International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH14) 28 & 29 November 2014

Odometry and Motion Planning for Omni Drive Robots

Paril Jain
R & D Department
Azoi Inc
Ahmedabad, India
paril9000@gmail.com

Abstract— **Omni drive robots operate by combined motion of more than 2 wheels leading to a force vector in resultant direction and thus enabling the robot to travel in that direction. Omni drive systems are quite efficient from the travelling point of view as compared to non-holonomic (non omni drive) systems, which require stopping at a position and taking a turn. This paper describes a very sophisticated yet simple approach to motion planning of such omni drive robots that would not only decrease the time taken by these robots to perform several tasks, but also increase the efficiency to quite a greater extent. Moreover the system being less complex, it can be implemented on a low cost processing unit and there is no need for high end processors or computers to run such robots. These systems, if developed properly, can bring a quite breakthrough by involving robotics in our daily life commercial applications.**

Keywords—Motion planning, Navigation, Omni wheeled robots, Trajectory planning, odometry

I. Introduction

Omni drive robots are those which can travel in any direction in any orientation without rotating beforehand. These robots posses more than 2 wheels such that a proper combination of the forces due to all these wheels can result in proper motion of the robot in any direction. Comparing it with conventional 2 wheeled robots that require rotating in desired direction initially before travelling forward, it is quite obvious that omni drive robots are time saving as well as efficient. We do have synchro-drive system robots that can also travel in any direction, but the driving motors in such robots rotate themselves and thus these require extra actuations. Whereas, omni drive systems, can simply adjust the force components of each individual wheels and carry out the same task.

Motion planning of wheeled robots is an active area of research even though it has been in progress since many years. New ideas and usage of new sensors keep getting introduced in this field of research. In my previous work [5] I presented the odometry and path planning for 2 wheeled robots. But for the way the applications of robots are increasing, usage of omni drive robots seems quite essential. A large amount of research has been done in this area as well. Relations of motor velocities on omni robots with their travelling direction has been presented by Mark Ashmore [1]. People have also tried to integrate several methods like fuzzy logics and artificial potential field (APF) as shown in [2]. Research has also been done on usage of wavelets to simplify the motion planning algorithms [3]. A wide range of sensors have been used for this task of self-localization (odometry) of omni-drive robots such as vision sensors or intertial sensors [4]. But along with all these it is very essential to define a simpler less complex odometry system that would allow us to perfectly control the motion of robot even in case of driving wheels slippage or some minor obstruction.

This paper brings forward a quite easy to implement and effective odometry system for omni-drive robots, matching the above mentioned criteria. In addition to this, few extremely efficient ways of motion planning of these robots have been developed that would make the control of these omni-drive robots quite flexible.

The paper outline is as follows: Section I describes about the basic concept of omni drive robots and their need. It introduces to the flow and the need of this paper Section II is aimed at explaining a novel approach to odometry generation for omni-drive robots using a combination of rotary encoders and Inertial Measurement Unit(IMU). Section III introduces us to the way we can control the motors of the robot perfectly to travel as per requirement in desired direction. Section IV is the main idea of this paper, where we will talk about the ways of liberalizing the motion of omni drive robots and use them to their full efficiency by an accurate controlling system. Section V includes the results of various tests carried out while developing this algorithm.

II. Odometry For Omni Drive Robots

The odometry system being described here is meant for all wheeled robots travelling on flat surface, be it 3 wheeled, 4 wheeled, mecanum or other configurations. The system consists of 2 rotary encoders suspended on a dummy wheel in center of the robot base (Independent on the shape of base) such that both the encoders are fitted perpendicular to each other, one in the direction accepted as 0 degree travelling

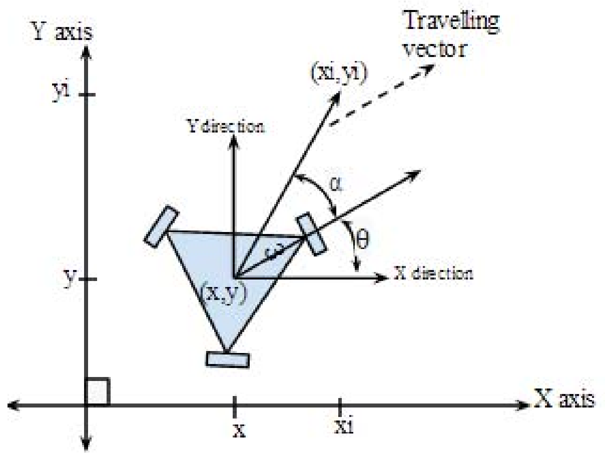

Figure 1. Odometry Calculation

direction, and the other perpendicular to it. The robot is also fitted with Inertial Measurement Unit at the exact centre of the robot base.

Before we start with the mathematical equations, it is essential to peruse the robot dynamics. All the explanations have been made taking 3-wheeled robots as an example. The system can be extended to other configurations. By properly controlling the speed of the wheels, omni wheeled robots can travel in any direction without changing their orientation.(Explained in Section III). Thus, while discussing about odometry calculations, we need to take 2 angles into consideration – orientation angle and the travelling angle. As the robot can change its orientation as well as travel in any direction, both these angles are essential to determine to estimate the position of the robot at any instant.

As shown in figure 1, we call θ as the main orientation angle of the robot and α as the instantaneous travelling angle of the robot. The θ is obtained by integrating the angular velocity given by the inertial measurement unit. It can also be made more accurate by combining it with external magnetometer sensor. The 2 rotary encoders fitted perpendicular to each other provide fixed number of pulses per revolution and hence by knowing the diameter of the wheels on which the encoders are fitted, we can find net distance travelled by each encoder. Let us denote the distance travelled by the encoder in direction of 0 degree/180 degree θ as x_enc and the distance travelled by the other encoder as y_enc.

$$\alpha = \frac{(y\_enc_i - y\_enc_{i-1})}{(x\_enc_i - x\_enc_{i-1})} \tag{1}$$

$$\Delta r = \sqrt{(x\_enc_i - x\_enc_{i-1})^2 + (y\_enc_i - y\_enc_{i-1})^2} \tag{2}$$

$$\theta = \int \omega dt \tag{3}$$

$$x = \int \Delta r \cos(\theta + \alpha)\, dx \tag{4}$$

$$y = \int \Delta r \sin(\theta + \alpha)\, dy \tag{5}$$

$$xi = x + l \times \cos(\theta + \alpha) \tag{6}$$

$$yi = y + l \times \sin(\theta + \alpha) \tag{7}$$

α = instantaneous travelling angle with respect to instantaneous orientation
ω = angular velocity from Inertial measurement unit
θ = Orientation angle of robot
Δr = Net distance travelled in very small time interval (1 sample)
x, y = Co-ordinates of a robot at any instant with respect to start position
xi, yi = The projected point for trajectory following and PID input(Explained in later sections)

As observed in the above equations, the difference in the distances travelled by both the encoders is calculated and net distance travelled is measured by equation (2). The instantaneous travelling angle of robot – α is calculated using equation (1). Now when the robot moves in 2 D plane, it may rotate as well as travel in any direction. Hence the net distance measured in equation (2) is in the direction at an angle of (θ + α) with respect to the actual) degree axis. The robot is aligned with our assumed co-ordinate axes such that the initial 0 degree travelling angle coincides with the X axis.

In order to find the co-ordinates of the robot travelling in 2 D plane, each such small segments are split along X and Y direction and integrated from the start position as shown in equations (4) and (5). At any point the co-ordinates of the robot represent the co-ordinates of the centroid(centre of the 2 encoders and IMU – placed at centroid) with respect to the start position.

The co-ordinates calculated in equation (6) and (7) are the co-ordinates projected in direction of travelling of robot w.r.t X axis. This is the point that will follow the desired trajectories.(Described in trajectory estimation section). Mechanical construction should be given special care while mounting the sensors for odometry. [9]

III. Motor Control For Omni Directional Motion

In order to enhance the motion of the omni wheeled robots providing them the liberty to move in any manner of their orientation on any desired path it is necessary to establish a control system that provides appropriate command to the motors such that robot remains on the desired trajectory. Moreover for motions such as rotation while moving on a desired path it is necessary that we establish the control system enabling the robot to move in any desired angle.

For this purpose the desired trajectory is given as error to the PID algorithm [10]. The robot may have 3 wheeled motion for our explanation, but from the point of view of trajectory

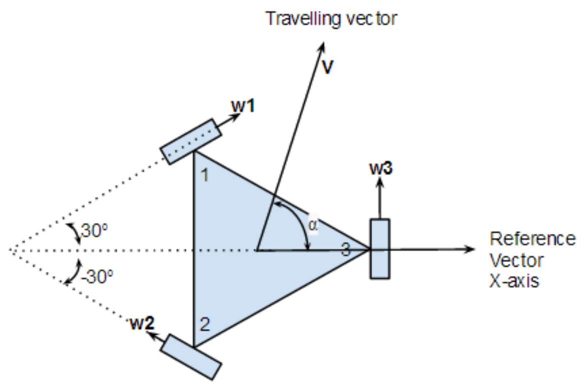

Figure 2. Force components for Motor speeds

following, we can split the wheels as left side acting wheel and right side acting wheel at any instant on the path depending upon the resulting force exerted by each wheel as shown in figure 2.

Once we have classified the wheels as left side acting and right side acting, we can generate appropriate pwm output from the PID control logic assuming a 2 wheeled robot. Once this is obtained, the free body diagram of forces due to the motors on the robot's net motion can be obtained as shown in figure 2.[1]

Suppose the net motion of the robot is in direction of vector **R.** Let V be net velocity of the robot in direction of vector **R**. After PID control logic, this velocity is split into 2 parts and then appropriate components are calculated in the direction of the motion of the wheels as shown in figure 2 and below given equations. In these we have V1, V2, V3 shown by different variables, but two of them will have same values depending on the travelling direction of the robot.

$$w1 = V1\left(\frac{\sqrt{3}}{2}\cos\alpha + \frac{1}{2}\sin\alpha\right) \qquad (8)$$

$$w2 = V2\left(\frac{\sqrt{3}}{2}\cos\alpha - \frac{1}{2}\sin\alpha\right) \qquad (9)$$

$$w3 = V3\,(\sin\alpha) \qquad (10)$$

The wheels are numbered 1, 2 ,3 for easy explanation. Here w1, w2 and w3 are the individual wheel velocities assigned to each wheel depending upon the correction in motion done by PID logic to maintain the robot on proper trajectory.

IV. Motion Planning

As discussed in the previous section, the robot is capable of travelling in any desired direction without changing the orientation. As we aimed at travelling on a straight path in the previous section, the error input for the PID control logic was the difference of the current orientation angle and a fixed angle, thus leading the robot in a fixed direction. We ignored the scenario of slippage of wheels in this case. But for surfaces where slippage of driving wheels may occur we need a fixed trajectory to follow rather than simply a direction where the machine would be heading.

Now suppose we want the robot to travel on a fixed straight line derived from the desired co-ordinates through which the line should pass. The equation of the line can be given as the input to the PID control logic.

For example,

$$\text{error} = yi - m*xi - c \qquad (11)$$

where xi and yi are obtained during each iteration from equation (6) and (7)

Thus when the points xi and yi of the robot are on line of pre-decided slope m and intercept c, the error will be zero and positive or negative in other 2 cases and hence the robot will act accordingly.

In a similar manner equation of any kind of path can be provided as error equation of PID control logic. For example, for a circle of known 3 points on the circle, we can find the radius and centre of that circle and hence we can generate the equation of the circle. The variables of this equation are xi and yi generated in equation (6) and equation (7).

The vector joining centre of the robot to the points- xi and yi is termed as the travelling vector of the robot. In order to allow the robot to travel on a line of any possible slope with any possible orientation, it is necessary to find the angle α. Looking w.r.t X axis, it is quite clear that the argument of cos or sin in equations (6) and (7) should be same as the angle made by line with the X-axis. Thus α can be evaluated.

In both the above cases the robot followed the desired trajectory without having a special control over its orientation while in motion. The robot simply followed the trajectory such that the travelling angle (α) remained constant w.r.t the orientation of the robot.

The following 2 cases will demonstrate the liberty of the robot motion where in we have a control over the orientation of the robot dynamically while the robot is moving. Thus the robot can change its orientation in a controlled manner while moving over its trajectory.

A. Robot changing its orientation while moving over a straight path or any other path

Suppose a robot at orientation θ has to travel along a line with a slope m passing through its centre. This initial travelling angle will be

$$\alpha = \tan^{-1}(m) - \theta \qquad (12)$$

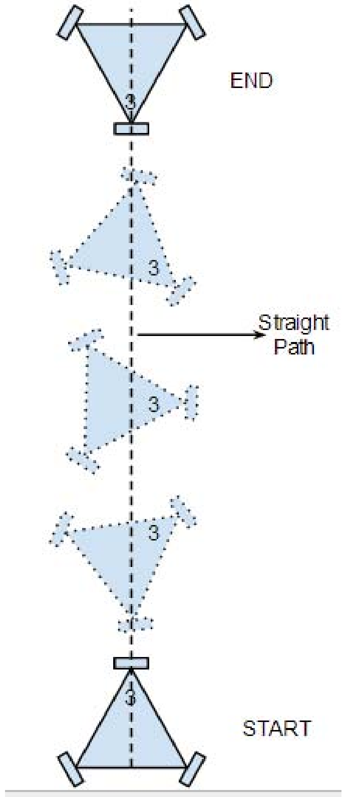

Figure 3. Rotational Motion over a straight line

Now the target is to rotate the robot by 180 degrees while travelling over this line as shown in figure 3.

The projected co-ordinates of the travelling vector are used in the path equation are provided along with the travelling angle to the PID. But now as we are willing to rotate the robot over this predefined path, the travelling angle will be a combination of α and a variable component. In our case, the orientation measuring sensor has been set such that the angle in counter-clockwise direction when the robot is observed from top view and vice-versa. Thus in order to rotate the robot by 180 degrees clockwise direction while the robot travels on the desired path, we can increment the variable component of α from 0 to 180 degrees with increments of 0.2 to 0.5 degree(for smoothness) over fixed distances. To understand this more clearly, refer figure 3**.**

As shown in figure 3 the robot is at start position intially. With increase in the variable component, the travelling vector rotates counter-clockwise**,** but due to PID, the motors try to keep the projected xi and yi on the desired line and hence the robot rotates even when travelling on straight path. By changing the equation of path in above case, we can carry out this method for any other trajectories.

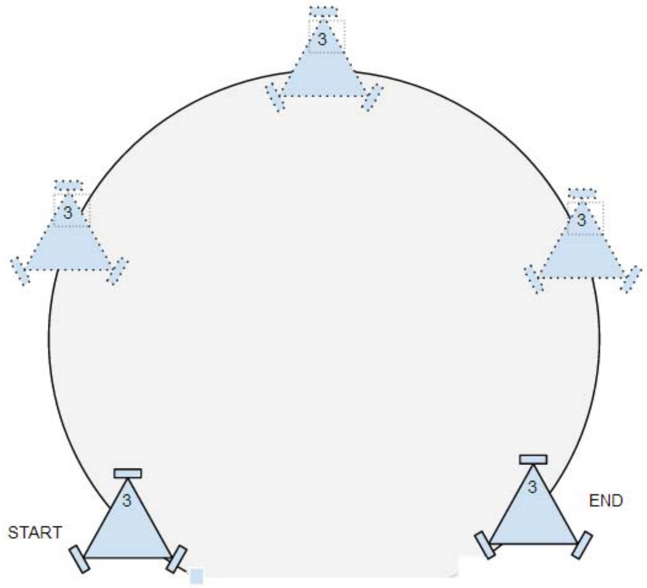

Figure 4. Motion over a curved surface without changing orientation

B. *Motion of Robot over any curved path without changing its orientation*

Imagine a robot moving on a circular path without changing its orientation on any point of its journey. This is based on similar concept as mentioned in the above case. Consider a circular path as shown in figure 4. Its standard equation is fed into the error of PID where xi and yi are used as varying terms. Now as we are aware of the centre of the circle, and we get the projected points from equations (6) and (7), the slope of the line joining the projected points and the centre of the circle can be evaluated. The travelling vector is perpendicular to line joining the points and the centre and hence using the slope relation between a line and its perpendicular, the slope of the travelling vector can be calculated. Thus α is obtained by subtracting the orientation angle from the slope of travelling vector.

As the robot moves in the direction of the tangent to the circle, the travelling angle is updated at each iteration. For the entire path, the robot adjusts its travelling angle such that there is no change in its orientation and the robot can travel as shown in figure 4. This method can be extended to any kind of paths, for example, an S shaped path. We simply need to find the slope of the tangent.

V. Results

All the above algorithms were tested using a commercial inertial measurement unit and Rotary encoders with 1024 PPR from Pepper n Fuch. The driving motors were maxon DC motors that allowed us to test the algorithm at different speeds. A micro-controller unit operating at 120 MHz was used as the central processing unit and hence we were able to carry out calculations at each change in count of the encoders used. A drift was observed in the gyroscope of IMU sensor as it was a low cost commercially available sensor. Usage of high end inertial measurement units for this task would increase the accuracy to a very large extent.

The path planning and motion planning algorithms worked quite perfect at slow speed, but at high speed over very curved surfaces, even though robot could calculate that it is travelling slight offset from the desired path, several factors like wheel slippage, too small radius of curved path to take an immediate curve and robot inertia came into effect. In order to solve this issue, intelligence was added such that the maximum speed of robot would decrease by a small amount when the offset from the path is beyond a limit. Also, deceleration at the end portion of the path resulted in an accurate final position.

The algorithms of motion planning are not only limited to wheeled robots. Once we determine the odometry of any other robotic system, these algorithms can be easily implemented on other systems such as aerial robots, underwater robots and many more. For future works, in the above system if the robot is completely off track, it may appear to wobble before coming on its actual trajectory. Specific algorithms can be easily developed and added to the above system which would enable the robot to determine the error and generate a completely new trajectory to get back smoothly on its actual path without and jerks or wobbles.

VI. Conclusion

The motion of the robot with its orientation in all cases as described in the paper worked perfect, thus providing a very good control over the robot navigation. The motion planning algorithms described here can be implemented in all such applications. Thus, days are not far when we may have robots transporting goods within factories from one place to another or we carrying different requirements in hospitals from one room to another or they might be helping in household activities.

References

[1] Mark Ashmore, Nick Barnes, "Omni-drive Robot Motion on Curved Paths: The Fastest Path between Two Points Is Not a Straight-Line", in Proc. 15th Australian Joint Conference on Artificial Intelligence Canberra, pp 225-236 , Dec 2-6 2002

[2] A. Melingui, T. Chettibi, R. Merzouki,J.B. Mbede, "Adaptive navigation of an omni-drive autonomous mobile robot in unstructured dynamic environments" in IEEE International Conference on Robotics and Biomimetrics, pp 1924-1929 , Dec 12-14, 2013

[3] Liu Peng-yu, He Yong-yi, "Omni-directional opera robot motion planning based on wavelet", in International Conference on Artificial Intelligence and Education, pp 525-528, Oct 29-30 2010

[4] H.R. Moballegh, P. Amini, Y. Pakzad, M. Hashemi, M. Nanniani, "An improvement of self-localization for omnidirectional mobile robots using a new odometry sensor and omnidirectional vision", in Canadian Conference on Electrical and Computer Engineering, pp 2337-2340 Vol 4, May 2-5 2014

[5] Paril Jain, Dhruv Kakadiya, M. Chauhan, A Mecwan, D.K. Kothari, "Navigation System for wheeled robots", Journal of Engineering Science and Management, Vol 7, no. 1, Jan-March 2014

[6] Eui-Jung Jung, Ho Yul Lee, Jae Hoon Lee, Byung-ju Yi, "Navigation of an omni-directional mobile robot with active caster wheels", in ICRA 2008, pp 1659-1665, May 19-23 2008

[7] S. Amagai, T Tsuji, J Samuel, H Osumi, " Control of Omni-directional mobile platform with four driving wheels using torque redundancy", in International Conference on Intelligent Robots and Systems, pp 1996-2002, Sept 22-26, 2008

[8] S. Ziaie-Rad, F. Janabi-Sharifi, M. M. Danesh-Panah, A. Abdollahi, " A practical approach to control and self-localization of Persia omni directional mobile robot", in International Conference on Intelligent Robots and Systems, pp 3473-3479, Aug 2-6 2005

[9] J. Borenstein, Liqiag Feng, " Measurement and correction of systematic odometry errors in mobile robots", pp 869-880, Dec 1996

[10] " PID controller ", http://en.wikipedia.org/wiki/PID_controller, August 20, 2014