicle's susceptibility to non-systematic errors. Consider, for example, Path A and Path B in Fig. 3.5. If the 10 bumps required by the extended UMBmark test were concentrated at the beginning of the first straight leg (as shown in exaggeration in Path A), then the *return position error* would be very small. Conversely, if the 10 bumps were concentrated toward the end of the first straight leg (Path B in Fig. 3.5), then the *return position error* would be larger. Because of this sensitivity of the *return position errors* to the exact location of the bumps it is not a good idea to use the *return position error* as an indicator for a robot's susceptibility to non-systematic errors. Instead, we suggest to use the *return orientation error*, $\varepsilon\theta$. Although it is more difficult to measure small angles, we found measurement of $\varepsilon\theta$ to be a more consistent quantitative indicator for comparing the performance of different robots. Thus, we measure and express the susceptibility of a vehicle to non-systematic errors in terms of the its *average absolute orientation error* defined as

$$\epsilon\theta^{nonsys}_{avrg} = \frac{1}{n}\sum_{i=1}^{n} |\epsilon\theta^{nonsys}_{i,cw} - \epsilon\theta^{sys}_{avrg,cw}| + \frac{1}{n}\sum_{i=1}^{n} |\epsilon\theta^{nonsys}_{i,ccw} - \epsilon\theta^{sys}_{avrg,ccw}| \tag{3.5}$$

where $n = 5$ is the number of experiments in cw or ccw direction, superscripts "*sys*" and "*nonsys*" indicate a result obtained from either the regular UMBmark test (for systematic errors) or from the *extended* UMBmark test (for non-systematic errors). Note that Eq. (3.5) improves on the accuracy in identifying non-systematic errors by removing the systematic bias of the vehicle, given by

$$\begin{aligned} \epsilon\theta^{sys}_{avrg,cw} &= \frac{1}{n}\sum_{i=1}^{n} \epsilon\theta^{sys}_{i,cw} \\ \epsilon\theta^{sys}_{avrg,ccw} &= \frac{1}{n}\sum_{i=1}^{n} \epsilon\theta^{sys}_{i,ccw} \end{aligned} \tag{3.6}$$

Note that the arguments inside the *Sigmas* in Eq. (3.5) are absolute values of the bias-free *return orientation errors*. This is so because we want to avoid the case in which two return orientation errors of opposite sign cancel each other out. For example, if in one run $\varepsilon\theta = 1°$ and in the next run $\varepsilon\theta = -1°$, then we should not conclude that $\epsilon\theta^{nonsys}_{avrg} = 0$. Using the average absolute return error as computed in Eq. (3.5) would correctly compute $\epsilon\theta^{nonsys}_{avrg} = 1$. By contrast, in Eq. (3.6) we compute the actual arithmetic average, because we want to identify a fixed bias.

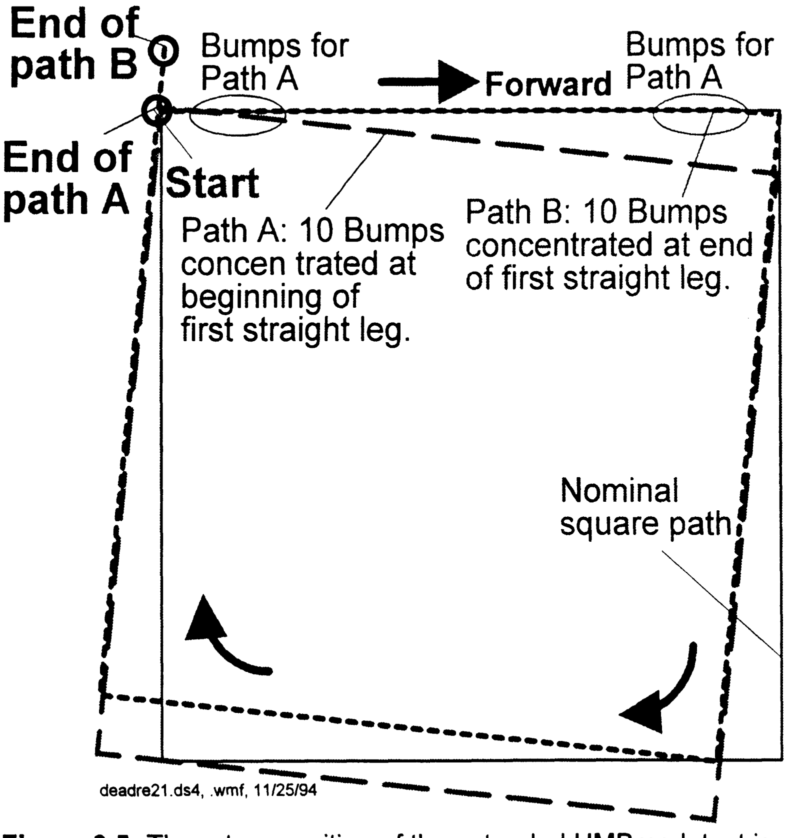
End of path B
Bumps for Path A
Forward
Bumps for Path A
End of path A
Start
Path A: 10 Bumps concen trated at beginning of first straight leg.
Path B: 10 Bumps concentrated at end of first straight leg.
Nominal square path
deadre21.ds4, .wmf, 11/25/94

Figure 3.5: The return position of the extended UMBmark test is sensitive to the exact location where the 10 bumps were placed. The return *orientation* is not.

3.5 Summary of the UMBmark Procedure

In summary, the UMBmark procedure is defined as follows:

1. At the beginning of the run, measure the absolute position (and, if measurement of non-systematic errors is planned, orientation) of the vehicle and initialize the onboard dead-reckoning starting position to that position.

2. Run the vehicle through a 4×4 m square path in cw direction, making sure to
   - stop after each 4 m straight leg;
   - make a total of four 90°-turns on the spot;
   - run the vehicle slowly to avoid slippage.

3. Upon return to the starting area, measure the absolute position (and, optionally, orientation) of the vehicle.

4. Compare the absolute position to the robot's *calculated* position, based on dead-reckoning and using Eqs. (1).

5. Repeat steps 1-4 for four more times (i.e., a total of five runs).

6. Repeat steps 1-5 in **ccw** direction.

7. Use Eqs. (2) and (3) to express the experimental results quantitatively as the *measure of dead-reckoning accuracy for systematic errors,* $E_{\max,\text{syst}}$.

8. Optionally, use a plot similar to Fig. 3.4 to represent εx_i and εy_i graphically.

9. If an estimate for the vehicle's susceptibility to non-systematic errors is needed, then perform steps 1-6 again, this time placing a round 10 mm diameter object (for example, an electrical household cable) under the inside wheel of the robot. The object must be placed there 10 times, during the first leg of the square path.

10. Compute the *average absolute orientation error*, $\epsilon\theta_{avrg}^{nonsys}$ according to Eqs. (5) and (6).

4. Measuring Systematic Errors — Experiments

In this section we present experimental results from testing five different mobile robot systems with the UMBmark procedure. A sixth system, called the *Smart Encoder Trailer* (SET), is currently under development at the University of Michigan. Results from *simulations* of the SET are included in this chapter. The six tested systems were:

1. TRC LabMate
2. Cybermotion K2A
3. CLAPPER — a unique 4-degree-of-freedom (4-DOF) platform developed at the University of Michigan [Borenstein, 1994a, 1994b, 1995].
4. Remotec Andros
5. Remotec Andros with the *Basic Encoder Trailer*
6. Smart Encoder Trailer (SET) — Simulation results only

Table I below summarizes the properties of the six different vehicles that were tested, and the following sections discuss each vehicle and result in detail.

4.1 TRC LabMate

Table I: Summary of properties and UMBmark results for the six different vehicles tested

Tested vehicle			Result in [mm]	
Platform Name	Comment	Calibration	$E_{max,syst}$	σ
1. TRC LabMate	none	none (b=340.0, D_R/D_L=1)	310	50
2.Cybermotion K2A	Slightly worn-out, in service since 1987	Original, from manufacturer	63	60
3. University of Michigan CLAPPER	4-DOF vehicle, made from 2 TRCs with compliant linkage	yes. manual	22	11
4. Remotec Andros	Tracked vehicle	none	11,500 (11.5m)	6,000 (6m)
5. Andros with Basic Encoder Trailer	Tracked vehicle, wheeled trailer	Trial and error	74	200
6. Smart Encoder Trailer	Simulation results only	N/A in simulation	30	5

the nominal wheelbase b=340 mm and a wheel diameter ration of D_R/D_L = 1.000). The LabMate shown in Fig. 4.1 is equipped with ultrasonic sensors that were not used in this experiment. An onboard 486/50 MHZ PC compatible single board computer controls the LabMate. On our LabMate platforms we bypass TRC's original onboard control computer completely. This is done by means of a set of two HCTL 1100 [Hewlett Packard] motion control chips that connect our 486 computer directly to the motors' PWM amplifiers. Generally we do this in order to achieve a very fast control loop, one that is not impeded by the relatively slow serial interface required by the original onboard computer. In the particular case of the UMBmark experiments described here, the bypass assures that the measurements are not affected by the manufacturer's dead-reckoning method and, possibly, software-embedded calibration factors. However, we emphasize that our bypass of the original onboard computer is in no way necessary for performing the UMBmark procedure.

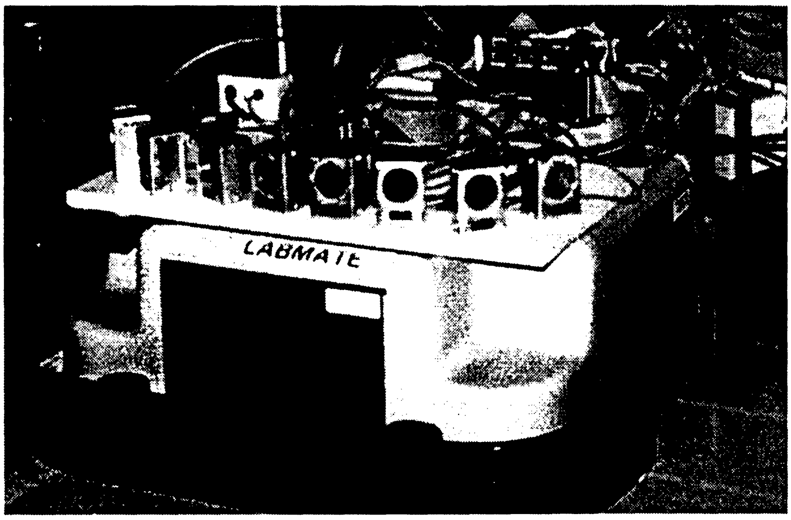

Figure 4.1: One of the four TRC LabMates at the University of Michigan. The system shown here is equipped with 8 ultrasonic sensors that were not used in the UMBmark experiment.

In the 4×4 m square path experiments, the robot traveled at 0.2 m/s during the four 4 m straight legs of the path and stopped before turning. During the four *on-the-spot* turns the robot's wheels had a maximum linear speed of ±0.2 m/s. Figure C-1 (in Appendix C) shows the *return position errors* (defined in Section 3.1.) for the unmodified/uncalibrated TRC LabMate. In this test $E_{\text{max,sys}}$ = 310 mm and σ = 50 mm.

4.2 Cybermotion

The Cybermotion K2A platform is a smart implementation of the synchro-drive (see [Everett, 1995] for a more detailed discussion on synchro-drives). We believe that the implementation of the synchro-drive on the Cybermotion K2A provides the inherently best dead-reckoning performance among all *commonly* used mobile robot drive kinematics. This is especially true with regard to *non-systematic* errors. For example, if the K2A encounters a bump on the ground, then the wheel in contact with the bump would have to turn slightly more than the other two wheels. However, since all the wheels are powered by the same motor and have the same speed, the wheel on the bump will

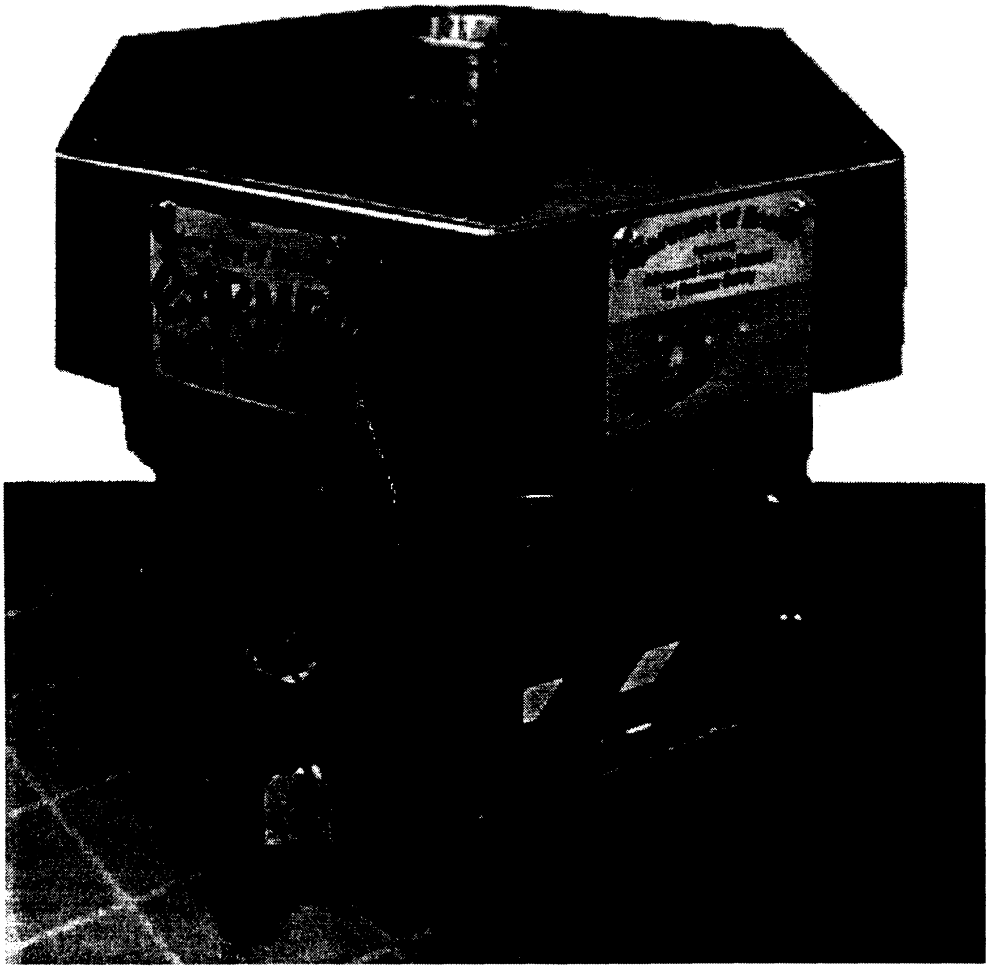

Figure 4.2: CARMEL, the University of Michigan's oldest mobile robot has been in service since 1987. CARMEL is a Cybermotion K2A platform with many custom additions. In 1992 CARMEL was voted one of the 100 "Best of What's New" products in *Popular Science*.

slip (at least, this is more likely than to assume that *both* other wheels on the ground will slip). Thus, if slippage occurs in the wheel that is "off," then the dead-reckoning information from the "correct" wheels remains valid, and only a small error (if at all) is incurred.

The Cybermotion K2A platform shown in Fig. 4.2 is called CARMEL. CARMEL was the first robot to be placed into service at the University of Michigan's Mobile Robotics Lab when the lab was created in 1987. Since then, CARMEL has had many collisions, was disassembled several times, and has survived generally rough treatment. For these reasons, one should regard CARMEL's UMBmark performance with caution. In our test, we found $E_{\mathrm{max,sys}}$ = 63 mm and σ = 60 mm. Although we have not studied in depth the kinematics of the K2A with regard to systematic errors, we believe that it is susceptible to some of the same systematic errors as differential-drive mobile robots. This is evident from the clearly defined separate clusters for the cw and ccw runs in Fig. C-2 in Appendix C. CARMEL traveled at 0.2 m/s during the four 4 m straight legs of the path and stopped before turning.

4.3 CLAPPER

The *Compliant Linkage Autonomous **P**latform with **P**osition **E**rror **R**ecovery* (CLAPPER) is a 4-Degree-of-Freedom (4DOF) vehicle developed and built at the University of Michigan [Borenstein, 1994a; 1994b, 1995]. The CLAPPER comprises two off-the-shelf TRC LabMates (here called "trucks") connected by a so-called *compliant linkage* (see Fig. 4.3). The vehicle is instrumented with two rotary absolute encoders that measure the rotation of the trucks relative to the compliant linkage. And a linear encoder measures the relative distance between the centerpoints of the two trucks. The unique ability to ***measure and correct*** *non-systematic* dead-reckoning errors during motion. A more detailed description of the CLAPPER is given in Appendix F, where we have included a reprint of reference Borenstein 1994a.

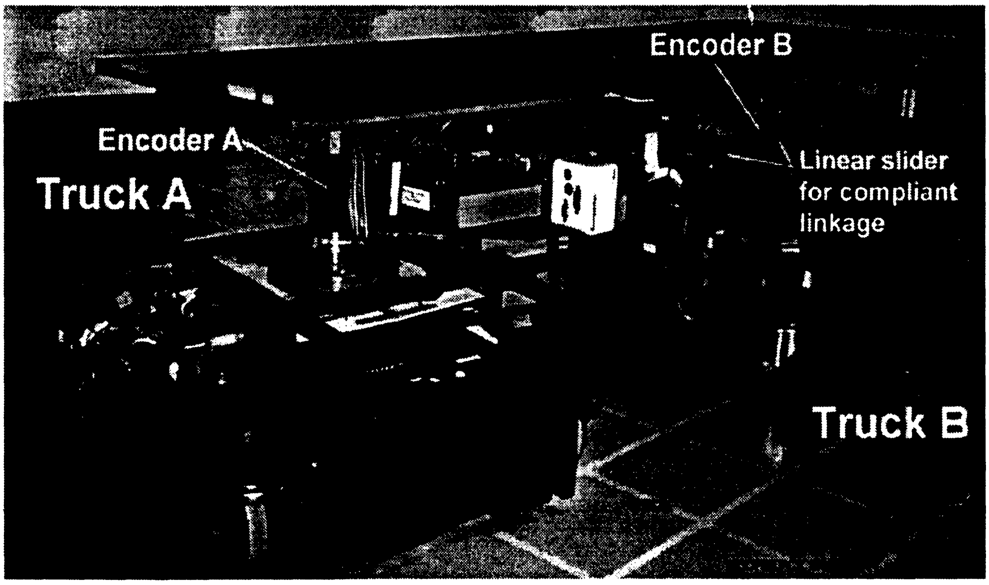

Figure 4.3: The CLAPPER is a unique 4-DOF mobile robot developed at the University of Michigan. The CLAPPER can measure and correct *non-systematic* dead-reckoning errors during motion.

The CLAPPER can also detect and correct the two dominant *systematic* dead-reckoning errors (i.e., unequal wheel-diameters and uncertainty about the effective wheelbase). However, the CLAPPER also introduces some new systematic errors related to its unique configuration (see more detailed explanation in [Borenstein, 1994b]. These new systematic errors were reduced by extensive trial-and-error calibration before running the UMBmark test. Figure C-3 in Appendix C shows the results of the UMBmark test with the CLAPPER. In this test $E_{\mathrm{max,sys}}$ = 22 mm and σ = 11 mm . Note that σ = 11 mm is substantially lower than the results for the other vehicles. This fact demonstrates the successful correction of *non-systematic* errors.

We should note that the test condition for the UMBmark test of the CLAPPER differed somewhat from the exact UMBmark specifications: (a) The test path was of rectangular shape with 7×4 m dimensions; (b) the vehicle also made some additional maneuvers in order to approach the stopping position properly (see [Borenstein, 1994b] for details); (c) The CLAPPER's average speed was

dimensions; (b) the vehicle also made some additional maneuvers in order to approach the stopping position properly (see [Borenstein, 1994b] for details); (c) The CLAPPER's average speed was 0.45 m/s, and (d) and the vehicle did not come to a complete halt before turns. These deviations from the UMBmark specifications are of little impact for a well calibrated system. If these differences had any effect on the CLAPPER's UMBmark performance it was one of deterioration and of not improvement.

Digressing for a moment from this section's focus on *systematic* errors, we observe from comparing Figs. C-2 and C-3 that the *active* error correction of the CLAPPER provides substantially better correction of *non-systematic* errors than the *passive* error correction inherently present in the K2A's synchro-drive. This observation is also supported by the two standard deviations, $\sigma_{CLAPPER}$ = 11 mm and σ_{K2A} = 60 mm, which provide some indication for the magnitude of *non-systematic* errors. However, this comparison is probably slightly biased in favor of the CLAPPER, because of the less-than-perfect physical condition of our K2A platform.

4.4 Remotec Andros

The Remotec Andros (the University of Michigan's modified model is shown in Fig. 4.4) is a uniquely designed tracked vehicle for tele-operation on difficult terrain. The most unique feature of the Andros is the set of two auxiliary tracks that can be raised or lowered individually. These auxiliary tracks allow the vehicle to climb over large obstacles and move up and down stairs. When the auxiliary tracks are lifted the vehicle can turn even in tight spots.

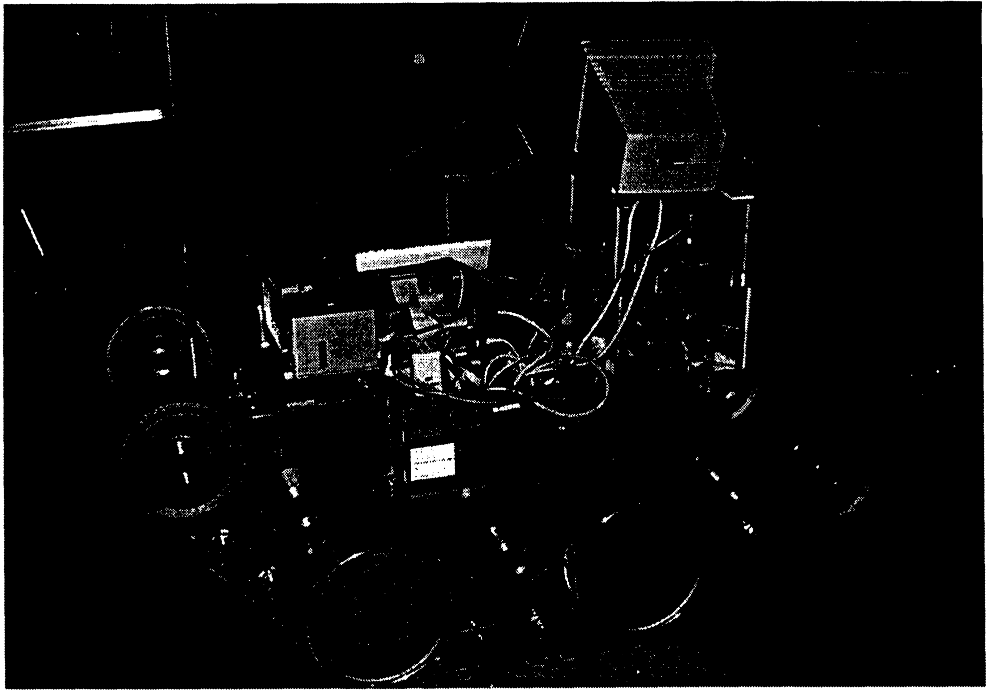

Figure 4.4: The University of Michigan's Remotec Andros is custom equipped with incremental optical encoders on both drive motors and on the auxiliary track motors. The original onboard computer and motor controllers were replaced by a 486-66 MHZ PC-compatible single board computer and our own HCTL 1100-based motor controllers.

While tracked vehicles allow operation in rough terrain, a tracked vehicle is very difficult to control under computer guidance and without a human operator. This is so because the widely used odometry-based position feedback cannot be implemented easily on a tracked vehicle: It is an inherent property of tracked vehicle steering that the tracks slip substantially during turning. Under tele-operator control, the lack of dead-reckoning information is not a problem, because the operator has visual feedback.

In recent years there has been growing interest, especially in the DOE Robotics community, in converting the Remotec Andros and other tracked vehicles to fully autonomous operation. The University of Michigan procured an Andros in 1993, and, in anticipation of autonomous operation, had the vehicle motors custom-equipped with incremental encoders. These encoders provide feedback on track position and rotational velocity for accurate velocity control. One currently ongoing research effort at our lab aims at developing a complex slippage model that would allow to correct — to a limited extend — the huge dead-reckoning errors caused by track slippage during turning (see [Fan et al., 1995] for more details).

To assess the extend of dead-reckoning errors in our encoder-equipped Andros we performed the UMBmark test. However, the nominal test-procedure had to be modified: The vehicle could not be *programmed* to perform a square path, because the dead-reckoning errors were so large that the nominal path could not even be approximated. Instead, we drove the vehicle by joystick through the 4×4 m path and compared its computed (i.e., using dead-reckoning) position with the vehicle's actually measured stopping position. Figure C-4 shows the resulting dead-reckoning errors: Typically, we encountered errors of 10 to 12 meters(!) for a square path of nominally 16 meters total length. The orientation error was typically on the order of 120 - 130°! It is quite clear from these results that odometric dead-reckoning is not a viable option with tracked vehicles. One interesting observation is that many of the run ended with very similar errors, indicating a high repeatability of the error mechanism. However, there were also runs where the final errors differed substantially from the set of "repeatable" errors.

4.5 Remotec Andros with Basic Encoder Trailer

In order to investigate and potentially improve dead-reckoning performance of tracked vehicles, the University of Michigan has developed a unique attachment for the Remotec Andros (or any robot with poor dead-reckoning), called the "Basic Encoder Trailer" (BET) (see [Fan et al., 1995] for more details). The BET differs in function from the "Smart Encoder Trailer," as will be discussed in Section 2.5 below.

The BET is a small two-wheeled trailer that drags behind the main vehicle (see Fig. 4.5). Mounted on each wheel is an optical incremental encoder. A rotary joint allows the trailer to rotate horizontally about a fixed point located behind the pulling vehicle. An optical absolute encoder measures the angle between the trailer and the main vehicle. The trailer can rotate around this joint without limitation because of a slip ring that provides the electrical connections to the wheel

encoders. When the Andros travels backward, the BET is designed to swivel and lead. The BET is also designed to be raised off the floor when the tracked vehicle drives over obstacles, although this function is not yet implemented on our system. Using the same dead-reckoning algorithm as is used for differential-drive robots, one can easily compute the incremental position of the center-point of the trailer. Then, using the angular measurements of the absolute encoder on the trailer/vehicle joint, one needs to apply only a simple coordinate transformation to compute the incremental position of the main vehicle.

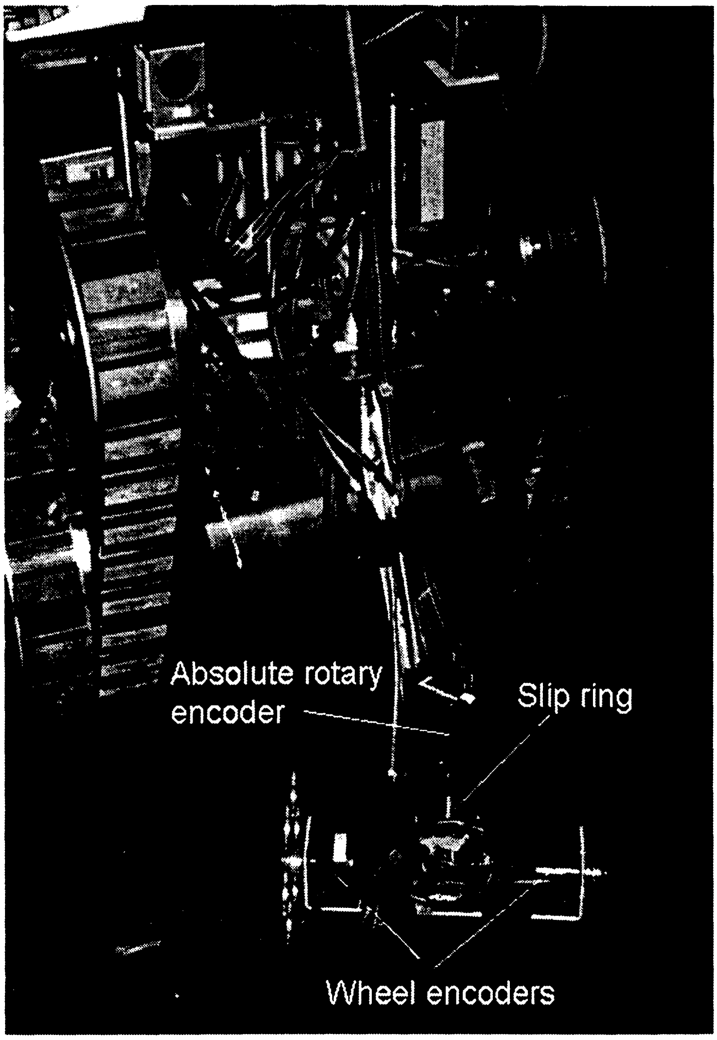

Figure 4.5: This encoder trailer was recently developed at the University of Michigan for use with tracked robots. The trailer is here shown temporarily attached to a TRC LabMate, prior to its permanent installation on the Remotec Andros.

The rational for using a device like the BET, despite its obvious disadvantages, is the fact that many DOE applications might require a tracked vehicle because of occasional obstacles, but would offer relatively smooth concrete floors during most of the robot's travel. For autonomous operation, a "conventional" tracked robot system would require nearly continuous absolute position updates, to provide control feedback. With conventional techniques, this would require the costly installation and maintenance of beacon systems in which several of the beacons can be seen at all times. We believe that the installation cost for such a system could be dramatically lower, if dead-reckoning information was available most of the time.

The dead-reckoning accuracy of the Andros with BET attached was tested with the help of the UMBmark test. The results are shown in Fig. C-5. As expected, the average Return Position Error was relatively small, on the order of 60-70 mm. However, Fig. C-5 shows an unexpected large spread of errors, larger than what could be explained as caused by irregularities on the floor. We will further investigate this matter in the future, but our current assumption is that electrical noise from the low-cost slip ring causes random distortions of the encoder pulses, thus resulting in erratic encoder readings.

4.6 Smart Encoder Trailer

The *Smart Encoder Trailer* (SET) is an innovative method for accurate dead-reckoning with mobile robots. This approach is based on a method called ***Internal Position Error Correction*** (IPEC), which was recently developed at the University of Michigan. This is the same method as the one used on the CLAPPER (discussed in Section 4.3, above). The SET is designed to be attached to an existing mobile robot (see Fig. 4.6), unlike the CLAPPER, which is a special purpose custom-built vehicle.

We have just completed the construction of the SET (see Fig. 4.7), but the system is not functional yet. However, we will present recent simulation results, which, combined with experimental results from the similarly configured CLAPPER, strongly suggest the feasibility of the SET implementation.

4.6.1 Validity of the SET Simulation

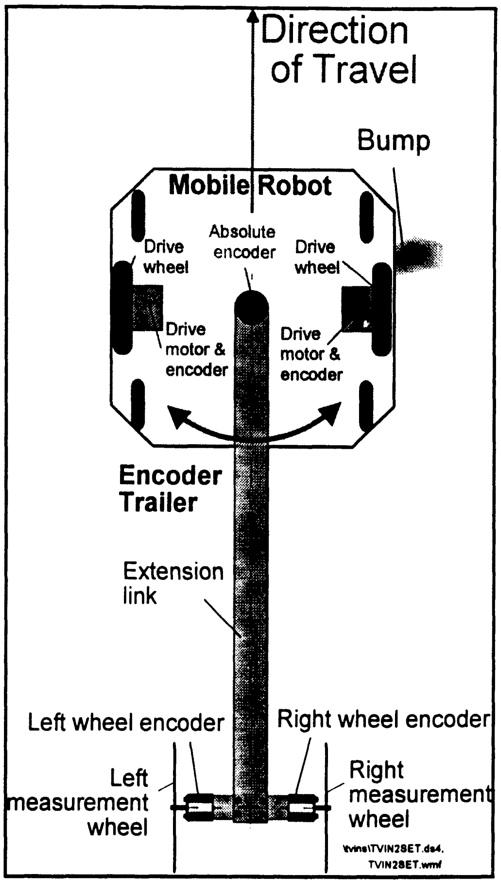

Figure 4.6: Concept drawing of the University of Michigan's *Smart Encoder Trailer* (SET).

Many experimentalists (including the authors here) are often critical of simulation results. This is particularly true when new concepts are tested for which the theoretical

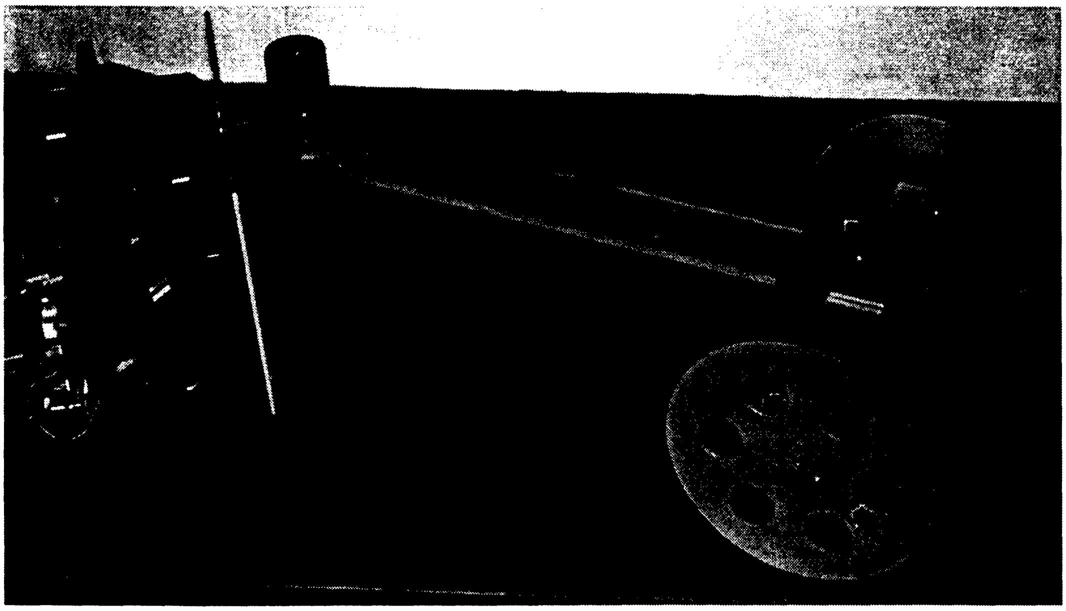

Figure 4.7: Prototype of the University of Michigan's *Smart Encoder Trailer* (SET). In the photo here, the SET is attached to a TRC LabMate, but the SET's encoders are not interfaced yet. Note that the location of the absolute encoder is behind the robot, not at the robot's centerpoint. U of M plans to implement the IPEC method in early 1995. Then, if successful, U of M will develop algorithms designed to allow fully transparent backward travel with a trailer.

models used in the simulation have not been verified experimentally. In the case of our SET simulation, however, we are fairly confident in the validity of our simulation. The reason is as follows: The concept and actual design of the SET described here is directly related to the concept and design of the CLAPPER (see Section 4.3). The CLAPPER is an actually working piece of hardware that has been tested under realistic (in our lab) and under tougher-than-realistic (on the lawn in the backyard of our lab) conditions. In addition, a simulation program for the CLAPPER has been developed, which models the CLAPPER and its method for measuring and correcting dead-reckoning errors very accurately. We have compared and validated the accuracy of results from the CLAPPER simulation against results of the actual experiments with the CLAPPER.

We are explaining all this because the SET simulation in this report is directly derived from the CLAPPER simulatioṅ. Comparing Fig. 4.6 with Figure 2 of the CLAPPER paper in Appendix F, it is quite obvious that the SET is very similar in design yet substantially less complex. For these reasons we are quite confident in the accuracy of the SET simulation presented here.

4.6.2 Implementation Details of the SET Simulation

In this section we will briefly present the principle of operation of our error correction method, called Internal Position Error Correction (IPEC), and its implementation on the robot/trailer system of Fig. 4.6. We recall that the primary purpose of the SET is to measure and correct dead-reckoning errors. We further recall that the foremost problem in dead-reckoning is the orientation error that results from traversing a bump or crack or object on the floor.

Figure 4.8 shows a line labeled "direction after traversing the bump," which is the (unintended) direction of the robot after it cleared a bump. Since this direction differs from the intended (straight ahead) direction as the result of a dead-reckoning error, the robot still "believes" it was traveling straight ahead. Consequently, the robot would expect the center of the trailer to be straight behind, along the dotted line labeled L_e in Fig. 4.8. Using dead-reckoning data from both the robot and the trailer, the robot can always compute this *expected direction* to the center of the trailer, whether both are traveling straight or along a curved path. This *expected* direction can then be compared to the *measured* direction, which is readily available from the absolute rotary encoder on the center of the robot. The difference between the *expected* direction and the measured direction is the *measured* orientation error $\Delta\theta_m$. $\Delta\theta_m$ can then be used (as will be explained below) to correct the computed orientation of the robot, which was based on dead-reckoning. The orientation error of the trailer can

be determined in a similar way, relative to the center of the robot.

It is evident from Fig. 4.8 that the *measured* orientation error $\Delta\theta_m$ is not identical to the *actual* orientation error $\Delta\theta_a$, and one must ask how we can correct the original dead-reckoning orientation if we don't know the *actual* error. The answer to this question is that $\Delta\theta_m$ is *almost* identical to $\Delta\theta_a$, and that for the kinematic configuration of the robot/trailer system this *near-identity* is guaranteed under all operating conditions (with a few known exceptions).

Our simulation program accurately models the kinematic design of a TRC LabMate robot with a SET attached (see Fig. 4.9). The simulation program models a wheelbase error by giving the simulated TRC an "actual" wheelbase of 337 mm instead of 340 mm, as assumed by the programs dead-reckoner. Similarly, the vehicle is simulated to have a wheel diameter ratio of 0.9985, while the dead-reckoner assumes the nominal ratio of 1. For the non-systematic error experiments (see Fig. D-6), the program also models a series of 10 bumps, each 10 mm high, during the first leg of the square path prescribed by the UMBmark procedure. The bump is "applied" to the wheel that faces the inside of the square path. Furthermore, the roughness of a concrete floor is simulated by random bumps, which are "applied" to the drive wheels approximately every 2 cm. Each wheel experiences a bump of different height, but always between 0-1 mm. The heights of each bump are determined by a random number generator. Although we simulate these small random bumps, this part of the simulation program is not very effective. This is so because the random bumps average more or less evenly over the complete run, causing only a

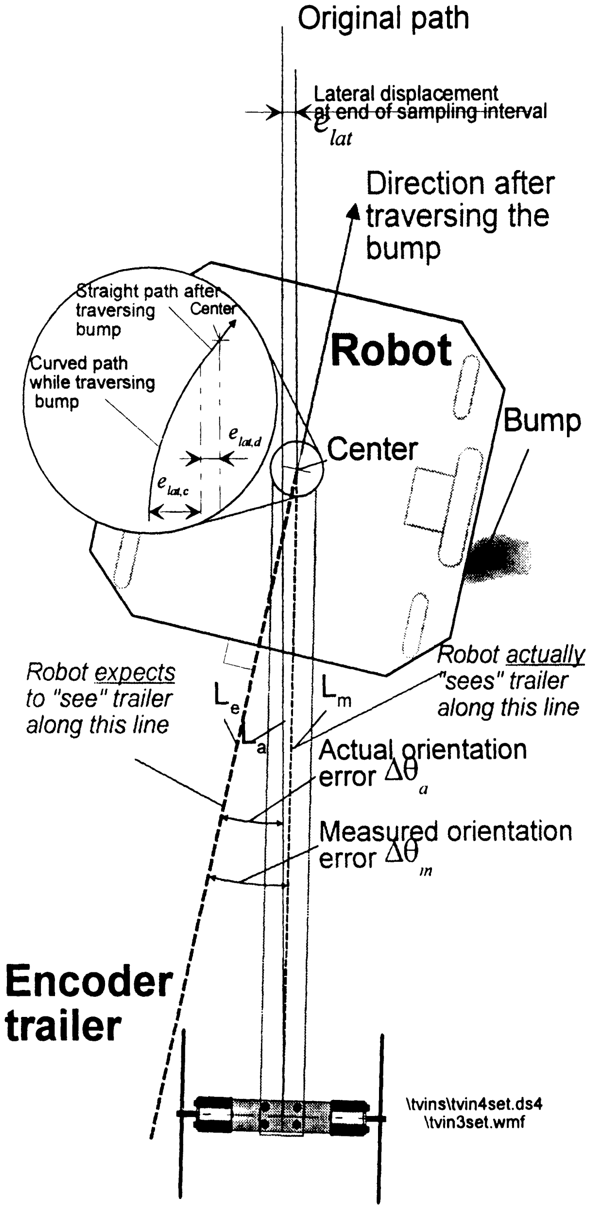
Original path
Lateral displacement
~~at end of sampling interval~~
e_{lat}
Direction after traversing the bump
Straight path after traversing bump
Center
Curved path while traversing bump
$e_{lat,d}$
$e_{lat,c}$
Robot
Center
Bump
Robot expects to "see" trailer along this line
L_e
L_m
L_a
Robot actually "sees" trailer along this line
Actual orientation error $\Delta\theta_a$
Measured orientation error $\Delta\theta_m$
Encoder trailer
\tvins\tvin4set.ds4
\tvin3set.wmf

Figure 4.8: After traversing a bump, the resulting change of orientation of the robot can be measured relative to the trailer.

minimal spread of the *return error positions*. This fact is not obvious in the SET plot in Fig. D-6, since the SET would minimize the spread anyway (because of the IPEC method).

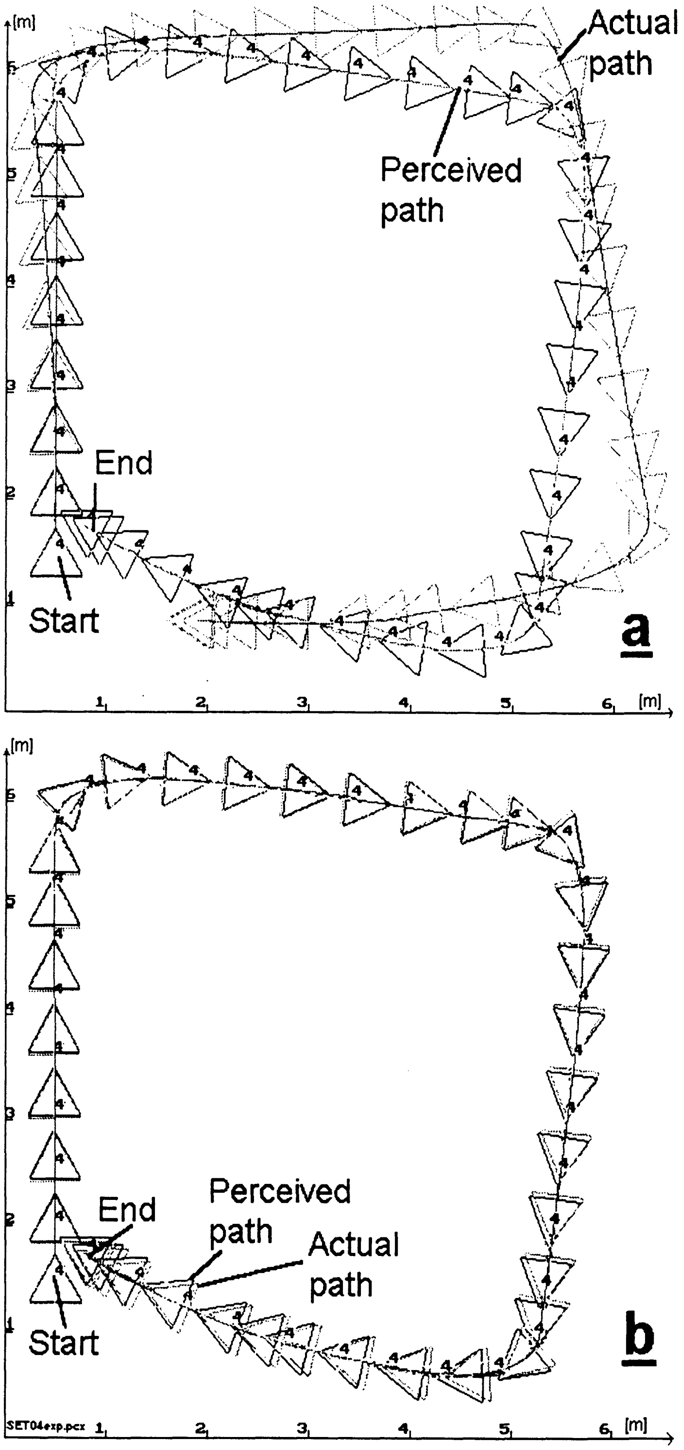

Figure 4.9: The screen shot here was taken after an earlier experiment in which the simulated robot/SET traversed bumps during all 4 legs of the square path. The double images show the difference between "actual" and "perceived" positions of the robot, due to dead-reckoning errors. (a) **IPEC disabled**, and (b) **IPEC enabled**.

5. Measuring Non-systematic Errors — Experiments

In this section we present results of measurements of non-systematic errors using the *extended* UMBmark test (explained in Section 3.4). Table II lists the results for the three robots that were tested. As explained in Section 3.4, ten 10-mm bumps were introduced during the first leg of each run. The resulting *average return orientation errors,* $\epsilon\theta_{avrg}^{nonsys}$(5 each in cw and ccw direction) are compared in Table II.

We recall that according to the explanation in Section 3.4, a vehicle's susceptibility to non-systematic errcrs is best expressed in terms of the *average return orientation error,* $\epsilon\theta_{avrg}^{nonsys}$, which can be computed from the raw orientation data of the extended UMBmark test, using Eq. (3.5). Note that this computation requires the *average of the systematic return orientation errors,* $\epsilon\theta_{avrg,cw}^{sys}$and $\epsilon\theta_{avrg,ccw}^{sys}$, in order to remove the systematic bias from the result of the non-systematic error tests, as shown in Eq. (3.5). All of these measurements, along with the "bottom line" of 0 - 1 mm height result $\epsilon\theta_{avrg}^{nonsys}$. X-Y plots of the *return position errors* of these experiments — although not very useful as a tool for comparison — are presented in Figures D-1 through D-6, in Appendix D.

The results in Table II show that the Cybermotion with its inherently resilient synchro-drive is only half as sensitive to non-systematic errors than the LabMate. However, the CLAPPER with active error correction is one order of magnitude less sensitive than the Cybermotion and 24 times less sensitive than the LabMate. The simulation results from the Smart Encoder Trailer (SET) are, as expected, equal in magnitude to the results from the CLAPPER.

Table II: Comparing experimental results of non-systematic error measurements with the help of the *extended* UMBmark test. A smaller *average absolute return orientation error* $\epsilon\theta^{nonsys}_{avrg}$ in the bottom line indicates better resilience to non-systematic errors. The CLAPPER's *average absolute return orientation error* $\epsilon\theta^{nonsys}_{avrg}$ is 24 times smaller than that of the LabMate and 11 times smaller than the error of the Cybermotion.

		Return Orientation Errors [°]					
		TRC LabMate	**Cyber-motion**	**U of M CLAPPER**	**Remotec Andros**	**Remotec Andros with Basic Encoder Trailer**	**Smart Encoder Trailer (SET) Simulation results only**
$\epsilon\theta^{sys}_{avrg,cw}$		0.3	3.8	-0.2	137	-0.31	0.34
$\epsilon\theta^{sys}_{avrg,ccw}$		-2.0	1.0	-0.1	-125	-0.71	-0.06
$\epsilon\theta^{nonsys}_{i,cw}$	cw 1	7.1	7.8	0.1	A test with 10 8-9 mm high bumps is not applicable to the Remotec Andros because the grooves on the vehicle's tracks are much larger than 9 mm. Furthermore, the dead-reckoning errors of the tracked vehicle are so large that dead-reckoning is totally unfeasible.	-0.1	0.4
	cw 2	6.4	4.2	0.1		5.6	0.3
	cw 3	5.6	1.7	0.1		-0.3	0.3
	cw 4	6.6	4.2	-0.7		5.1	0.3
	cw 5	5.9	3.6	0.2		-1.4	0.4
$\epsilon\theta^{nonsys}_{i,ccw}$	ccw 1	-7.5	-11.4	-0.6		-1.0	0.0
	ccw 2	-8.8	-2.5	-0.4		-1.0	0.3
	ccw 3	-6.6	-4.6	-0.5		-2.2	0.3
	ccw 4	-8.8	-6.3	0.1		-1.7	0.2
	ccw 5	-8.7	-2.3	0.2		1.1	0.2
$\epsilon\theta^{nonsys}_{avrg}$		**8.35**	**3.91**	**0.35**		**2.53**	**0.16**

6. Correction of Systematic Dead-reckoning Errors

One interesting aspect of the error distribution pattern in the UMBmark experiment (see Fig. 3.4, in Section 3.3, above) is the fact that one can analytically derive correction factors from the experimental results. Before we do so, let us first define two new error characteristics that are meaningful only in the context of the Bi-directional Square Path experiment. These characteristics, called Type A and Type B, represent dead-reckoning errors in orientation. Type A is defined as an orientation error that ***reduces (or increases)*** the total amount of rotation of the robot during the square path experiment in ***both cw and ccw direction***. By contrast, Type B is defined as an orientation error that ***reduces (or increases)*** the total amount of rotation of the robot during the square path experiment in **one direction**, but ***increases (or reduces)*** the amount of rotation when going in the ***other direction***. As examples consider Figures 6.1 and 6.2, further below. Figure 6.1 shows a case where the robot turns four times for a nominal amount of 90° per turn. However, because the actual wheelbase of the vehicle was larger than the nominal value, the vehicle actually turned only 85° in each corner of the square path. In the example of Fig. 6.1 the robot will actually turn only $\theta_{total} = 4 \times 85° = 340°$, instead of the desired $\theta_{nominal} = 360°$. We observe that in ***both the cw and the ccw*** experiment the robot ends up turning ***less*** than the desired amount , i.e., $|\theta_{total,\ cw}| < |\theta_{nominal}|$ ***and*** $|\theta_{total,\ ccw}| < |\theta_{nominal}|$. Thus, the orientation error is of Type A. In Fig. 6.2 the trajectory of a robot with unequal wheel diameters is shown. This error expresses itself in a curved path that adds to the overall orientation at the end of the run in ccw direction, but it reduces the overall rotation in the ccw direction, i.e., $|\theta_{total,\ ccw}| > |\theta_{nominal}|$ ***but*** $|\theta_{total,\ cw}| < |\theta_{nominal}|$. Thus, the orientation error in Fig. 6.2 is of Type B.

In an actual run Type A and Type B errors will of course occur together. The problem is therefore how to distinguish and compute Type A and Type B errors from the measured final position errors of the robot in the Bi-directional Square Path experiment. We approach this problem by defining a simplified model for systematic dead-reckoning errors. This model assumes that

1. E_d and E_b are the dominant sources of systematic dead-reckoning errors.
2. An incorrect wheelbase (E_b) causes errors only during turning but not during straight line motion.
3. Unequal wheel diameters (E_d) cause errors only during straight line motion but not during turning. This assumption is substantiated in Appendix B.
4. E_b causes only Type A errors but not Type B errors.
5. E_d causes only Type B errors but not Type A errors.
6. Because of assumption #1, eliminating E_b eliminates the system's Type A error *almost* completely
7. Because of assumption #1, eliminating E_d eliminates the system's Type B error *almost* completely
8. Orientation errors are the main source of concern because once they are incurred they grow without bound into lateral position errors [Crowley, 1989; Feng et al., 1993].

Because of the close association between E_b and Type A errors and between E_d and Type B errors (according to assumptions #6 and #7) we will use the terms E_b and Type A, as well as the terms E_d and Type B, interchangeably.

6.1 Analysis of Type A and Type B Errors

Having defined a model, we will now analyze the characteristics of the UMBmark procedure with regard to that model. To simplify the mathematical treatment, we note that the following approximations are valid for small angles:

$$\begin{aligned} L\sin\gamma &\approx L\gamma \\ L\sin2\gamma &\approx 2L\gamma \\ \mathrm{L}\sin3\gamma &\approx 3L\gamma \end{aligned} \qquad (6.1)$$

$$\begin{aligned} L\cos\gamma &\approx L \\ L\cos2\gamma &\approx L \\ L\cos3\gamma &\approx L \end{aligned} \qquad (6.2)$$

where
L - the length of each side of the square path.
γ - any incremental orientation error caused by either E_d or E_b, measured in [rad].

For simplicity, we assume that the starting position (x_0, y_0) of the robot is at (0,0). At first we will analyze and examine the contribution of Type A and Type B errors separately. Then, we will superimpose both errors to represent the actual conditions.

Figure 6.1 shows the contribution of Type A errors. We recall that according to assumptions #1 and #4 Type A errors are caused mostly by E_b. We also recall that Type A errors cause too much or too little turning at the corners of the square path. The (unknown) amount of erroneous rotation in each nominal 90° turn is denoted as α. Because of the approximations in Eqs. (6.1) and (6.2), α is measured in [rad].

a. For Type A errors in ccw direction:

$$x_1 = x_0 + L \tag{6.3a}$$
$$y_1 = y_0 \tag{6.3b}$$

$$x_2 = x_1 + L\sin\alpha \approx L + L\alpha \tag{6.4a}$$
$$y_2 = y_1 + L\cos\alpha \approx L \tag{6.4b}$$

$$x_3 = x_2 - L\cos2\alpha \approx L\alpha \tag{6.5a}$$
$$y_3 = y_2 + L\sin2\alpha \approx L + 2L\alpha \tag{6.5b}$$

$$x_4 = x_3 - L\sin3\alpha \approx -2L\alpha \tag{6.6a}$$
$$y_4 = y_3 - L\cos3\alpha \approx 2L\alpha \tag{6.6b}$$

b. For Type A errors in cw direction:

$$x_1 = x_0 + L \tag{6.7a}$$
$$y_1 = y_0 \tag{6.7b}$$

$$x_2 = x_1 + L\sin\alpha \approx L + L\alpha \tag{6.8a}$$
$$y_2 = y_1 - L\cos\alpha \approx -L \tag{6.8b}$$

$$x_3 = x_2 - L\cos2\alpha \approx L\alpha \tag{6.9a}$$
$$y_3 = y_2 - L\sin2\alpha \approx -L - 2L\alpha \tag{6.9b}$$

$$x_4 = x_3 - L\sin3\alpha \approx -2L\alpha \tag{6.10a}$$
$$y_4 = y_3 + L\cos3\alpha \approx -2L\alpha \tag{6.10b}$$

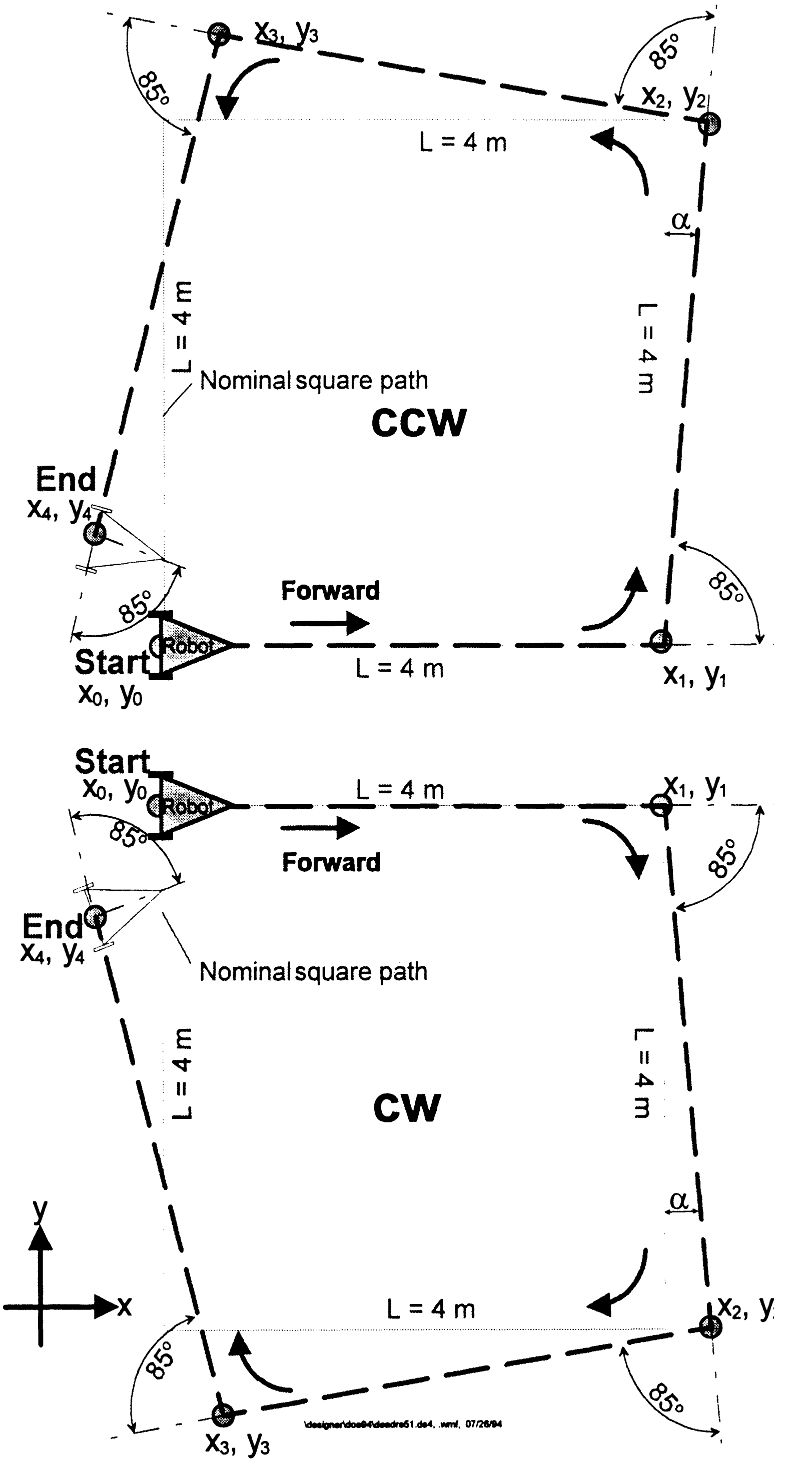

Figure 6.1: Type A errors in ccw and cw direction. Type A errors are caused almost exclusively by the wheelbase error E_b.

Figure 6.2 shows the contribution of Type B errors. We recall that according to our assumptions #1 and #5 Type B errors are caused mostly by the ratio between wheel diameters, E_d. We also recall that Type B errors cause a slightly curved path instead of a straight one during the four straight legs of the square path. Because of the curved motion, the robot will have gained an incremental orientation error, denoted β, at the end of each straight leg. Note that the auxiliary line c'_1, which connects the corner points of the actual path, has a slope of β/2 because it is parallel to the tangent to the midpoint of arc c_1. With respect to the unknown parameter β (in [rad]), we obtain:

a. For Type B errors in ccw direction:

$$x_1 = x_0 + L\cos(\beta/2) \approx L \tag{6.11a}$$
$$y_1 = y_0 + L\sin(\beta/2) \approx L\beta/2 \tag{6.11b}$$

$$x_2 = x_1 - L\sin(3\beta/2) \approx L - 3L\beta/2 \tag{6.12a}$$
$$y_2 = y_1 + L\cos(3\beta/2) \approx L\beta/2 + L \tag{6.12b}$$

$$x_3 = x_2 - L\cos(5\beta/2) \approx -3L\beta/2 \tag{6.13a}$$
$$y_3 = y_2 - L\sin(5\beta/2) \approx -2L\beta + L \tag{6.13b}$$

$$x_4 = x_3 + L\sin(7\beta/2) \approx 2L\beta \tag{6.14a}$$
$$y_4 = y_3 - L\cos(7\beta/2) \approx -2L\beta \tag{6.14b}$$

b. For Type B errors in cw direction:

$$x_1 = x_0 + L\cos(\beta/2) \approx L \tag{6.15a}$$
$$y_1 = y_0 + L\sin(\beta/2) \approx L\beta/2 \tag{6.15b}$$

$$x_2 = x_1 + L\sin(3\beta/2) \approx L + 3L\beta/2 \tag{6.16a}$$
$$y_2 = y_1 - L\cos(3\beta/2) \approx L\beta/2 - L \tag{6.16b}$$

$$x_3 = x_2 - L\cos(5\beta/2) \approx 3L\beta/2 \tag{6.17a}$$
$$y_3 = y_2 - L\sin(5\beta/2) \approx -L(2\beta + 1) \tag{6.17b}$$

$$x_4 = x_3 - L\sin(7\beta/2) \approx -2L\beta \tag{6.18a}$$
$$y_4 = y_3 + L\cos(7\beta/2) \approx -2L\beta \tag{6.18b}$$

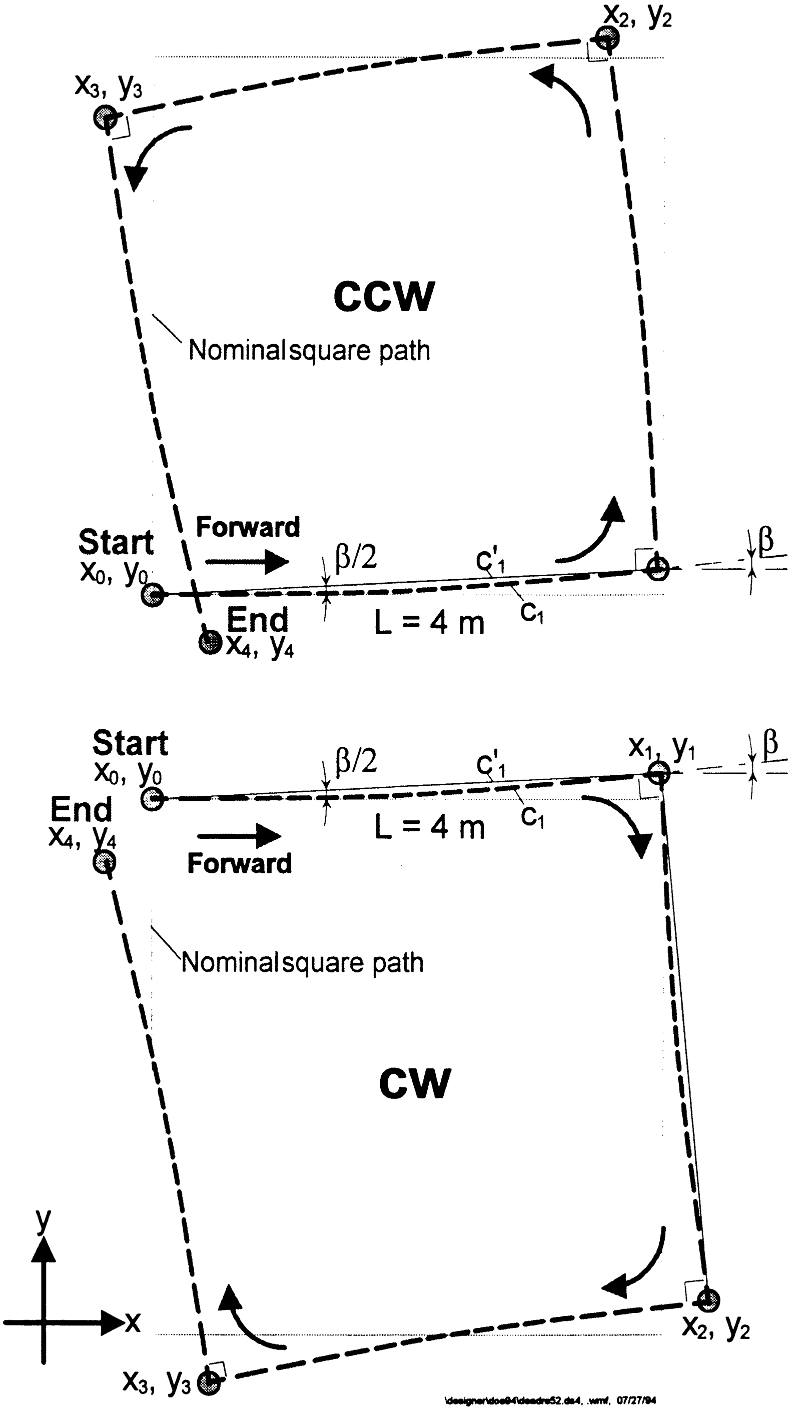

Figure 6.1: Type B errors in ccw and cw direction. Type B errors are caused almost exclusively by unequal wheel diameters (E_d).

Superimposing Type A and Type B errors for the cw experiment in x-direction yields

$$x_{cw}\text{:}\quad -2L\alpha - 2L\beta = -2L(\alpha + \beta) = x_{c.g.,cw} \tag{6.19}$$

$$x_{ccw}\text{:}\quad -2L\alpha + 2L\beta = -2L(\alpha - \beta) = x_{c.g.,ccw} \tag{6.20}$$

Subtracting (6.20) from (6.19) yields

$$-4L\beta = x_{c.g.,cw} - x_{c.g.,ccw} \tag{6.21}$$

or

$$\beta = \frac{x_{c.g.,cw} - x_{c.g.,ccw}}{-4L}\,\frac{(180^\circ)}{\pi} \tag{6.22a}$$

for β in degrees.

Comparing terms in y-direction yields a similar result

$$\beta = \frac{y_{c.g.,cw} + y_{c.g.,ccw}}{-4L}\,\frac{(180^\circ)}{\pi} \tag{6.22b}$$

Using simple geometric relations, the radius of curvature R of the curved path of Fig. 6.2 can be found from triangle ABM in Fig. 6.3.

$$R = \frac{L/2}{\sin(\beta/2)} \tag{6.23}$$

Once the radius R is computed, it is easy to determine the ratio between the two wheel diameters that caused the robot to travel on a curved, instead of a straight path (see Fig. 6.4):

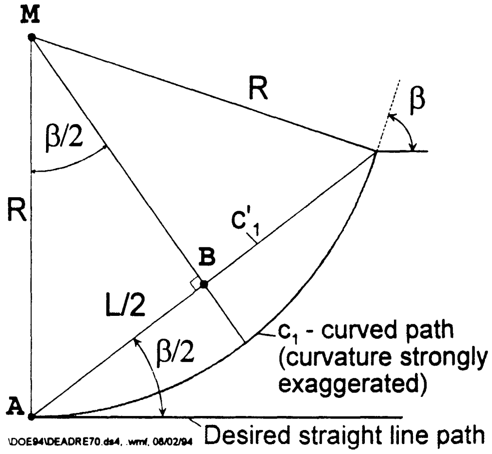
M
R
β
β/2
R
c'1
B
L/2
β/2
c1 - curved path (curvature strongly exaggerated)
A
\DOE94\DEADRE70.ds4, .wmf, 08/02/94
Desired straight line path

Figure 6.3: Geometric relations for finding the radius of curvature.

$$E_d = \frac{D_R}{D_L} = \frac{R + b/2}{R - b/2} \tag{6.24}$$

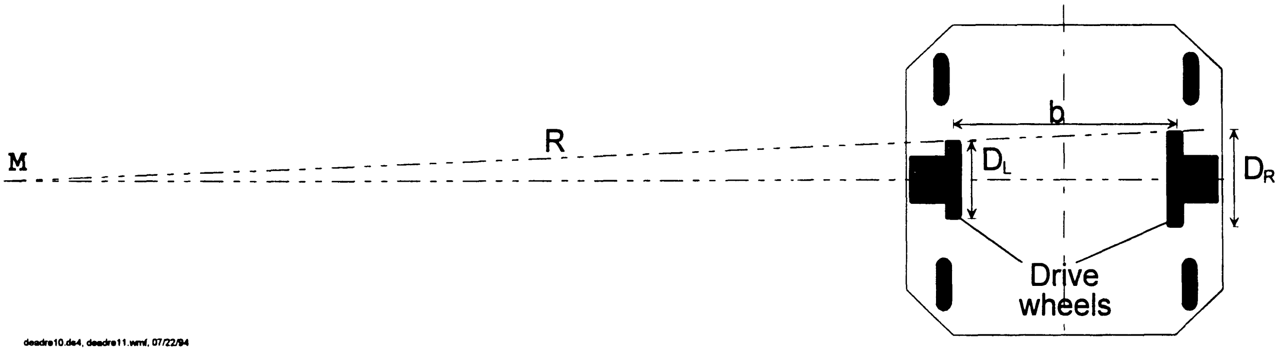

Figure 6.4: Unequal wheel diameters cause the robot to travel on a curved path of radius *R* (curvature is exaggerated for better illustration).

The ratio of Eq. (6.24) can be used to correct Type B errors as will be explained in Section 6.2. Similarly, α can be found by adding Eq. (6.19) and Eq. (6.20)

$$-4L\alpha = x_{c.g.,cw} + x_{c.g.,ccw} \tag{6.25}$$

or

$$\alpha = \frac{x_{c.g.,cw} + x_{c.g.,ccw}}{-4L} \frac{(180^\circ)}{\pi} \tag{6.26a}$$

solves for α in [degrees].

Again, comparing terms in y-direction yields a similar result for α

$$\alpha = \frac{y_{c.g.,cw} - y_{c.g.,ccw}}{-4L} \frac{(180^\circ)}{\pi} \tag{6.26b}$$

We can now compute the wheelbase error E_b. Since the wheelbase b is directly proportional to the actual amount of rotation as shown by Eq. (2.4), we can use the proportion:

$$\frac{b_{actual}}{90^\circ} = \frac{b_{nominal}}{90^\circ - \alpha} \tag{6.27}$$

so that

$$b_{actual} = \frac{90^\circ}{90^\circ - \alpha} \; b_{nominal} \tag{6.28}$$

where, per definition of Eq. (2.8)

$$E_b = \frac{90^\circ}{90^\circ - \alpha} \tag{6.29}$$

6.2 Compensation for systematic dead-reckoning errors

Once we know the quantitative values of E_d and E_b, it is easy to compensate for these errors in software. The correction for the wheelbase error E_b is trivial: the wheelbase b is redefined in software according to Eq. (6.28). The correction for the unequal wheel diameters, E_d, is slightly more complex: After performing the UMBmark procedure, we know the *actual* wheel diameter ratio $E_d = D_R/D_L$ from Eq. (6.24). However, when applying a compensation factor, we must make sure not to change the *average* wheel diameter D_a, since one would then have to recalibrate that parameter. D_a will remain unchanged if we consider it as a constraint

$$D_a = (D_R + D_L)/2 \tag{6.30}$$

Solving Eqs. (6.24) and (6.30) as a set of two linear equations with two unknowns, D_R and D_L, yields

$$D_L = \frac{2}{E_d + 1} D_a \tag{6.31}$$

and

$$D_R = \frac{2}{(1/E_d) + 1} D_a \tag{6.32}$$

We can now define the two correction factors

$$c_L = \frac{2}{E_d + 1} \tag{6.33}$$

and

$$c_R = \frac{2}{(1/E_d) + 1} \tag{6.34}$$

which can be implemented in the dead-reckoning algorithm by rewriting Eq. (2.2) as

$$\Delta U_{L/R,\,I} = c_{L/R}\, c_m\, N_{L/R,\,I} \tag{6.35}$$

We have thus corrected both dominant systematic errors.

6.3 Correction of Systematic Errors

In this section we describe experiments that validate the above described method for correcting Type A and Type B errors by changing the effective wheelbase b and the effective wheel-diameter ratio D_R/D_L. The experiments were performed with a LabMate robot equipped with an onboard AMPRO 486/50 MHZ PC compatible single-board computer.

The robot was programmed for both a cw and a ccw 4×4 m square path. To avoid slippage, the robot was traveling slowly, at a speed of 0.2 m/s during the straight legs of the square path. At the end of each leg the robot came to a complete stop and rotated on-the-spot through 90°. This means that the robot made a fourth 90° turn after returning to its starting area. The linear speed of the two drive wheels during turning was approximately 0.2 m/s and -0.2 m/s. The robot started and stopped near an L-shaped corner and used a so-called "*sonar calibrator*" [Borenstein 1993] to determine its position and orientation relative to the L-shaped corner. We will refer to this as the *absolute* position. The sonar calibrator comprises three standard POLAROID ultrasonic sensors. Two sensors were facing the long side of the L-shaped corner, the third sensor faced the short side. The ultrasonic sensor system allowed measurement of the absolute position of the vehicle to within ±2 millimeters in the x and y directions, and to about ±0.4° in orientation.

At the beginning of each run a sonar measurement was taken to determine the starting position of the vehicle. The robot then traveled through the programmed path and returned to the L-shaped corner, where the *perceived* position (i.e., the position the vehicle "thought" it had, based on dead-reckoning) was recorded. Then, a second sonar measurement was taken to determine the *absolute* position. The difference between the absolute position and the perceived position is called the *return position error* ε; ε is defined by Eqs. (3.1), above.

The uncalibrated robot (i.e., $D_R/D_L = 1.0000$ and $b = b_{\text{nominal}} = 340.00$ mm) made five cw trips and five ccw trips. As expected, the *return position errors* were clearly grouped in a cw cluster and a ccw cluster, as was shown in Fig. 5. For each of the two clusters the x and y components of the respective centers of gravity were computed according to Eq. (3.2). The resulting $x_{\text{c.g.}}$ and $y_{\text{c.g.}}$ were used to compute E_d according to Eqs. (6.22) - (6.24). Then, correction factors c_L and c_R were computed according to Eqs. (6.33) and (6.34) and introduced into the dead-reckoning program. Similarly the corrected wheelbase b_{new} was computed according to Eqs. (6.26) - (6.28)[2].

At this time the calibration procedure was complete. In order to verify the results we ran the UMBmark experiment for a second time, this time with the correction factors in place. Figure 6.5 shows the results of both the uncalibrated runs and the runs with the calibrated vehicle.

As explained in Section 3, Eqs. (3.2) and (3.3) were used to express the experimental results quantitatively as the *measure of dead-reckoning accuracy for systematic errors,* $E_{\text{max,syst}}$. In the example of Fig. 6.5, $E_{\text{max,syst}}$ was 317 mm before compensation and 21 mm after compensation. This represents a 15-fold improvement.

In order to assure that the experiment shown in Fig. 6.5 was not an isolated case, we performed another seven carefully monitored experiments. Table I lists the results from all eight experiments. We emphasize that Table I lists **all** experiments we ever made, it is not a selection of the best runs. We further emphasize that in each experiment we used all runs, without eliminating "outliers" (with the exception of four or five runs where the errors reported by the *sonar calibrator* were absurdly large, presumably due to a malfunctioning of the *sonar calibrator*).

The seemingly large fluctuations in improvement, especially among experiments #3, #4, and #5 (which all used the same correction factors) are due to the fact that the *centers of gravity* (c.g.s) for the runs after calibration are all very close to the origin (as seen in Fig. 6.5). Thus, the arbitrary spread of return position errors caused by non-systematic error sources has greater impact on the c.g.s. For example, the c.g. of Experiment 4 is only 17 mm (5/8") closer to the origin than the c.g. of Experiment #3 — a difference that is easily attributable to the arbitrary spread of non-systematic errors.

[2] Hoping to reduce the effect of non-systematic errors further, we actually computed E_d and E_b in two ways: (1) based on the values for $x_{\text{c.g.}}$, according to Eqs. (6.22a) and (6.26a); and (2) based on the values for $y_{\text{c.g.}}$, according to Eqs. (6.22b) and (6.26b). We then averaged $E_{d,x}$ and $E_{d,y}$, as well as $E_{b,x}$ and $E_{b,y}$. This measure may not be necessary in general, because the respective correction values (based on $x_{\text{c.g.}}$ or $y_{\text{c.g.}}$) differed by less than 1% in all cases.

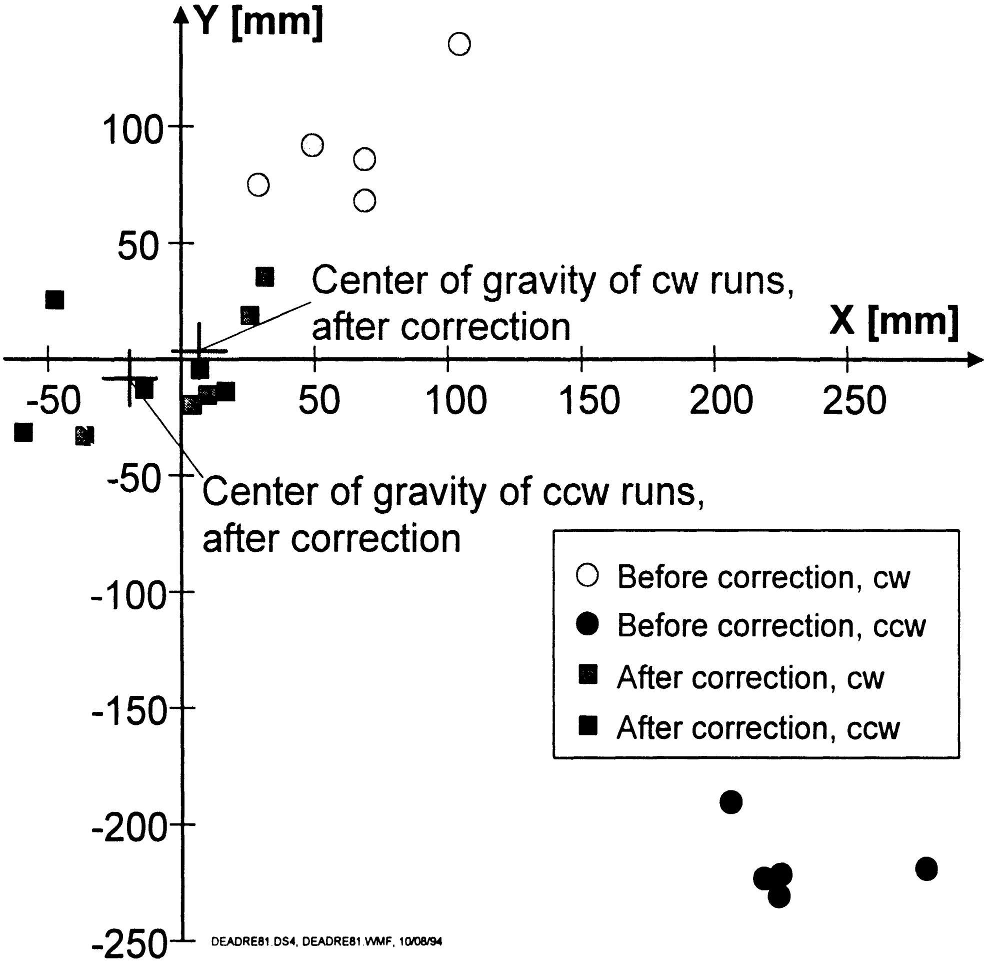

Figure 6.5: Position errors of a TRC LabMate after completing the UMBmark test (4 x 4 m).
Before calibration: b=340.00 mm, D_R/D_L = 1.00000
After calibration: b=336.17 mm, D_R/D_L = 1.00084

In principle, it is possible to achieve even better results by performing the compensation procedure for a second time, "on top of" the first compensation. This is so because a compensated robot can be treated as though it was a "new" uncompensated robot, but with different initial parameters. Using the *standard deviation* (σ) of the 5 runs in each direction it is easy to decide when a second compensation run will be beneficial. The standard deviation of the *return position errors* in the UMBmark test was about σ = 25 mm. The *Standard Error of the Mean* (SEM), defined as SEM = σ/sqrt(*n*), was 11.2 mm (*n* is the number of runs). As a rule-of-thumb sometimes used in small sample statistics [Walpole and Myers, 1985], one can say that if $E_{max,syst}$ < 3×SEM it is unlikely (here: 5%) that the result can be improved by a second compensation. We put this rule-of-thumb to the test in Experiment #7, where $E_{max,syst}$ = 66 mm was notably worse (the improvement over the uncompensated run was only 6.4-fold) than in the other experiments. Applying the above rule-of thumb, it is evident that 66 mm > 3×SEM = 33.6 mm, so that a second compensation run was indicated. After the second compensation, the vehicle's error was $E_{max,syst}$ = 20 mm, i.e., a 21-fold reduction relative to the uncompensated systematic error.

Table III: The *Measure of Dead-reckoning Accuracy for Systematic Errors*, $E_{max,syst}$, before and after compensation.

Experiment # and Fig.	$E_{max,syst}$ before compensation [mm]	$E_{max,syst}$ after compensation [mm]	Improvement	Comment
E-1	317	21	15-fold	Details also shown in Fig. 6.5
E-2	349	32	11-fold	
E-3	310	31	10-fold	These 3 experiments used the same set of uncalibrated results and identical correction factors.
E-4	310	14	22-fold	
E-5	310	26	12-fold	
E-6	403	35	11-fold	
E-7	423	after 1st comp: 66 after 2nd comp: 20	21-fold*	In this experiment the diameter of the right wheel was slightly increased by winding three loops of masking tape around the wheel perimeter.
E-8	232	12	19-fold	In this experiment the diameter of the left wheel was slightly increased by winding five loops of masking tape around the left wheel perimeter.

*) *Two compensation runs were performed. See explanation in main text.*

7. CONCLUSIONS

This report deals with the measurement and correction of systematic dead-reckoning errors, and with the measurement and comparison of non-systematic errors in mobile robots. The contribution of the work described in this report falls into three areas: (1) measurement of systematic errors, (2) measurement of non-systematic errors, and (3) correction of systematic errors. We will summarize our conclusions for each one of these areas below.

7.1 Measurement of Systematic Errors

The report identifies specifically the errors due to the *wheel diameter ratio,* E_d, and the *uncertainty about the wheelbase*, E_b as the foremost source of systematic dead-reckoning errors in differential drive mobile robots. A third — potentially significant — error is the *scaling error*, E_s. E_s is the ratio between the average of the *actual* wheel diameters and the *nominal* wheel diameter. However, this error is so easy to measure and correct (see Appendix B) that we have removed it from consideration.

The focus on E_d and E_b is based on our error model, which assumes that systematic orientation errors are either of Type A or Type B. Type A errors are directly affected by E_b and Type B errors are directly affected by E_d. Other systematic errors may also affect the overall Type A and Type B error. However, we do not need to worry about this, because, in principle, both Type A and Type B errors can be eliminated completely by changing the effective wheelbase and wheel-diameter ration in software.

Based on this model we define a benchmark test for dead-reckoning accuracy in differential-drive robots. This test, called *UMBmark*, assures that different dead-reckoning errors don't compensate for each other, as may be the case with other dead-reckoning test. The UMBmark procedure yields a single numeric value, $E_{max,sys}$, that represents a *quantitative* measure of a vehicle's systematic dead-reckoning errors. This makes UMBmark an effective tool for evaluating the dead-reckoning performance of a vehicle with different parameters or for the comparison of dead-reckoning performance between different mobile robots.

One interesting result from the UMBmark tests is that the Cybermotion K2A (which is a *synchro-drive* platform) appears to have similar error characteristics as do differential drive vehicles. This observation is made evident by the UMBmark test in Fig. C-2. In this result one can clearly identify the characteristic cw and ccw clusters of *return position errors*. The formation of such clusters points to the existence of underlying error mechanisms that are subject to classification into Type A and Type B errors. We will further investigate this matter in future work. It may be possible to derive a calibration procedure that is similar to the one described in Section 6, but for the Cybermotion platform instead of the LabMate.

7.2 Measurement of Non-systematic Errors

To measure non-systematic dead-reckoning errors, we defined a variation of our benchmark test, called the *extended UMBmark.* One interesting insight gained from the work on the *extended UMBmark* test is that the *average return orientation error* is the relevant criterium for expressing non-systematic errors, while the *return position error* is the relevant criterium for systematic errors.

The measurement of non-systematic errors by means of the *extended UMBmark* test is of interest because it clearly shows the benefits of our previously developed *Internal Position Error Correction* (IPEC) method. The results produced by the implementations of the IPEC method in the CLAPPER and in the (simulated) Smart Encoder Trailer are consistently one order of magnitude better than the results of "conventional" mobile robots subjected to substantial floor irregularities.

7.3 Correction of Non-systematic Errors

Another contribution of this report is the definition of a systematic procedure for measuring and correcting Type A and Type B dead-reckoning errors. The effectiveness of this procedure and the validity of its underlying model are supported by the experimental results. The results show that by changing only the effective wheelbase and the effective wheel-diameter ratio the vehicle's dead-reckoning accuracy (with respect to systematic errors only) increased by at least one order of magnitude. This improvement was consistent when tested repeatedly for the same vehicle and when tested on the same vehicle but with artificially altered wheelbases and wheel-diameter ratios.

One should note that dead-reckoning calibration factors are used by many researchers. However, to date such factors were usually found by some form of trial-and-error and some intuition on the part of the experimenter. This type of approach is very time consuming and yields inferior results. By contrast, the UMBmark procedure offers a systematic approach that yields near-optimal results. The strength of the UMBmark calibration procedure lies in the fact that even minute mechanical inaccuracies, such as wheel diameters that differ by as little as 0.1% can be isolated and identified. Yet, a conventional measuring tape is all that is needed to conduct the experiment.

A higher level of sophistication can be reached by mounting a *sonar calibrator* (see Section 6.3). With the help of the *sonar calibrator* the UMBmark procedure lends itself to be implemented as an *automated* self-calibration procedure. U of M is now beginning to develop such an automated approach. If successful, this method would require only two human interventions: (1) manual measurement of the scaling error E_s (with an ordinary tape measure); and (2) initial placement of the robot in a n L-shaped corner of the testing site. The robot would then run the fully automated self-calibration routine (UMBmark), compute the calibration factors, and insert the calibration factors into its dead-reckoning program. This method should be of interest for all manufacturers of differential-drive autonomous vehicles. Similarly, end-users who are concerned with accurate dead-reckoning would want to run the self-calibration routine periodically to correct for different loads and tire-wear.

7.4 Future Work

The work presented in this report offers some highly attractive possibilities for future work. Encouraged by the results of this report, we can assume that it is possible to reduce the susceptibility of mobile robot to *non-systematic* errors by one to two orders of magnitude (as shown by the CLAPPER and the SET). Based on the results of this report we can further assume that it is possible to reduce the effect of *systematic* errors by at least one order of magnitude (as shown by our results with the UMBmark calibration procedure). Then we can conclude that ***all*** dead-reckoning errors are effectively reduced, thereby potentially yielding a platform that can navigate accurately over long distances with only sporadic absolute positioning updates. We must note, though, that these results have been obtained on *different* platforms, but have not been integrated yet into one single platform.

Integration of the above named features is our goal for the near future. The benefits of a successful integration of our error reducing measures are significant. For example, a conventional vehicle requires an absolute positioning update after 10 m of linear travel. Even then, a severe disturbance might cause the vehicle to lose track of its position to the extend that it cannot re-establish its position. By contrast, a vehicle with the dead-reckoning improvements as we envisage will be able to travel 100 m without absolute position correction. If one finds this thought unlikely, it is because in the back of our minds is buried the question: "but what if there is a *really* big bump...?" This question is justified for conventional mobile robots, because we have all seen such robots getting lost upon the encounter with a sizable disturbance. Yet, we are simply not accustomed to seeing a vehicle navigate that immediately detects and corrects errors caused by large disturbances. For such a vehicle orientation errors due to floor irregularities are *virtually* bounded.

Based on the results and insights gained from this report, we feel strongly motivated to pursue our goal of developing a highly accurate and reliable, yet commercially feasible dead-reckoning robot. Many colleagues who see a demo of our CLAPPER vehicle are impressed by the dead-reckoning performance, but wonder about the commercial feasibility of this complex piece of equipment. Yet, the CLAPPER is not all that complex: it is the combination of two standard LabMates, connected by a sliding-table (which also serves as the loading deck). The end-user price of the two LabMates is $10K each, the sliding table and encoders can easily be integrated by the manufacturer of the LabMates (TRC) for — perhaps — another $10K. Further required to build the CLAPPER is a linear encoder and two absolute encoders with rotary bearings and slip-rings for, say, an additional $10K. Thus, one can reasonably expect that a commerically produced CLAPPER vehicle could sell for $40K. By comparison, the basic Cybermotion K3A platform costs $50K. One additional advantage of the CLAPPER, not mentioned in this report, is the CLAPPER's ability to be recovered from actuator failure *remotely* based on its actuator redundancy.

Because of these potential benefits we will continuing our work with both the CLAPERand the SET. Combining the *UMBmark calibration* (for the reduction of *systematic* errors) with our method for detecting and rejecting *non-systematic* errors (as implemented in the CLAPPER and — hopefully soon — in the Smart Encoder Trailer) promises a new level of completely reliable dead-reckoning performance with two-to-three orders of magnitude improved accuracy!

8. REFERENCES

1. Barshan, B. and Durrant-Whyte, H.F., 1993, "An Inertial Navigation System for a Mobile Robot." *Proceedings of the 1st IAV*, Southampton, England, April 18-21, pp. 54-59.

2. Barshan, B. and Durrant-White, H. F., 1994, "Orientation Estimate for Mobile Robots Using Gyroscopic Information." *1994 International Conference on Intelligent Robots and Systems (lROS '94)*. München, Germany, September 12-16, pp. 1867-1874.

3. Borenstein, J. and Koren, Y., 1985, "A Mobile Platform For Nursing Robots." *IEEE Transactions on Industrial Electronics*, Vol. 32, No. 2, pp. 158-165.

4. 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.

5. Borenstein, J., 1993, "Multi-layered Control of a Four-Degree-of-Freedom Mobile Robot With Compliant Linkage." *Proceedings of the 1993 IEEE International Conference on Robotics and Automation,* Atlanta, Georgia, May 2-7, pp. 3.7-3.12.

6. Borenstein, J., 1994a, "The CLAPPER: a Dual-drive Mobile Robot With Internal Correction of Dead-reckoning Errors." *Proceedings of the 1994 IEEE International Conference on Robotics and Automation,* San Diego, CA, May 8-13, pp. 3085-3090.

7. Borenstein, J., 1994b, "Internal Correction of Dead-reckoning Errors With the Compliant Linkage Vehicle." Accepted for publication in the *Journal of Robotic Systems*, November 1994.

8. Borenstein, J., 1995, "The CLAPPER: A Dual-drive Mobile Robot With Internal Correction of Dead-reckoning Errors." *Submitted for inclusion in the **Video Proceedings** of the 1995 IEEE International Conference on Robotics and Automation.*

9. Byrne, R.H., Klarer, P.R., and Pletta, J.B., 1992, "Techniques for Autonomous Navigation," *Sandia Report SAND92-0457*, Sandia National Laboratories, Albuquerque, NM, March.

10. Byrne, R.H., 1993, "Global Positioning System Receiver Evaluation Results." Sandia Report SAND93-0827, Sandia National Laboratories, Albuquerque, NM, September, 1993.

11. Chenavier, F. and Crowley, J., 1992, "Position Estimation for a Mobile Robot Using Vision and Odometry." *Proceedings of IEEE International Conference on Robotics and Automation*, Nice, France, May 12-14, pp. 2588-2593.

12. Congdon, I. et al., 1993, "CARMEL Versus FLAKEY — A Comparison of Two Winners." *AI Magazine Winter 1992*, pp. 49-56.

13. Cox, I. J., 1991, "Blanche — An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle." *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, April, pp. 193-204.

14. Crowley, J. L., 1989, "Asynchronous Control of Orientation and Displacement in a Robot Vehicle." *Proceedings of the 1989 IEEE International Conference on Robotics and Automation.* Scottsdale, Arizona, May 14-19, pp. 1277-1282.

15. Cybermotion, 1987, "K2A Mobile Platform." *Commercial Sales Literature*, 115 Sheraton Drive, Salem, VA 24153.

16. Evans, J. M., 1994, "HelpMate: An Autonomous Mobile Robot Courier for Hospitals." *1994 International Conference on Intelligent Robots and Systems (lROS '94)*. München, Germany, September 12-16, 1994, pp. 1695-1700.

17. Everett, H.R., 1995, "*Sensors for Mobile Robots*," A K Peters, Ltd., Wellesley, expected publication date Spring 1995.

18. Fan, Z., Borenstein, J., Wehe, D., and Koren, Y., 1995, "Positioning of Mobile Robots." *University of Michigan, Department of MEAM, Technical Report* #95-01, January.

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

20. Hewlett Packard, "Optoelectronics Designer's Catalog, 1991-1992."

21. Hollingum, J., 1991, "Caterpillar make the earth move: automatically." *The Industrial Robot*, vol. 18, no. 2, pp. 15-18.

22. Komoriya, K. and Oyama, E., 1994, "Position Estimation of a mobile Robot Using Optical Fiber Gyroscope (OFG)." *International Conference on Intelligent Robots and Systems (lROS '94)*. München, Germany, September 12-16, pp. 143-149.

23. Kortenkamp, D. et al., 1992, "Integrating Obstacle Avoidance, Global Path Planning, Visual Cue Detection, and Landmark Triangulation in a Mobile Robot." Presented at the *1992 SPIE Symposium on Advances in Intelligent Systems, Mobile Robots VII*, Boston, MA, Nov. 15-20.

24. Rencken, W. D., 1994, "Autonomous Sonar Navigation in Indoor, Unknown, and Unstructured Environments."*1994 International Conference on Intelligent Robots and Systems (lROS '94)*. Muenchen, Germany, September 12-16, pp. 431-438.

25. Skewis et al., 1991, "Motion Planning for Hospital Transportation Robot." *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, April, pp. 58-63.

26. TRC (Transition Research Corporation), Shelter Rock Lane, Danbury, CT, 06810-8159.

27. Tsumura T., Fujiwara, N., Shirakawa, T. and Hashimoto, M., 1981, "An Experimental System for Automatic Guidance of Roboted Vehicle Following the Route Stored in Memory." *Proc. of the 11th Int. Symp. on Industrial Robots*, Tokyo, pp. 18-193.

28. Walpole, R. E. and Myers, R. H., 1985, "*Probability and Statistics for Engineers and Scientists*," 3rd edition. Macmillan Publishing Company, New York, New York 10022.

Appendix A: The Effect of Unequal Wheel-diameters During Turning

In this Appendix we investigate how unequal wheel diameters affect on-the-spot turning of a differential-drive mobile robot. Figure A1 shows the two drive wheels of the robot before and after the nominal 90° turn. Since on-the-spot-turning requires that both wheels rotate at the same speed, we can assume that the angular velocity of both wheels is equal. However, due to the unequal wheel diameters the actual linear velocities of the wheels are proportional to the actual wheel diameters D_R and D_L. Thus, the *instantaneous center of rotation* (ICR) 'O' can be found easily as shown in Fig. A1. Note that 'O' does not coincide with the vehicle centerpoint C. At the completion of this turn point c will have moved to c'. The "on-the-spot" turn is therefore accompanied by a lateral displacement. However, in the square path experiment with four "on-the-spot" turns the four resulting lateral displacements balance and can be ignored.

We now wish to derive a relation between the actual wheel diameters D_L and D_R, and the actual angle of rotation τ. From Figure A1 we obtain

$$\frac{r_R}{r_L} = \frac{D_R}{D_L} \qquad \text{(A.1)}$$

where

$r_{R/L}$ is the distance from the ICR 'O' to the right or left wheel.

rewriting Eq. (A.1) yields

$$r_R = (D_R/D_L)\, r_L \qquad \text{(A.2)}$$

Under normal driving conditions the ICR is always on the drive axis (or along its imaginary extension beyond the wheels), so that

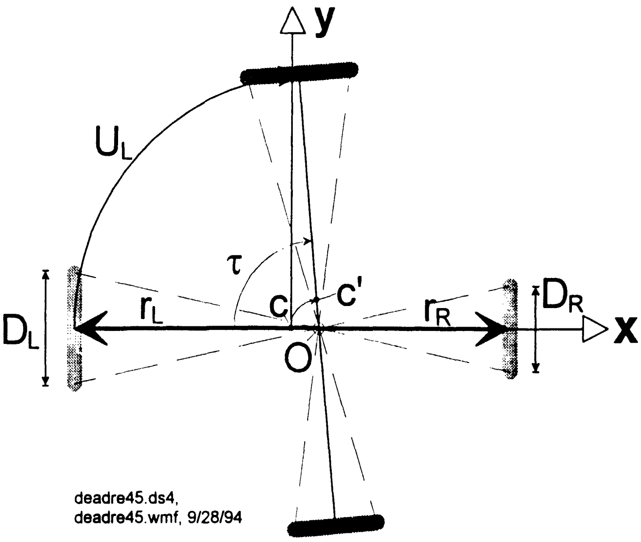

Figure A1: When turning through a nominal 90° turn, it is not the "unequal wheel diameter" error, but rather the "average actual wheel diameter" error, that affect the amount of turning.

$$r_R + r_L = b \qquad \text{(A.3)}$$

Substituting Eq. (A.2) into Eq. (A.3) and solving for r_L yields

$$r_L = \frac{D_L}{D_R + D_L} b \qquad \text{(A.4a)}$$

Next, we compute the nominal linear travel distance and the nominal number of encoder pulses that correspond to a 90° turn.

$$\frac{U_{n,90}}{\pi b} = \frac{90^\circ}{360^\circ} \tag{A.6}$$

or

$$U_{n,90} = \frac{\pi b}{4} \tag{A.7}$$

where the index *n* indicates a nominal value and *U* is the length of the arc prescribed by either one of the wheels (assuming nominal diameters) during turning.

Substituting Eq. (A.7) into Eq. (2.2) we can compute $N_{n,90}$, the nominal number of encoder pulses required for a 90° turn.

$$N_{n,90} = \frac{n\,C_e}{4\,D_n}\,b \tag{A.8}$$

Finally, we derive from the geometric relations in Fig. A1

$$\frac{U_L}{\tau} = \frac{2\pi r_L}{360^\circ} \tag{A.9}$$

or

$$\tau = \frac{360^\circ}{2\pi r_L}\,U_L \tag{A.10}$$

where U_L is the actual distance traveled by the left wheel, which has a diameter D_L. U_L is not known but we know that the left wheel is programmed to accumulate $N_{n,90}$ pulses, so that U_L can be computed as follows (again from Eq. (2.2)):

$$U_L = \frac{\pi D_L}{n\,C_e}\,N_{n,90} \tag{A.11}$$

Substituting Eq. (A.8) into Eq. (A.11)

$$U_L = \frac{\pi D_L}{4\,D_n}\,b \tag{A.12}$$

and substituting Eq. (A.12) in Eq. (A.10) we obtain the actual angle of rotation, τ:

$$\tau = \frac{360^\circ}{8 r_L}\,\frac{D_L}{D_n}\,b \tag{A.13}$$

Substituting Eq. (A.4b) yields

$$\tau = \frac{45^\circ (D_R + D_L)}{D_n} \qquad (A.14)$$

To interpret this result more easily, we define the average actual wheel diameter

$$D_{avrg} = \frac{(D_R + D_L)}{2} \qquad (A.15)$$

and we rewrite Eq. (A.14) as

$$\frac{D_{avrg}}{\tau} = \frac{D_n}{90^\circ} \qquad (A.16)$$

Equation (A.16) can be expressed in words as "The average actual wheel diameter relates to the actual angle of turning as the nominal wheel diameter relates to the nominal angle of turning."

Three important conclusions can be drawn from Eq. (A.16):

1. The orientation error due to unequal wheel diameters is independent of which wheel is larger or smaller than the nominal diameter. Therefore, the same error will be experienced in both cw and ccw turning. The error is thus of Type A.

2. The orientation error depends on the *average* actual wheel diameter $D_{avrg} = (D_L+D_R)/2$.
   If $D_{avrg} > D_n$, then the vehicle will turn more than the nominal amount.
   If $D_{avrg} < D_n$, then the vehicle will turn less.

3. The orientation error during turning does ***not*** depend on the *ratio* of the actual wheel diameters, $E_d = D_R/D_L$. Rather, E_d has a minor effect on the x and y position of centerpoint C, because the actual center of rotation, C', does not coincide with C, as shown in Fig. A.1.

Appendix B: Measurement and Correction of the Scaling Error E_s

In Section 2.3 we mentioned that the *scaling error* E_s is easily measured with an ordinary tape measure. One simple procedure for doing so requires that the robot be programmed to go straight for a distance of, say, 3 m. The experimenter marks the robot's starting and stopping positions on the floor. Using the tape measure, the distance between the two marks can be measured with an accuracy of typically ±1 mm or 0.03% of full scale. The actual distance is then compared to the calculated distance, based on the robot's dead-reckoning. E_s is thus defined as

$$E_s = L_{calc}/L_{actual} \tag{B.1}$$

Compensation for the linear scaling error is achieved by replacing the original nominal wheel diameter D_n in the deaa-reckoning program by the compensated nominal diameter D_n*:

$$D_n^* = E_s D_n \tag{B.2}$$

With the simple procedure explained above, the scaling error E_s will be $\leq$ 0.03%. For example, in a 10 m straight run, the position error due to E_s will be $\leq$ 3 mm. This is roughly one order of magnitude less than the typical error-spread due to *non-systematic* errors when traveling over reasonably smooth concrete floors.

APPENDIX C: UMBMARK EXPERIMENTAL RESULTS FOR SYSTEMATIC ERRORS

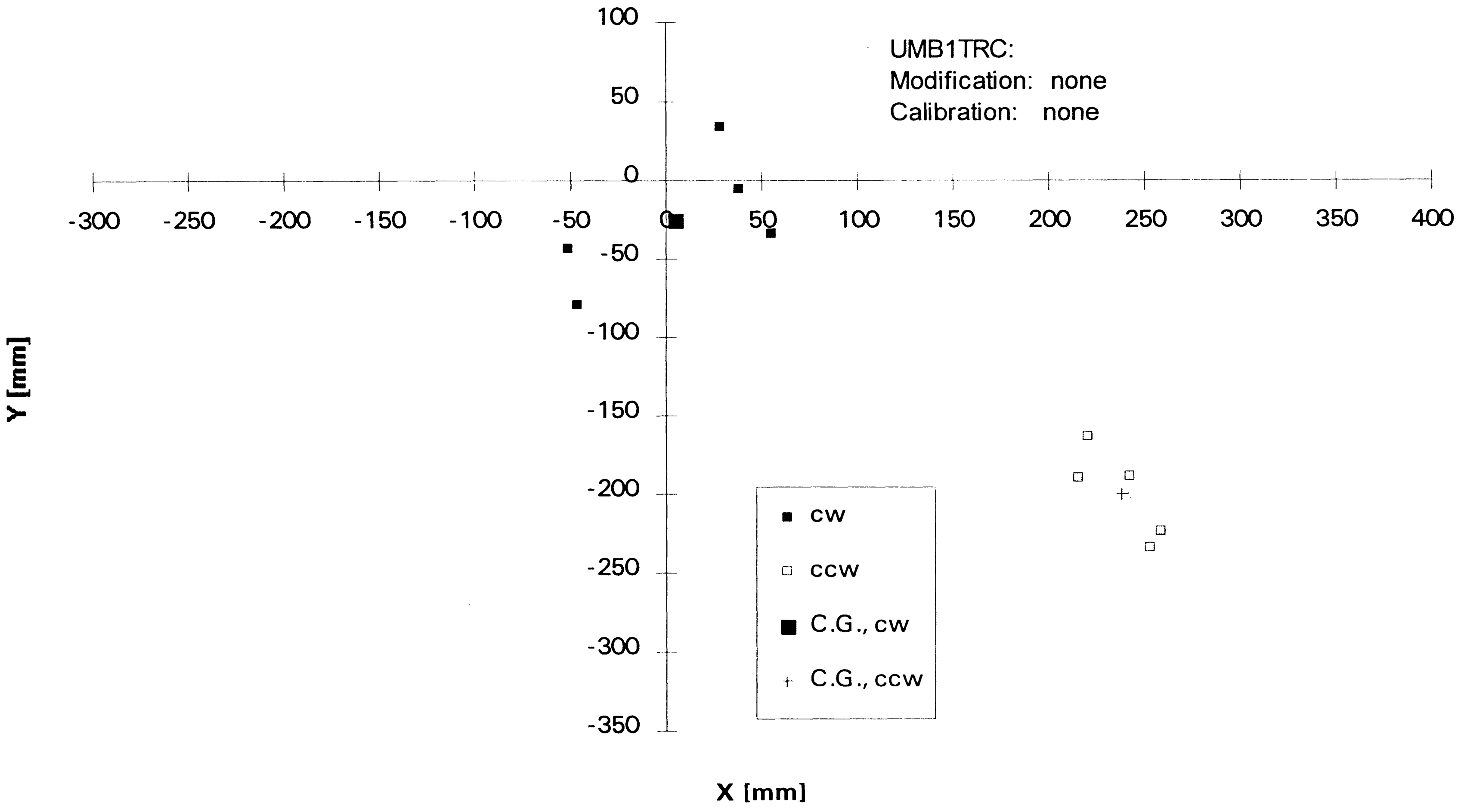

Figure C-1: Return Position Errors for the TRC LabMate. UMBmark test with 4×4 m bi-directional square path. Smooth concrete floor, no bumps.

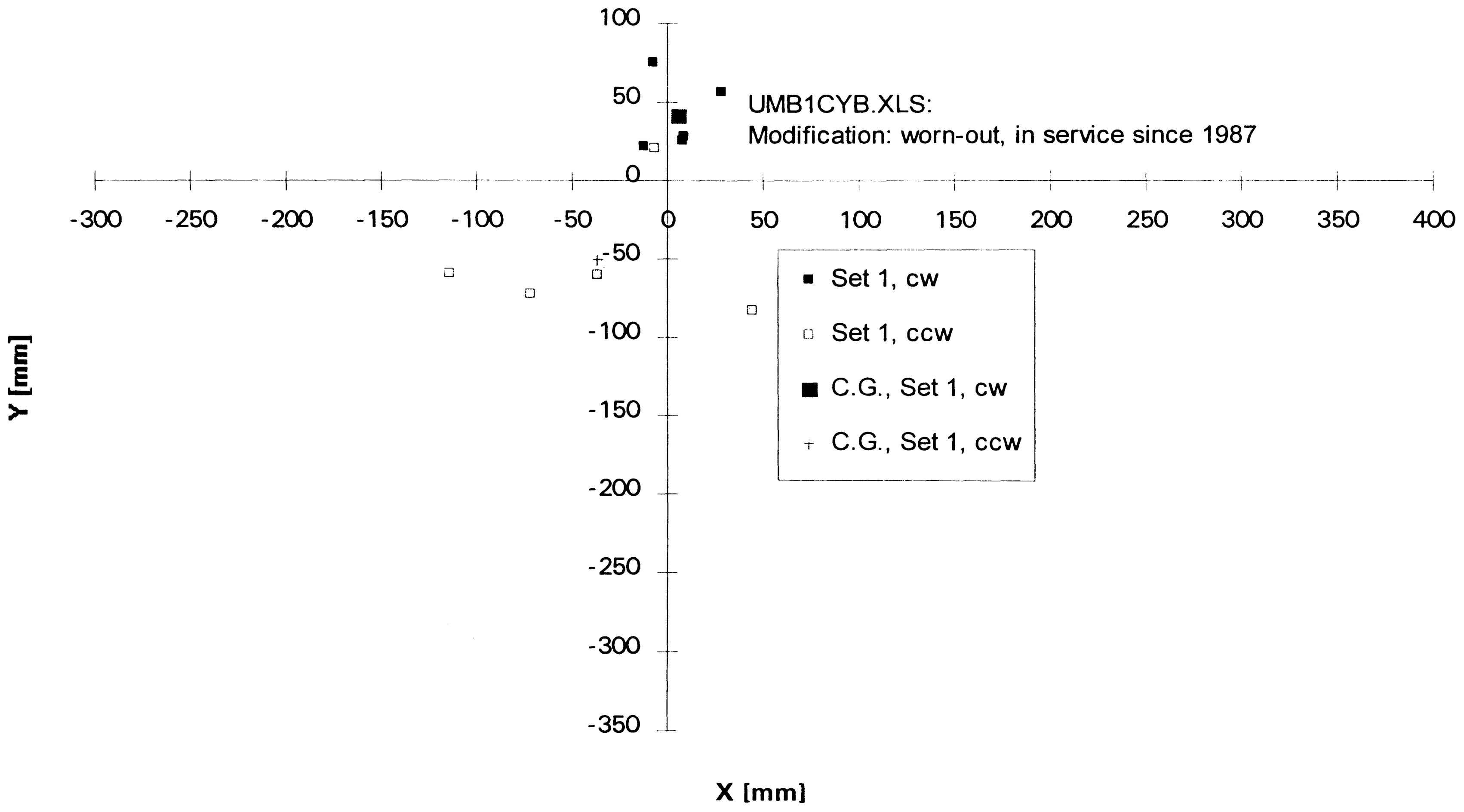

Figure C-2: Return Position Errors for the Cybermotion K2A. UMBmark test with 4×4 m bi-directional square path. Smooth concrete floor, no bumps.

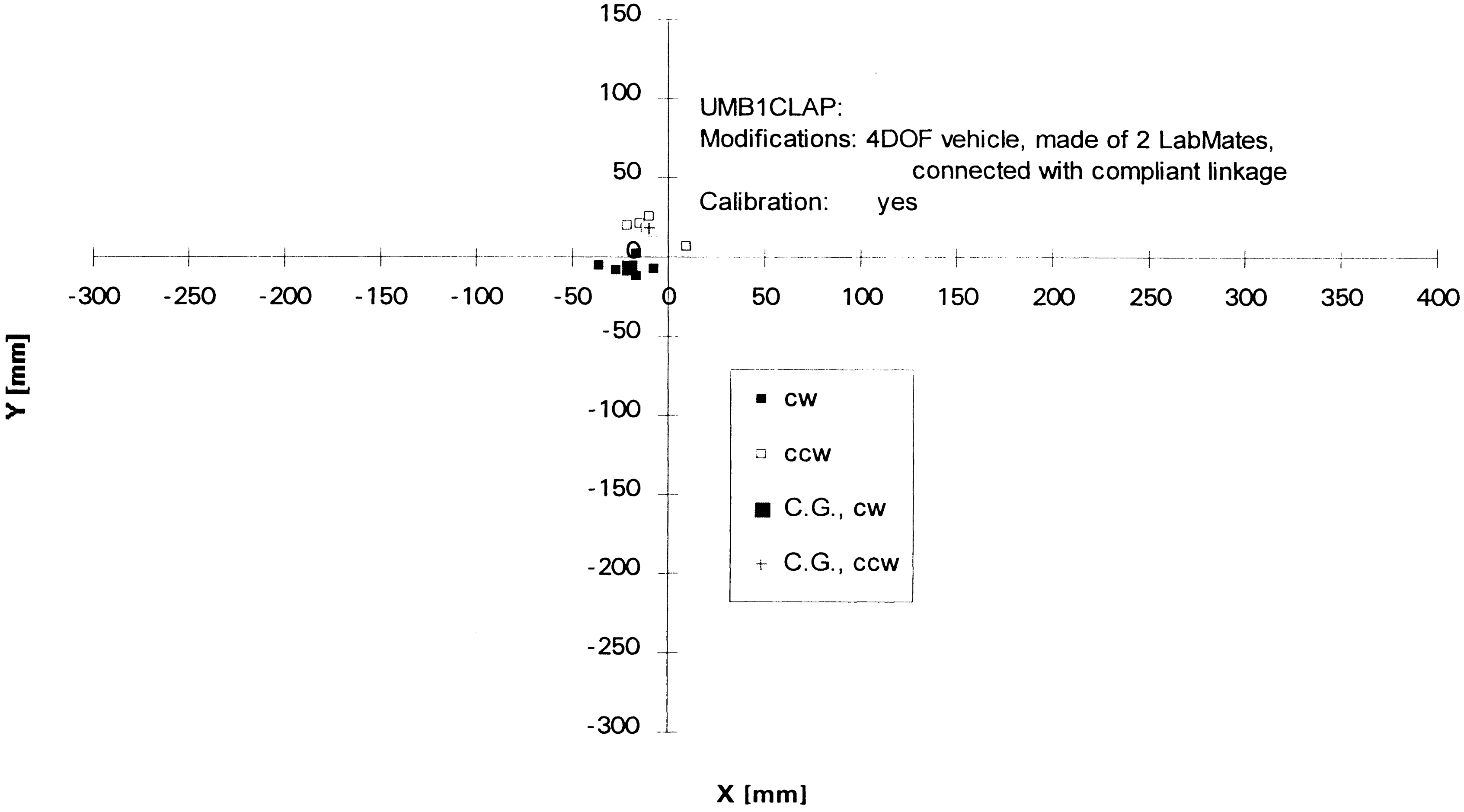

Figure C-3: Return Position Errors for the University of Michigan's CLAPPER. UMBmark test with 4×4 m bi-directional square path. Smooth concrete floor, no bumps.

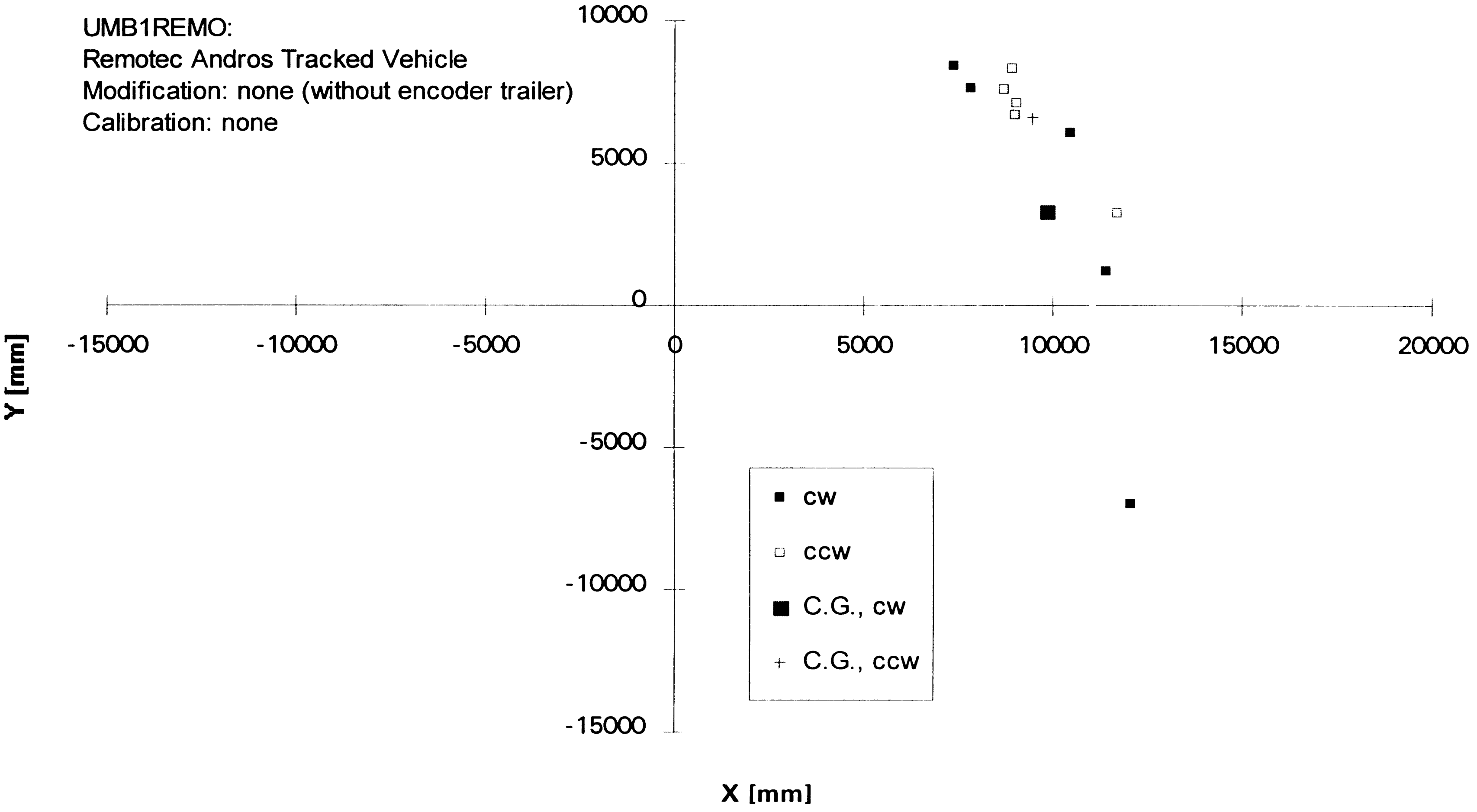

Figure C-4: Return Position Errors for the Remotec Andros. UMBmark test with 4×4 m bi-directional square path. Smooth concrete floor, no bumps. **Note that the scale of this plot is 50 times coarser than that of all other plots. The typical return position error of the tracked Andros vehicle was on the order of 8-10 meters! The average orientation error was 130°! Clearly, dead-reckoning with a tracked vehicle is out of the question.**

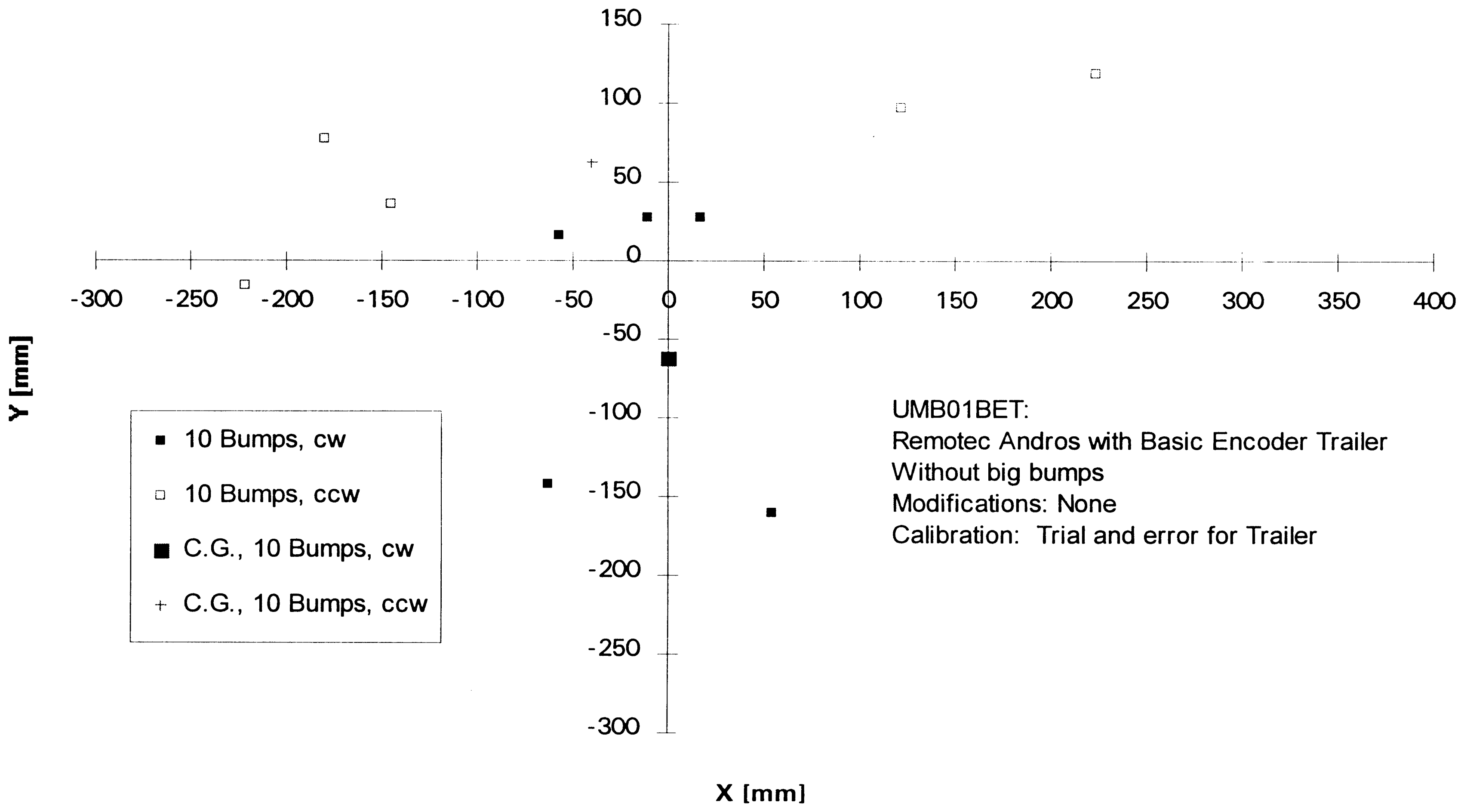

Figure C-5: Return Position Errors for the Remotec Andros with Basic Encoder Trailer. UMBmark test with 4×4 m bi-directional square path. Smooth concrete floor, no bumps. We hypothesize that the large spread of return position errors is a measuring error caused by wear in the slip-ring mounted on the joint between Andros platform and encoder trailer.

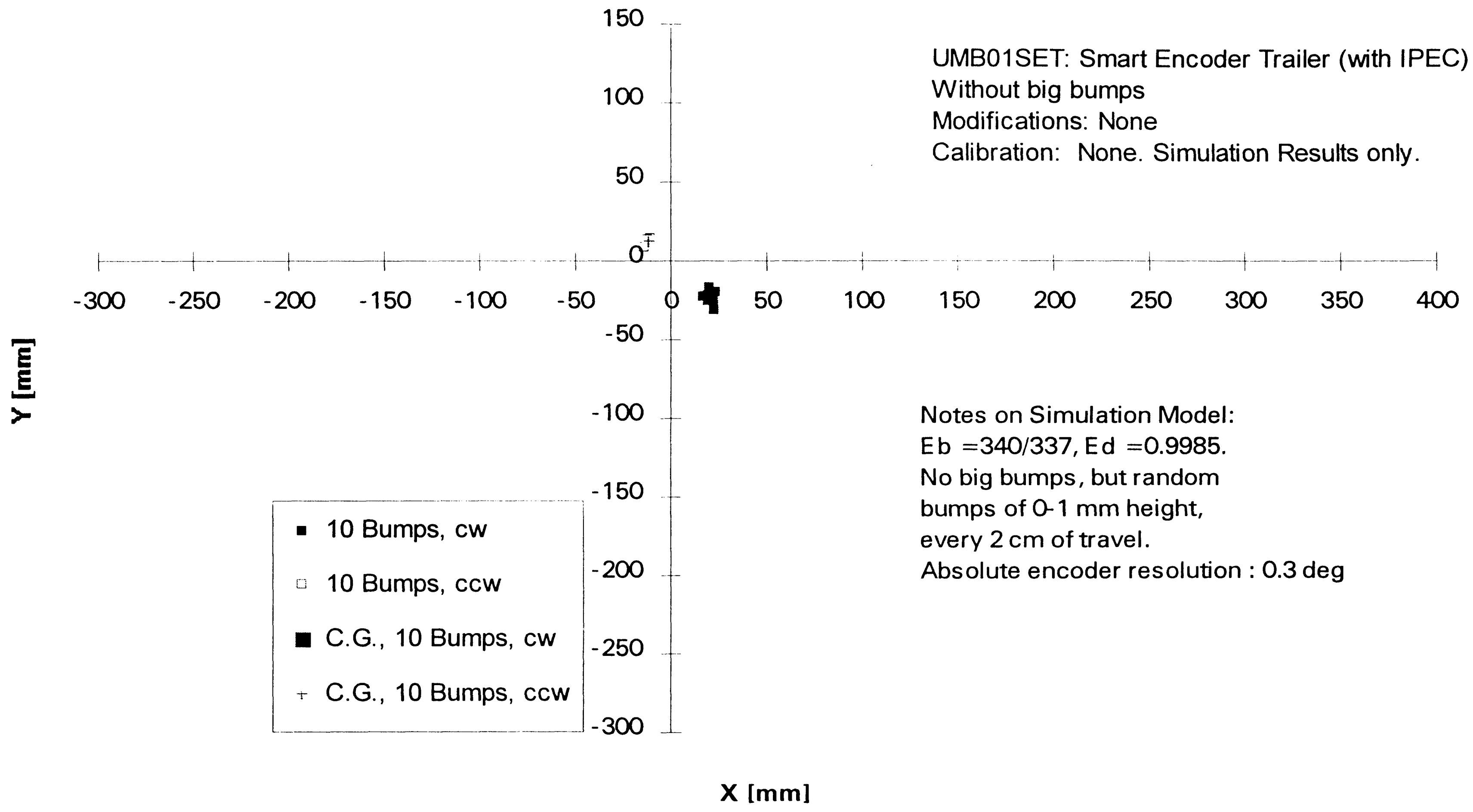

Figure C-6: *Simulation result* — return position errors for a TRC Labmate combined with the Smart Encoder Trailer (SET). UMBmark test with 4×4 m bi-directional square path. Smooth concrete floor, no bumps. Note how the accurately modeled systematic errors of the LabMate (we modeled the actually measured values from the uncalibrated LabMate in experiment E-2) are almost completely corrected by the SET (compare to Fig. E-2 with results "before correction"). Also note the (expected) similarity of the *simulated* result herewith the actual results from the CLAPPER in Figure

Appendix D:
Extended UMBmark Experimental Results for Non-systematic Errors

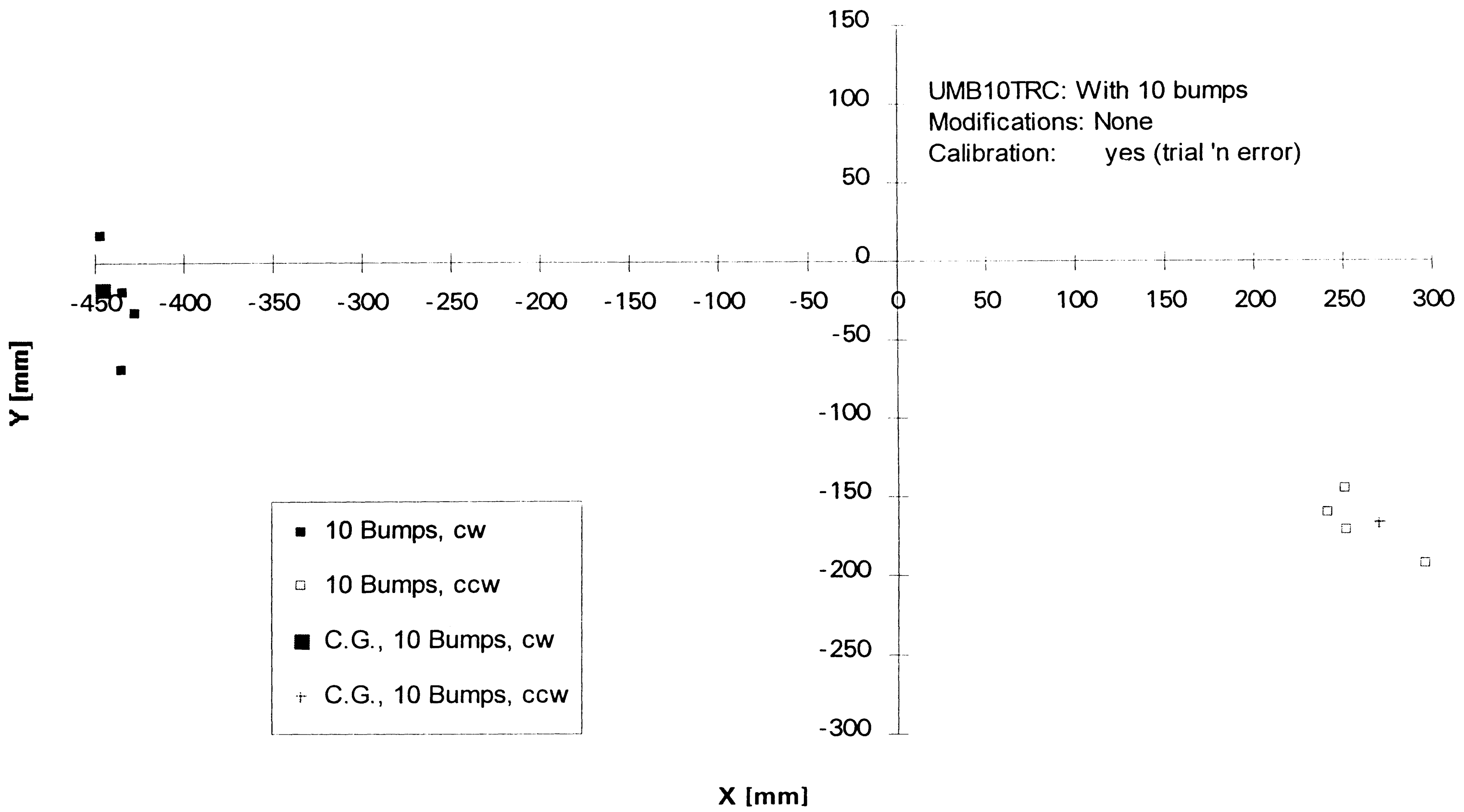

Figure D-1: Return Position Errors for the TRC Labmate. *Extended* UMBmark test with 4×4 m bi-directional square path. 10 bumps, each 10 mm high, were introduced during the first leg.

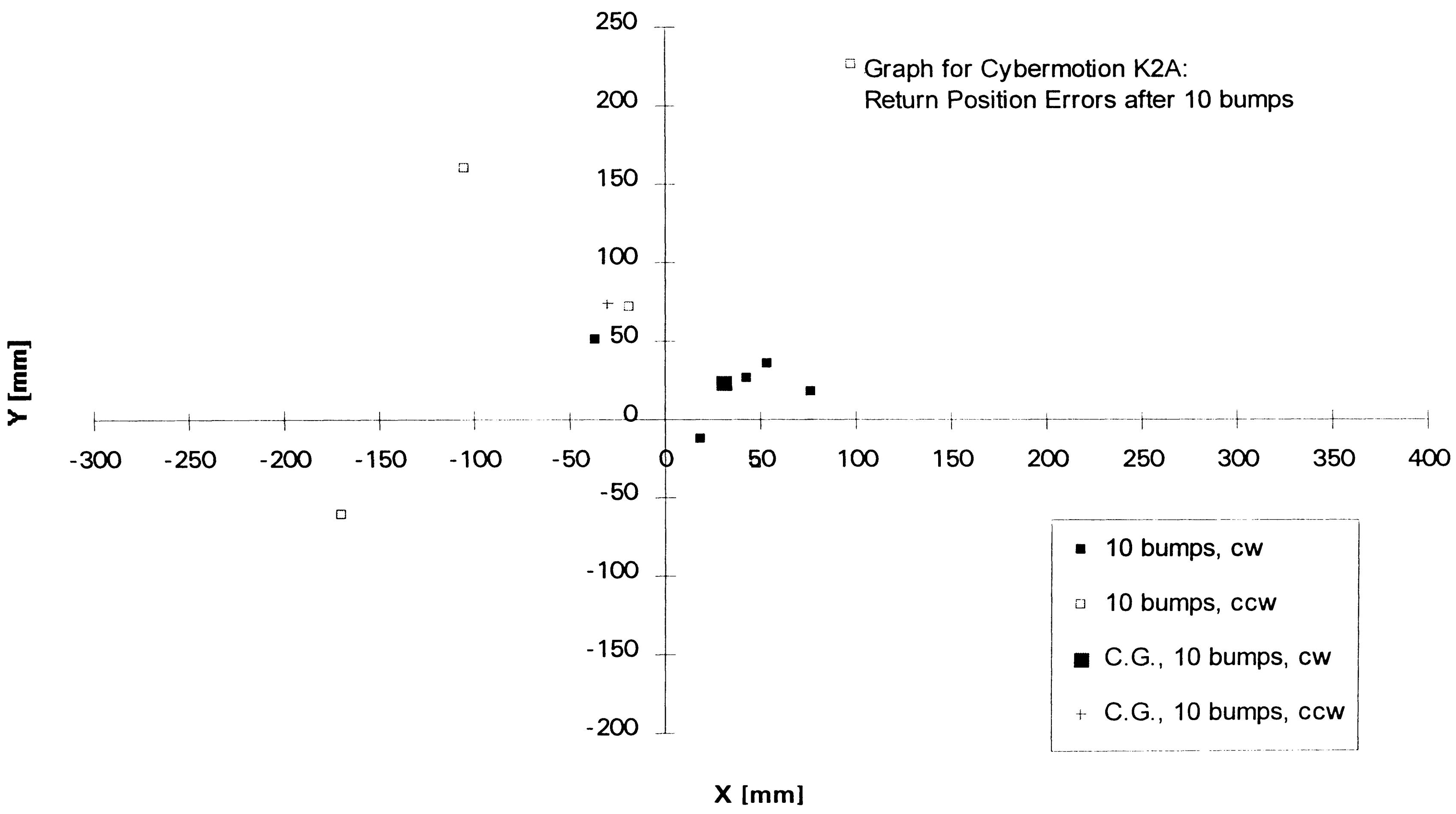

Figure D-2: Return Position Errors for the Cybermotion K2A. *Extended* UMBmark test with 4×4 m bi-directional square path. 10 bumps, each 10 mm high, were introduced during the first leg. It is difficult to explain the exceedingly wide spread of the return position errors in ccw direction.

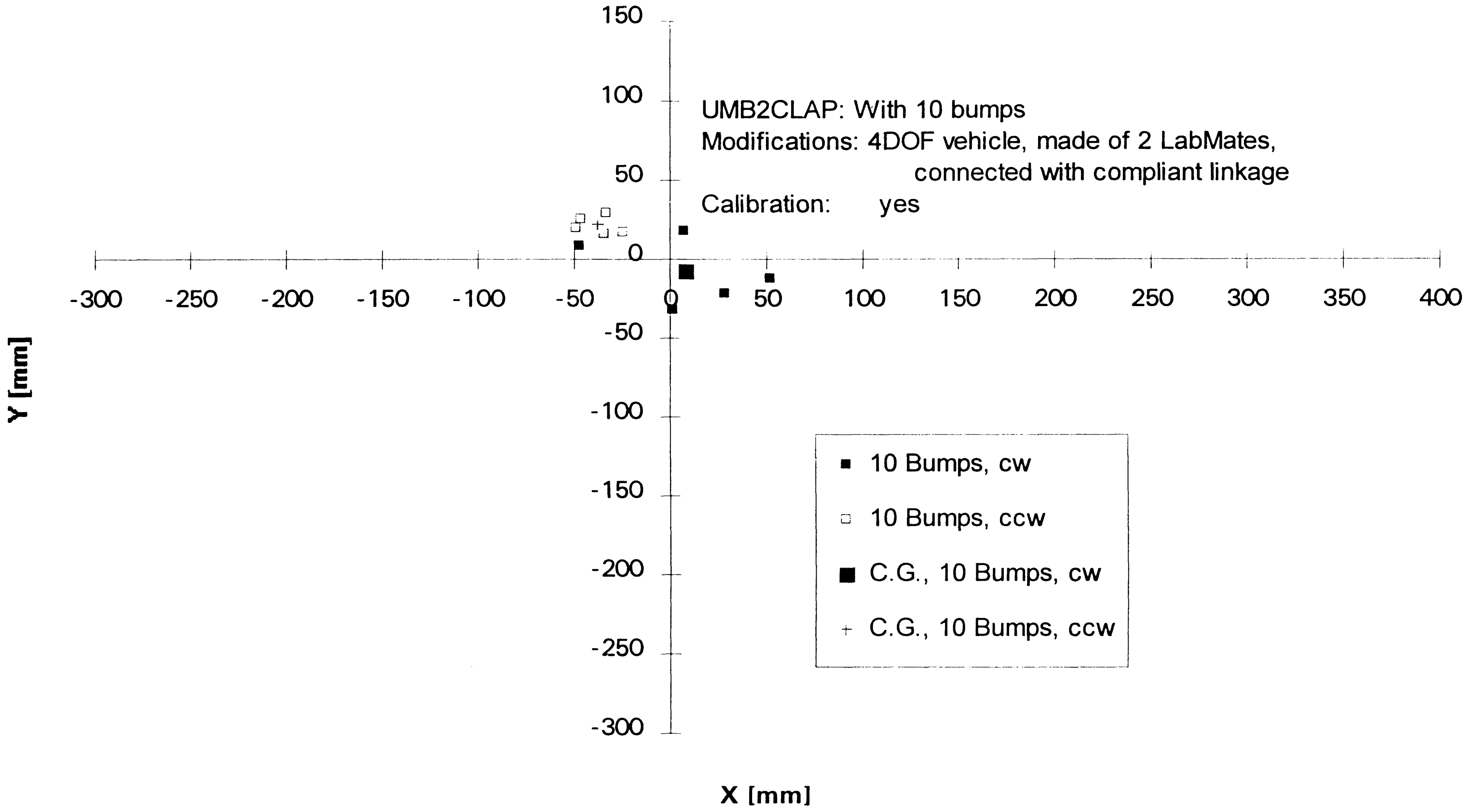

Figure D-3: Return Position Errors for the University of Michigan's CLAPPER. *Extended* UMBmark test with 4×4 m bi-directional square path. 10 bumps, each 10 mm high, were introduced during the first leg. The CLAPPER is especially designed to detect and correct non-systematic dead-reckoning errors. The result: a vehicle that is almost completely resilient to errors due to bumps, cracks, or other irregularities of the floor.

Figure D-4: This Figure should show the Return Position Errors for the Remotec Andros for the 4×4 m bi-directional square path test, with 10 bumps. However, this test is not applicable because the tracks of the Andros vehicle have grooves that are much larger than the 8-9 mm diameter bump used in our tests. The test result would therefore be essentially identical to the result without bumps, shown in Fig. C-4.

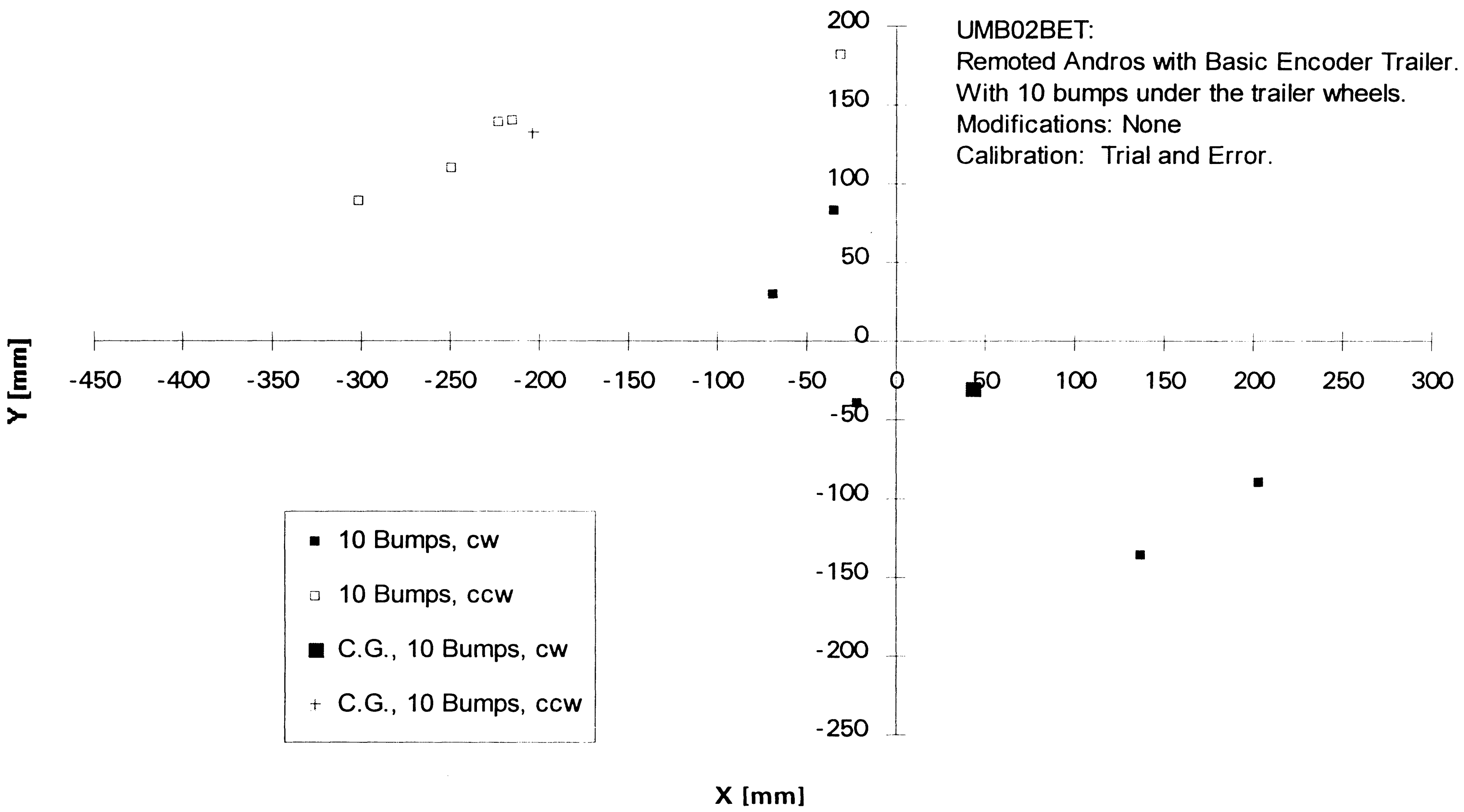

Figure D-5: Return Position Errors for the Remotec Andros with Basic encoder Trailer attached. This test comprised the nominal 4×4 m bi-directional square path. 10 bumps, each 8-9 mm high, were introduced under the trailer's wheels during the first leg of the run.

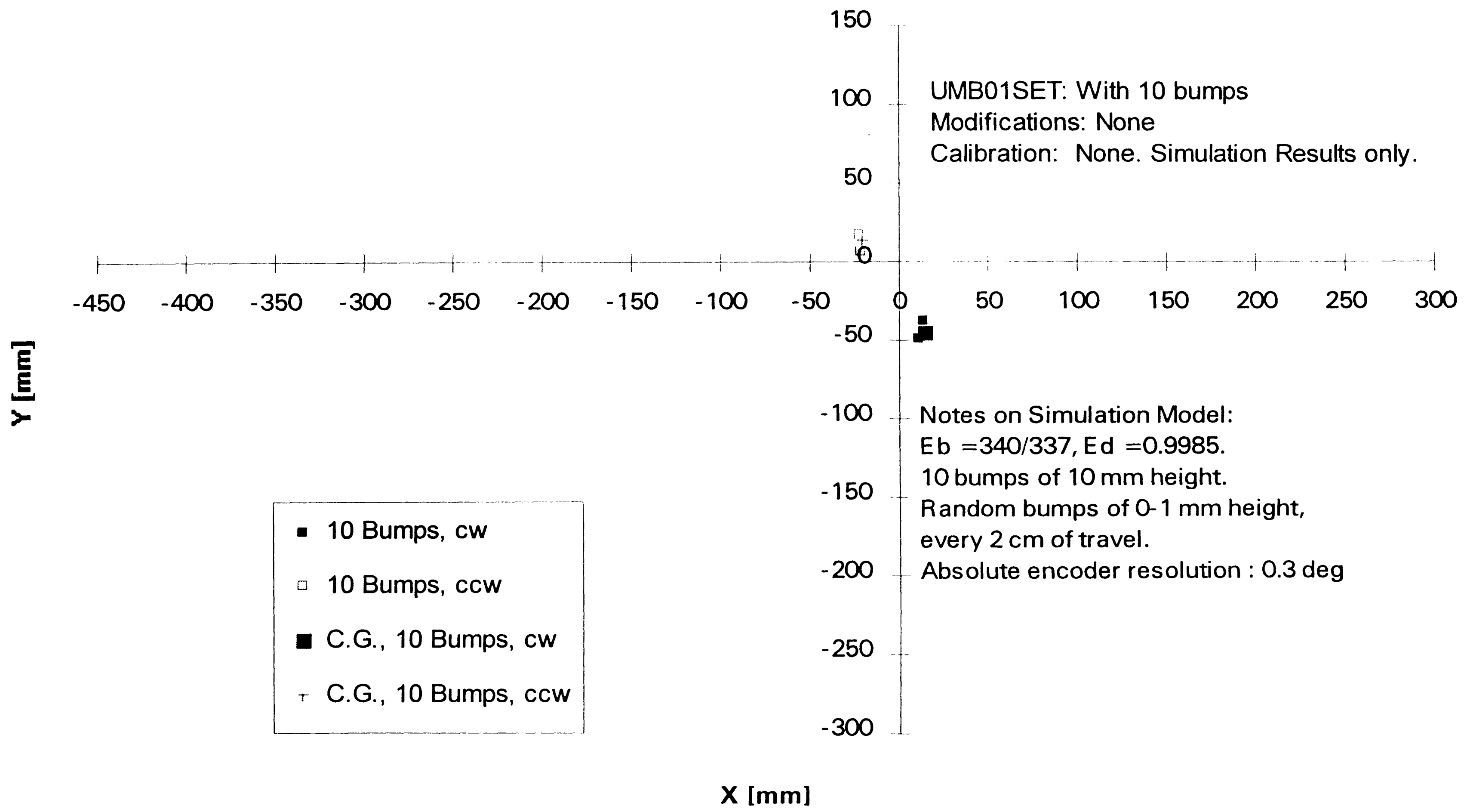

Figure D-6: Return Position Errors for a simulation run with the Smart Encoder Trailer (SET). The *Extended* UMBmark test with a 4×4 m bi-directional square path was applied, and, in addition, 10 bumps, each 10 mm high, were introduced during the first leg of the square path. The SET uses the same principle of error correction as the CLAPPER. The simulation program is directly derived from a similar simulation for the CLAPPER. Results from the CLAPPER simulation were consistently similar to the actual CLAPPER, increasing our trust in the accuracy of the SET simulation.

Appendix E: Correction of Systematic Errors — Experimental Results

(All experiments in Appendix E were performed with the same TRC LabMate)

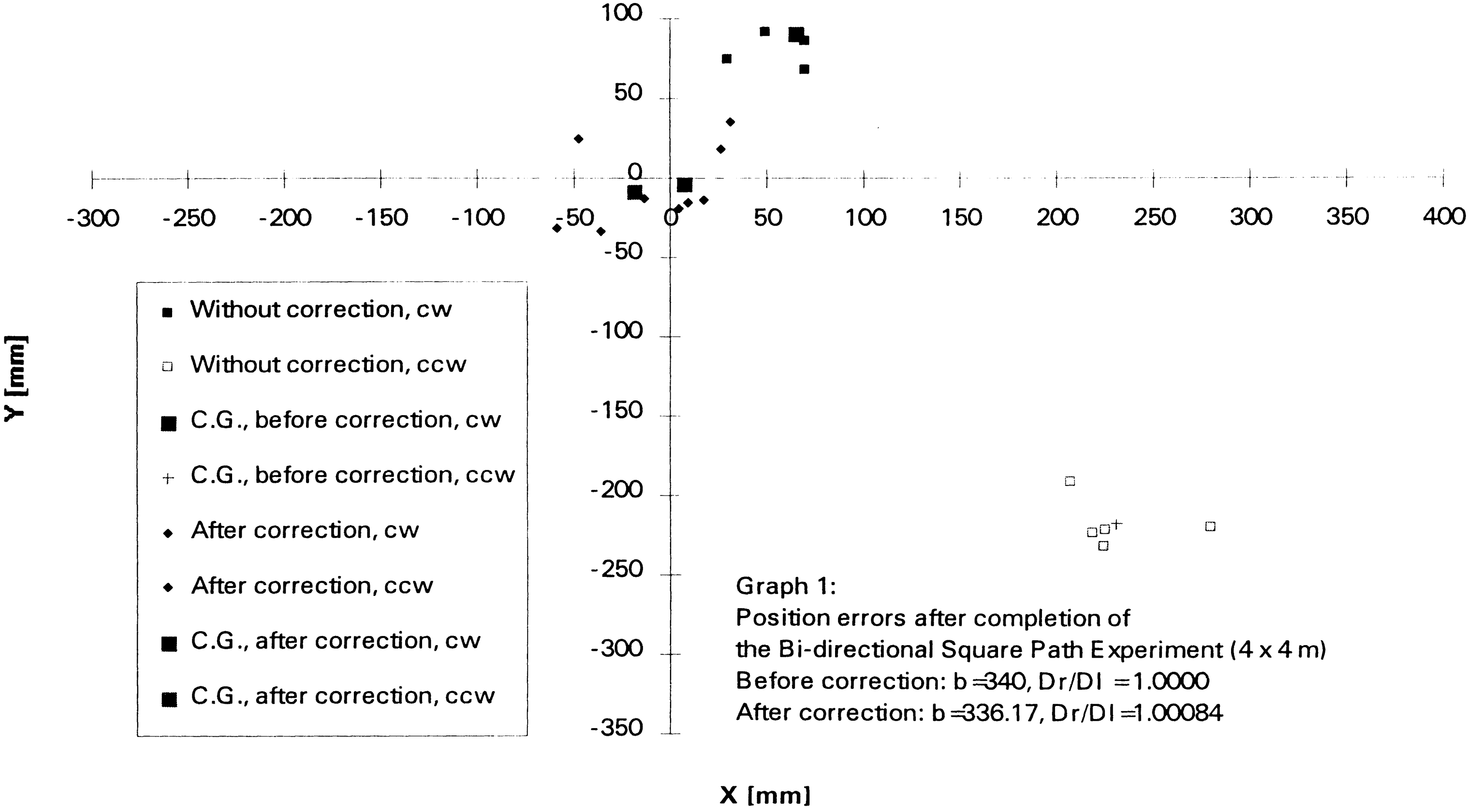

Figure E-1: Experiment E-1 shows how the *return position error* of the LabMate is reduced dramatically after compensation with the analytical compensation factors derived in Section 6. Here the improvement is 15-fold (Improvement is measured as the ratio between the furthest center of gravity (C.G.) of the uncalibrated runs and the furthest C.G. of the runs after calibration).

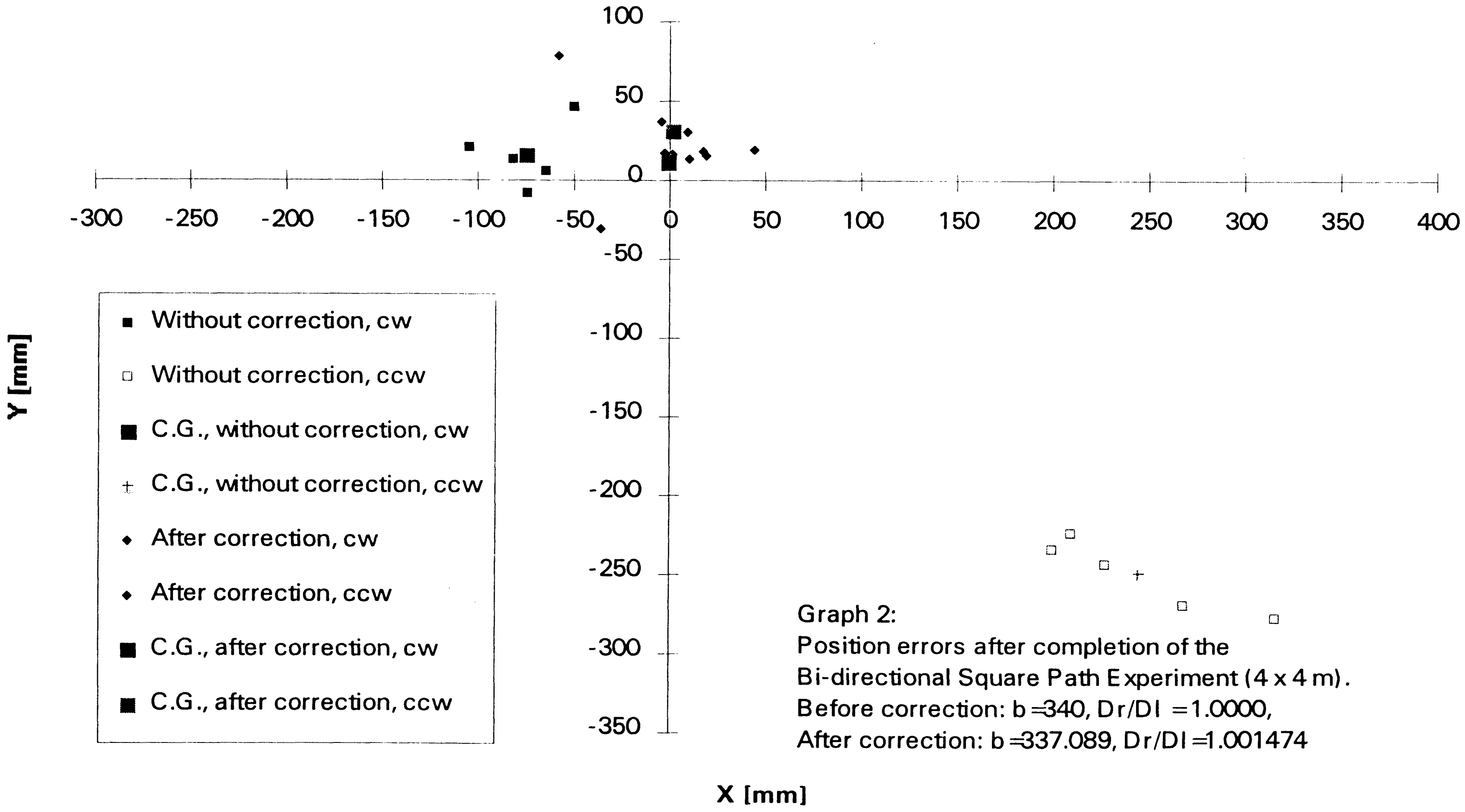

Figure E-2: Experiment E-2 shows another set of runs before and after calibration. Here the improvement is.

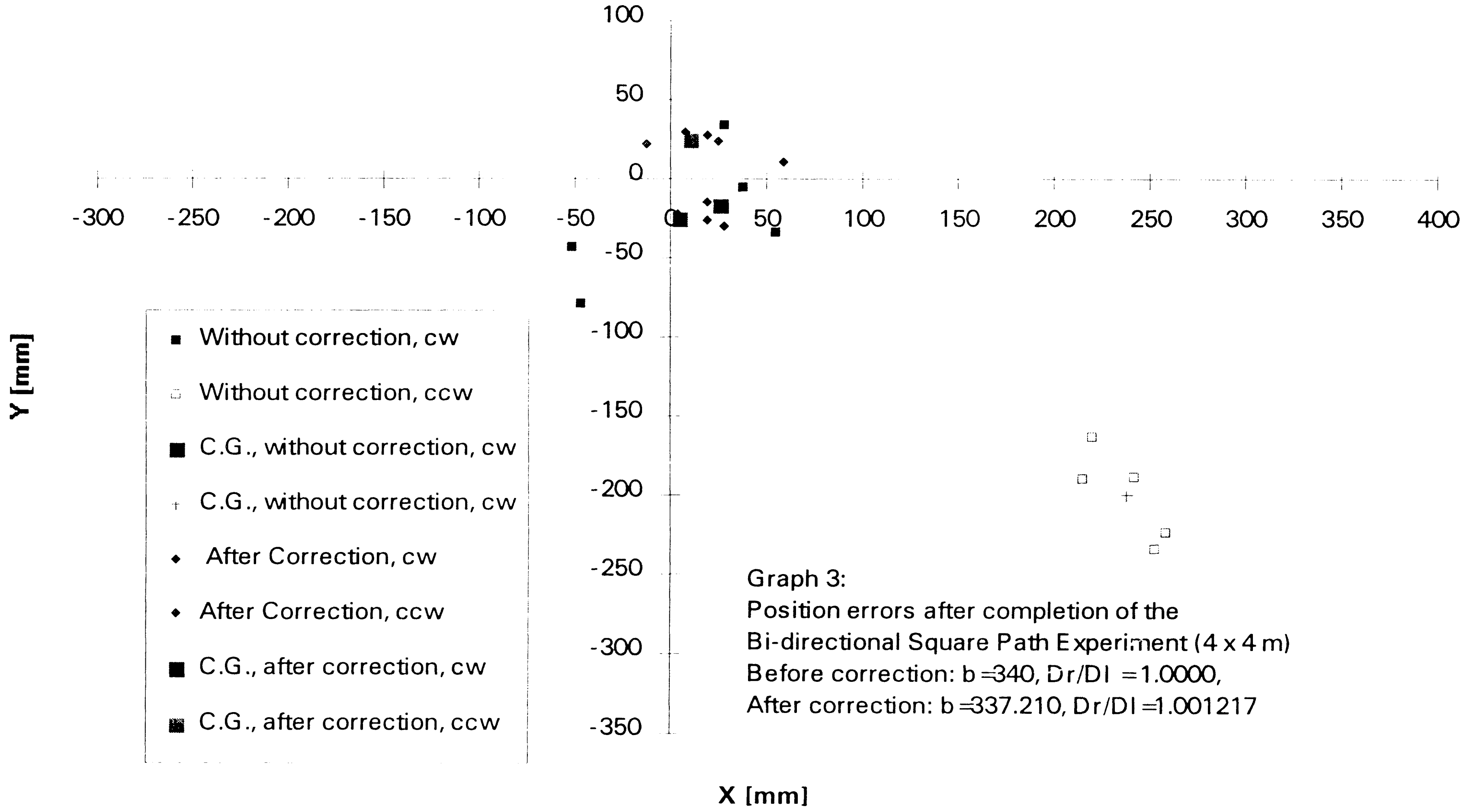

Figure E-3: Experiment E-3 shows yet another set of runs before and after calibration. Here the improvement is 10-fold (the least successful run in this series of eight experiments).

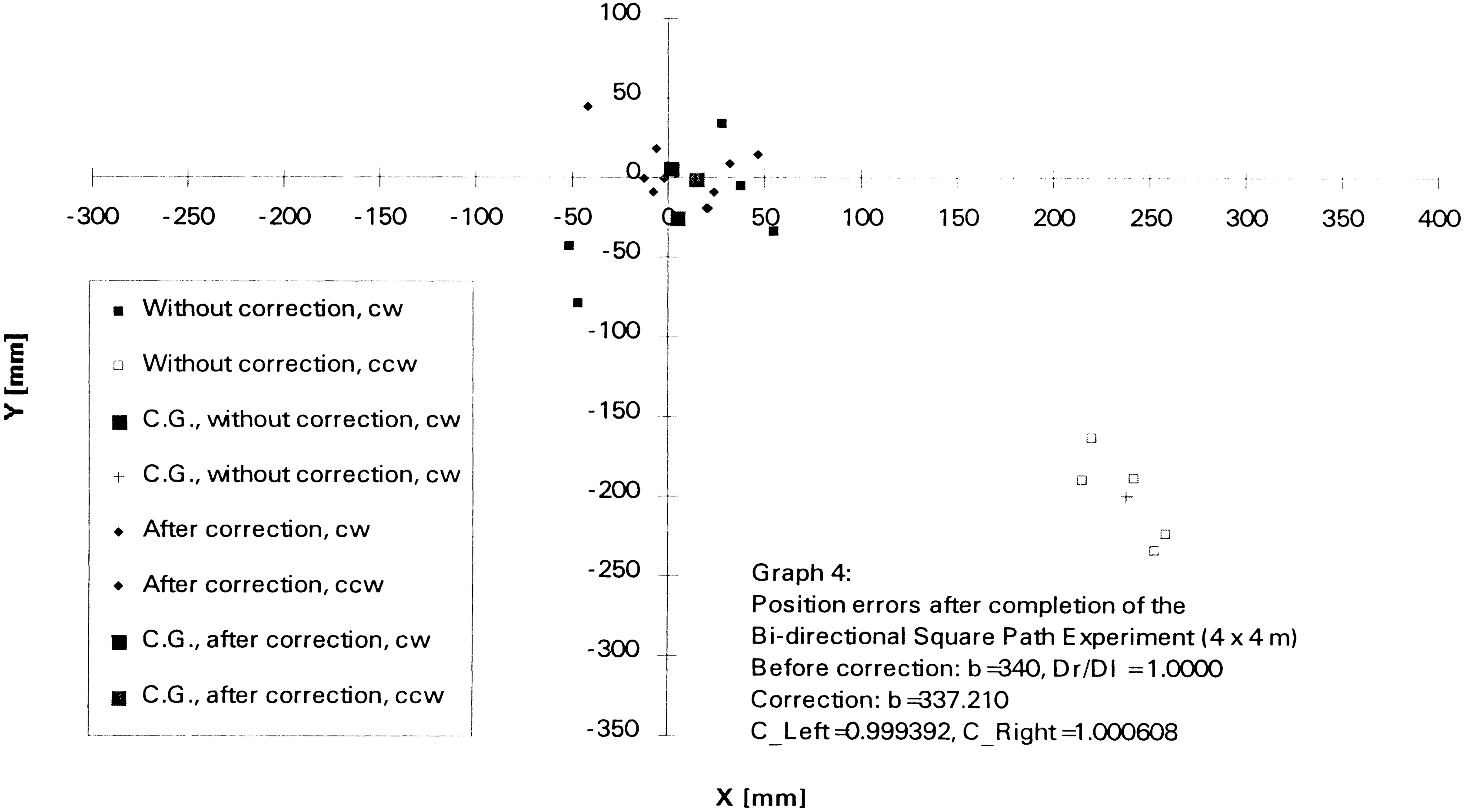

Figure E-4: Same set of correction factors used as experiment 3, but new set of "after-calibration runs.".Here the improvement is 21-fold (see text inSection 6.3 for explanation of this *seemingly.*different result, as compared to experiment E-3

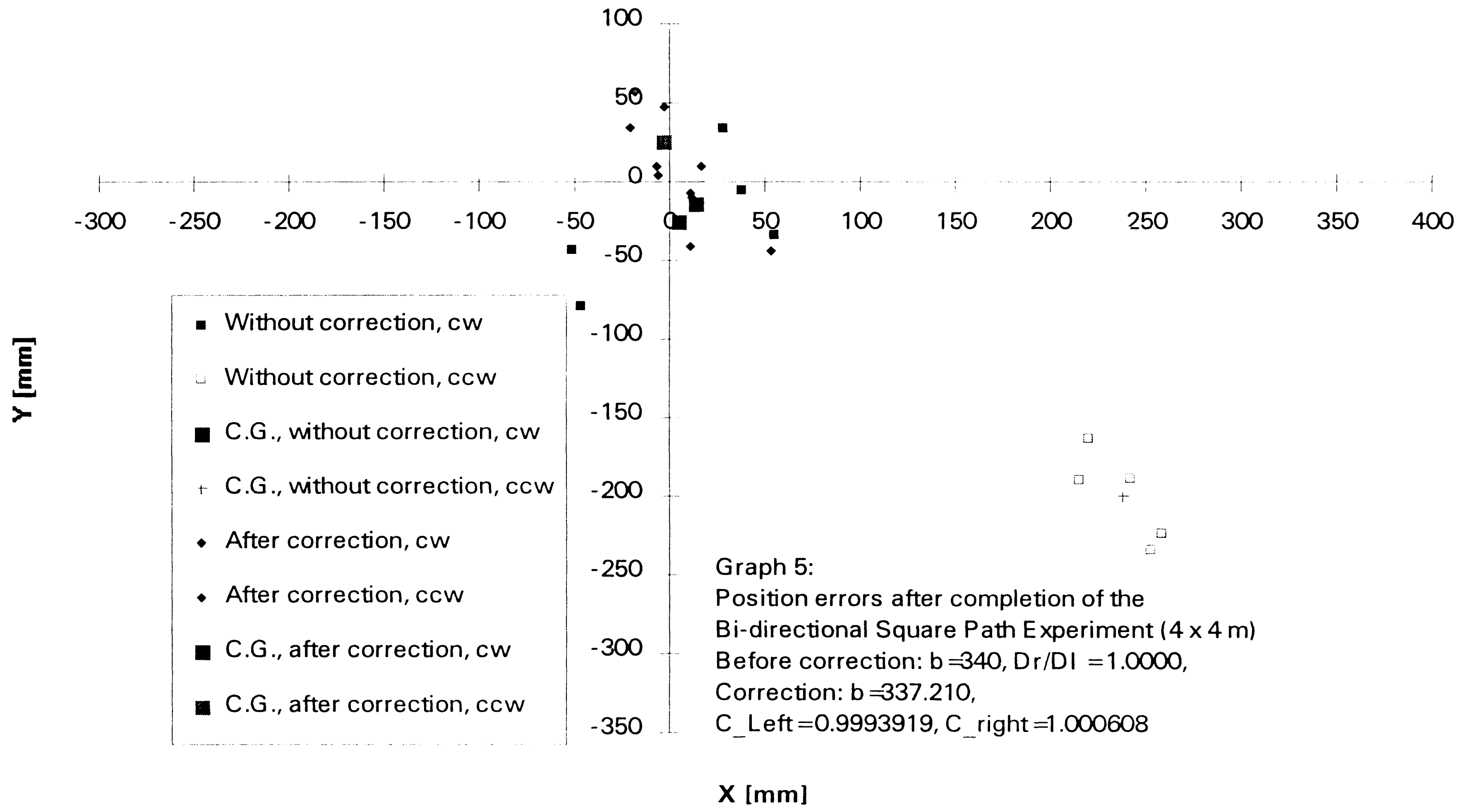

Figure E-5: Once more, the same set of correction factors as in E-3 and E-4 was used. The improvement this time is 12-fold. Experiments E-3, E-4, and E-5 reinforce our observation that the UMBmark calibration procedure provides *consistently* one order of magnitude improvment in dead-reckoning accuracy (with respect to systematic errors).

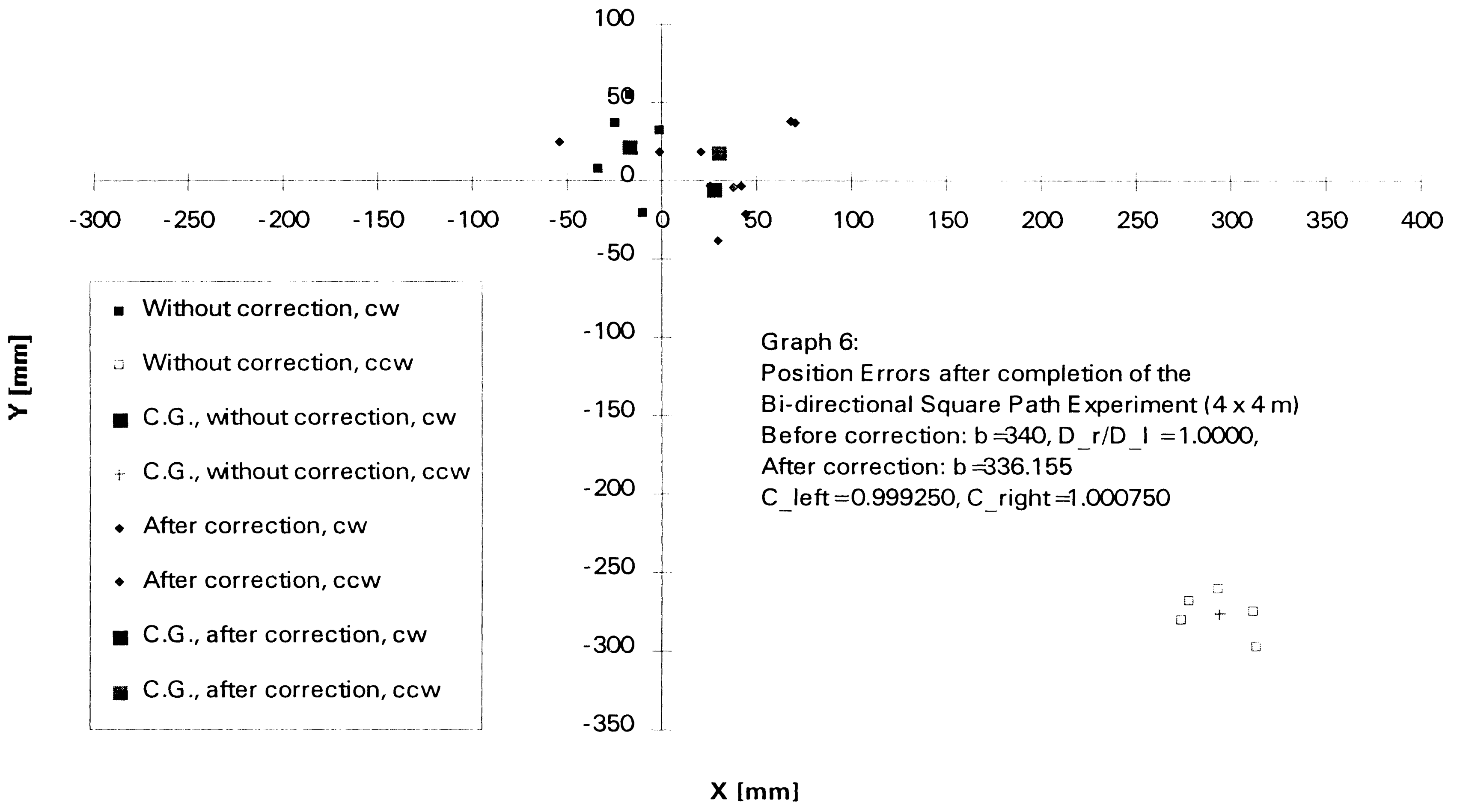

Figure E-6: To test the repeatability of the UMBmark calibration procedure, we ran the procedure again, from the beginning: we ran UMBmark without any correction, derived the correction factors from the results of the UMBmark run, and ran UMBmark again with the computed correction factors. Here the improvement in accuracy was 11-fold.

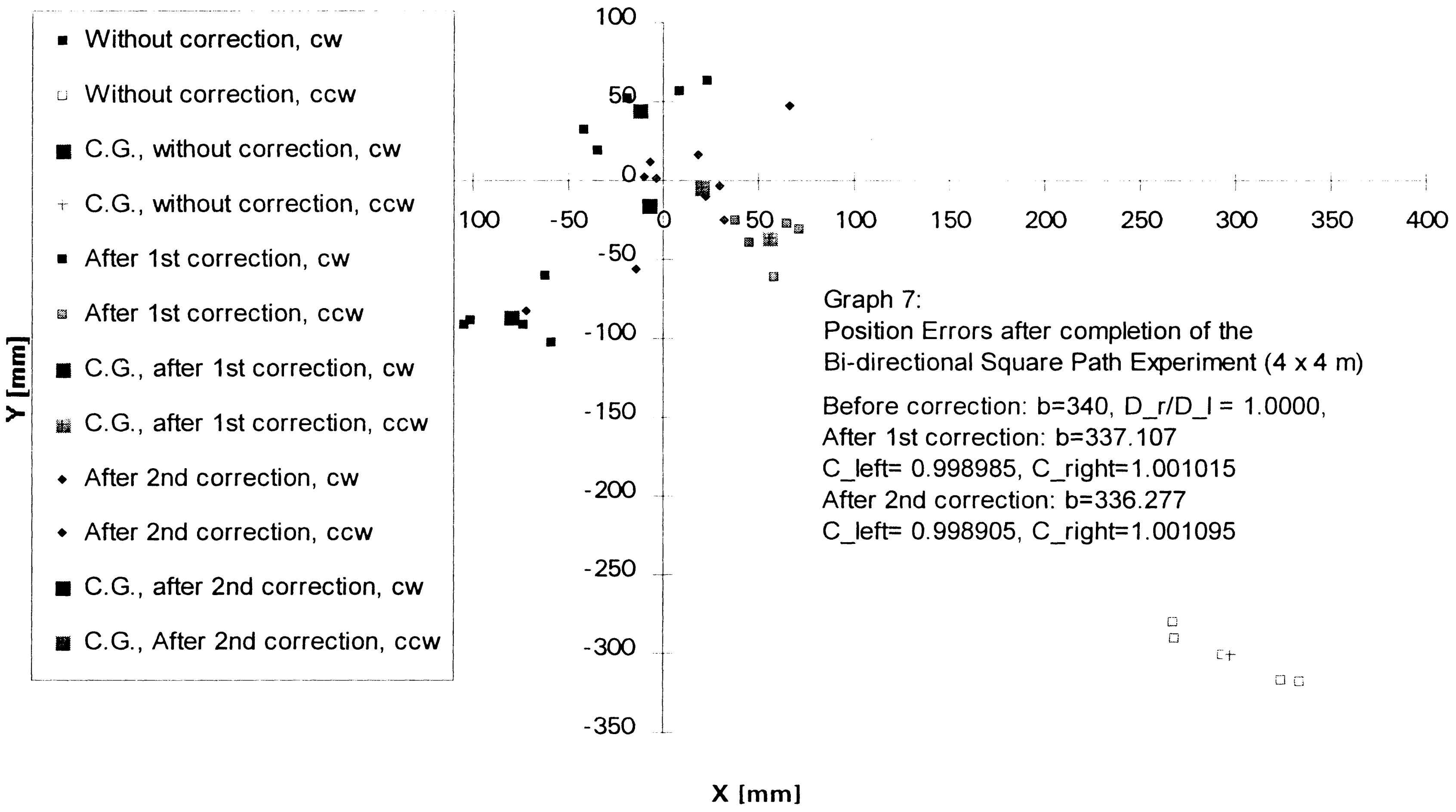

Figure E-7: In this experiment the diameter of the right wheel was slightly increased by winding three loops of masking tape around the wheel perimeter. This results effectively in a "different" vehicle.In this experiment, the initial improvement was only 6-fold. However, applying a simple test (explained in Section 6.3) it is immediately clear that further improvement can be achieved. Applying the UMBmark calibration procedure for a second time, the overall improvement becomes 21-fold.

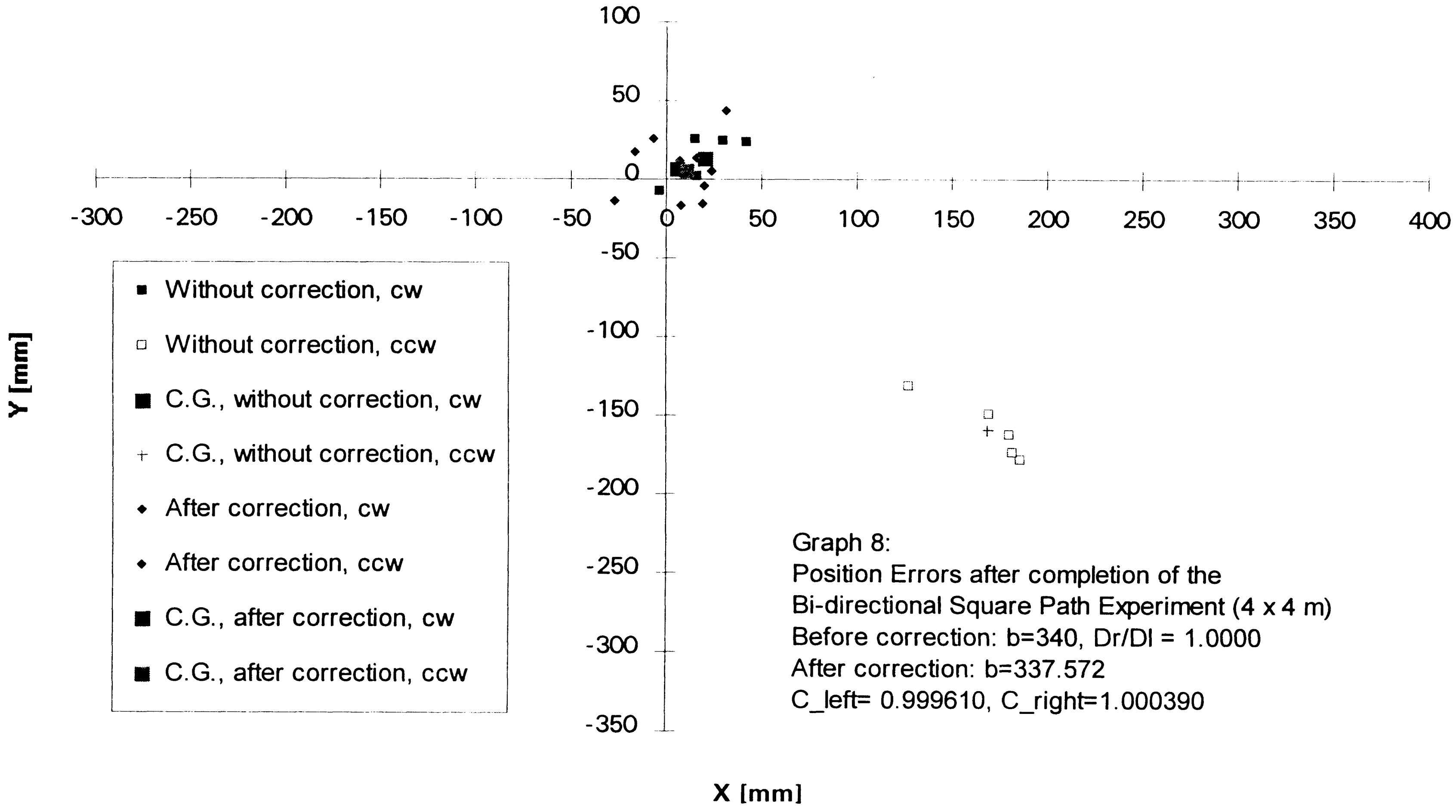

Figure E-8: In this experiment the diameter of the left wheel was slightly increased by winding five loops of masking tape around the left wheel perimeter. The improvement after applying the UMBmark calibration procedure was 19-fold.

Reprint of: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, May, 1994, pp. 3085-3090.

The CLAPPER: A Dual-drive Mobile Robot With Internal Correction of Dead-reckoning Errors

Johann Borenstein
The University of Michigan, MEAM Mobile Robotics Lab

ABSTRACT

This paper presents a new approach to accurate and reliable dead-reckoning with mobile robots. The approach makes use of special properties of our recently developed *Multi-Degree-of-Freedom* (MDOF) mobile platform, in which two differential-drive mobile robots (called "trucks") are physically connected through a *compliant linkage*. Using one linear and two rotary encoders, the system can measure the relative distance and bearing between the two trucks. During operation, both trucks perform conventional dead-reckoning with their wheel encoders, but, in addition, use information about their relative position to correct dead-reckoning errors.

Our system, called *Compliant Linkage Autonomous Platform with **P**osition **E**rror **R**ecovery* (CLAPPER), requires neither external references (such as navigation beacons, artificial landmarks, known floorplans, or satellite signals), nor *inertial navigation aids* (such as accelerometers or gyros). Nonetheless, the experimental results included in this paper show one to two orders of magnitude better positioning accuracy than systems based on conventional dead-reckoning.

1. INTRODUCTION

In most mobile robot applications two basic position-estimation methods are employed together: *absolute* and *relative* positioning [Borenstein and Koren, 1987; Hongo et al, 1987]. Relative positioning is usually based on dead-reckoning (i.e., monitoring the wheel revolutions to compute the offset from a known starting position). Dead-reckoning is simple, inexpensive, and easy to accomplish in real-time. The disadvantage of dead-reckoning is its unbounded accumulation of errors.

Absolute positioning methods usually rely on (a) navigation beacons, (b) active or passive landmarks, (c) map matching, or (d) satellite-based navigation signals. Each of these absolute positioning approaches can be implemented by a variety of methods and sensors. Yet, none of the currently existing systems is particularly elegant. Navigation beacons and landmarks usually require costly installations and maintenance, while map-matching methods are either very slow or inaccurate [Cox, 1991], or even unreliable [Congdon et al, 1993]. With any one of these measurements it is necessary that the work environment be either prepared or be known and mapped with great precision. Satellite-based navigation can be used only outdoors and has poor accuracy (on the order of several meters) when used in real-time, during motion.

Another approach to the position determination of mobile robots is based on inertial navigation with gyros and/or accelerometers. Our own experimental results with this approach, as well as the results published in a recent paper by Barshan and Durrant-Whyte [1993], indicate that this approach is not advantageous. Accelerometer data must be integrated twice to yield position, thereby making these sensors exceedingly sensitive to *drift*. Another problem is that accelerations under typical operating conditions can be very small, on the order of 0.01 g. Yet, fluctuation of this magnitude already occur if the sensor deviates from a perfectly horizontal position by only 0.5°, for example when the vehicle drives over uneven floors. Gyros can be more accurate (and costly) but they provide information only on the rotation of a vehicle.

This paper introduces a new method for correcting dead-reckoning errors without external references. This method requires two collaborating mobile robots that can accurately measure their relative distance and bearing during motion. Our previously developed MDOF vehicle [Borenstein, 1993; 1994a] meets these requirements and we were able to implemented and test our error correction method on this vehicle with only minor modifications. Section 2 summarizes the relevant characteristics of our MDOF vehicle. Because of the new error correction capability, we now call our vehicle the *Compliant Linkage Autonomous Platform with **P**osition **E**rror **R**ecovery* (CLAPPER). Section 3 describes the error correction method in detail, and Section 4 presents experimental results.

2. THE MDOF COMPLIANT LINKAGE VEHICLE

In previous research we have developed an innovative *Multi-Degree-of-Freedom* (MDOF) vehicle with *compliant linkage* (Fig. 1).

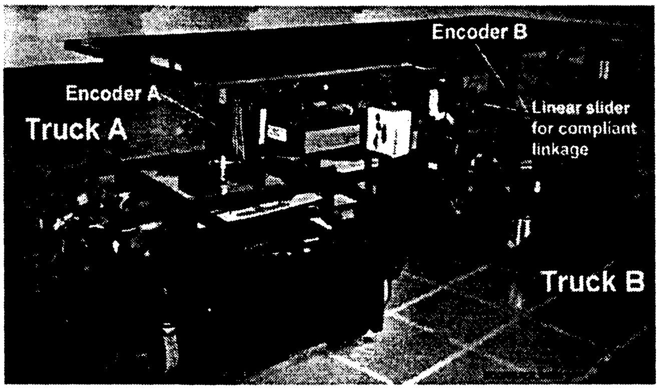

Figure 1: The University of Michigan's experimental dual differential drive vehicle with compliant linkage.

The advantage of MDOF vehicles over conventional mobile robots is that they can travel sideways and they can negotiate tight turns easily. However, existing MDOF vehicles have been found difficult to control because of their overconstrained nature [Reister, 1991; Killough and Pin, 1992]. These difficulties translate into severe wheel slippage or jerky motion under certain driving conditions. Because of this excessive wheel-slippage existing MDOF vehicles are not very suitable for mobile robot applications that rely heavily on dead-reckoning. Our MDOF vehicle overcomes these difficulties by introducing the *compliant linkage* design (Fig. 2). The compliant linkage accommodates momentary controller errors and thereby successfully eliminates the excessive wheel slippage reported by other makers of MDOF vehicles.

The schematic drawing in Fig. 2 shows the essential components of the compliant linkage vehicle. The vehicle comprises of two trucks (in our prototype, these are commercially available *LabMate* robots from TRC [1993]). The two trucks are connected by the compliant linkage, which allows force-free relative motion within its physical range. A linear encoder measures the momentary distance between the two trucks, and two absolute rotary encoders measure the rotation of the trucks relative to the compliant linkage. Each of the four drive wheels in the system has a shaft encoder to allow conventional dead reckoning.

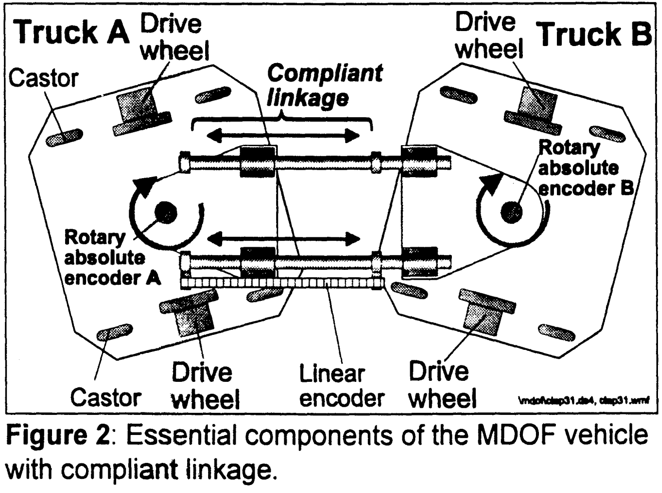

Figure 2: Essential components of the MDOF vehicle with compliant linkage.

The linear incremental encoder has a resolution of 0.1 mm, but the actual accuracy of distance measurements between the two trucks is only ±5 mm because of mechanical inaccuracies in our prototype vehicle. The resolution of the rotary absolute encoders is 0.3°. We will call these the three *"internal"* encoders.

The experiments with our MDOF vehicle [Borenstein, 1993V1] showed that control errors are effectively absorbed by the compliant linkage, resulting in smooth and precise motion without excessive wheel slippage. In a series of 4×4 m square path experiments we found typical dead-reckoning errors to be less than 6.5 cm in x and y direction, and orientation errors were less than ±1° [Borenstein, 1994a]. This dead-reckoning accuracy is comparable with that of conventional 2-DOF robots. Of course, these results were obtained on smooth floors without irregularities, and with well calibrated parameters to minimize systematic errors.

3. INTERNAL CORRECTION OF DEAD-RECKONING ERRORS

At first glance, it may appear impossible to obtain accurate position corrections from a "floating reference point," such as another mobile robot ***in motion***. Yet, our method is designed to overcome this problem: it exploits the fact that certain dead-reckoning errors develop slowly while others develop quickly. For example, when a robot traverses a bump or crack in the floor, it will experience an appreciable orientation error within just a few centimeters of travel ("*fast-growing*" error). The lateral position error, on the other hand, is very small at first ("*slow-growing*" error), although it will grow with distance as a function of the orientation error. Our method performs relative position measurements very frequently, allowing each truck to detect errors in its orientation (which can have changed significantly during one sampling interval), while relying on the fact that the lateral position error of both trucks was only small during the same interval.

Before we present the details of our method we should make a clear distinction between two types of dead-reckoning errors found in mobile robot navigation: (a) *systematic errors*, which are related to properties of the vehicle, that is, they are independent of the environment; and (b) *non-systematic errors*, which are usually caused by *irregularities* or roughness of the floor.

Note that non-systematic errors can neither be avoided nor can they be compensated for in conventional dead-reckoning. By contrast, the CLAPPER can detect and reduce both systematic and non-systematic errors by one to two orders of magnitude, even with both trucks in motion.

In this Section we explain our approach for the simple case of straight-forward motion, but the method works equally well (and without any modification) for curved trajectories. We will develop a numeric example throughout this section, to show why certain assumptions are valid.

At first, we assume that both trucks are longitudinally aligned and travel forward. For the sake of the numeric example, let us assume that both trucks are traveling at $V = 0.5$ m/s, and that the sampling time of the *internal encoders* is $T_s = 40$ ms. Thus, during a sampling interval both trucks travel a distance $D_s = VT_s = 20$ mm.

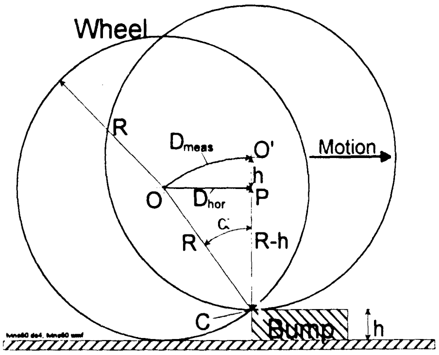

Figure 3: Simplified geometry of wheel traversing a bump.

Next, we consider the geometry of a wheel of radius R traversing a bump of height h (see Fig. 3). Making the simplifying assumption that the wheel was perfectly rigid, the wheel will traverse the bump by rotating around the point of contact C until the wheel's center point O is right above C (at O'). During this motion the wheel encoder measures a rotation α, which is interpreted as the linear travel distance D_{meas}. Yet, the actual travel distance in *horizontal* direction is only D_{hor}. This discrepancy creates a linear error $\Delta D = 2(D_{meas} - D_{hor})$ (not shown in Fig. 3). Note that the factor '2' is used because the wheel travels up *and down* the bump.

For straight-line motion, the low-level controller of a conventional differential-drive mobile robot will try to keep the rotational velocities of both wheels equal. Thus, the *horizontal* distance traveled by the wheel that traversed the bump (let us assume it is the right wheel, in our example) will be ΔD less than that of the left wheel, causing a curved motion to the right, as shown in Fig. 4.

Applying simple geometric relations (based on Figs. 3 and 4, but not derived here in detail), the numeric sample results shown in Table I are obtained (all physical dimensions correspond to the *LABMATE* robot from TRC). We will use these sample numbers in the following discussion.

Table 23: Sample path errors after traversing a bump

Physical Dimensions	
Wheelbase *b*	340 mm
Wheel radius *R*	75 mm
Height of bump *h*	10 mm
Computed Results	
Linear error ΔD	2.63 mm
Orientation error $\Delta\theta_a$ (see Fig. 3)	0.44°
Lateral error after 10m travel	
e_{lat}(D=10m)	77 mm

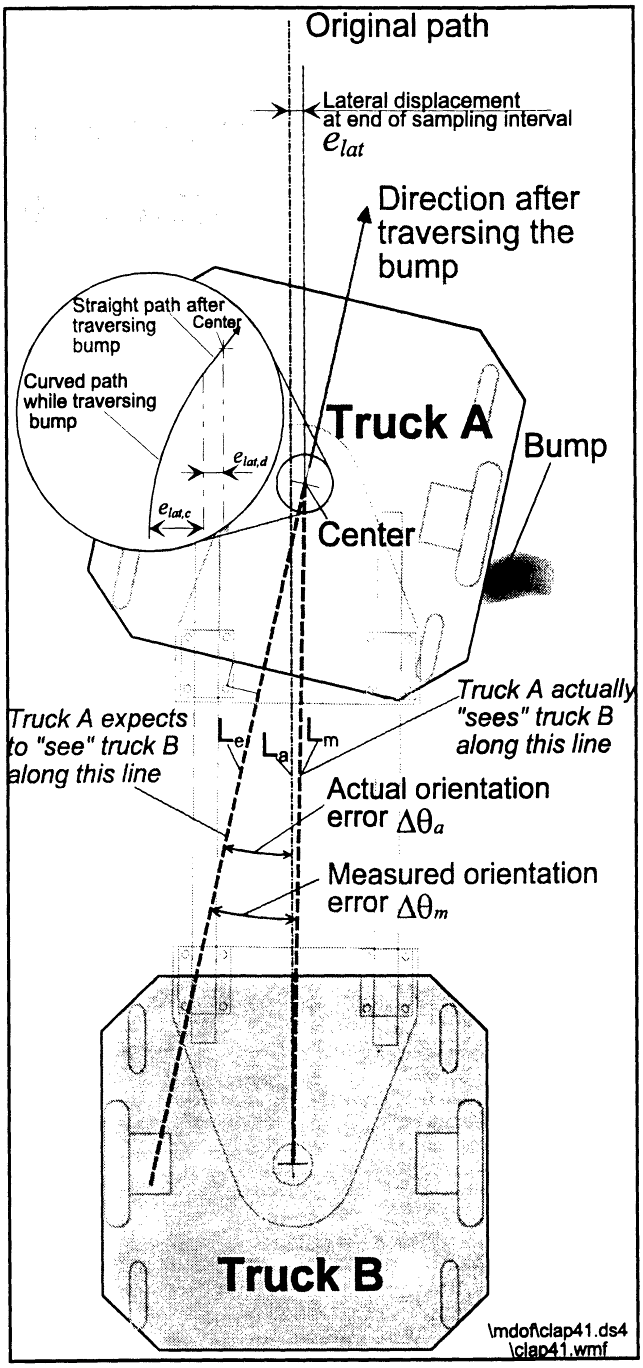
Original path
Lateral displacement at end of sampling interval
e_{lat}
Direction after traversing the bump
Straight path after traversing bump
Center
Curved path while traversing bump
$e_{lat,d}$
$e_{lat,c}$
Truck A
Bump
Center
Truck A expects to "see" truck B along this line
L_e
L_a
L_m
Truck A actually "sees" truck B along this line
Actual orientation error $\Delta\theta_a$
Measured orientation error $\Delta\theta_m$
Truck B
\mdof\clap41.ds4
\clap41.wmf

Figure 4: After traversing a bump, the resulting change of orientation of truck A can be measured relative to truck B.

The resulting orientation error $\Delta\theta_a$ (see Fig. 4) is the *most significant error* in the system [Feng et al, 1993], because it will cause an ***unbounded*** lateral error, e_{lat}, which grows proportionally with distance at a rate of

$$e_{lat}(D) = D\cdot\Delta D/b = D\sin\Delta\theta_a \qquad (1)$$

where

D - Distance traveled since clearing the bump
b - wheelbase

For example, Table I shows that the lateral error of truck A after only 10 m travel would be $e_{lat}(D{=}10\text{ m}) = 77$ mm.

The method for detecting dead-reckoning orientation errors is based on our new concept of *fast-growing* and *slow-growing* dead-reckoning errors. The CLAPPER performs relative position measurements very frequently, allowing each truck to detect *fast-growing* errors in its orientation (which can have changed significantly during one sampling interval), while

The practical implementation of this approach works as follows: Figure 4 shows the direction in which truck A "expected" truck B, based on the dead-reckoning data from both trucks. If, however, truck A had traversed a bump, it would have acquired an orientation error $\Delta\theta_a$. Comparing the direction reading from absolute encoder A with the "expected" direction, the system can uncover this orientation error. Subsequently the internal world model of truck A can be corrected accordingly. One problem with this approach is the fact that even a perfectly accurate measuring system cannot reveal the actual orientation error $\Delta\theta_a$. Rather, because of the lateral offset e_{lat} the orientation error is measured (incorrectly) as $\Delta\theta_m$. This allows us to correct the momentary orientation θ (based on dead-reckoning) in the internal world model of truck A by adding the measured orientation error $\Delta\theta_m$. The corrected orientation of truck A is therefore $\theta_{corrected} = \theta + \Delta\theta_m$.

In order to illustrate the validity of our approach we must show that the difference between $\Delta\theta_a$ and $\Delta\theta_m$ is indeed negligibly small under all normal driving condition. To do so, let us consider the enlarged area of Fig. 4. It is easy to compute the lateral position error $e_{lat}(D)$ *after* traversing the bump, because it increases at a constant rate as shown in Eq. (1). However, *while* traversing the bump the lateral position error changes as a function of the orientation error, which, in turn, is a function of the shape of the bump. Nonetheless, it can be shown that the orientation error increases monotonously while traversing a bump [Borenstein, 1994b]. Thus, any time we sample the orientation error $\Delta\theta_s$, we can be sure it is the largest orientation error from the time the bump was first encountered (i.e., $\Delta\theta_s \equiv \Delta\theta_{s,max}$). This holds true even if the wheel had not yet cleared the bump at the end of the sampling interval.

With this explanation in mind, an upper bound for the lateral orientation error while traversing a bump can be defined as

$$e_{lat}(D_s) \leq D_s \sin(\Delta\theta_s) \qquad (2)$$

We recall that in our numeric example $D_s = 20$ mm and $\Delta\theta_s = 0.44°$. Substituting these values into Eq. (2) yields $e_{lat}(D_s) = 0.15$ mm after traversing the bump.

Next, we can show that this lateral error has no significant influence on the accuracy of the relative orientation measurement between the two trucks. For example, the CLAPPER maintains a distance of $L = 1$ m between the two trucks. One can easily compute from the geometry of Fig. 4 that the small lateral error $e_{lat}(D_s) = 0.15$ mm will reduce the *actual* orientation error $\Delta\theta_a = 0.76°$ by $\varepsilon = \sin^{-1}(e_{lat}(D_s)/L) = \sin^{-1}(0.15/1000) = 0.01°$ and result in a *measured* orientation error $\Delta\theta_m = \Delta\theta_a - \varepsilon = 0.76° - 0.01° = 0.75°$.

Thus, the lateral error e_{lat} does reduces the accuracy of the orientation error measurement, but only by $\varepsilon = 0.01°$, i.e., much less than the resolution of the *internal encoders*.

This numeric example illustrates how our approach exploits the concept of *slow-growing* and *fast-growing* dead-reckoning errors: Most floor irregularities will cause an appreciable, immediately measurable orientation error (*fast-growing* error), while the resulting lateral error e_{lat} remains negligibly small during the sampling interval (*slow-growing* error).

Figure 4 shows the simple case in which only truck A encountered a bump while truck B retained its heading. However, even in the worst case, (i.e., if truck B also encountered a bump during the same sampling interval), its lateral error $e_{lat}(D_s)$ would be similarly small. Neither this lateral error nor the orientation error of truck B would cause a significant error in the orientation measurement of truck A relative to B or vice versa. Yet, even in this extreme case, the inaccuracy of the orientation error measurement would only be $\varepsilon = 2\times0.01° = 0.02°$, or $0.02/0.76\times100 = 2.6\%$.

The method described above can detect and reduce only rotational errors, but not translational errors. However, rotational errors are much more severe than translational errors, because orientation errors cause ***unbounded*** growth of lateral position errors. In the numeric example above, the translational error resulting from traversing a bump of height 10 mm was $\Delta D = 2.63$ mm. By comparison, the lateral error due to the rotational error $\Delta\theta$ is $e_{lat} = 77$ mm after only 10 m of further travel.

Another important strength of the CLAPPER's error correction system is the fact that ***orientation errors do not accumulate***. This is so because the error correction (i..e., adding $\Delta\theta_m$ to the internal world model of truck A is done in every sampling interval, Our experiments show that over-correction or under correction in one sampling interval is simply "caught" in the next interval. Furthermore, it doesn't matter whether the discrepancy between expected and measured relative direction is the result of bumps, cracks, or systematic errors.

In principal the total orientation error of each truck is ***bounded*** by the resolution of the *internal encoders*. This is a major advantage compared to conventional dead-reckoning, where orientation errors do accumulate. In practice, however, our system cannot guarantee an error to be bounded by the encoder resolution. Our system is sensitive to systematic measurement errors from the encoders. For example, if

rotary encoder A is constantly biased by, say, 0.5°, then the error correction function will assume a steady state in which truck A has a constant orientation error of 0.5° relative to the compliant linkage. Such an error would cause slightly curved motion of the CLAPPER (even on a perfectly smooth surface). Fortunately this systematic error can be detected experimentally and corrected by calibration with excellent results.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the CLAPPER's error correction method we performed numerous sets of experiments.

In this paper we present the results of the basic *straight-line experiment*, where the CLAPPER was programmed to travel straight forward for 18 m, stop, and return straight-backward for 18 m, to the starting position.

In order to automate the evaluation of the experiments, all experiments started and ended near an L-shaped *reference corner*. Three ultrasonic sensors were mounted on the vehicle, two sensors were facing the long side of the L-shaped corner, the third sensor faced the short side. The ultrasonic sensor system allowed measurement of the absolute position of the vehicle to within ±2 millimeters in the *x* and *y* directions, and to about ±0.25° in orientation.

At the beginning of each run a sonar measurement was taken to determine the starting position of the vehicle. The vehicle then traveled through the programmed path and returned to the L-shaped corner, where the *perceived* position (i.e., the position the vehicle "thought" it had, based on dead-reckoning) was recorded. Then, a sonar measurement was taken to determine the *absolute* position. The difference between the absolute position and the perceived position was the *position error.*

We performed three runs for each one of the following four conditions: (a) without error correction, without disturbances; (b) without error correction, with disturbances; (c) with error correction, without disturbances; and (d) with error correction, with disturbances.

In the runs "without disturbances" one can assume disturbance-free motion because our lab has a fairly smooth concrete floor. In the runs "with disturbances" bumps were created by placing a 10 mm diameter cable placed under the wheels. We used bumps only on the return leg of the 2×18 m round-trip and only under the right-side wheels of the vehicle (to avoid mutual cancellation of errors). In the runs with error correction we used 20 bumps that were evenly spaced along the 18 m return-path. Some bumps affected both the front and rear truck, some affected only one of the two trucks. In the runs without error correction we used only 10 bumps, because our cluttered lab could otherwise not accommodate the large path deviations. Without error correction, each bump caused an orientation error of approximately 0.6°.

Figure 5 summarizes the results from the straight-line experiment. shown are the stopping positions and orientations of the vehicle after compleing the 36 m journey back and forth along the x-axis. Each one of the four conditions of this experiment was performed three times. Note that without disturbances, the ending positions with error correction are only slightly better than those without correction. We relate the almost uniform error of approx. -1.7° in the run without error correction to systematic errors. The run with error correction shows how the systematic error is overcome. The more important results are those from runs with disturbances. Here the non-error corrected runs average -7.7°, out of which -1.7° are the result of the systematic error. The remaining error average of -6° were caused by applying identical 0.6° disturbances along the return path. Also note that the lateral position errors (without correction) would have been larger if the disturbances had been applied in the beginning of the return path.

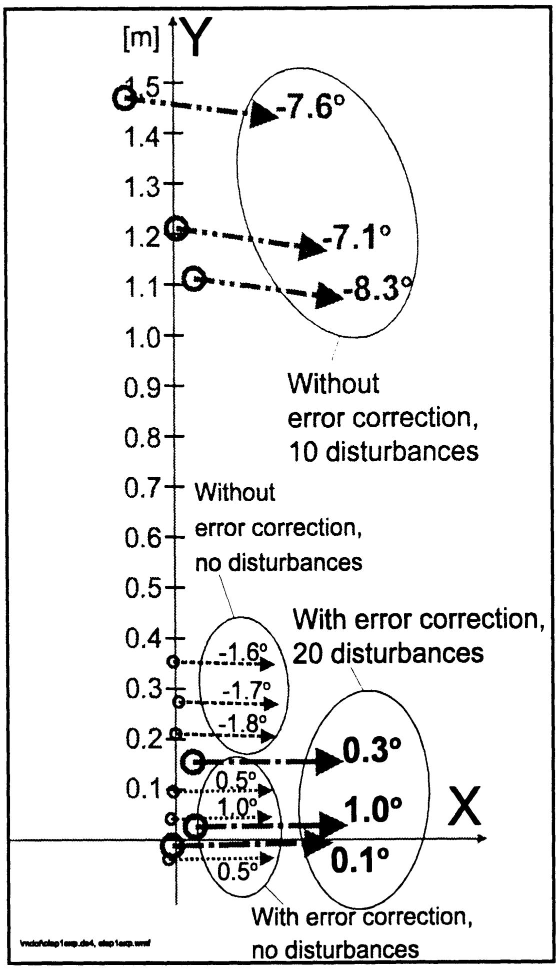

Figure 5: CLAPPER stopping positions after completing the straight-path experiment.

We performed many more experiments than the ones documented here. In all runs the orientation error with the CLAPPER was less than ±1.0° for the 36 m path. The experiment described in this section, as well as several other experiments, are documented in the video proceedings of this conference [Borenstein,1994V2].

5. ALTERNATIVE IMPLEMENTATIONS

We are currently investigating the possibilities of implementing our error correction method in a different kinematic configuration. This configuration comprises only one differential drive mobile robot, which ***tows*** a small trailer with encoder

wheels. Simulation results for this possible implementation indicate the feasibility of this approach [Borenstein, 1994c]. From a commercial point of view, the encoder trailer may be more attractive to manufacturers of mobile robots and AGVs, because the trailer can be attached to most existing vehicles.

6. CONCLUSIONS

The method described in this paper is applicable to many other autonomous vehicles. Vehicles used in construction or agricultural applications, where dead-reckoning has been impossible in the past because of the large amount of slippage on soft soil, may benefit directly from our new method. Furthermore, it may is possible to expand the *growth-rate concept* to tracked vehicles (like tanks or bulldozers) and possibly even to those watercraft and aircraft that have significantly different growth-rates in their positioning errors.

These features are made possible by exploiting the new concept of ***growth-rate*** of dead-reckoning errors that is introduced in this paper for the first time. The growth-rate concept distinguishes between certain dead-reckoning errors that develop slowly while other dead-reckoning errors develop quickly. Based on this concept, truck A frequently measures a property with *slow-growing* error characteristics on reference truck B (thus admitting a small error) to detect a *fast-growing* error on truck A (thus correcting a large error), and vice versa.

In summary, the advantage of the CLAPPER system are:

1. The ***immediate*** correction of orientation errors, which would otherwise cause unbounded growth of lateral position errors.
2. Prevention of accumulation of orientation errors, to the limit determined by the calibration of the internal position measurement accuracy, and provided that none of the wheels slipped sideways.

Acknowledgements:

This research was funded by NSF grant # DDM-9114394 and in part by the Department of Energy. Special thanks to Dr. Liqiang Feng who provided comments and suggestions for this manuscript.

7. REFERENCES

1. Barshan, B. and Durrant-Whyte, H.F., 1993, "An Inertial Navigation System for a Mobile Robot." *Proceedings of the 1st IAV*, Southampton, England, April, pp. 54-59.
2. Borenstein, J. and Koren, Y., 1985, "A Mobile Platform For Nursing Robots." *IEEE Transactions on Industrial Electronics*, Vol. 32, No. 2, pp. 158-165.
3. 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.
4. Borenstein, J., 1993, "Multi-layered Control of a Four-Degree-of-Freedom Mobile Robot With Compliant Linkage." *Proceedings of the 1993 IEEE International Conference on Robotics and Automation,* Atlanta, Georgia, May 2-7, pp. 3.7-3.12.
5. Borenstein, J., 1994a, "Control and Kinematic Design for Multi-degree-of-freedom Mobile Robots With Compliant Linkage." Accepted for publication. To appear in the *IEEE Transactions on Robotics and Automation*, February 1995.
6. Borenstein, J., 1994b, "Internal Correction of Dead-reckoning Errors With the Compliant Linkage Vehicle." Accepted for publication in the *Journal of Robotic Systems*, November 1994.
7. Borenstein, J, 1994c, "Internal Correction of Dead-reckoning Errors With the Smart Encoder Trailer." *Proceedings of the International Conference on Intelligent Robots and Systems (IROS '94)- Advanced Robotic Systems and the Real World.* September 12-16, 1994, München, Germany, pp. 127-134.
8. Borenstein, J., 1994V1, "Four-Degree-of-Freedom Redundant Drive Vehicle With Compliant Linkage." ***Video Proceedings*** *of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, 1994. (Note: Video Proceedings booklet lists this entry under incorrect title and abstract).
9. Borenstein, J., 1995, "The CLAPPER: A Dual-drive Mobile Robot With Internal Correction of Dead-reckoning Errors." *Submitted for inclusion in the* ***Video Proceedings*** *of the 1994 IEEE International Conference on Robotics and Automation.*
10. Congdon, I. et al., 1993, “CARMEL Versus FLAKEY — A Comparison of Two Winners.” *AI Magazine Winter 1992*, pp. 49-56.
11. Cox, I. J., 1991, “Blanche — An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle.” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, April, pp. 193-204.
12. Feng, L, Koren, Y., and Borenstein, J., 1993, "A Cross-Coupling Motion Controller for Mobile Robots." *IEEE Journal of Control Systems Magazine*. December pp. 35-43.
13. Killough, S. M. and Pin, F. G., 1992, "Design of an Omnidirectional Holonomic Wheeled Platform Prototype." *Proceedings of the IEEE Conference on Robotics and Automation*, Nice, France, May 1992, pp. 84-90.
14. Hongo, T., Arakawa, H., Sugimoto, G., Tange, K., and Yamamoto, Y., 1987, “An Automated Guidance System of a Self-Controlled Vehicle.” IEEE Transactions on Industrial Electronics, Vol. IE-34, No. 1, 1987, pp. 5-10.
15. Reister, D. B., 1991, "A New Wheel Control System for the Omnidirectional HERMIES-III Robot." *Proceedings of the IEEE Conference on Robotics and Automation* Sacramento, California, April 7-12, pp. 2322-2327.
16. TRC (Transition Research Corp), Shelter Rock Lane, Danbury, Connecticut, 06810.