Automated high speed omnidirectional navigation using closed loop implementation of four wheel holonomic mecanum drive

Prof. Savita G. Kulkarni

Gururaj N. Mulay
Dept. Of Electronics and Telecommunication
MIT
Pune,India

Tanmay Patil ,Charudatta Parkhe
Dept. Of Computer Science
MIT
Pune,India

parkhecharudatta@gmail.com

***Abstract*—Omnidirectional navigation is being increasingly used to maneuver in space constrained environments. Mecanum wheels is one of the ways to achieve omnidirectional navigation and have several advantages over other alternatives, however they are prone to slippage and loss of control at high speeds. Firstly this paper describes an automated navigation system which extends the concept of dead reckoning and applies it to a four wheel omnidirectional drive system using mecanum wheels. Secondly it improves the system by proposing a technique to negate the slippage of a wheel, by using an additional sensor, a gyroscope. Further, to compensate for the increased processing required for slip negation, the paper proposes a nested closed loop control structure and also briefs about layout of the embedded system on which it was implemented. This system can be applied for high speed as well as for precision automated omnidirectional navigation. This system was implemented and tested in various conditions, results compared with a standard open loop system showed significant improvement in navigation accuracy.**

Index Terms*—Closed-loop holonomic drive, position control ,RPM control(key words)*

I. Introduction

Holonomic omni-directional navigation is being increasingly used in industries and for mobile robotics application. The focus now shifts for such systems to be applied from manual control to complete autonomous navigation. Conventionally any navigation system is automated using two approaches internal feedback, and external references. External references include such as guiding lines on ground, or sensing some other external parameters, however they cannot be employed in fields without any references. Automation through internal feedback is achieved by taking feedback of the internal parameters such as rotation counts of wheels.

Many approaches have been proposed over the years. Raul Rojas and Alexander Gloye Forster [1] in their proposition stated that it is possible to derive a forward kinematic equation for an n-wheel holonomic vehicle/robot with omni wheels [1].The paper also addresses the problem of identifying the slipping wheel and compensating for the same. Many other approaches have proposed PID control loop and fuzzy control loop for the navigation of the omnidirectional vehicle. Though research and work in the automation of an omnidirectional drive is available automation of a omnidirectional vehicle having a mecanum drive using closed control loops is not well known.

This paper addresses the issue of the automation of a four wheel omnidirectional drive system consisting of mecanum wheels using nested closed control loops. The proposed method extends the concept of dead reckoning by describing an automated navigation system with only its driving motors' internal encoder counts and the angle provided by a gyroscope. Secondly, it also addresses the problem of slippage identification and compensation using a rather simple yet effective approach by using the gyroscope and the internal encoder counts of the driving motors[5]. This method also compensates the increased processing required for slip negation. The embedded system used for the testing of the proposed method is also explained along with the results compared with a standard open loop system[2].

This system was developed for high speed omnidirectional navigation requirements of the ROBOCON 2013 theme. The system was developed for achieving navigating of a robot through space constrained regions and at the same time achieves high linear speed. Though the system is described for mecanum wheels, omni wheels may be used as well, with a few changes in the holonomic constraint equations. Also the system is applicable for n-wheel omnidirectional drive system however the procedure described for inconsistent equations may not be applicable.

II. Closed loop holonomic drive

A. Four-wheel Holonomic Drive

Holonomic omnidirectional navigation allows a mobile robot/vehicle to rotate and navigate freely in any possible direction without any constraints hence used for navigation in space restricted environments. Such a system is widely operated manually using open loop control systems. To automate such a drive a closed loop system must be realized which will require inter-conversion between wheel velocity and resultant velocity.

To obtain relation between the velocities to be given to the wheels and the resultant velocity (linear and angular) of the robot, holonomic constraint equations are used. The constraint equation for a three wheel system reduces to three equations, and inverse of the equations is computable. However any drive having greater than three wheels introduces redundant equations, consequently the equation may or may not be consistent. This paper proposes a method for automation of four wheel drive system handling consistent as well as inconsistent equations. Further a nested control loop is proposed for improving the performance of the practical implementation of such a system.

B. Internal Feedback: Angular velocity of each wheel

The only feedback the control system demands is the angular velocity of each individual wheel. Practically such a system can be realized by attaching rotary encoders with the shafts of each wheel. For each execution of control loop executing in time interval dt, encoder counts obtained would imply angular velocity of that wheel (ω_n). The angular velocity is directly proportional to the linear velocity of the wheel in the direction roller of that wheel. Hence the rotations per minute (RPM) count provides for the instantaneous linear velocity of each wheel which will be used for position tracking which is the upper layer closed loop system.

III. Holonomic Constraint Equations

Figure.2 shows the distribution of velocity vectors along the X and Y axes of Cartesian coordinate system. Here x' and y' are the axes of robot frame.

Velocity vector V1*, corresponding to first wheel consists of three components one in X direction, second in Y direction and third is angular velocity component Rω.

$$V1 = V1_{(x')} + V1_{(y')} + R\omega \qquad (1)$$

*all the velocities described henceforth are vector in nature

Where $V1_{(x')}$ is the component of velocity vector V1 along X axis and $V1_{(y')}$ is the component of velocity vector V1 along Y axis .

Now, the X and Y components to the wheel producing V1 are resolved below. There is also a Rω which contributes to V1. Hence the equation for V1 can be described as,

$$V1 = -V_x \sin(45+\theta) + V_y \cos(45+\theta) + R\omega \qquad (2)$$

Similarly, the equations can be derived for V2, V3 and V4 as follows,

$$V2 = -V_x \sin(45-\theta) - V_y \cos(45-\theta) + R\omega \qquad (3)$$
$$V3 = V_x \sin(45+\theta) - V_y \cos(45+\theta) + R\omega \qquad (4)$$
$$V4 = V_x \sin(45-\theta) + V_y \cos(45-\theta) + R\omega \qquad (5)$$

All the above equations can be expressed in matrix form as below,

$$\begin{bmatrix} V1 \\ V2 \\ V3 \\ V4 \end{bmatrix} = \begin{bmatrix} -\sin(45+\theta) & \cos(45+\theta) & 1 \\ -\sin(45-\theta) & -\cos(45-\theta) & 1 \\ \sin(45+\theta) & -\cos(45+\theta) & 1 \\ \sin(45-\theta) & \cos(45-\theta) & 1 \end{bmatrix} * \begin{bmatrix} Vx \\ Vy \\ R\omega \end{bmatrix}$$

So it is possible to find the velocities given the required V_x, V_y and Rω. The above stated matrix is nothing but the forward constrained equations. In order to find the reverse constrained equations we need to find the inverse of the above stated matrix.

Now, considering the four equations (2)(3)(4) and (5), it is observed that there are three variables and four equations. Hence, the system is over-consistent. Any three equations can be chosen for solving for V_x, V_y and Rω, we can get the current vectors of the wheels due to its previous taken correction.

$$\begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix} = \begin{bmatrix} -\sin(45+\theta) & \cos(45+\theta) & 1 \\ -\sin(45-\theta) & -\cos(45-\theta) & 1 \\ \sin(45+\theta) & -\cos(45+\theta) & 1 \end{bmatrix} * \begin{bmatrix} Vx \\ Vy \\ R\omega \end{bmatrix}$$

The system may become inconsistent if any of the wheels slips. So the matrix now becomes, After getting the inverse, we get following equations,

$$V_x = V1(M_{11}) + V2(M_{21}) + V3(M_{31})$$
$$V_y = V1(M_{12}) + V2(M_{22}) + V3(M_{32})$$
$$R\omega = V1(M_{13}) + V2(M_{23}) + V3(M_{33}) \qquad (6)$$

Where M_{mn} is respective cofactor matrices of given matrix.

The equations for V_x and V_y are obtained as follows,

$$V_x = (c1-c2)\ V1 + (-2c1)\ V2 + (c1+c2)\ V3 \qquad (7)$$
$$V_y = (s1+s2)\ V1 + (-2s1)\ V2 + (s1-s2)\ V3 \qquad (8)$$

The equations above can be solved by substituting the values of the counts per dt (delta time) obtained from the lower level control loop. The counts resemble the distance traveled in the V_x and V_y direction in delta time. It is observed that if these are integrated over time we get the distance traveled along the respective X and Y axes.

The angle (theta) translates these co-ordinates of the system to the field i.e .real Cartesian co-ordinate system. Theta represents the angle between the two co-ordinate systems at time t.

Now in order to convert the counts in terms of distance traveled in measurable quantities like centimeter or meter, calculate the counts obtained when the wheel of the robot complete one complete revolution. This gives us the counts obtained when the wheel travels a distance equivalent to its circumference. The accumulated counts after integration are divided by the counts obtained per revolution of the wheel. Hence, the precise X and Y co- ordinates are obtained on the field (real) Cartesian co-ordinate system.

A. Slipping Wheel Identification and Negation

The major problem at high speeds for mecanum wheels is slipping. Since slippage in wheels will introduce deviation from true values of the RPM measured from the encoders, it would significantly impact the accuracy of the dead reckoning. In case of slippage, equation (V1+V2 = V3+V4) for consistency check will fail [1]. Hence choosing different combinations (of three equations) from V1, V2, V3, and V4 would result in different results.

Firstly, we compute the V_x, V_y, $R\omega$ for all the four possible combinations of choosing three equations.

$(V2,V3,V4) \rightarrow (V_x, V_y, R\omega)_1$ …(9)
$(V1,V3,V4) \rightarrow (V_x, V_y, R\omega)_2$ …(10)
$(V1,V2,V4) \rightarrow (V_x, V_y, R\omega)_3$ …(11)
$(V1,V2,V3) \rightarrow (V_x, V_y, R\omega)_4$ …(12)

To eliminate the slipping we introduce a fairly accurate gyroscope into the system. Polling the gyroscope within the same interval in which the RPM was polled we obtain ω_{gyro} from it. R being know for the machine, or computed experimentally (shown below) R ω_{gyro} is computed. This should ideally match with each of the $R\omega_n$ obtained. However if a single wheel slips, error will be introduced in three of the $R\omega_n$ computed, while only one of them is error free. The $R\omega_n$ that matches the R $\omega_{gyro,}$ will hence be the error free triplet.

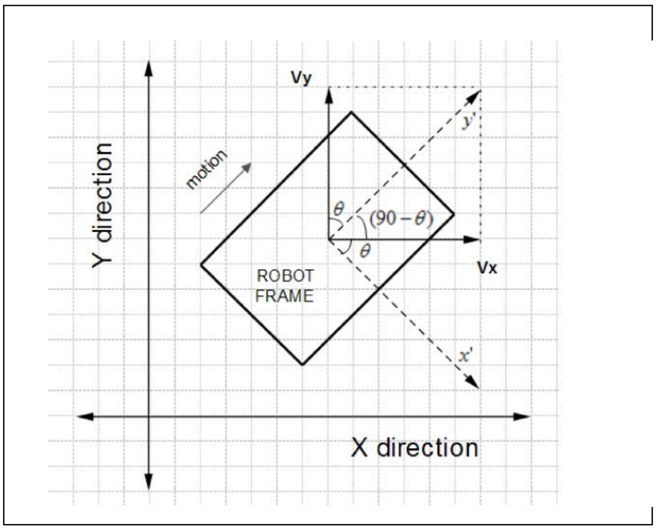

Fig.2. axes conversion

Hence this triplet will be used for further calculations while the rest are discarded.

- Example: Consider wheel-1 slips, RPM measured will be V1+d. This will affect the output of (10),(11),(12) however (9) will be unaffected since V1 is not used to compute (9). Since (9) is error free, $R\omega_1 = R\ \omega_{gyro}$.

- Limitations: The above method works only if a single wheel slips. In case of multiple wheel slippages we may select the triplet having its $R\omega_n$ closest to $R\omega_{gyro}$. This will minimize the error due to slippage; however this will not completely eliminate it.

B. Experimental Computation of Rω

If R is not known for the system it can also be computed experimentally. However we must ensure that no slippage occurs during this experiment. The absence of slippage implies that the consistency check will always be true and hence any $R\omega_n$ chosen must match $R\ \omega_{gyro}$.

$$R\ \omega_{gyro} = R\omega_n$$
$$\text{i.e.}\quad R = R\omega_n / \omega_{gyro}$$

IV. CONTROL LOOP SYSTEM

The above derived equations show that for a complete closed loop implementation each iteration of control loop must compute the forward as well as reverse constraint equations multiple times in case of slippage. This would require significant processing time on an embedded system for a mobile robot. Higher processing time implies lesser frequency of operation of control loop, resulting in poor response time of system. Hence a dual layer control loop has been proposed dividing the task to be performed, into position control and rpm control. While the task of the position computation is done is one loop, the other loop which requires few computations, runs much faster and controls the rpm of each motor. Hence a smooth and improved rpm control is obtained. Also the PID factors for the RPM control loop are so tuned that the slippage due to jerks is avoided.

A. Upper Layer Closed Loop System- Position Control

The upper layer control system works as a position control. This control loop dynamically tracks the change in the current position and orientation from starting position. To obtain this the feedback of the control loop is the rpm of all the four wheels. The rotations counts within a Δt time interval (r1, r2, r3, r4) are converted to the ΔX, ΔY and $\Delta\vartheta$ applying reverse constraint equations as shown in Fig.4 and Fig.5.

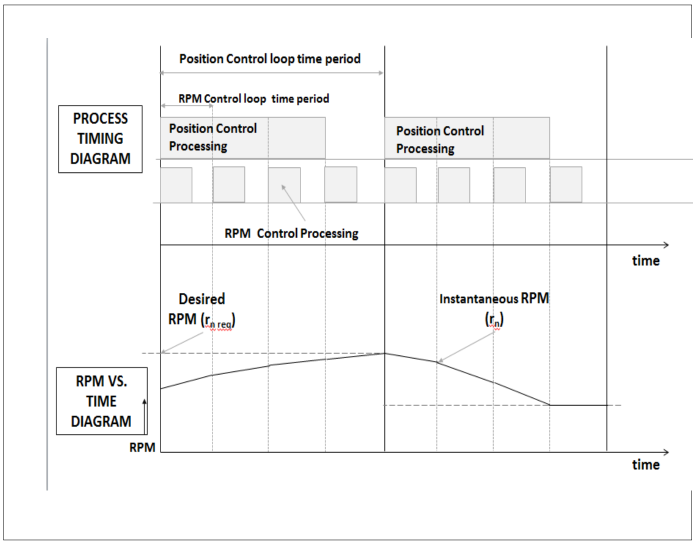

Fig.3.process timing diagram

The ΔX, ΔY and $\Delta\vartheta$ are integrated over the time interval to obtain X, Y, ϑ. The current position and orientation of the bot being now available, the destination velocity its direction and required orientation (Vx, Vy, ω) is dynamically adjusted to reach the destination.

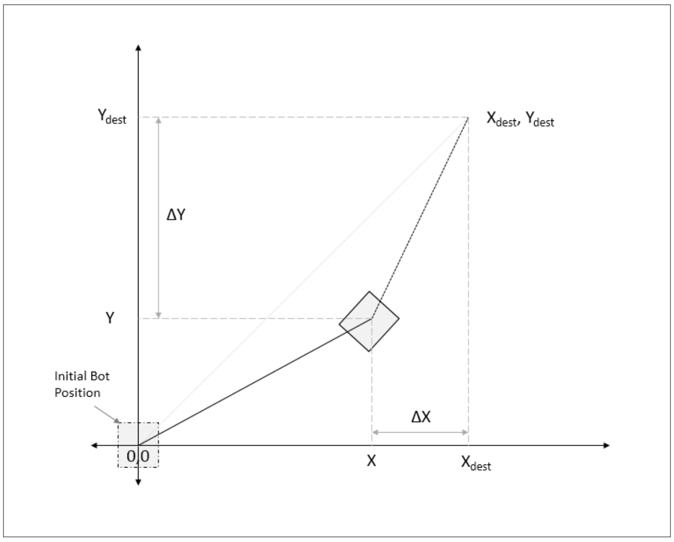

Fig.4.position control

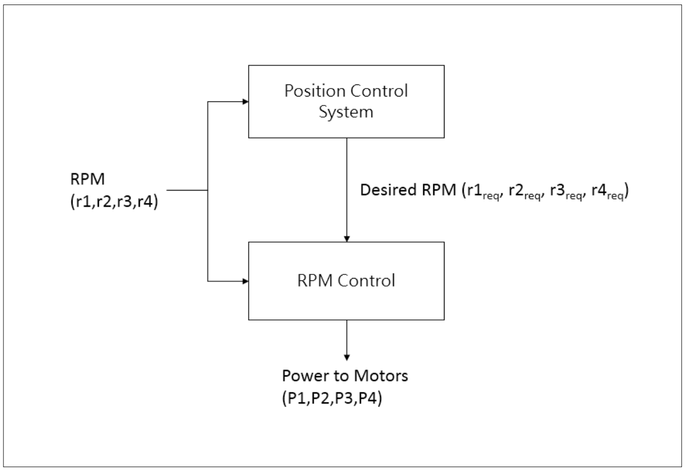

Fig.5 flowchart

The forward holonomic constraint equations are then used to obtain desired RPM of each wheel (r1req, r2req, r3req, r4req).

B. Lower Layer Closed Loop System- RPM Control

The lower layer RPM control loop enables the upper layer control system to work on an abstract level. The Position Control Loop provides the desired set of rpm for each wheel as an output, this is fed to the RPM Control System. The RPM control System ensures that each wheel rotates at the desired

RPM. To maintain the RPM at a desired value, feedback of the RPM is taken from the motor encoders and the power (P_n) supplied to the motor is accordingly adjusted by applying a standard PID control loop.

The PID coefficients are adjusted such that there are no sudden changes in the output which would result in violent acceleration and slipping of wheels. Hence it may take several cycles to achieve the desired RPM. To facilitate this RPM Control System runs at a frequency much higher that the Position Control System. This implies that the RPM Control Loop runs several cycles within a single Position Control cycle and ensures that the RPM will be achieved and stabilized within a single cycle of the Position Control System as shown in Fig.3.

V. Electronic Implementation

The algorithm mentioned is successfully implemented on a motherboard divided in three distinct blocks. First block - the master block- implements the upper layer closed loop using master microcontroller and second block-the slave block- implements the lower layer closed loop system using two slave microcontrollers, as shown in figure 6.

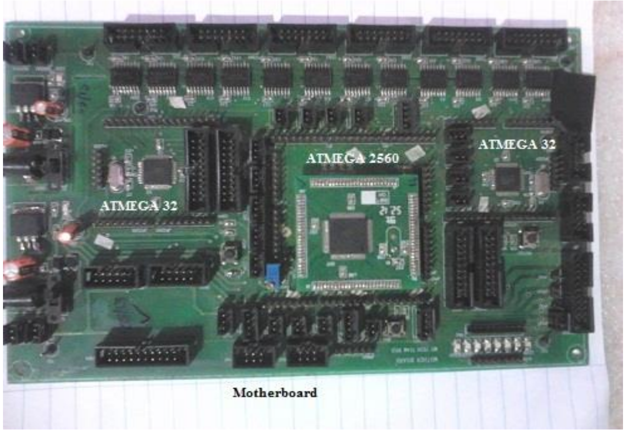

Fig. 6.Motherboard for Electronic implementation

The third block consists of motor drivers of respective motors.

A. Master block

The master controller ATMEGA-2560 is a microcontroller from ATMEL Corporation. Working at 16 MHz frequency, it communicates with two slaves using Universal Asynchronous Receiver Transmitter (UART) protocol. A gyroscope from ANALOG DEVICES was interfaced with ATMEGA2560 using Serial Peripheral Interface protocol, which updated current angle at specific intervals of time. It also has capability interface different sensors at its ADC port.

5 Volt (V) regulated from 8 V battery was used for digital circuit. To ensure general-purpose usage, the motherboard has different connectors brought up as in Fig.6 .

B. Slaves block

Two ATMEGA-32 used as the slaves on the motherboard. Each of the ATMEGA32 drives two out of four motors

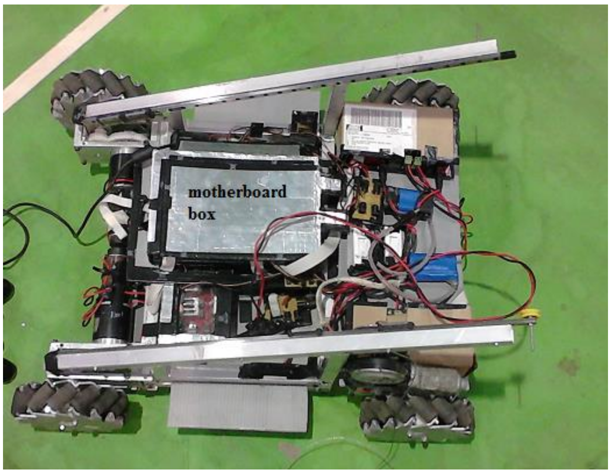

Fig.7 Physical Realization of Mecanum Drive

with RPM control algorithm. These motors were MAXON RE40 motors provided with gear reduction. Their encoders have 512 Parts Per Million (PPR) resolution. Two encoders in feedback loop were connected to each ATMEGA32. These encoders continuously updated ATMEGA32 with two (90 degree) phase shifted signals (counts) enabling to find speeds as well as directions of the respective motors. The RPMs maintained by the lower level algorithm are converted into a variable pulse i.e. Pulse Width Modulation (PWM).

This PWM was routed to motor drivers of each motor facilitating the RPM control the motors.

C. Motor Drivers

Well known LMD18200 driver is a linear motor driver Integrated Circuit (IC) which was fed with the calculated PWM for respective motor. By changing the width of PWM, it is possible to change the speed of motors. Working at 24V battery supply, LMD controlled the voltage supplied to motors, which in turn controlled the RPM. To maintain noise free environment at digital circuit, the motor driver circuitry was isolated by placing Isolator ICs in between the motherboard and motor driver.

VI. Results and Cconclusions

A. Results

The results were collected to demonstrate the necessity of a closed loop system which takes RPM feedback. The open loop system was operated manually upon reaching X_{dest}. While the closed loop system was programmed to reach destination coordinates. The closed loop system in this experiment does not include slip elimination and nested control loop. The average linear speed provided to the robot was 50 cm/sec.

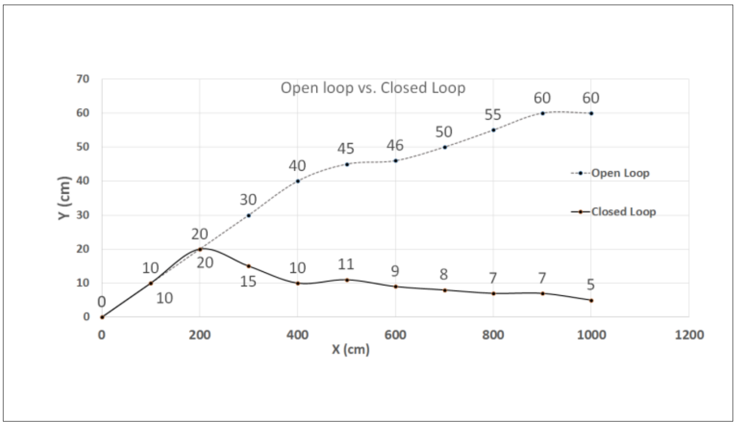

Fig.9.open loop vs. closed loop

We observe that as the open loop does not take any feedback it fails to correct itself and as error accumulates it deviates from the destination, while the closed loop system Position control ensures that the error is corrected. The ($Y - Y_{dest}$) taken as error shows a 91 % improvement in accuracy (refer Fig.9).

Experiment performed shows the improvement due to the nested control loop. The same setup as above was used however the average speed was increased to 150 cm/sec as in Fig.9.The observations show that with the normal control loop the robot oscillates, indicating that the response time of the control loop is large, while the smooth path taken by robot with the nested control loop depicts an improvement in response of control. The overall percentage increase in accuracy is 80%.

This experiment shows the improvement due to addition of the slip negation algorithm. The same setup as above was used, the average speed was increased at 200 cm/sec for deliberately increasing slip. The observations show that (Fig.8) with the normal control loop the robot oscillates, indicating that the response time of the control loop is large, while the smooth path taken by robot with the nested control loop depicts an improvement in response of control.

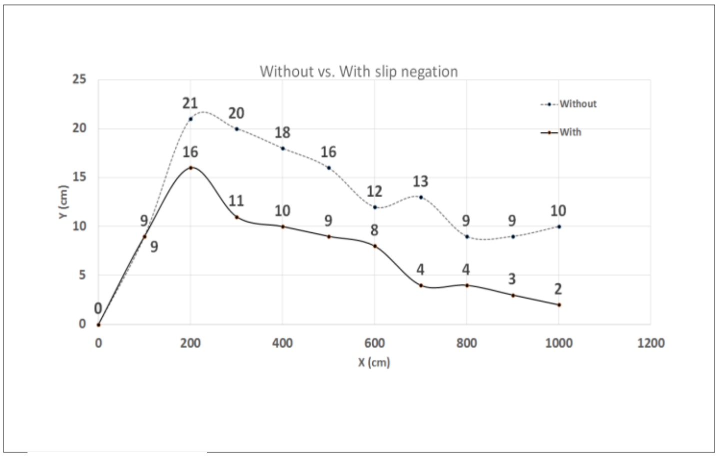

Fig.8.slippage graph

B. Conclusions

This system was implemented on a self-fabricated mecanum robot with least mechanical errors that could occur in the structure of robot frame. Testing in various conditions of surface friction as well as in various paths of navigation, showed significant improvement in navigational accuracy as compared with results of a standard open loop system.

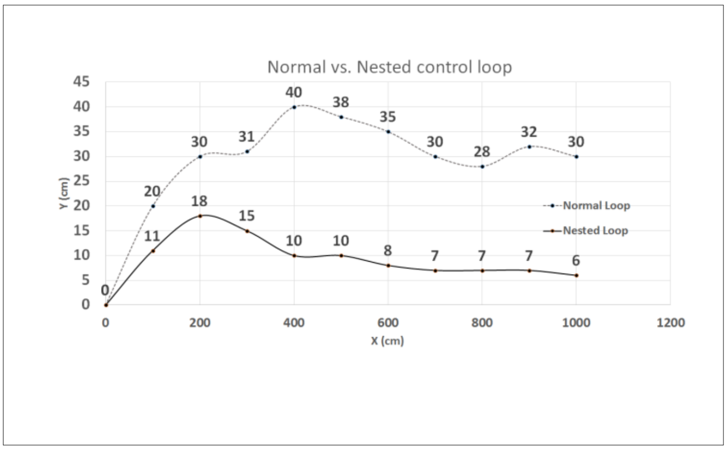

Fig.10.normal vs. nested loop

Acknowledgment

All the hardware, including the laboratory for testing was provided by Maharashtra Institute of Technology, Pune along with sponsorship for all the equipment used for practical experiments.

References

[1] Raul Rojas, Alexander Gloye Forster, "Holonomic Control of a robot with an omnidirectional drive.," To appear in KI - K¨unstliche Intelligenz, B¨ottcherIT Verlag, 2006.

[2] D. J. Daniel et al.: "Kinematics and Open-Loop Control of an Ilonator-Based Mobile Platform", Proc. of the IEEE Int. Conf. on Robotics and Automation, 1985.

[3] L. Huang et al.: “Design and Analysis of a Four-wheel Omnidirectional Mobile Robot”, Proc. of the 2nd Int. Conf. on Autonomous Robots and Agents, 2004.

[4] O. Purwin et al.: “Trajectory Generation and Control for Four Wheeled Omnidirectional Vehicles”, Proc. American Control Conference, 2005.

[5] R. L. Williams et al.: “Dynamic Model With Slip for Wheeled Omnidirectional Robots”, IEEE Tr. on Robotics and Automation,vol. 18, no. 3, June 2002.