Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

A Kinematic-independent Dead-reckoning Sensor for Indoor Mobile Robotics

Andrea Bonarini and Matteo Matteucci and Marcello Restelli
Department of Electronics and Information
Politecnico di Milano
Milan, Italy
Email: {bonarini,matteucci,restelli}@elet.polimi.it

***Abstract*— In this paper, we present a dead reckoning sensor to support reliable odometry on mobile robots. This sensor is based on a pair of optical mice rigidly connected to the robot body and its main advantages are 1) this localization system is independent from the kinematics of the robot, 2) the measurement given by the mice is not subject to slipping, since they are independent from the traction wheels, nor to crawling, since they measure displacements in any direction 3) it is a low-cost solution with a precision comparable to classical shaft encoders. We present the mathematical model of the sensor, its implementation, and some experimental evaluations using the standard UMBmark benchmark for odometry.**

I. Introduction

Since the very beginning of mobile robotics, dead reckoning was used to estimate the robot pose, i.e., its position and its orientation with respect to a global reference system placed in the environment. Dead reckoning is a navigation method based on measurements of distance traveled from a known point used to incrementally update the robot pose. This leads to a relative positioning method, which is simple, cheap and easy to accomplish in real-time. However this approach has the main disadvantage of an unbounded accumulation of errors.

The majority of the mobile robots use dead reckoning based on wheels velocity in order to perform their navigation tasks (alone or combined with other absolute localization systems [6]). Typically, odometry relies on measures of the space covered by the wheels gathered by encoders which can be placed directly on the wheels or on the engine-axis. These measures are then combined to compute robot movement along the x and y coordinates of a global frame of reference and its change of orientation. It is well-known that this approach to odometry is subject to:

- *systematic* errors, caused by factors such as unequal wheel-diameters, imprecisely measured wheel diameters and wheel distance [6];
- *non-systematic* errors, caused by irregularities of the floor, bumps, cracks or by wheel-slippage.

Despite its intrinsic limitations, many researchers agree that odometry is an important part of a robot navigation system and that navigation tasks would be simplified having an improved odometry providing good short-term accuracy at very high sample rates.

In this paper, we present a new dead reckoning sensor, based on the measures taken by two optical mice fixed on the bottom of the robot, which is very robust towards non-systematic errors, since it is not coupled with the driving wheels while measuring the effective robot displacement. In the following section, we briefly introduce the motivations for using this new method for dead reckoning. Section III, describes the geometrical derivation that allows to compute the robot movement on the basis of the readings of the mice while Section IV describes a simple procedure to calibrate the odometry system. The main characteristics of the mice and how they affect the accuracy and the applicability of the system are described in Section V, where we also report results related to some experiments showing the effectiveness of our approach. Section VI closes the paper by introducing a brief comparison of the system with related works.

II. Motivations for a new odometry sensor

Classical dead reckoning methods, which use the data measured by encoders on the wheels or on the engine-axis, suffer from two main non-systematic source of errors: *slipping*, which occurs when the encoders measure a movement which is larger than the actually performed one (e.g., when the wheels lose the grip with the ground), and *crawling*, which is related to a robot movement that is not measured by the encoders (e.g., when the robot is pushed by an external force, and the encoders cannot measure the displacement because wheels are blocked and do not turn according to the wielded external force).

Especially with recent developments in non-holonomic kinematics featuring omnidirectional wheels for traction, these methods measuring wheels displacement do not guarantee reliable performance. In this scenario, our sensor provide a device that is independent from the kinematics of the robot, and so we can use the same approach on several different robots. In fact our dead reckoning method, based on mice readings, does not suffer from slipping problems, since the sensors are not bound to any driving wheel. Also the crawling problems are solved, since the mice go on reading even when the robot movement is due to a push, and not to the engines.

Furthermore, this is a very low-cost system which can be easily interfaced with any platform, thus can be used to integrate standard odometry methods. In fact, it requires

0-7803-8463-6/04/$20.00 ©2004 IEEE

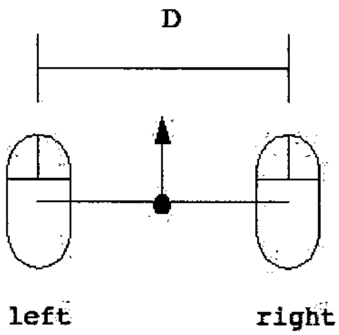


Fig. 1. The relative positioning of the two mice

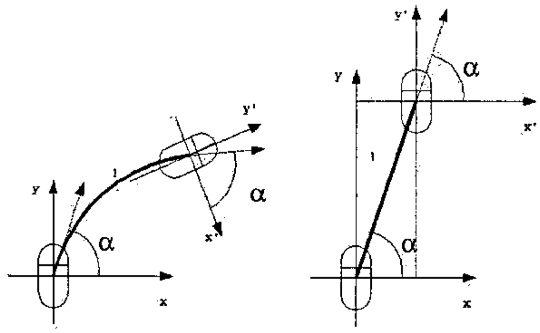


Fig. 2. Two different paths for which the mouse readings are the same

only two optical mice which can be placed in any position under the robot, and can be connected using the USB interface. This allows to build an accurate dead reckoning system, which can be employed on and ported to all the mobile robots which operate in an environment with a ground that allows the mice to measure the movements (indoor environments typically meet this requirement). In fact, the only issue by using this method is related to missed readings due to a floor with a bad surface or when the distance between the mouse and the ground becomes too large because of bumps and holes.

III. How to Compute the Robot Pose

In this section we present the geometrical derivation that allows to compute the pose of a robot using the readings of two mice placed below it in a fixed position. For sake of ease, we place the mice at a certain distance D, so that they are parallel between them and orthogonal w.r.t. their joining line (see Figure 1). We consider their mid-point as the position of the robot and their direction (i.e., their longitudinal axis pointing toward their keys) as its orientation.

Each mouse measures its movement along its horizontal and vertical axes. If the robot makes an arc of circumference, it can be shown that also each mouse will make an arc of circumference, which is characterized by the same center and the same arc angle (but with a different radius). During the sampling time, the angle α between the x-axis of the mouse and the tangent to its trajectory does not change. This implies that, when a mouse moves along an arc of length l, it measures always the same values independently from the radius of the arc (see Figure 2). So, considering an arc with an infinite radius (i.e., a segment), we can write the following relations:

$$\overline{x} = l\cos(\alpha) \quad (1)$$

$$\overline{y} = l\sin(\alpha). \quad (2)$$

From Equations 1 and 2, we can compute both the angle between the x axis of the mouse and the tangent to the arc:

$$\alpha = \arctan\left(\frac{\overline{y}}{\overline{x}}\right), \quad (3)$$

and the length of the covered arc:

$$l = \begin{cases} |\overline{x}|, & \alpha = 0, \pi \\ \frac{\overline{y}}{\sin\alpha}, & otherwise \end{cases} \quad (4)$$

We hypothesize that, during the short sampling period, the robot moves with constant translational and rotational speeds. This implies that the robot movement during a sampling period can be approximated by an arc of circumference. So, we have to estimate the 3 parameters that describe the arc of circumference (i.e., the x and y coordinates of the center of instantaneous rotation and the rotation angle $\Delta\theta$), given the 4 readings taken from the two mice. We call $\overline{x}_r$ and $\overline{y}_r$ the measures taken by the mouse on the right, while $\overline{x}_l$ and $\overline{y}_l$ are those taken by the mouse on the left. Notice that we have only 3 independent data; in fact, we have the constraint that the respective position of the two mice cannot change. This means that the mice should read always the same displacement along the line that joins the centers of the two sensors. In particular, if we place the mice as in Figure 3, we have that the x values measured by the two mice should be always equal: $\overline{x}_l = \overline{x}_r$. In this way, we can compute how much the robot pose has changed in terms of Δx, Δy, and $\Delta\theta$.

In order to compute the orientation variation we apply

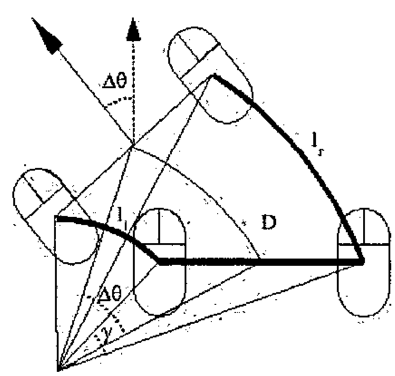


Fig. 3. The angle arc of each mouse is equal to the change in the orientation of the robot

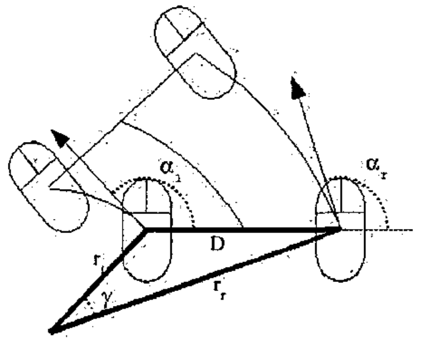


Fig. 4. The triangle made up of the joining lines and the two radii

the theorem of Carnot to the triangle made by the joining line between the two mice and the two radii between the mice and the center of their arcs (see Figure 4):

$$D^2 = r_r^2 + r_l^2 - 2\cos(\gamma) r_r r_l, \quad (5)$$

where r_r and r_l are the radii related to the arc of circumferences described respectively by the mouse on the right and the mouse on the left, while γ is the angle between r_r and r_l. It is easy to show that γ can be computed by the absolute value of the difference between α_l and α_r (which can be obtained by the mouse measures using Equation 3): $\gamma = |\alpha_l - \alpha_r|$.

The radius r of an arc of circumference can be computed by the ratio between the arc length l and the arc angle θ. In our case, the two mice are associated to arcs under the same angle, which corresponds to the change in the orientation made by the robot, i.e. $\Delta\theta$ (see Figure 3). It follows that:

$$\begin{aligned} r_l &= \frac{l_l}{|\Delta\theta|} && (6) \\ r_r &= \frac{l_r}{|\Delta\theta|}. && (7) \end{aligned}$$

If we substitute Equations 6 and 7 into Equation 5, we can obtain the following expression for the orientation variation:

$$\Delta\theta = \frac{\sqrt{l_l^2 + l_r^2 - 2\cos(\gamma) l_l l_r}}{D} \cdot sign(\overline{y}_r - \overline{y}_l) \quad (8)$$

The movement along the x and y axes can be derived by considering the new positions reached by the mice (w.r.t. the reference system centered in the old robot position) and then computing the coordinates of their mid-point (see Figure 5). The mouse on the left starts from the point of coordinates ($-\frac{D}{2}$;0), while the mouse on the right starts from ($\frac{D}{2}$;0). The formulas for computing their coordinates at the end of the sampling period are the following:

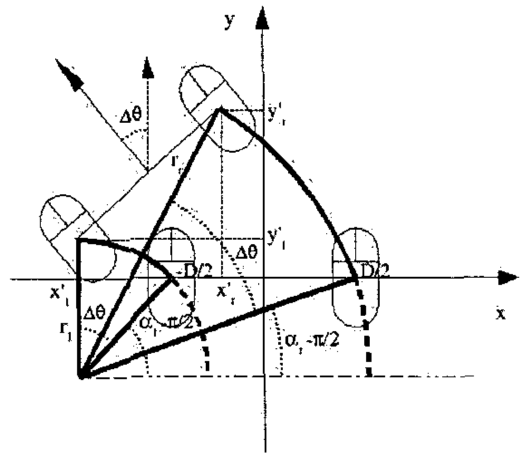


Fig. 5. The movement made by each mouse

$$\begin{aligned} x'_r &= r_r\left(\sin(\alpha_r + \Delta\theta) - \sin(\alpha_r)\right) sign(\Delta\theta) + \frac{D}{2} \\ y'_r &= r_r\left(\cos(\alpha_r) - \cos(\alpha_r + \Delta\theta)\right) sign(\Delta\theta) \\ x'_l &= r_l\left(\sin(\alpha_l + \Delta\theta) - \sin(\alpha_l)\right) sign(\Delta\theta) - \frac{D}{2} \\ y'_l &= r_l\left(\cos(\alpha_l) - \cos(\alpha_l + \Delta\theta)\right) sign(\Delta\theta). \end{aligned}$$

From the mice positions, we can compute the movement executed by the robot during the sampling time with respect to the reference system centered in the old pose using the following formulas:

$$\begin{aligned} \Delta x &= \frac{x'_r + x'_l}{2} && (9) \\ \Delta y &= \frac{y'_r + y'_l}{2}. && (10) \end{aligned}$$

The absolute coordinates of the robot pose at time $t+1$ $(X_{t+1}, Y_{t+1}, \Theta_{t+1})$ can be compute by knowing the absolute coordinates at time t and the relative movement carried out during the period $(t; t+1]$ $(\Delta x, \Delta y, \Delta\theta)$ through these equations:

$$\begin{aligned} X_{t+1} &= X_t + \sqrt{\Delta x^2 + \Delta y^2}\cos\left(\Theta_t + \arctan\left(\frac{\Delta y}{\Delta x}\right)\right) \\ Y_{t+1} &= Y_t + \sqrt{\Delta x^2 + \Delta y^2}\sin\left(\Theta_t + \arctan\left(\frac{\Delta y}{\Delta x}\right)\right) \\ \Theta_{t+1} &= \Theta_t + \Delta\theta. \end{aligned}$$

IV. SENSOR CALIBRATION PROCESS

In the introduction, we have mentioned the fact that odometry is affected by two kind of errors: *systematic* and *non-systematic*. In this section, we analyze the systematic errors that can affect our system and propose a calibration procedure in order to correct them.

In our approach systematic errors can be generated by:

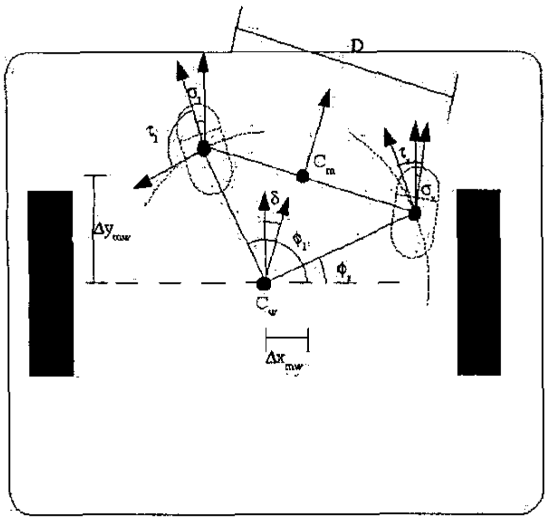


Fig. 6. Two mice fixed under a mobile robot. This figure points out all the parameters that we must estimate in order to use odometry with two mice.

- imperfections in the measurements of the positions and orientations of the two mice with respect to the robot;
- the resolution of the mouse, which depends from the surface on which the robot is moving;
- different resolutions of the two mice.

Our odometric system needs to know the value of some parameters related to the positioning of the two mice (see Figure 6): the distance between the two mice D, the orientation of the two mice σ_l and σ_r w.r.t. the robot heading, and the angle δ between the robot heading and the direction orthogonal to the mice joining line. If these parameters are not correctly estimated, systematic errors will be introduced. Moreover, we can estimate the displacement between the midpoint of the two mice and the midpoint of the two wheels, in terms of Δx_{MW} and Δy_{MW}.

In order to estimate these parameters we define a calibration procedure which consists in two practical measurements: the *translational* measurement and the *rotational* measurement. The translational measurement consists in making the robot travel (manually) 500 mm forward for ten times, and, for each time, storing the mice readings. At the end of the measurement we can compute the averages of the four readings: $\hat{x}_r, \hat{y}_r, \hat{x}_l, \hat{y}_l$, which allow us to estimate the mice resolutions $\hat{k}_l, \hat{k}_r$ and the angle between the robot and the mouse heading $\hat{\sigma}_l$ and $\hat{\sigma}_r$:

$$\hat{k}_l = \frac{500}{\sqrt{\hat{x}_l^2 + \hat{y}_l^2}} \tag{11}$$

$$\hat{k}_r = \frac{500}{\sqrt{\hat{x}_r^2 + \hat{y}_r^2}} \tag{12}$$

$$\hat{\sigma}_l = \arctan\left(\frac{\hat{x}_l}{\hat{y}_l}\right) \tag{13}$$

$$\hat{\sigma}_r = \arctan\left(\frac{\hat{x}_r}{\hat{y}_r}\right). \tag{14}$$

$$\tag{15}$$

In the rotational measurement we take the mice readings after a counter-clockwise 360^o revolution that the robot makes around its rotational axis; this process is repeated five times. The averages of these readings allow us to estimate the distances between the center of rotation C_w and the two mice ($\hat{r}_l$ and $\hat{r}_r$), the angle between the mouse heading and the tangential to the circumference described during the revolution ($\hat{\tau}_l$ and $\hat{\tau}_r$), and the distance between the two mice projected to the x and y-axes ($\hat{\Delta x}_m$ and $\hat{\Delta y}_m$).

$$\hat{r}_l = \frac{\hat{k}_l\sqrt{\hat{x}_l^2 + \hat{y}_l^2}}{2\pi} \tag{16}$$

$$\hat{r}_r = \frac{\hat{k}_r\sqrt{\hat{x}_r^2 + \hat{y}_r^2}}{2\pi} \tag{17}$$

$$\hat{\tau}_l = \arctan\left(\frac{\hat{y}_l}{\hat{x}_l}\right) \tag{18}$$

$$\hat{\tau}_r = \arctan\left(\frac{\hat{y}_r}{\hat{x}_r}\right) \tag{19}$$

$$\hat{\Delta x}_m = \hat{r}_r\cos(\hat{\phi}_r) - \hat{r}_l\cos(\hat{\phi}_l) \tag{20}$$

$$\hat{\Delta y}_m = \hat{r}_r\sin(\hat{\phi}_r) - \hat{r}_l\sin(\hat{\phi}_l), \tag{21}$$

where $\hat{\phi}_l = \hat{\sigma}_l + \hat{\tau}_l$ and $\hat{\phi}_r = \hat{\sigma}_r + \hat{\tau}_r$.

From these data we can compute the other required parameters, i.e., the distance between the mice $\hat{D}$, the angle $\hat{\delta}$, and the displacement $\hat{\Delta x}_{mw}$ and $\hat{\Delta y}_{mw}$:

$$\hat{D} = \sqrt{\hat{\Delta x}_m^2 + \hat{\Delta y}_m^2} \tag{22}$$

$$\hat{\delta} = \arctan\left(\frac{\hat{\Delta y}_m^2}{\hat{\Delta x}_m^2}\right) \tag{23}$$

$$\hat{\Delta x}_{mw} = \frac{\hat{r}_r\cos(\hat{\phi}_r) + \hat{r}_l\cos(\hat{\phi}_l)}{2} \tag{24}$$

$$\hat{\Delta y}_{mw} = \frac{\hat{r}_r\sin(\hat{\phi}_r) + \hat{r}_l\sin(\hat{\phi}_l)}{2}. \tag{25}$$

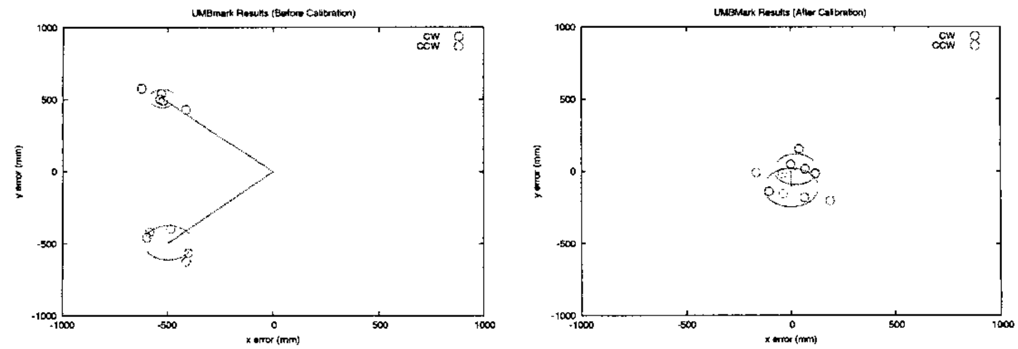
We can generalize the equations derived in the previous section, for odometry with two mice in arbitrary fixed positions. In order to achieve this, we need only to modify, from Equation 5 till the end of the corresponding section, the value of α_l and α_r by adding respectively the angle $\hat{\sigma}_l - \hat{\delta}$ and $\hat{\sigma}_r - \hat{\delta}$.

In this way, we compute the movement of the middle-point of the two mice $C_m =< X_m, Y_m, \Theta_m >$. If we want to compute the odometry w.r.t. the rotational center $C_w =< X_w, Y_w, \Theta_w >$, we have to apply the following formulas:

$$\Theta_w = \Theta_m - \hat{\delta} \tag{26}$$

$$X_w = X_m - D_{mw}\cos(\Theta_w + \theta_{mw}) \tag{27}$$

$$Y_w = Y_m - D_{mw}\sin(\Theta_w + \theta_{mw}), \tag{28}$$



(a) (b)

Fig. 7. The errors measured in the UMBmark before (a) and after (b) the sensor calibration process described in Section IV

where $D_{mw} = \sqrt{\hat{\Delta x}_{mw}^2 + \hat{\Delta y}_{mw}^2}$ and $\theta_{mw} = \arctan\left(\frac{\hat{\Delta y}_{mw}}{\hat{\Delta x}_{mw}}\right)$.

V. Experimental Results

The performance of the odometry system we have described depends on the characteristics of the solid-state optical mouse sensor used to detect the displacement of the mice. Commercial optical mice use the Agilent ADNS-2051 sensor [1] a low cost integrated device which measures changes in position by optically acquiring sequential surface images (frames hereafter) and determining the direction and magnitude of movement.

The ADNS-2051 sensor is essentially a tiny, high-speed video camera coupled with an image processor, and a quadrature output converter [2]. A light-emitting diode (LED) illuminates the surface underneath the sensorreflecting off microscopic textural features in the area. A plastic lens collects the reflected light and forms an image on a sensor. If you were to look at the image, it would be a black-and-white picture of a tiny section of the surface represented as a matrix of 256 pixels. The sensor continuously takes pictures as the mouse moves at 1500 frames per second or more, fast enough so that sequential pictures overlap. Through an Agilent proprietary image-processing algorithm, the sensor identifies common features between these frames and determines the distance between them; this information is then translated into Δx and Δy values to indicate the sensor displacement.

By looking at the sensor characteristics, available through the data sheet, it is possible to estimate the precision of the measurement and the maximum working speed of the device. In fact the Agilent ADNS-2051 sensor is programmable to give mouse builders (this is the primary use of the device) 400 or 800 cpi resolution, a motion rate of 14 inches per second, and frame rates up to 2,300 frames per second. At recommended operating conditions this allows a maximum operating speed of 0.355 m/s with a maximum acceleration of 1.47 m/s^2.

These numbers reflect only nominal values for the sensor characteristics since its resolution can vary depending on the surface material and the height of the sensor from the floor. This variation in sensor readings calls for an accurate mounting on the robot and the calibration procedure described in Section IV to reduce systematic errors in odometry before to apply the formulas described in Section III.

In order to validate our approach, we take two USB optical mice featuring the Agilent ADNS-2051 sensor, which can be commonly purchased in any commercial store. We fix the mice to the robot body, trying to positioning them as described in Section III, and taking care of making them stay in contact with the ground. We made our experiments on a carpet, like those used in the RoboCup Middle Size League. In order to identify the parameters of the odometric sensor, we performed the calibration procedure described in the previous section; the values of the parameters estimated through this process are reported in Table I.

The preliminary test we made is the UMBmark test, which was presented by [5]. The UMBmark procedure consists of measuring the absolute actual position of the robot in order to initialize the on-board dead reckoning starting position. Then, we make the robot travel along a $4x4$ m square in the following way: the robot stops after each $4m$ straight leg and then it makes a 90^o turn on the spot. When the robot reaches the starting area, we measure its absolute position and orientation and compare them to the position and orientation calculated by the dead reckoning system. We repeated this procedure five times in clockwise direction and five times in counter-clockwise.

TABLE I

The parameters estimated with the calibration procedure

$\hat{k}_l$	$425cpi$	$\pm 0.97\%$
$\hat{k}_r$	$425cpi$	$\pm 0.64\%$
$\hat{\sigma}_l$	1.69^o	$\pm 0.12\%$
$\hat{\sigma}_r$	1.81^o	$\pm 0.09\%$
$\hat{D}$	$219mm$	$\pm 1.09\%$
$\hat{\delta}$	-2.07^o	$\pm 1.02\%$
$\hat{\Delta x}_{wm}$	$94mm$	$\pm 1.13\%$
$\hat{\Delta y}_{wm}$	$57mm$	$\pm 1.22\%$

In Figure IV we can see the results obtained with the robot before the execution of the calibration procedure and after the sensor parameters have been identified. In the uncalibrated case, we set the parameters to default values, i.e., we assume that the mice are parallel and orthogonal w.r.t. their joining line, we assume $425cpi$ for the resolution of both mice and we manually measure the distance between them and set it to $217\ mm$.

As we can see, if we do not calibrate our system we can suffer from large systematic errors, while after calibration we manage to drastically reduce their influence. Borenstein proposed to evaluate the influence of dead reckoning systems, by dividing the errors into two clusters, one for the clockwise runs and one for the counter-clockwise runs, and then computing $E_{max,syst}$, i.e., the maximum between the distances of the center of gravity of the two clusters from the origin. The measure of dead reckoning accuracy for systematic errors that we obtained is $E_{max,syst} = 729\ mm$ for the uncalibrated system and $E_{max,syst} = 114\ mm$ after the calibration procedure. This last result is comparable with those achieved by other dead reckoning systems tested with the UMBmark.

VI. Discussion & Conclusion

The fundamental idea of dead reckoning as the integration of incremental motion information over time, which leads inevitably to the unbounded accumulation of errors, has been throughly investigated in literature. Specifically, orientation errors will cause large lateral position errors, which increase proportionally with the distance traveled by the robot. To address this issue, there have been a lot of work in this field for systematic error measurement, comparison and correction [6] especially for differential drive kinematics.

First works in odometry error correction were done by using and external compliant linkage vehicle pulled by the mobile robot. Being pulled this vehicle does not suffer from slipping and the measurement of its displacing can be used to correct pulling robot odometry [4]. In [6] the authors propose a practical method for reducing, in a typical differential drive mobile robot, incremental odometry errors caused by kinematic imperfections of mobile encoders mounted onto the two drive motors.

Little work has been done for different kinematics like the ones based on omnidirectional wheels. In these cases, slipping is always present during the motion and classical shaft encoder measurement leads to very large errors. In [3] the Persia RoboCup team proposes a new odometric system which was employed on their full omni-directional robots. In order to reduce non-systematic errors, like those due to slippage during acceleration, they separate odometry sensors from the driving wheels. In particular, they have used three omni-directional wheels coupled with shaft encoders placed 60^o apart of the main driving wheels. The odometric wheels are connected to the robot body through a flexible structure in order to minimize the slippage and to obtain a firm contact of the wheels with the ground. Also this approach is independent from the kinematics of the robot, but its realization is quite difficult and, again, it is affected by (small) slippage problems.

An optical mouse was used in the localization system presented in [10]. In their approach, the robot is equipped with an analogue compass and an optical odometer made out from a commercially available mouse. The position is obtained by combining the linear distance covered by the robot, read from the odometer, with the respective instantaneous orientation, read from the compass. The main drawback of this system is due to the low accuracy of the compass which results in another source for systematic errors.

Being odometry inevitably affected by the unbounded accumulation of errors, there are several works that propose methods for fusing odometric data with absolute position measurements to obtain more reliable position estimation [7], [8]. However, despite its limitations, a reliable odometry system providing good short-term accuracy simplifies the navigation task and this is the reason we have presented this new dead reckoning sensor based on a pair of optical mice. As we said, the main advantages are good performances w.r.t. the two main problems that affect dead reckoning sensors: slipping and crawling. Due to its characteristics, the proposed sensor can be successfully applied with many different robot architectures, being completely independent from the specific kinematics. In particular, we have developed it for our omnidirectional Robocup robots, which will be presented at Robocup 2004 and we are going to integrate in on our self-localization framework MUREA [9].

References

[1] Agilent Technologies Semiconductor Products Group. Optical mice and hiw they work: The optical mouse is a complete imaging system in a tiny package. Technical Report 5988-4554EN, Agilent Technologies Inc., November 2001.

[2] Agilent Technologies Semiconductor Products Group. Agilent adns-2051 optical mouse sesor data sheet. Technical Report 5988-8577EN, Agilent Technologies Inc., May 2003.

[3] P. Amini, M. D. Panah, and H. Moballegh. A new odometry system to reduce asymmetric errors for omnidirectional mobile robots. In *RoboCup 2003: Robot Soccer World Cup VII*, 2003.

[4] J. Borenstein. Internal correction of dead-reckoning errors with the compliant linkage vehicle. *Journal of Robotic Systems*, 12(4):257–273, April 1995.

[5] J. Borenstein and L. Feng. UMBmark - a method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. In *Proceedings of SPIE Conference on Mobile Robots*, Philadelphia, October 1994.

[6] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(6):869–880, December 1996.

[7] F. Chenavier and J. Crowley. Position estimation for a mobile robot using vision and odometry. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2588–2593, Nice, France, May 12–14 1992.

[8] I. Cox. Blanche - an experiment in guidance and navigation of an autonomous mobile robot. *IEEE Transactions on Robotics and Automation*, 7(3):193–204, 1991.

[9] M. Restelli, D. G. Sorrenti, and F. M. Marchese. Murea: a multi-resolution evidence accumulation method for robot localization in known environments. In *Proceedings of IROS 2002*, pages 415–420, Piscayaway, NJ, 2002. IEEE Computer press.

[10] F. M. Santos, V. F. Silva, and L. M. Almeida. A robust self-localization system for a small mobile autonomous robot. In *Preceedings of ISRA*, 2002.