

On Differential Drive Robot Odometry with Application to Path Planning

Evangelos Papadopoulos and Michael Misailidis

Abstract— Localization and path planning for obstacle avoidance are two fundamental aspects of mobile robots navigation. In this paper, we improve the localization ability of a robot through odometry, and present and extend a path planning method for such robots. To improve the odometry accuracy of the robot, we propose a new odometry calibration method, and we evaluate the replacement of the differential drive robot caster with an omniwheel. A path planning method is implemented which yields path planning with simultaneous obstacle avoidance, with extended applicability. The odometry improvement method is applied to a differential drive mobile robot but it can be also applied to any other mobile robot if the corresponding kinematic model is used. Experiments show that the calibration method yields improved results.

I. INTRODUCTION

INDUSTRIAL robots have been used by industry during the last five decades, demonstrating their usefulness. However, most of these have their base fixed, and hence a limited workspace. Mobile robots represent an evolution of these, since they can move freely in dynamic environments, see for example Fig. 1. The proliferation of mobile robots is subjected to tackling two major problems. The first problem is localization, i.e. where the robot is at some time instance, while the second is its path planning with obstacle avoidance, i.e. how to reach a destination avoiding collision with obstacles.



Figure 1. Differential drive mobile robot.

Manuscript received October 16, 2006

E. Papadopoulos is with the Department of Mechanical Engineering, National Technical University of Athens (NTUA), Greece (corresponding author, phone: +(30) 210-772-1440; fax: +(30) 210-772-1455; e-mail: egpapado@central.ntua.gr).

M. Misailidis is with the Department of Mechanical Engineering, National Technical University of Athens, Greece (e-mail: m_misailidis@yahoo.gr).

During the past few years many suggestions have been made to address the localization problem. One of the first methods introduced, and still used mainly as a subsidiary method in many projects, is odometry. Its advantages are the low cost of the sensors needed, its good accuracy for short distances, and its compatibility with other positioning methods. Up to date, two main efforts have been made in order to improve the odometry accuracy. For many years, the only available method for odometry calibration was the UMBmark, proposed by Borenstein and Feng, [3], [4]. According to this method, developed for differential drive robots, the robot moves along a rectangular shape path two times, one clockwise and one counterclockwise. At the end of each path, the distance from the initial position is measured, and the odometry parameters are corrected accordingly. The advantages of the method are its simplicity and the fact that there is no need for additional sensors. One of its disadvantages is the use of a path, which consists only of straight paths and on-the-spot rotations. Another disadvantage is that the initial and final positions of the robot must coincide. Also, it is assumed that the mean value of the wheel radius is considered known. This results in an uncertainty about the length of the traveled path.

Aiming at a correction of odometry errors, a new method, called the PC-Method, has been proposed by Hod, Choset and Kyun. This method can be used for systematic and non systematic odometry errors, [7]. According to it, the mobile robot moves along a path and its position is estimated by both the odometry and another localization method. The odometry model is calibrated so that the difference between the estimated position with odometry and the other localization method is the least. The advantage of this method is its great accuracy and the fact that it is not an end point localization method as the UMBmark, but it uses all trajectory points. A drawback of this method is the necessity of employing additional sensors and the use of a closed path with the same initial and final point as in the UMBmark method.

Concerning the path planning with obstacle avoidance problem, several researchers have proposed various methods. Jacobs and Canny have proposed the design of paths as a combination of arcs and straight lines, [5]. Mirtich and Canny developed a method that keeps a robot at the maximum distance from obstacles and considers the nonholonomic constraint of mobile robots, [6].

C. Schlegel has developed a method that takes into account kinematic constraints and the dynamics of the robot and which achieves velocities up to 1m/s, [1]. Quinlan and Khatib developed a method that produces smooth

trajectories and can be used for obstacle avoidance, [11]. Fox, Burgard and Thrun have proposed a method that considers a robot's velocity and allows its motion at high speeds, [2].

Finally, Philippsen and Siegwart have developed a method that is based on the combination of DWA (Dynamic Window), elastic band and NF1, [10]. The method performs smooth motions efficiently, both computationally and in the sense of goal-directedness.

In this paper, we propose a method to improve the localization ability of a robot that relies on odometry and improve a method of path planning in order to become more flexible and easily applicable. To address the localization problem, we first apply an odometry calibration method. According to that method, we integrate odometry errors throughout a path traveled by the robot and we produce new improved odometry parameters. The method employed is an end point method with different initial and final points, which does not need additional sensors. An important advantage of the method is that it can fit to any odometry model.

The accuracy of the proposed method is similar to the one achieved by PC-Method, but in contradiction to it, it does not use sensors other from the wheel encoders. Our studies showed that to some extent, odometry errors are due to the caster wheel used in mobile robots. Therefore, we examine the influence of the caster to the odometry errors. We consider the caster as a systematic odometry error source, and replace it by an omnidirectional wheel. Comparison results are presented.

Next, the path planning method proposed by Papadopoulos and Poulakakis is studied. This method takes into account the workspace obstacles, and the nonholonomic constraint of the differential drive mobile robot to produce a smooth path. Up to now the path was determined by the initial and final points of the path. The method proposed here, allows the definition of intermediate points, which makes the resulted path shorter, more natural, while avoiding the need to have the robot stop at intermediate path points.

II. ODOMETRY CALIBRATION METHOD

In this section, we first estimate the odometry parameters of an experimental mobile robot. The robot we employ is a Pioneer 3 DX differential drive robot, shown in Fig. 1. This robot has two independently driven wheels with tires and a caster wheel for stability. The driven wheels are equipped with encoders and the angular readings become available through simple routine calls.

The three major odometry parameters of such a mobile robot are the radius of its right and left wheels, R_r and R_l respectively and the distance D between its wheels, see Fig. 2. In order to estimate these parameters, we express the velocity of the robot in terms of these parameters as well as of the angular velocities of its right and left wheels, ω_r and ω_l respectively. Odometry is based on the integration of the following kinematic equations:

$$\begin{aligned}\dot{x} &= V_s \cdot \cos\theta \\ \dot{y} &= V_s \cdot \sin\theta \\ \dot{\theta} &= \omega\end{aligned}\quad (1)$$

where,

$$\begin{aligned}V_s &= \frac{\omega_r \cdot R_r + \omega_l \cdot R_l}{2} \\ \omega &= \frac{\omega_r \cdot R_r - \omega_l \cdot R_l}{D}\end{aligned}\quad (2)$$

The parameters R_r and R_l are the actual radii of the right and left wheel respectively, and D the actual distance between the center of the wheels. Integrating (1) yields the position and orientation of the mobile robot. However, the results depend heavily on the values of the parameters in (1). If these are not known accurately, then large and growing estimation errors result.

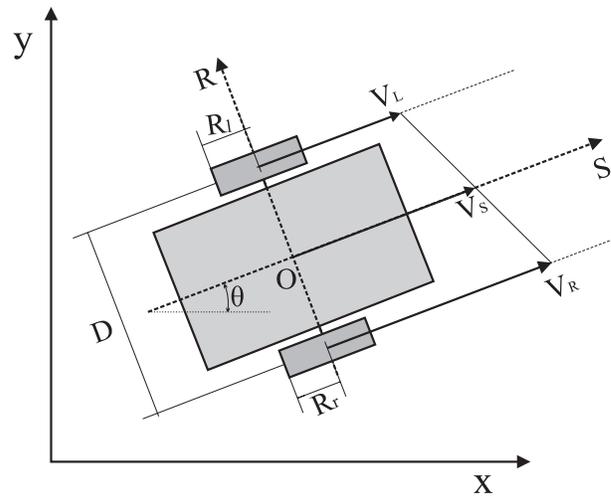


Figure 2. Mobile manipulator schematic and variable definitions.

Denoting by \hat{R}_r , \hat{R}_l , and \hat{D} the estimates of the corresponding parameters, the errors in these are defined as,

$$\begin{aligned}\delta R_r &= \hat{R}_r - R_r \\ \delta R_l &= \hat{R}_l - R_l \\ \delta D &= \hat{D} - D\end{aligned}\quad (3)$$

Using (1) and the errors in (3), the errors in linear and angular velocities due to parameter estimation errors are given by,

$$\begin{aligned}\delta \dot{x} &= \frac{\partial(V_s \cdot \cos\theta)}{\partial R_r} \delta R_r + \frac{\partial(V_s \cdot \cos\theta)}{\partial R_l} \delta R_l + \frac{\partial(V_s \cdot \cos\theta)}{\partial D} \delta D \\ \delta \dot{y} &= \frac{\partial(V_s \cdot \sin\theta)}{\partial R_r} \delta R_r + \frac{\partial(V_s \cdot \sin\theta)}{\partial R_l} \delta R_l + \frac{\partial(V_s \cdot \sin\theta)}{\partial D} \delta D \\ \delta \dot{\theta} &= \frac{\partial \dot{\theta}}{\partial R_r} \delta R_r + \frac{\partial \dot{\theta}}{\partial R_l} \delta R_l + \frac{\partial \dot{\theta}}{\partial D} \delta D\end{aligned}\quad (4)$$

or, in matrix form,

$$\begin{bmatrix} \delta \dot{x} \\ \delta \dot{y} \\ \delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial(V_s \cdot \cos \theta)}{\partial R_r} & \frac{\partial(V_s \cdot \cos \theta)}{\partial R_l} & \frac{\partial(V_s \cdot \cos \theta)}{\partial D} \\ \frac{\partial(V_s \cdot \sin \theta)}{\partial R_r} & \frac{\partial(V_s \cdot \sin \theta)}{\partial R_l} & \frac{\partial(V_s \cdot \sin \theta)}{\partial D} \\ \frac{\partial \dot{\theta}}{\partial R_r} & \frac{\partial \dot{\theta}}{\partial R_l} & \frac{\partial \dot{\theta}}{\partial D} \end{bmatrix} \begin{bmatrix} \delta R_r \\ \delta R_l \\ \delta D \end{bmatrix} \quad (5)$$

Setting,

$$\dot{B} = \begin{bmatrix} \delta \dot{x} \\ \delta \dot{y} \\ \delta \dot{\theta} \end{bmatrix} \quad (6)$$

$$\dot{A} = \begin{bmatrix} \frac{\partial(V_s \cdot \cos \theta)}{\partial R_r} & \frac{\partial(V_s \cdot \cos \theta)}{\partial R_l} & \frac{\partial(V_s \cdot \cos \theta)}{\partial D} \\ \frac{\partial(V_s \cdot \sin \theta)}{\partial R_r} & \frac{\partial(V_s \cdot \sin \theta)}{\partial R_l} & \frac{\partial(V_s \cdot \sin \theta)}{\partial D} \\ \frac{\partial \dot{\theta}}{\partial R_r} & \frac{\partial \dot{\theta}}{\partial R_l} & \frac{\partial \dot{\theta}}{\partial D} \end{bmatrix} \quad (7)$$

$$X = \begin{bmatrix} \delta R_r \\ \delta R_l \\ \delta D \end{bmatrix} \quad (8)$$

and using (2), matrix \dot{A} is written as,

$$\dot{A} = \begin{bmatrix} \frac{\omega_r}{2} \cos \theta - V_s \sin \theta \frac{\theta_r}{D} & \frac{\omega_l}{2} \cos \theta + V_s \sin \theta \frac{\theta_l}{D} & V_s \sin \theta \frac{\theta}{D} \\ \frac{\omega_r}{2} \sin \theta + V_s \cos \theta \frac{\theta_r}{D} & \frac{\omega_l}{2} \sin \theta - V_s \cos \theta \frac{\theta_l}{D} & -V_s \cos \theta \frac{\theta}{D} \\ \frac{\omega_r}{D} & -\frac{\omega_l}{D} & -\frac{\omega}{D} \end{bmatrix} \quad (9)$$

We assume that the actual values of R_r , R_l and D are constant, thus X is fixed. Then, arithmetic integration with time along a path yields:

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta \theta \end{bmatrix} = A \cdot \begin{bmatrix} \delta R_r \\ \delta R_l \\ \delta D \end{bmatrix} \quad (10)$$

or,

$$B = A \cdot X \quad (11)$$

Equation (11) describes the fact that small variations in the wheel radius R_r , R_l and the distance D between the wheel centers result in errors in estimated robot position and orientation.

Exploiting the above analysis, the robot is commanded to move along a path, while its wheel rotational velocities $\omega_r(t)$, and $\omega_l(t)$ are recorded. In addition, the robot's final position and orientation is measured with respect to the global coordinate system. Next, the robot's wheel rotational

velocities are imported in a robot simulator software, such as the one provided by ActivMedia, and the expected position and orientation of the robot is found. In this way, the vector B is found. The array A is subsequently calculated from $\omega_r(t)$ and $\omega_l(t)$. Then, the vector X is computed from (11), by inverting matrix A . In this way we obtain a better estimation of R_r , R_l and D .

In order to reduce the influence of random errors, we employ the above procedure more than once. Each time the robot travels a different path and we get different arrays A_i and B_i . Then (11) becomes:

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} \cdot X \Rightarrow B = A \cdot X \quad (12)$$

To solve (12), we use the pseudoinverse of A , which solves (12) with the least squares method. Since A is invertible, this method always yields a solution for the unknown parameter errors X . Finally, we obtain the real values of the robot parameters through equations

$$\begin{aligned} R_r &= R_r + \delta R_r \\ R_l &= R_l + \delta R_l \\ D &= D + \delta D \end{aligned} \quad (13)$$

Because of the fact that the difference between real and estimated values of R_r , R_l and D is not small enough, we have to employ the previous method repeatedly until R_r , R_l and D converge to certain values which we consider the real values of the parameters.

Another problem that has to be faced is the uncertainty about the parallelism of robots longitudinal axis and x-axis of the global coordinates system. This uncertainty is modeled with an additional unknown parameter ϕ_o , which represents the initial angle between the robots longitudinal axis and the x-axis of the global coordinates system. The angle ϕ_o has to be found and its nominal value is zero. (1) becomes:

$$\begin{aligned} \dot{x} &= V_s \cdot \cos(\theta + \phi_o) \\ \dot{y} &= V_s \cdot \sin(\theta + \phi_o) \\ \dot{\theta} &= \omega \end{aligned} \quad (14)$$

and array A :

$$A = \begin{bmatrix} \frac{\omega_r}{2} \cos(\theta + \phi_o) - V_s \sin(\theta + \phi_o) \frac{\theta_r}{D} & \frac{\omega_l}{2} \cos(\theta + \phi_o) + V_s \sin(\theta + \phi_o) \frac{\theta_l}{D} \\ \frac{\omega_r}{2} \sin(\theta + \phi_o) + V_s \cos(\theta + \phi_o) \frac{\theta_r}{D} & \frac{\omega_l}{2} \sin(\theta + \phi_o) - V_s \cos(\theta + \phi_o) \frac{\theta_l}{D} \\ \frac{\omega_r}{D} & \frac{\omega_l}{D} \\ V_s \sin(\theta + \phi_o) \frac{\theta}{D} & -V_s \sin(\theta + \phi_o) \frac{\theta}{D} \\ -V_s \cos(\theta + \phi_o) \frac{\theta}{D} & V_s \cos(\theta + \phi_o) \frac{\theta}{D} \\ \frac{\omega}{D} & 0 \end{bmatrix}$$

Equation (11) becomes:

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta \theta \end{bmatrix} = A \cdot \begin{bmatrix} \delta R_r \\ \delta R_l \\ \delta D \\ \delta \phi_o \end{bmatrix} \tag{15}$$

and is used in obtaining an equation of the form of (12). The solution for the errors is as above.

The method described above has a number of advantages when compared to the UMBmark and the PC-methods. Firstly, the proposed method allows the calculation of the exact values of each robot wheel whereas with the UMBmark method we can find only their ratio, resulting in an uncertainty in the estimation of the distance traveled. Another advantage of the method described here, is that it does not require a particular shape of path like the UMBmark. Instead, it can use paths with the same characteristics as the ones that the robot will follow in its designated use. Finally an important advantage of the method versus the UMBmark, is that it can be implemented in every mobile robot whose kinematic model is known. An advantage of the method versus the PC-method is that it does not require the employment of additional sensors able to estimate accurately the robot's position over the whole path traveled. This is due to the end-point type of the method employed.

III. CASTER REPLACEMENT WITH OMNIWHEEL

In the previous section, we assumed that the difference between the real and the nominal values of R_r , R_l and D are sources of systematic errors and we modeled the robot's motion so as to find a better estimation of these values. In this section, we examine the caster wheel and assume that it is a source of systematic errors for reasons described below. As a complete modeling of the caster wheel would be a very complex task, instead we propose its replacement with an omniwheel. Finally, we discuss the advantages and drawbacks of using an omniwheel in a mobile robot. After careful observation and analysis, it became evident that the

systematic odometry errors originating from the use of a caster, appear due to the fact that the caster revolution axis (not the wheel axis) is not absolutely vertical as it is supposed to be, see Fig. 3.

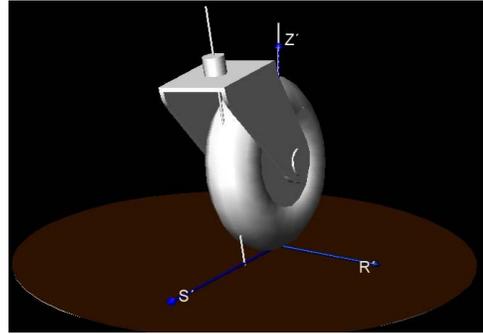


Figure 3. Due to constructive inaccuracies the caster axis is not vertical to the ground.

The reaction F of the ground to the wheel is analyzed in two components F_1 and F_2 , see Fig. 4. F_2 is parallel to the caster axis where as F_1 is perpendicular to it and tends to revolve the caster. As soon as the caster turns, a friction force T appears, see Fig. 5, which tends to make the caster parallel to the movement plane. Although small, this force T influences the robot throughout its whole path. The magnitude and direction of T depends on the angle between the caster plane and the vertical plane which passes through the caster axis.

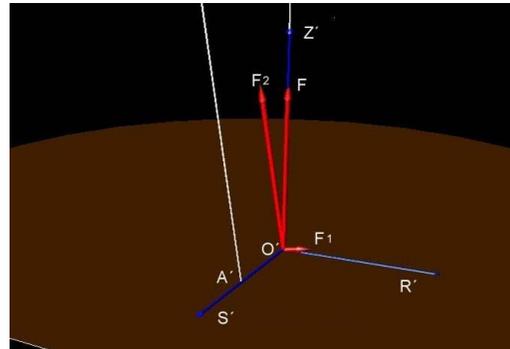


Figure 4. The reaction F of the ground is decomposed to a parallel and vertical force.

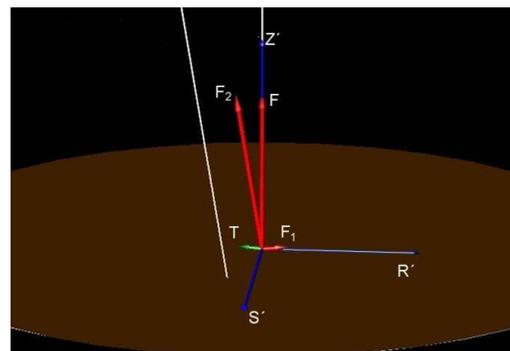


Figure 5. The vertical force tends to rotate the caster and friction T appears in order to make caster's plane perpendicular to movement direction.

Because modeling this influence is a complex task with uncertain practical gains, we decided to replace the caster with an omniwheel, i.e. a wheel that can be moved in directions parallel to its wheel axis by rolling and without sliding, see Fig. 6.

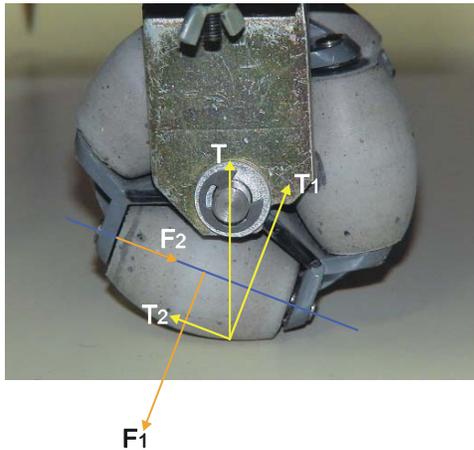


Figure 6. The ground reaction component pushes the barrel to the left.

The main advantage of the omniwheel is that it acts as a support point with a negligible shift and that the effects described above are minimized.

Another significant advantage is that the disturbance forces caused by the omniwheel are random and consequently the resulting errors are non-systematic and therefore less severe for odometry.

The main disadvantage of the particular omniwheel used was its low quality. Firstly, its projection was not completely circular. Secondly, its barrels were allowed to move along its longitudinal axis. This movement was quite rough and occurred when the barrel's axis changed inclination.

This is demonstrated in Figs. 6 and 7. When the inclination of the barrel's axis is as shown in Fig. 6, the component T_2 of the reaction of the ground pushes the barrel on the left until it reaches the left edge. When the omniwheel rotates, the inclination of the barrel's axis changes, and a force F_2 is needed to counterbalance the new force T_2 .

This force cannot be exerted until the barrel reaches its right edge and therefore the axis of the barrel, and along with it the whole omniwheel, slips on the barrel until the barrel contacts the right edge.

These wheel imperfections cause disturbance forces that oppose odometry accuracy. However, they appear only when the inclination of the barrel's axis changes and not continuously.

In addition, their appearance is rather random unlike the forces that appear due to the caster's imperfections, which influence the robot throughout its entire motion and in a systematic way.

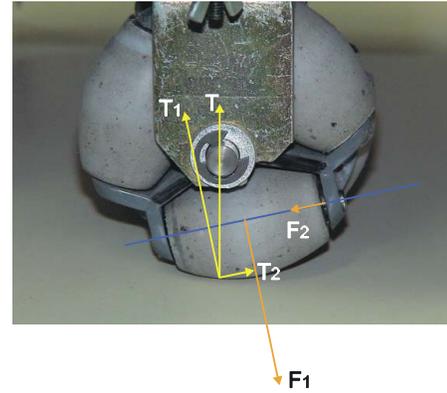


Figure 7. When the omniwheel rotates, T_2 changes direction and pushes the barrel to the right.

IV. EXPERIMENTAL RESULTS

Using a Pioneer 3 DX mobile robot, we conducted four groups of measurements aiming at (a) the calibration of the odometry, (b) the calculation of R_r , R_l and D , (c) the evaluation of the odometry improvement achieved by calibration and by the use of the omniwheel.

Each group of measurements included following five different trajectories for the calculation of R_r , R_l and D , and a final trajectory for the evaluation of the obtained accuracy. In all five calibration trajectories, the robot was placed at exactly the same initial location and with the same initial orientation on a smooth surface. At the end of each trajectory, the final position and orientation of the robot was marked and the differences δx , δy and $\delta \theta$ were calculated. The calibration trajectories were of arbitrary shape and length. After all the five trajectories were executed the errors δR_r , δR_l , δD and $\delta \phi_o$ as well as the corresponding parameters R_r , R_l , D and ϕ_o , were calculated with the method described in section II.

To evaluate the accuracy of the obtained parameters, the error between the real and the estimated position of the robot was calculated using the following equation,

$$d = \sqrt{(x_{real} - x_{estimated})^2 + (y_{real} - y_{estimated})^2} \quad (16)$$

In the two first groups of measurements, the caster wheel was used, while in the third and fourth group, the omniwheel replaced the caster wheel. Also, in the first and third group, the path's radii of curvature was greater than one meter. In the second and fourth group the path included parts of smaller radii of curvature and on-the-spot rotations.

To evaluate the accuracy of the obtained parameters we had the robot travel a long path of at least 120 m. At the end of the path the distance between the real position of the robot and the position estimated by odometry before and after the calibration was calculated. This is shown in Table I.

Looking only at the second column of Table I, the following conclusions can be drawn. By comparing the results of the first to the second group, and those of the third to the fourth one, we conclude that decreasing the curvature radii of the path, increases the resulting odometry errors. In

our experiments, the error increases drastically when the radii of curvature are smaller than 1 m. Furthermore, comparing one and three, or two and four, in Table I, we can conclude that the use of an omniwheel improves the accuracy of the robot's odometry.

Table I. Average odometry experimental results.

Group No	Odometry error before calibration (cm)	Odometry error after calibration (cm)	Average Path length (m)	Notes
1	35.59	4.17	126.5	$r > 1$ m, caster
2	60.87	44.48	124.9	$r < 1$ m, caster
3	16.10	7.44	126.2	$r > 1$ m, omniwheel
4	37.03	14.14	124.8	$r < 1$ m, omniwheel

As far as the calibration method is concerned, a comparison between the second and third column in Table I shows that odometry improvement is achieved in all cases. However, when comparing the first to the second group, we notice that odometry improvement is much better for the first group, i.e. when the radius of curvature is greater than 1 meter. Taking into account that these two paths have been traveled with the caster, we conclude that odometry errors caused by the caster are much more prevalent when the radius of curvature is small. This is to be expected, since according to the explanation given in Section III, the variations of the force F_1 are bigger when the caster turns for a wider angle. Also, if we compare the third to the fourth group, we see that the odometry error in both cases is about half the error without calibration.

The inability of the calibration method to improve the accuracy of the odometry in the 2nd, 3rd, and 4th group beyond the accuracy obtained for the 1st group, is attributed to non-systematic errors, especially in the case of the 3rd and 4th group.

V. PATH PLANNING METHOD IMPROVEMENT

Apart from the localization challenge that we discussed above, one has to deal with the path planning problem. A method has been developed by [9] which is fast in computing point-to-point trajectories that can avoid obstacles.

However, one disadvantage of the method is that in some cases, the paths that occur can be very long. This problem can be tackled by using intermediate points from which the path should pass. If such points were used, the robot should stop at these before resuming its motion. Obviously, this was a drawback that needed to be rectified. Here, we suggest an improved version of the method in [9], which allows the creation of more flexible paths with continuous speeds.

The original path planning method is based on the transformation of the nonholonomic constraint of the differential drive robots

$$\sin\theta \cdot dx - \cos\theta \cdot dy = 0 \quad (17)$$

to the equation

$$du + vdw = 0 \quad (18)$$

where u , v , w are functions of x , y and θ . This can be achieved with the following invertible transformation:

$$u(x, y, \theta) = x \cdot \sin\theta - y \cdot \cos\theta \quad (19)$$

$$v(x, y, \theta) = -x \cdot \cos\theta - y \cdot \sin\theta \quad (20)$$

$$w(x, y, \theta) = \theta \quad (21)$$

The idea here is that the planning is done in the u - v - w space, without the need to tackle the nonholonomic constraint, and then the result is converted back to the x - y - θ space.

Function w is selected to be some time function, while u and v are functions of w , as is shown below

$$w = f(t) \quad (22)$$

$$u = g(w) \quad (23)$$

$$v = -\frac{du}{dw} = -g'(w) \quad (24)$$

Notice that due to (24), this selection automatically satisfies the nonholonomic constraint. Function f is selected to be polynomial function and its coefficients are determined by the boundary conditions:

$$\begin{aligned} f(t_o) &= w_o = \theta_o = \theta(t_o) \\ f'(t_o) &= \dot{w}_o = \dot{\theta}_o = \dot{\theta}(t_o) \\ f''(t_o) &= \ddot{w}_o = \ddot{\theta}_o = \ddot{\theta}(t_o) \\ f(t_f) &= w_f = \theta_f = \theta(t_f) \\ f'(t_f) &= \dot{w}_f = \dot{\theta}_f = \dot{\theta}(t_f) \\ f''(t_f) &= \ddot{w}_f = \ddot{\theta}_f = \ddot{\theta}(t_f) \end{aligned} \quad (25)$$

where the subscript o indicates the beginning and f the end of the path. Given the above boundary conditions, f must be a fifth order polynomial function, of the form,

$$f(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (26)$$

The boundary conditions for g are:

$$\begin{aligned} g(w_o) &= u_o = x_o \cdot \sin\theta_o - y_o \cdot \cos\theta_o \\ g'(w_o) &= -v_o = x_o \cdot \cos\theta_o - y_o \cdot \sin\theta_o \\ g(w_f) &= u_f = x_f \cdot \sin\theta_f - y_f \cdot \cos\theta_f \\ g'(w_f) &= -v_f = x_f \cdot \cos\theta_f - y_f \cdot \sin\theta_f \end{aligned} \quad (27)$$

and g therefore is a third order polynomial:

$$g(w) = b_3 w^3 + b_2 w^2 + b_1 w^1 + b_0 w \quad (28)$$

This method allows the use of intermediate points, under the assumption that at the end of each sub-path, the vehicle angular velocity is zero. This results in a zero robot translational velocity, as is shown beneath.

After differentiation, (19) and (20) yield

$$\dot{x} = \cos w \cdot \dot{w} \cdot u + \sin w \cdot \dot{u} + \sin w \cdot \dot{w} \cdot v - \cos w \cdot \dot{v}$$

$$\dot{y} = \sin w \cdot \dot{w} \cdot u - \cos w \cdot \dot{u} - \cos w \cdot \dot{w} \cdot v - \sin w \cdot \dot{v}$$

and using (18) we get

$$\dot{x} = \cos w \cdot \dot{w} \cdot u - \cos w \cdot \dot{v}$$

$$\dot{y} = \sin w \cdot \dot{w} \cdot u - \sin w \cdot \dot{v}$$

Replacing w from (21)

$$\begin{aligned} \dot{x} &= \cos \theta \cdot \dot{\theta} \cdot u(\theta) - \cos \theta \cdot v'(\theta) \cdot \dot{\theta} \\ \dot{y} &= \sin \theta \cdot \dot{\theta} \cdot u(\theta) - \sin \theta \cdot v'(\theta) \cdot \dot{\theta} \end{aligned} \quad (29)$$

As shown by (29), setting $\dot{\theta}$ to zero, also results in a zero translational velocity of the robot. If it is not acceptable to have the robot stop at intermediate points, then it must be ensured that there is continuity of the robot's translational velocity.

Next, we consider the motion from one point to another, with the requirement of passing through a number of intermediate points without a stop. The resulting path is comprised of a number of sub-paths whose end-points are the desired intermediate points. If we examine the problem variables, we can observe that the continuity of $\theta(t)$, $\dot{\theta}(t)$ and $u(\theta)$ can be ensured by the use of proper boundary conditions. In order to ensure the continuity of v' , another boundary condition must be added to (27) so that the initial value of v' at the $i+1$ sub-path is the equal to the final value of v' at the i^{th} part of the path, i.e.,

$$\begin{aligned} v'_{o_{-(i+1)}} &= v'_{f_{-i}} = (-g'_i(w_{f_{-i}}))' = \\ &= -g''_i(w_{f_{-i}}) = -g''_i(\theta_{f_{-i}}) = -g_d \end{aligned} \quad (30)$$

But,

$$\begin{aligned} v'_{o_{-(i+1)}} &= (-g'_{i+1}(w_{o_{-(i+1)}}))' = \\ &= -g''_{i+1}(w_{o_{-(i+1)}}) = -g''_{i+1}(\theta_{o_{-(i+1)}}) \end{aligned} \quad (31)$$

Therefore, the additional boundary condition is

$$g''(w_o) = g_d = -v'_{f_{-i}} \quad (32)$$

and hence, the order of the g function is increased to four:

$$g(w) = b_4 w^4 + b_3 w^3 + b_2 w^2 + b_1 w^1 + b_0 w$$

VI. SIMULATION RESULTS

The implementation of the extended method is simulated here. The robot commanded to move from its initial position $(x,y,\theta) = (0,0,0)$ to a final position $(90,10, 3\pi/2)$. Employing the original method, the robot travels the path shown in

Figure 8 and reaches its destination. Although this path may have a shape that is undesirable, there is no way to change it. To modify the path's shape, two intermediate points $(x,y) = (35,20)$ and $(70,30)$ are inserted and applying the extended method presented above, the path shown in Figure 9 is obtained. This is smoother and very different from the one in Fig. 8. Therefore, the extended method offers better control over the shape of the mobile robot trajectory.

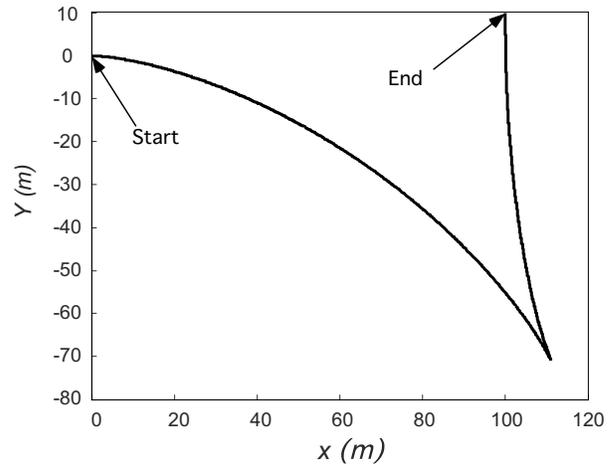


Figure 8. Robot's path without intermediate points.

Figs. 10 and 11 show the time evolution of the robot orientation with the original and extended methods. It can be seen that the disadvantage of the original method is that the function f and consequently the robot orientation θ are defined exclusively by the initial and final orientation of the robot regardless of the velocity of the robot or even of the obstacles in the surrounding space. This problem is alleviated by the proposed extended method.

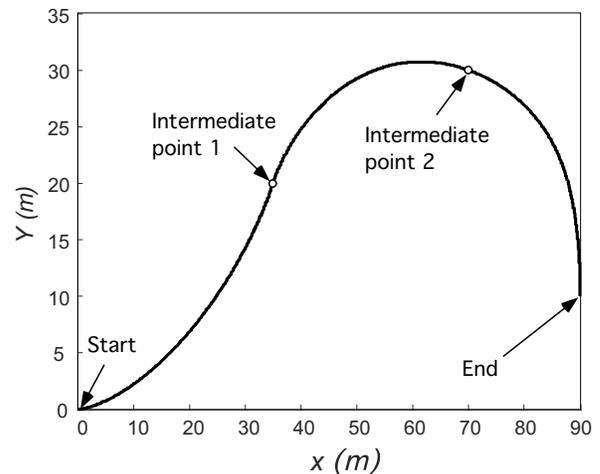


Figure 9. Robot's path with two intermediate points.

The proposed method can also be used to improve the obstacle avoidance capabilities of the original method. To this end, the procedure proposed in [9] can be followed but appended with some intermediate points that will prevent the generated paths from having undesired shapes.

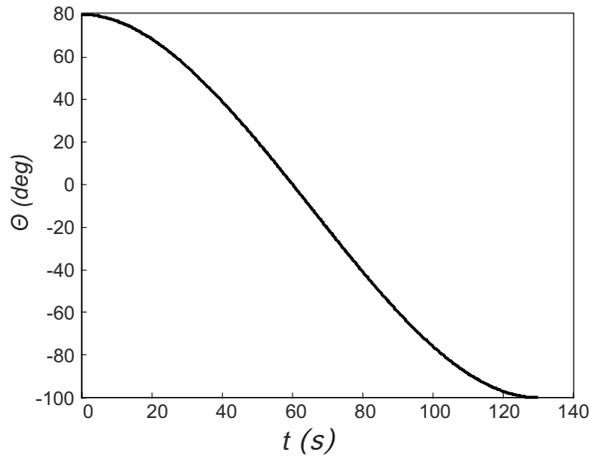


Figure 10. Robot orientation history. Without intermediate points, the robot orientation is a monotonous function.

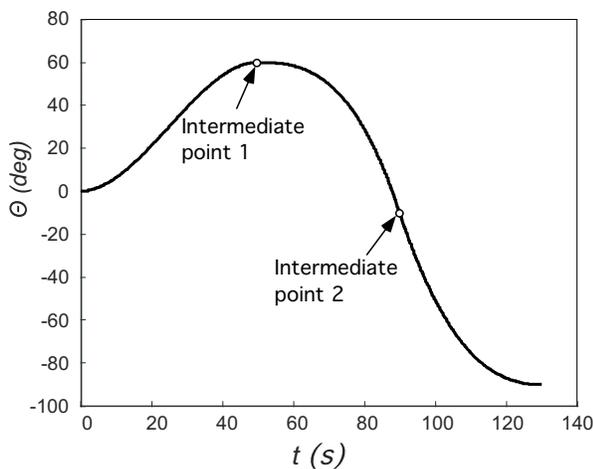


Figure 11. Robot orientation history with intermediate points.

Another point that must be noted is that not all of the boundary equations (25) have to be used in every path. For example, if we are not interested in the angular velocity and acceleration of the robot at an intermediate point, the two last equations of (25) do not need to be used.

In conclusion, the proposed extension makes the method more flexible, allowing the definition of intermediate points from which the robot must pass, and yielding smoother and shorter trajectories.

VII. CONCLUSIONS

In this paper, techniques for differential drive robots were developed. First, the localization accuracy of such robots employing odometry was considered. To address the localization problem, an odometry calibration method was used. The odometry errors were integrated along the entire path and new improved odometry parameters were calculated. An end-point method, with different initial and final points was used while no sensors beyond encoders were used. An important advantage of the proposed method is that it can fit any odometry model. The influence of the caster to odometry errors was also studied, and it was found

that in general, the omniwheel yields improved accuracy.

As far as the path planning problem is concerned, we extended the method proposed in [9] so that it can accept intermediate points without bringing the robot to a stop at these. The result was a path shape that is more controllable and avoids long robot excursions.

REFERENCES

- [1] Schlegel, C., "Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot", *Proceedings IEEE/RJS International Conference on Intelligent Robots and Systems*, Victoria, B. C. Canada, 1998, pp. 594-599.
- [2] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine*, 4(1):23-33, March 1997.
- [3] j. Borenstein, L. Freg, "Correction of Systematic Odometry Errors in Mobile Robots", *Proc. International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, August 5-9, 1995, pp. 569-574.
- [4] J. Borenstein, L. Freg, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots", *IEEE Transactions on Robotics and Automation*, Vol 12, No 6, December 1996, pp. 869-880.
- [5] Jacobs P., Canny J., "Planning Smooth Paths for Mobile Robots", *Proc. International Conference on Robotics and Automation*, Scottsdale, AZ, 1989, pp. 2-7.
- [6] Mirtich B. and Canny J., "Using Skeleton for Nonholonomic Motion Planning among Obstacles", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nice, France, May 1992, pp. 2533-2540.
- [7] Nakju Doh, Howie Choset, Wan Kyun Chung, "Accurate Relative Localization Using Odometry", *Proc. International Conference on Robotics and Automation*, Taipei, Taiwan, September 14-19, 2003.
- [8] Papadopoulos E. G., Poulakakis I., "Planning and Model-Based Control for Mobile Manipulators", *Proc. International Conference on Intelligent Robots and Systems (IROS '00)*, Kagawa University, Takamatsu, Japan, October 30 - November 5, 2000, pp. 1810-1815.
- [9] Papadopoulos E. G., Poulakakis I., Papadimitriou I., "On Path Planning and Obstacle Avoidance for Nonholonomic Platforms with Manipulators: A Polynomial Approach", *The International Journal of Robotics Research*, Vol. 21, No. 4, April 2002, pp. 367-383.
- [10] Roland Philippsen and Roland Siegwart, "Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot", *Proc. International Conference on Robotics and Automation*, (ICRA 2003), Taipei, Taiwan, September 14-19, 2003, pp 446-451.
- [11] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," *Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, GA, 1993, pp. 802-807.