Design of a position and orientation measurement robot

A Lakhani, I Gorlach, F Smith
Department of Mechatronics
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
s207063172@live.nmmu.ac.za

Abstract—**Mobile robotics plays an ever increasing role not only in an industrial environment but in all aspects of modern life. Mobile robots can be found in offices, shops and hospitals. As an important engineering task when designing mobile robots is the navigation strategy, this research presents a dead-reckoning system for an accurate localisation of mobile robots. The dead-reckoning system is based on the calculations of the increments of the robot movements, which utilise the odometry parameters obtained from optical sensors. The proposed solution for controlling mobile robots has the advantage of low-cost as it utilises an optical mouse sensor.**

Keywords- dead reckoning; mobile robot; optical mouse sensor.

I. INTRODUCTION

The aim of this research project is to develop a mobile robot that moves in any required direction, while keeping record of its actual position and angle with respect to a reference starting point, without any lines or ground marks.

The positional data is computed using the principle of dead-reckoning. This includes capturing of odometry parameters from sensors attached to the robot, and calculating the new position based on the previous position [1] [2].

Previous attempts to build dead-reckoning robots have shown that they have problems with accurate positioning owing to low-resolution shaft encoders and, hence, cumulative calculation errors [2] [3]. Another problem encountered was using a pre-built chassis because it makes modifying the robot difficult [3]. However, the proposed design of this robot will help understand the concept of dead-reckoning and so improve future designs. This project outlines the design of a position measurement robot using a more reliable odometry sensor and its construction from modular parts.

In this research, different configurations were explored, including odometric sensors for data capture, microcontrollers to calculate absolute position and a mechanical chassis for motion. The proposed method calculates the absolute position of the robot using data acquired from the odometric sensor.

The integration of the three Mechatronics subsystems, namely, mechanical, electrical and IT, are utilised in this research project. Testing is conducted on individual components and on the entire dead-reckoning navigation system. Further recommendations are provided to increase the accuracy of the navigation system.

II. PROBLEM ANALYSIS AND SYSTEM REQUIREMENTS

The purpose of developing a mobile robot is to demonstrate robot positioning and mapping systems without any ground reference, except for the starting point. Robot positioning is an integral aspect of many robot designs as it is not always possible to have ground marks. However, this concept allows for the design of robots with artificial intelligence [4] [3]. Therefore, the robot is able to calculate a path and travel to its destination while taking into account its surroundings and any obstruction it may encounter.

There are many algorithms (such as genetic and evolutionary robotic algorithms) that can be used to generate paths for the movement of robots [5]. Unfortunately, there are systematic and non-systematic errors such as unequal wheel diameters, irregular floor surface and slippage which cause inaccurate robot movement [2]. Thus, odometry sensors are used to provide feedback to position systems to minimise these errors and produce a more accurate trajectory.

The proposed design will also assist with the understanding of issues related to the process of dead reckoning, including factors that reduce precision in position measurement. For further development, this feedback system can then be used to acquire high-accuracy positioning for mobile robots. The system requirements are:

- a robot that moves in any required direction, while keeping record of its actual position and angle
- the display of positional data ((x, y) coordinate and angle) on an LCD screen
- the interface of all peripherals by means of a microcontroller

III. THEORETICAL BACKGROUND

The robot works on the principle of dead reckoning. Dead reckoning is the process of calculating a position relying on a previously determined position [13]. To implement such algorithms on a robot, it is important to understand the theory

Presented at the 4th Robotics and Mechatronics Conference of South Africa (ROBMECH 2011)
23-25 November 2011, CSIR Pretoria South Africa.

on which it is based. The mathematical relations used for dead reckoning differ from one robot to another owing to the steering mechanism differences. For example, the car-like steering turns in a curvature of constant and known radius, while differential-drive robots can make very narrow turns and can even spin in their positions [6]. In this project, a differential-drive concept is used for the robot design.

A. *Dead reckoning*

Since the introduction of mobile robotics, dead reckoning has been used to estimate the robots' pose (position and orientation) with respect to a global reference system placed in the environment. Dead reckoning is a navigation method based on measurements of distance travelled from a known point and used incrementally to update the robot pose [8]. 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, since errors are carried forward.

The majority of mobile robots use dead reckoning based on wheels velocity to perform navigation tasks [14]. Typically, odometry relies on measurements of the space covered by the wheels, gathered by encoders placed directly on the wheels or on the engine-axis. These measurements are then combined to calculate robot movement along the x and y coordinates of a global frame of reference and its change in orientation (angle). 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 [2]
- non-systematic errors: caused by factors such as irregularities of the floor surface, bumps, cracks or wheel-slippage [2]

Although there are many limitations due to systematic and non-systematic errors, odometry is an important part of navigation systems. These limitations, however, may be overcome by making use of more accurate sensors and the precise calibration of parameters in the robot kinetic model. When errors occur, the estimated position can become very inaccurate. This is illustrated in Figure 1 which shows an estimate of the accumulation of dead-reckoning errors. The ellipses represent a certain percentage probability of the robot's actual position.

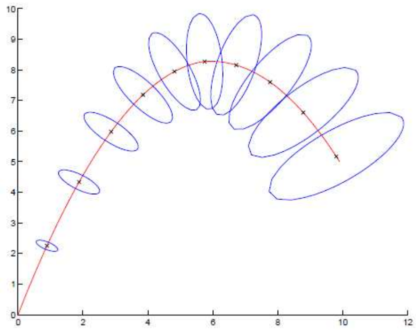

Figure 1. Accumulation of dead-reckoning errors [15]

The advantage of using optical mice is that they are very robust towards non-systematic errors, since they are not coupled with the driving wheels while measuring the effective robot displacement [2].

B. *Derivation of the equations for dead reckoning*

In this section, the geometrical derivation of the position and orientation of the robot is presented.

The two optical mice are fixed at the bottom of the robot. Each mouse is at a distance of D/2 from the centre of the robot so that they are parallel between them and orthogonal with respect to their joining line.

Each mouse measures its movement along its horizontal and vertical axes. If the robot makes an arc of circumference, each mouse will also make an arc of circumference, which is characterised by the same centre 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 means that when a mouse moves along an arc of length 1, it always measures the same values independently from the radius of the arc (see Figure 2) [2]. Figure 2 shows two different paths for which the mouse readings are the same.

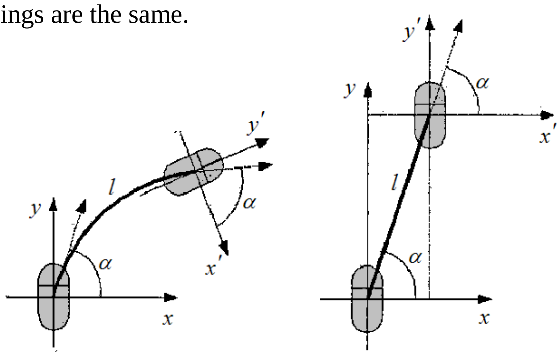

Figure 2. Optical mouse sensor position relative to robot [2]

So, considering an arc with an infinite radius (a segment), the following relations occur:

$$\bar{x} = l\cos(\alpha) \tag{1}$$

$$\bar{y} = l\sin(\alpha) \tag{2}$$

$$\therefore \alpha = \tan^{-1}\left(\frac{\bar{y}}{\bar{x}}\right) \tag{3}$$

$$l = \begin{cases} |\bar{x}|, & \alpha = 0, \pi \\ \dfrac{\bar{y}}{\sin\alpha}, & \text{otherwise} \end{cases} \tag{4}$$

It is assumed 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. The three parameters that describe the arc of circumference have to be estimated (namely, the x and y coordinates of the centre of instantaneous rotation and the rotation angle Δθ) from the four readings taken from the two mice. $\bar{x}_r$ and $\bar{y}_r$ are the measures taken by the mouse on the right, while $\bar{x}_l$ and $\bar{y}_l$ those taken by the mouse on the left. It is important to note that there are only three independent data, as 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 centres of the two sensors. In particular, if the mice are placed as in Figure 3, the *x* values measured by the two mice should be always equal $\bar{x}_r = \bar{x}_l$ .In this way, we can compute how much the robot pose has changed in terms of Δx, Δy and Δθ. Figure 3 shows the arc angle of each mouse as the robot moves in a curve.

To compute the orientation variation, the theorem of Carnot is applied to the triangle made by the joining line between the two mice and the two radii between the mice and the centre of their arcs (see Figure 3):

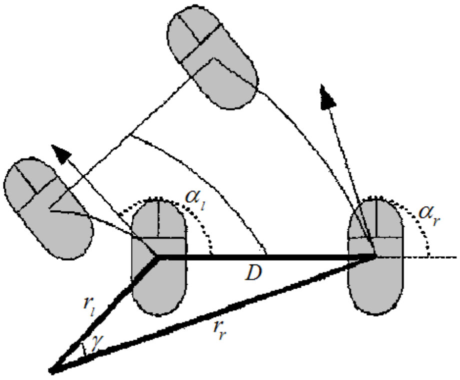

Figure 3. Carnot's triangle of optical mice radii [2]

$$D^2 = r_r^2 + r_l^2 - 2\cos(\gamma) r_r r_l \tag{5}$$

Where r_r and r_l are the radii related to the arc of circumference described by the mouse on the right and the mouse on the left respectively, while γ is the angle between r_r and r_l . γ can be computed by the absolute value of the difference between α_r and α_l (which can be obtained by the mouse measurements using Equation 1):

$$\gamma = \left|\alpha_r - \alpha_l\right| \tag{6}$$

The radius *r* of an arc of circumference can be computed by the ratio between the arc length 1 and the arc angle θ. In this case, the two mice are associated with arcs under the same angle, which corresponds to the change in the orientation Δθ made by the robot (see Figure 2).This implies that:

$$r_l = \frac{l_l}{|\Delta\theta|} \tag{7}$$

$$r_r = \frac{l_r}{|\Delta\theta|} \tag{8}$$

If Equations 7 and 8 are substituted into Equation 5, the following expression for the orientation variation can be obtained:

$$\Delta\theta = \frac{\sqrt{l_r^2 + l_l^2 - 2\cos(\gamma) l_r l_l}}{D}.sign \tag{9}$$

$$sign = \begin{cases} +1, & \bar{y}_r \geq \bar{y}_l \\ -1, & \bar{y}_r < \bar{y}_l \end{cases}$$

The movement along the x and y axes can be derived by considering the new positions reached by the mice (with respect to the reference system centred in the old robot position), and then computing the coordinates of their mid-point (see Figure 4). The mouse on the left, starts from the coordinates (-D/2, 0), while the mouse on the right, starts from (D/2, 0).

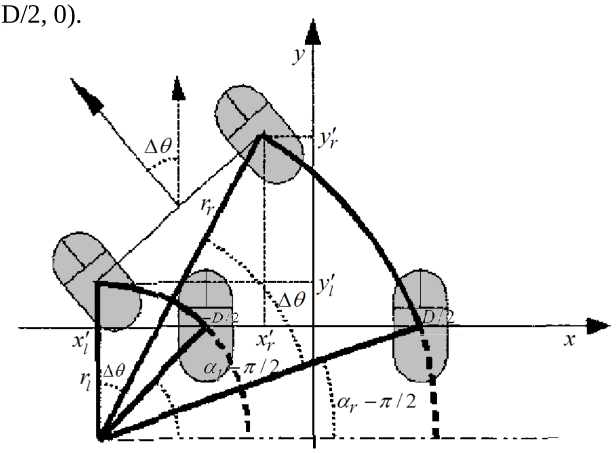

Figure 4. Angles onoptical mice path [2]

The formulas for computing their coordinates at the end of the sampling period are the following:

$$x'_r = r_r\left(\sin(\alpha_r + \Delta\theta) - \sin(\alpha_r)\right).sign + \frac{D}{2} \tag{10}$$

$$y'_r = r_r\left(\cos(\alpha_r) - \cos(\alpha_r + \Delta\theta)\right).sign \tag{11}$$

$$x'_l = r_l\left(\sin(\alpha_l + \Delta\theta) - \sin(\alpha_l)\right).sign - \frac{D}{2} \tag{12}$$

$$y'_l = r_l\left(\cos(\alpha_l) - \cos(\alpha_l + \Delta\theta)\right).sign \tag{13}$$

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

$$\Delta x = \frac{x'_r + x'_l}{2} \tag{14}$$

$$\Delta y = \frac{y'_r + y'_l}{2} \tag{15}$$

The absolute coordinates of the robot pose at time $t+1, (x_{t+1}, y_{t+1}, \theta_{t+1})$ can be computed by knowing the absolute coordinates at time t and the relative movement carried out during the period $t \rightarrow t+1 (\Delta x, \Delta y, \Delta \theta)$ through these Equations:

$$x_{t+1} = x_t + \sqrt{\Delta x^2 + \Delta y^2} \cos\left(\theta_t + \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right)\right) \quad (16)$$

$$y_{t+1} = y_t + \sqrt{\Delta x^2 + \Delta y^2} \sin\left(\theta_t + \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right)\right) \quad (17)$$

$$\theta_{t+1} = \theta_t + \Delta \theta \quad (18)$$

IV. DESIGN DESCRIPTION

The proposed design makes use of a differential drive chassis as it provides better control owing to the unique drive system and, therefore, more accurate positioning. The odometry sensor used is an optical mouse sensor as it has an image-capture sensor that translates sequential surface images into change in position, Δx and Δy. These changes in position are used with dead-reckoning formulae to calculate the positional data (x, y and angle). A minimum of two optical mice need to be mounted to measure translation and orientation of the robot [8]. The subsections below explain various components of the robot:

A. Component Selection

The components for this mobile robot had to be carefully selected due to the high accuracy required in measuring the positional data. Another concern was the selection of compatible components to ensure that interfacing the different systems was not a complex task.

The optical mouse sensor is the most important component as it is the odometry sensor used to obtain positional data for the robot's trajectory.

The reason an optical mouse sensor was used is that 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 sources of errors. Firstly, slipping, which occurs when the encoders measure a movement which is larger than actually performed, For example, when the wheels lose the grip with the ground, Secondly, crawling, which is related to a robot movement not measured by the encoders. For example, when the robot is pushed by an external force and the encoders cannot measure the displacement because the wheels are blocked and do not turn according to the wielded external force.

In the case of an optical mouse, the sensor provides a device that is independent from the kinematics of the robot. The same approach may be used on several different robots. The 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 motors.

A GPS-sensor, on the other hand, is able to give absolute coordinates of the robot, but is very expensive. Therefore, an optical mouse is used for the odometry measurements.

B. Mechanical

The mechanical component of the project focuses on the chassis and the mounts for the sensors. Although a pre-built chassis was acquired, it is relatively easy to build a chassis to match the requirements of the system. The motor and gearbox used were obtained from toy cars. This required designing a base to which the gearboxes and motors could be attached, and consequently, the electronics and sensors. The base was designed such that it had a large area allowing sufficient space on the top for all of the electronics to be mounted. This not only makes it visually appealing, but also easy to detect a fault if it occurs. The base was made of 2mm thick plastic sheet which is strong and light.

The other mechanical design was the mount for the optical mouse. There would have to be two brackets for the optical mice set at an equal distance from the midpoint of the chassis. Designing the brackets had to be carefully performed so that the mice would not be in the way of any moving parts. Brackets also had to be designed such that it would always push the mice firmly on the ground. This would ensure the readings from the mice were not influenced by sudden bumps or irregularities on the movement surface. This was achieved by placing springs over a rod so as to avoid their buckling. The brackets were manufactured from aluminium because it is light, strong and cheaply manufactured.

C. Electrical

The electrical component of the project focuses on the electrical power management of the entire system. Power management is of utmost importance when it comes to mobile robots as most mobile robots require batteries to power the electronics and motors. Therefore, the electrical system must be designed to consume the lowest possible power, but with best performance.

For the robot electrical design, two separate power supplies were used. A 9V square battery was used to power the PICDEM EXPLORER 2 demonstration board. This was because the demonstration board has a built-in voltage regulator circuit that steps down the 9V to 3.3V for the PIC 18F87J11 microcontroller. The demonstration board has another voltage regulator circuit to step down the 9V to 5V for the LCD screen. The demonstration board also has two 5V dc output terminals which were used as a power source for the optical mice. The optical mice were connected to the demonstration board terminals to simplify the wiring and to allow the optical mouse sensor and PIC microcontroller to communicate at the same logic levels. The 9V square battery selected had a 210mAh rating which is much higher than commercial square batteries, and enabled it to power all the on-board components for a longer period of time.

Rather than using the 9V square battery, a different power source was used for the motors because they required more power, therefore, four 1.5V batteries were used to power the motors as 1.5V batteries are rated at 2450mAh. An advantage of having a different power source is that the motors are isolated from the microcontroller at all times via the ULN2004. In the case of an electrical fault, the entire system is not affected.

D. IT and Software programming

The focus in this component was programming the PIC 18F87J11 microcontroller to perform the following tasks, including:

- reading odometry data from the optical mice
- calculating positional data (position and orientation)
- displaying positional data on LCD screen
- controlling motors using pulse width modulation (PWM) to follow any path

The software used to code the microcontroller was a free program provided by Microchip Inc., known as MPLAB IDE. The well-known C-language environment was used because it is high-level language and, hence has more capability and in-built functions. However, some assembler code was used for delays because it is more precise.

V. TESTING AND PERFOMANCE

A. Distance Test

Although the distance test is performed to determine the accuracy of the optical mouse sensor, no special program is written to the microcontroller for this test. When the robot is fully functional, it displays coordinates on the LCD in inches (distance is displayed in inches because optical mouse sensor is calibrated for a fixed number of counts per inch) for easy interpretation. For this test, a distance of 2” is marked on the floor, and the mouse is manually pushed for 2” in a straight line. The recorded y-coordinate is converted from inches to counts. Table I below shows the values recorded for 10 trials.

TABLE I. DISTANCE TEST DATA

Test number	***Distance (inches)***	***Distance (Counts)***
1	1.96	884
2	1.94	873
3	1.94	875
4	1.93	869
5	2.06	927
6	1.93	869
7	2.00	900
8	2.09	941
9	1.99	896
10	2.00	899
Sum	19.85	8933
Mean	1.99	893.30
Variance	1.36	612.23
Standard deviation	0.05	24.74

From Table I, it is evident that the optical mouse sensor has a relatively high accuracy. The error between the recorded mean value and typical mean value from the datasheet is about 7 counts, which is equivalent to 0.016”. This means the optical mouse sensor has an accuracy of approximately 98% per inch. Although the absolute maximum error from calculations is 41 counts compared to typical value, which is equivalent to 0.091”, the error can be explained owing to surface irregularities and variation in manual pushing of the sensor. Generally, this test shows that the sensor has high accuracy.

An important fact to note is, although relatively small errors have been recorded in the 2” distance test, when the robot is in motion, it covers a much greater distance and the errors start accumulating up to a point where the positional data loses significance.

To show the significance of the error accumulation, Figure 5 shows a graph of total error against the distance travelled. A small set of data is recorded and based on the trend (solid line); the data is then interpolated (dotted line).

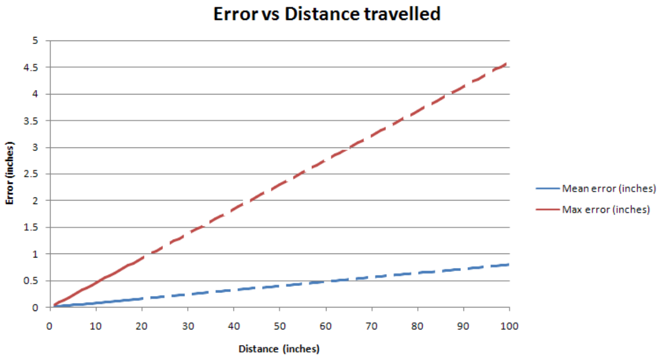

Figure 5. Error accumulation

From the graph above it can be deduced that as the distance travelled increases, so does the accumulated error. The maximum error that can occur is estimated at 4.6”/100”. This gives it an accuracy of about 95%.

The error estimated in Figure 6 is for distance in the **y**-direction. Similar results can be obtained for distance in the **x**-direction. This leads to an approximate total accuracy of about 90% per 100”, which can cause large navigational deviations. To minimise this problem, the optical mouse sensors need to be calibrated to take into account accumulating errors.

B. System simulation

The system simulation test shows the trajectory of the robot when a generated path parameters are applied. For this test, the MPLAB simulator is used to display coordinates through hyper-terminal. The program accepts inputs from the path generation table (Table II) and calculates the simulated coordinates according the algorithm in section III. These coordinates are plotted in MATLAB to show the simulated robot trajectory. Table II below shows the generated path for a circle.

TABLE II. PATH GENERATION TABLE

Counter	*Duty cycle 1 (%)*	*Duty cycle 2 (%)*
0	100	100
300	80	0
490	100	100
790	80	0
980	100	100
1195	80	0
1385	100	100
1685	0	0

This table shows how the motor speeds (duty cycles) need to be adjusted for a square path. It involves keeping both speeds at a fixed value for the straight path, followed by switching off motor two so that an arc can be made. These two sequences are repeated to generate a rectangular path. Figure 6 below shows the simulated path.

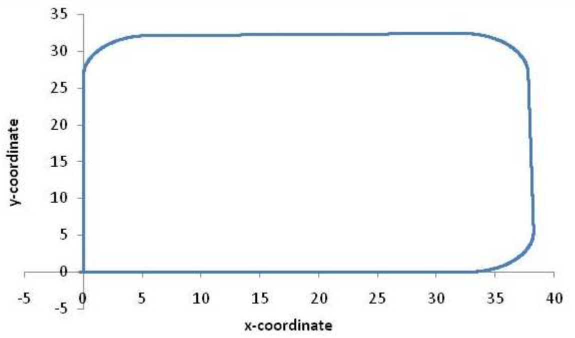

Figure 6. Simulated path for square with round edges

From the figure above, it can be deduced that the dead reckoning algorithm performs accordingly; coordinates calculated match the expected outcome.

C. Complete system test

The objective of the system test is to calculate the robot's absolute coordinates and angle. To test this objective, no special code is required, and the robot is manually pushed to a fixed point in space. For the purpose of this test, the fixed point has coordinates (36, 17) inches. The robot is pushed to the fixed point, and the coordinates are displayed on the LCD screen. Table III shows the data recorded.

TABLE III. DISTANCE TEST DATA

Test number	*x-coordinate (inches)*	*y-coordinate (inches)*	*Distance (inches)*	*Absolute error (inches)*
1	51.5	18.0	54.6	14.7
2	53.5	13.0	55.1	15.2
3	46.0	18.0	49.4	9.6
4	47.5	11.0	48.8	8.9
5	49.5	22.0	54.2	14.4
6	50.5	23.0	55.5	15.7
7	52.5	19.0	55.8	16.0
8	45.5	20.0	49.7	9.9
9	48.5	26.0	55.0	15.2
10	50.0	21.0	54.2	14.4
Sum	495.0	191.0	532.2	134.1
Mean	49.5	19.1	53.2	13.4
Variance	7.1	20.1	7.7	7.7
Standard deviation	2.7	4.5	2.8	2.8

The results of this test show how large the accumulation error is when the actual system is used to calculate absolute coordinates. The mean error is 13.4", which is equivalent to a 34% error. It is clear that owing to systematic and non-systematic errors, the accumulated error is large. Table II shows that the smallest error is even relatively large and this shows that if the sensors are not calibrated properly and systematic errors are not carefully minimised, the accumulated error will cause the measurements to be very inaccurate. Therefore, if the accumulated errors are not minimised from the start, they will increase exponentially causing measure data to be useless [15].

The system accuracy is primarily limited by the optical mouse sensor resolution, which is 400 counts per inch or 0.0025inch per count [7]. Due to accumulation of errors, this resolution is lost greatly as discussed in Section V.

The optical mouse sensor has a maximum forward speed of 12 inches per second [7]. This speed however, could not be achieved by the robot because the microcontroller code included compulsory delays due to the LCD screen. This speed reduced to approximately 7.5 inches per second.

The sensor is also highly affected by the surface it moves on. This is due to the fact that it processes images reflected from the ground [7]. As a result, the resolution of 400 cpi is not always constant and, therefore, the software code must be corrected according to the surface.

VI. CONCLUSION AND RECOMMENDATIONS

As explained in section IV, odometry sensory is prone to systematic and non-systematic errors [2]. The systematic errors in this robot design can be generated by imperfections in the measurement of position and orientation of the two mice with respect to the robot as well as the resolution of the mouse, which depends on the surface on which the robot is moving. Though it is expected that both optical mouse sensors would have the same resolution, there is often a slight variation owing to the nature of operation. Non-systematic errors that affect the robot include irregularities on the floor surface such as cracks and bumps, wheel slippage and crawling. As odometry is inevitably affected by the unbounded accumulation of errors, the actual robot is not able to accurately calculate the coordinates. These errors can be reduced by carefully calibrating the sensors with many repeated tests, but this serves to only minimise the error, not eliminate it. There are several strategies that propose methods for fusing odometric data with absolute position measurements to obtain more reliable position estimations [16] [17]. However, despite its limitations, a reliable odometry system providing good short-term accuracy simplifies the navigation task.

The developed robot system has a high accuracy of relative positioning. However, the drawback is a low absolute positioning accuracy, which can be overcome by incorporating additional sensors, such as: electronic compasses, gyroscopes and CCD cameras [8].

VII. ACKNOWLEDGEMENTS

The authors would like to acknowledge financial assistance provided by the General Motors South Africa Chair of Mechatronics and the Technology Innovation Agency (TIA) for supporting this project.

REFERENCES

[1] R Siegwart and I R Nourbakhsh, Introduction to Autonomous Mobile Robots. Massachusetts, USA: The MIT Press, 2004.

[2] A Bonarini, M Matteucci, and M Restelli, "A kinematic independent dead-reckoning sensor for indoor mobile robotics," in proceeding of the IEEE International conference on Intelligent robots and systems, Japan, Sendai, 2004.

[3] I Kamal. (2008) ikalogic. [Online]. http://www.ikalogic.com/wfr2.php

[4] T Johnson. (2003) Mousebot (How to extract quadrature encoders from computer mice). [Online]. Mousebot (How to extract quadrature encoders from computer micehttp://www.seattlerobotics.org/encoder/200311/johnson/Mousebot.html

[5] A Engelbrecht, Computational intelligence - an introduction, 2nd ed. Pretoria, South Africa: John Wiley & sons ltd, 2007.

[6] G W Lucas. (2001) A tutorial and elementary trajectory model for the differential steering system of robot wheel actuators. [Online]. A tutorial and elementary trajectory model for the differential steering system of robotwheel ahttp://rossum.sourceforge.net/papers/DiffSteer/DiffSteer.html.

[7] Avago Technologies, "SDNS-2083 optical mouse sensor", USA, datasheet 2006.

[8] D Sekimori and F Miyazaki, "Precise dead-reckoning for mobile robots using multiple optical mouse sensors," Akashi National college of technology, Graduate school of engineering science Osaka University, Japan, Thesis 2007.

[9] D Walter. (2000) Seattle Robotics. [Online]. http://www.seattlerobotics.org/encoder/200010/dead_reckoning_article.html

[10] J Borenstein and L Feng, "Measurement and correction of systematic odometry errors in mobile robots," IEEE Transactions on Robotics and Automation, vol. 12, no. 6, pp. 869-880, 1996.

[11] D Sorensen, "Online optical flow feedback for mobile robot navigation," Texas A&M University, Texas, Thesis 2003.

[12] F Chanvier and J Crowley, "Position estimation for a mobile robot using vision and odometry," in Proceedings of IEEE International Conference on Robotics and Automation, France, 1992, pp. 2588-2593.

[13] I Cox and Blanche, "an experiment in guidance and navigation of an autonomous mobile robot," IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp. 12-14, 1991.