2015 IEEE 10th International Conference on Industrial and Information Systems, ICIIS 2015, Dec. 18-20, 2015, Sri Lanka

Simplified Controller for Three Wheeled Omni Directional Mobile Robot

G. I. R. K. Galgamuwa, L. K. G. Liyanage, M. P. B. Ekanayake *MIEEE,* B. G. L. T. Samaranayake *SMIEEE*

Department of Electrical and Electronics Engineering
University of Peradeniya
Peradeniya, Sri Lanka
isuru.r.g@ieee.org, l.kaveen.liyanage@ieee.org mpb.ekanayake@ee.pdn.ac, lilantha@ee.pdn.ac.lk

***Abstract*— This paper presents the work on developing a controller for the three wheeled omni-directional mobile robot for the operation on a flat terrain. Central to a robot base are omni-directional wheels. In addition to providing traction normal to the rotor axis like an ordinary wheel, they are capable of sliding parallel to the rotor axis without much friction. We have designed and implemented a robot utilizing three omni-directional wheels mounted 120^{O} apart. Trajectory of the robot is specified as the time variation of the position coordinates and the orientation of the robot with respect to a fixed frame. The trajectory is dictated by the speeds of the three omni-directional wheels. Been a holonomic robot, the designed robot can achieve any continuous trajectory. The trajectory is planned on the fixed world frame. Then, it is transformed to the time-varying robot body frame. The velocities of the wheels are computed with respect to the body frame. A PID controller is utilized for the control of each parameter. Consequently, several PID controllers are employed for trajectory control. The MATLAB® SIMULINK® platform is used to develop the kinetic model of the system. The designed robot has been implemented in hardware.**

Keywords*— *Holonomic Robot, Omnidirectional maneuvering, Trajectory control, Cascade control

I. INTRODUCTION

Design and control of omni-directional mobile robots has gained much enthusiasm worldwide as is evident from numerous robot competitions. The tasks of robots in those competitions cover a wide spectrum of challenges ranging from simple line following to more complex tasks such as robot soccer and other "sports for robots" [1]. Despite being a simple architecture, wheeled robots are often used in such competitions due to the convenience of construction and navigation. Moreover, wheeled mobile robots are used for various scientific, domestic and military applications [2, 3].

In such applications, it is often required to maneuver in tight spaces and obstacle avoidance with short range feedback. This requires a high degree of agility granted by being able to move and adjust its orientation simultaneously. This precisely is the advantage of the so called "holonomic robots".

This paper presents the work on developing a three wheeled omni-directional mobile robot, which is a special case of a holonomic robot. This robot is designed for two dimensional (2D) planar motions. The location of the robot on the 2D plane is specified by the (x,y) coordinates and the orientation of the robot is specified by the angle of rotation θ about its center of gravity. This gives rise to three degrees of freedom (DOF). The designed robot can control all three DOFs independently.

In addition to maneuverability in tight spaces, it provides convenient path planning, optimization and programming. Obstacle avoidance and exception handling is convenient with this architecture than the other approaches. For example, a non-holonomic robot will not be able to control orientation and position simultaneously with full agility giving rise to the "parallel parking" problem [2]. However, for a holonomic robot, this problem is a routine task. This type of robot platforms are commercially used in motorized wheel chairs, forklift trucks etc [2, 4].

Omni-directional robots can be further classified according to domain of operations, geometry etc [2]. In addition, from the construction point of view, they can be classified based on the type of wheels used and number of wheels [4]. Omni-directional robots can have three wheels like what is presented in this paper or can have four, five or higher number of wheels. They can be constructed using mecanumTM wheels [5, 6], using caster wheels or using omni-wheelsTM [7] as presented in this paper.

At the same time it should be noted that due to the special wheel arrangement of omni-directional robots, the full motor efficiency cannot be obtained. The reasons for these are explained later in this paper. Furthermore, the omni-directional robots heavily rely on sensors such as position sensors, accelerometer, speed and position encoders for individual motors as well as a cascade of sub-system controllers. However in non-holonomic robots, most of these components will only be auxiliary.

In this paper, the dynamic model of the robot platform is derived consistent with known literature and a simplified control algorithm is proposed considering the practical applications and the limited hardware and computational resources available on-board. The designed system and control algorithm is initially implemented and simulated on MATLAB® SIMULINK® platform and implemented in hardware.

II. OMNIBOT CONSTRUCTION & FORWARD KINEMATICS

The selection of wheels is imperative for the construction of an omni-directional robot. The wheels used should be able to provide traction in the tangential direction similar to a regular wheel. At the same time, the wheels should impose (nearly) no

978-1-4799-1876-8/15/$31.00 ©2015 IEEE

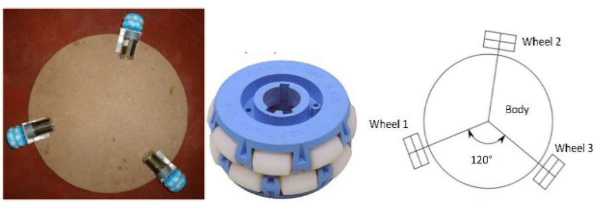
resistance to the axial movement [8]. The mecanum™ wheels and omni-wheels™ are therefore ideal for the construction of an omni-directional robot [5]. Due to the construction, mecanum™ wheels are easier to control and pose less friction in the axial direction compared to omni-wheels™ [5]. But mecanum™ wheels are comparatively more expensive. Therefore, omni-wheels™ were used in this design.

Though it is possible to construct omni-directional robots with asymmetrical structures, symmetrical omni-directional robots are easier to model, plan trajectories and control [9]. The simplest, in terms of structure and component use, symmetrical omni-directional robot is a three-wheeled robot where the wheels are mounted symmetrically 120° apart [4, 9]. So, the center of the wheels will lie on the vertices of an equilateral triangle. Due to this simplicity and ease of construction, a three wheeled symmetrical omni-directional robot was designed. The designed robot was named "OmniBot". The kinematic structure, the hardware construction of the OmniBot and an omni-wheel™ are shown in Fig. 1.

Due to the complex dynamics of the OmniBot, a step-by-step design approach is followed, assessing the performance of the system at each level. Once the geometry and the mechanical structure were finalized, the first step was to assign coordinate systems to the robot. Since it is a mobile robot, the "world frame" (frame 0) is attached to a fixed location on the terrain. Then the "body frame" (frame 1) was attached to the point of intersection of the three rotor axes. Therefore, the origin of frame 1 is attached to the geometric center of the circular base of the robot. The orientation of the robot that will have the axis of the first wheel parallel to the x– axis of the world coordinate frame and the second and third wheels to the right of the first wheel is designated to be the zero orientation. This coordinate frame allocation is shown in Fig. 2.

The next step is to derive the kinematic equations of the robot using its geometry. For the sake of mathematical modeling, the omni-wheel was modeled as free to rotate tangential to the wheel and move without any resistance in the axial direction. Then, the forward kinematics was simulated on a computer using the MATLAB® SIMULINK® platform. The slippage and other perturbations were modeled as a disturbance input to the model. This enabled to develop an intuition of the underlying operation of OmniBot. The basic types of motion of the OmniBot have been classified as 'rotation', 'forward direction' and 'normal direction' and they are summarized in Fig. 3.

Once the kinematic model was completed, the control strategy was developed assuming that the trajectory control problem is a tracking problem. In particular, the desired trajectory was prescribed with respect to the world frame as variation of the two position coordinates and the orientation in time $(x_0(t), y_0(t), \theta_0(t))$. Then it is transformed to the robot frames as a variation of the robot frame in a similar form: $(x_1(t), y_1(t), \theta_1(t))$. Then this specification is transformed to the speed profile of the three drive motors $(\omega_1(t), \omega_2(t), \omega_3(t))$. The rationale is therefore, that if it is possible to track the speed of the three motors it is possible to achieve the desired trajectory. Due to this architecture, the motor speed control loop will be the inner most control loop. It is typically operated using pulse width modulation (PWM) drive with frequency 1 kHz or more. Therefore, this inner control loop would have to be operated at a cycle time of about one tenth of this frequency, i.e, at 10ms.



(a) (b) (c)

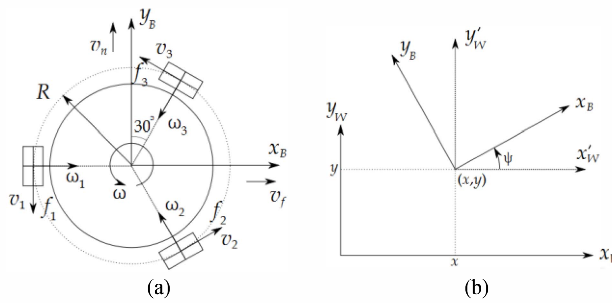
Fig. 1 . (a) The designed mechanical structure of the OmniBot (b) An omni-wheel (c) Kinematic structure of OmniBot

This enables the body-frame loop to be operated approximately ten times slower, i.e., at 100 ms cycle time. This enables the outermost loop to be operated a further ten times slower - in the order of 1 s cycle time. This makes it possible to design controllers based on the "dominant pole" argument [10]. It means that from the point of view of the outer loop, inner loop will be already at the steady state of the action of the previous commands [10, 11]. However, it should be noted that the system is essentially nonlinear. But, considering that the dynamics are "sufficiently slow" and "slowly varying", it is possible to assume a linearized control [11, 12].

We therefore, employ a cascade of Proportional-Integral-Derivative (PID) controllers for each of the control variables at each level. Since each level consists of three control variables, and there are three levels, there are nine PID controllers all together. The control algorithm was then simulated on the familiar MATLAB® SIMULINK® platform. This control architecture simplifies the control algorithm but increases the computational resource requirement. Hence a STM32F4 DISCOVERY processor board is used to implement the control laws of the OmniBot.

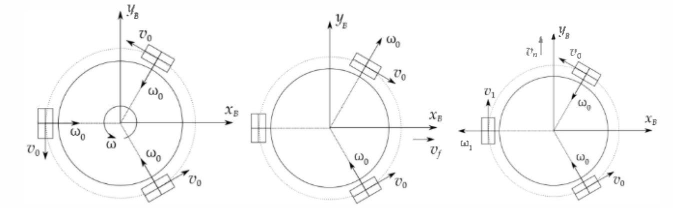
III. Dynamic Model of the OmniBot

Deriving the dynamic model of the OmniBot is the essential for deriving its controls. The first step in deriving the dynamic model is to derive the kinetic model purely by considering the geometry of the system dynamics. Then, the complete dynamics can be incorporated to obtain the full dynamic model. The (PID) controllers are applied considering a modular architecture. The geometry of the system along with the association of the relevant coordinate frames is shown in Fig. 2(b). The following notations and symbols, consistent with Fig. 2(a), are used in deriving the kinematic model, the full dynamic model and the control laws.



(a) (b)

Fig. 2. (a) Parameters defined on the OmniBot (b) Coordinate frames



(a) (b) (c)

Fig. 3. Basic motion types of OmniBot (a) rotation (b) along the "forward" direction (c) along the "normal" direction

- Body frame variables:

ω	Angular velocity of body rotation
v_f	Forward linear velocity
v_n	Normal linear velocity
v_1, v_2, v_3	Linear velocity of each wheel
f_1, f_2, f_3	Traction force of each wheel
ω_1, ω_2, ω_3	Angular velocity of motor shafts
τ_1, τ_2, τ_3	Torque generated by each wheel
V_1, V_2, V_3	Voltage applied to each motor

- World frame variables:

(x,y)	Position of the robot
ψ	Orientation of the robot

- Physical constants:

m	Mass of the robot
J_z	Moment of inertia of the robot
r	Radius of a wheel
R	Radius of the robot body
n	Gear ratio of the motors

In the fabricated OmniBot, R = 20 cm, r = 5 cm, n = 1.

A. Kinetic Model

First of all it is easy to note that the velocity terms defined on robot body frame, designated by B, and the world frame, designated by W, are simply related via the rotational transform [9, 13] as given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_f \\ v_n \\ \omega \end{bmatrix} \tag{1}$$

The "forward" and "normal" components of the linear velocity are simply the linear velocity components of the robot body parallel to the x- and y- axes of the body frame. Therefore, they are simply related to the linear velocities of the wheels by simple vector addition since the omni-wheels offers negligible friction for movements in the axial direction. Therefore, by the choice of body frame axis shown in fig. 02(a) the equations relating the linear velocities of each of the wheels to the forward, normal and tangential velocities of the robot body can be written.

In order to do this, it is necessary to consider the effect of the rotational kinematics and the linear kinetics of the robot body as a rigid body and add those two components together. The contribution of the rotational component is $R\omega$ for all the wheels due to the symmetry. The linear component can be found by projecting the vectors appropriately. Thus, the following equations are obtained:

$$v_1 = -v_n + R\omega \tag{2}$$

$$v_2 = v_f \cos\tfrac{\pi}{6} + v_n \cos\tfrac{\pi}{3} + R\omega \tag{3}$$

$$v_3 = -v_f \cos\tfrac{\pi}{6} + v_n \cos\tfrac{\pi}{3} + R\omega \tag{4}$$

Furthermore, $v_i = r\omega_i/n$ for i = 1,2,3 since the motor shafts are coupled through the wheels through a gear train. The radius of the wheels and the gear ratio are explicitly written because they can be changed while having the same set of motor.

Using (2), (3), (4), $v_i = r\omega_i/n$ and the fact that $\cos(\pi/3) = \sin(\pi/6)$, it is possible relate the shaft speeds of each of the motors to the forward, normal and the angular speeds of the robot body frame as follows:

$$\begin{bmatrix} v_f \\ v_n \\ \omega \end{bmatrix} = \frac{r}{n} \underbrace{\begin{bmatrix} 0 & -1 & R \\ \cos\frac{\pi}{6} & \sin\frac{\pi}{6} & R \\ -\cos\frac{\pi}{6} & \sin\frac{\pi}{6} & R \end{bmatrix}}_{B^T}{}^{-1} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{5}$$

After simplification, (5) becomes

$$\begin{bmatrix} v_f \\ v_n \\ \omega \end{bmatrix} = \frac{r}{3n} \begin{bmatrix} 0 & \sqrt{3} & -\sqrt{3} \\ -2 & 1 & 1 \\ 1/R & 1/R & 1/R \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{6}$$

Combining (1) with (6), it is possible to relate the speeds in the world frame to the speed of the motors. Both the forward and the inverse kinematics can be obtained easily.

B. Dynamic Model

The dynamics of the rotational components can be directly written by taken the moment about the center of gravity of the robot. Due to the Newton's law of rotational dynamics, this gives the familiar equation

$$J_z\dot{\omega} = \sum torque = Rf_1 + Rf_2 + Rf_3 \tag{7}$$

However, the linear dynamics are affected by pure linear dynamics as well as the influence of the rotational dynamics as given by the Newton-Euler equations [14,15]

$$\boldsymbol{F} = m\boldsymbol{a} + m\boldsymbol{\omega} \times \boldsymbol{v} \tag{8}$$

Where, $\boldsymbol{F}$ is the net force vector, $\boldsymbol{a}$, $\boldsymbol{\omega}$ and $\boldsymbol{v}$ are the vector representation of acceleration, angular velocity and linear velocity of the systems, respectively. The equations are simplified as the dynamics of the robot is limited to the x-y plane and the radius of the robot (R) does not change over time. Therefore, with the unit vectors in the body frame denoted by $\boldsymbol{i}_B$, $\boldsymbol{j}_B$ and $\boldsymbol{k}_B$. Then the dynamics can be written as

$$f_2 \cos\tfrac{\pi}{6} - f_3 \cos\tfrac{\pi}{6} = m\dot{u} - \omega v_n \tag{10}$$

$$-f_1 + f_2 \sin\tfrac{\pi}{6} + f_3 \sin\tfrac{\pi}{6} = m\dot{v} + \omega v_f \tag{11}$$

Combining (7), (10) and (11), the dynamic model of the robot can be written as a matrix equation in the convenient form:

$$\begin{bmatrix} \dot{v}_f \\ \dot{v}_n \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega v_n \\ \omega v_f \\ 0 \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix}}_{H} B \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{12}$$

Where B is defined in (5). The traction force of each wheel is related to the torque exerted by each motor by $\tau_i = rf_i/n$. The motor torque can be obtained using the DC motor model [10] using the DC motor parameters:

$$L_i \frac{dI_i}{dt} + R_i I_i + k_i \omega_i = V_i \tag{13}$$

$$k_i I_i - b_i \omega_i - \tau_i = J_i \dot{\omega}_i \tag{14}$$

Where, L_i, R_i, k_i, b_i and J_i denote the inductance, resistance, damping coefficient and inertia of the motor system respectively. The inertia contains the rotor inertia and the inertia due to wheels. Typically, the electrical constants of the motor are small compared to the mechanical systems. Therefore, the electrical dynamics are much faster. As a result, the motor dynamics can be reduced to a first order system as:

$$J_i R_i \dot{\omega}_i = k_i V_i \text{-} (k_i^2 + R_i b_i)\omega_i \text{-} R_i \tau_i \tag{15}$$

In principle, it is possible to use (12), (15) along with (6) to derive the dynamics relating the velocities in the robot body frame to the input voltages to the motors or even using (1) to relate the input voltages to the world coordinates as in [9, 13]. However, this paper proposes an alternate control strategy as outlined in section 2. This paper uses a "layered architecture" for controlling the designed OmniBot. The three stages: world frame, body frame and the motor systems are depicted in Fig. 4.

C. Control Strategy

The user or the trajectory generation algorithm provides the path in the world frame as a velocity profile. Then, using (1), it is translated to a velocity profile in the body frame. Finally, using (6), it is converted to the speed profile of the three motors. The cascade of properly tuned PID controllers the desired trajectory can be achieved indirectly rather than directly using the dynamics derived using the more complicated direct approach as derived using (12), (15) and (6) combined.

The indirect control strategy simplifies the controller design by reducing the burden on system identification. Due to the simplified approach, it suffices to obtain the motor dynamics. In particular, there is no aggregated effect of errors due to the mismatch between the motor model and the actual motor dynamics because, by the selection of controllers, each of the system levels are decoupled for "practical purposes". It is even possible to have different controllers for each layer. For example, one may opt to use a fuzzy controller without even obtaining the motor parameters but used PID controllers at the body frame and world frame levels.

The penalty for this simplified controller design is the timing and speed of operation. As described in section II, the world frame should have a sampling time in the order of 1s, the body frame, in the order of 100 ms and the motors, in the order of 10 ms so that the motors can be operated with PWM frequency 200 Hz to 10 kHz, which is typical. A typical DC motor used for robotics applications will have a speed in the order of 60 – 600 rpm after gear reduction. Wheel with diameter 4 – 20 cm is typical. Therefore, within a second, a typical mobile robot will move approximately a minimum of one centimeter to a maximum of about six meters. Therefore, if the path changes rapidly, the robot has to be slowed down. But if the path is close to linear, the speed of the robot may be increased.

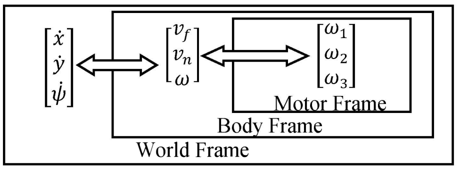


Fig. 4. Three layers of control

Furthermore, increasingly, robots rely on visual ranging and visual navigation. Vision algorithms are typically slow consuming easily up to half a second to process the visual information. Therefore, the timing constraint imposed by this control strategy falls well within the deadline because the control algorithm itself is simple.

IV. System Simulations

The first step of the practical implementation of this robot is to identify the dynamics of the motors. Therefore, a system identification experiment was carried out on the DC motors using the dSPACE hardware emulator platform on MATLAB®. The DC motors were excited using a PWM sequence of 1 kHz and 0 - 24 V amplitude, whose duty ratio was modulated by a pseudo-random sequence. Since the amplitude of the PWM sequence is fixed, the duty ratio will be proportional to the supply voltage. The speed feedback of the motor was recorded through an encoder. Using these observations, the state space model between the duty ratio and the speed encoder feedback was derived. This is actually what is necessary in the robot control.

The system identification was carried out in the discrete time domain with respect to the candidate second order state space model based on the least squares method. Consequently, the parameters were identified as $\phi_{11} = 0.7320$, $\phi_{12} = -0.0027$, $\phi_{21} = -0.8820$, $\phi_{22} = 0.9407$, $\gamma_1 = 0.4782$, and $\gamma_2 = 8.8787$. These parameters were used in conducting the simulations. The parameters identified for the three motors used were the same practically.

$$\begin{bmatrix} i[k+1] \\ w[k+1] \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \begin{bmatrix} i[k] \\ w[k] \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} d[k] \tag{16}$$

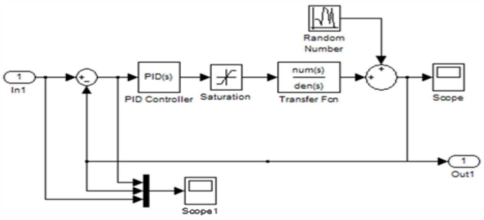


Fig. 5. Simulation model of the motor

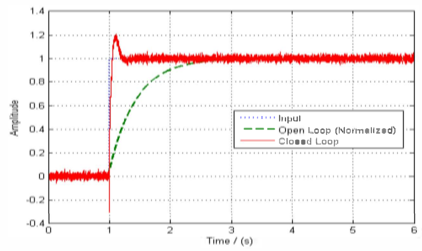


Fig. 6. Step response of the motor. The open loop system dynamics were simulated without noise and normalized to demonstrate the speed of operation

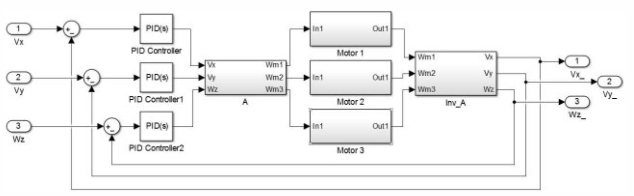
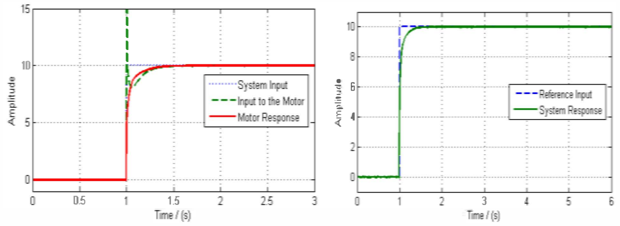
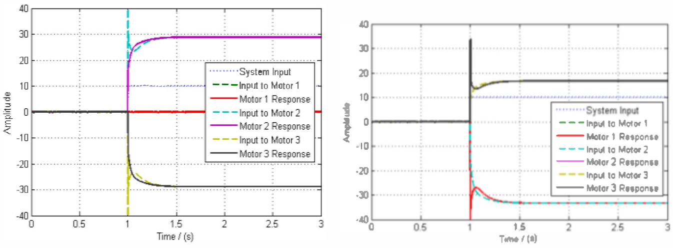


Fig. 7. Simulation model of the local frame. The motor dynamics are implemented as a subsystem with three inputs and three outputs



(a) (b)

Fig. 8. Responses to a rotation command in the local frame (a) system response (b) motor response and the command signal applied to the motors (time axis shortened to show the transient response clearly). All three motors have the same input-output profile.



(a) (b)

Fig. 9. Response of the three motors after a command to move in the (a) "forward" direction (b) "normal" direction

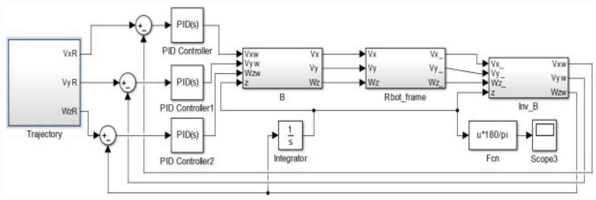


Fig. 10. Simulation model of the world frame

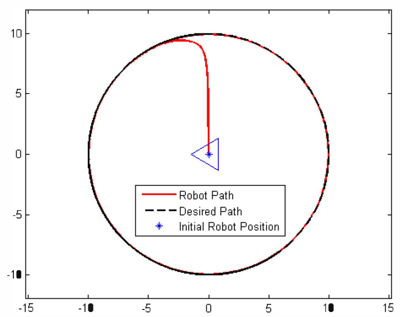


Fig. 11. Response of the world frame model to a reference trajectory of a circle

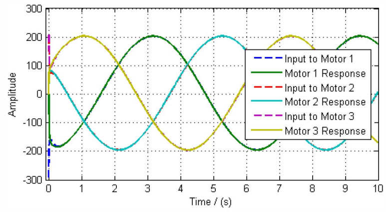


Fig. 12. Response of the three motors after a command to move in the circle in fig. 11

V. Hardware Implementation

Several hardware options were considered, and finally it was concluded to use a STM32F4 DISCOVERY board with the "waijung" SIMULINK block set for the hardware realization. This allowed easy integration of the SIMULINK model and the hardware via the DISCOVERY board. The developed SIMULINK model was slightly modified to discreet time in order to be compatible with the DISCOVERY board. The motor transfer function in discreet time, *H(z)* is given by (17). A Controller, *C(z)* was designed in discreet time in order to cancel out the un-desired zeroes and poles as given in (18).

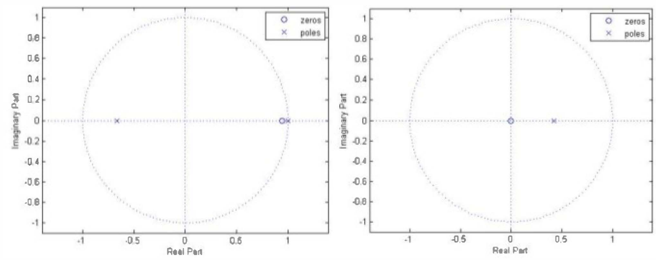
$$H_z = \frac{0.5764\,(z-0.9431)}{(z-0.9977)(z+0.6687)} \tag{17}$$

$$C(z) = \frac{(z+0.6687)}{(z-0.9977)} \tag{18}$$

$$H_{cl}(z) = \frac{0.5764}{(z-0.4213)} \tag{19}$$

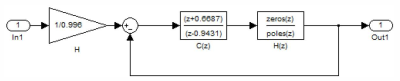
The Closed loop transfer function, $H_{cl}(z)$ with the controller can be derived as (19). The pole zero map of the motor transfer function and closed loop system is shown in Fig.13.

The derived controller is first simulated in the SIMULINK and then Modified with waijung blockset to be embedded in to the DISCOVERY board as shown in Fig. 14. The hardware connection for a single motor with Pololu MD01b motor driver is shown in Fig.16. Only the motor controllers are embedded in to the Discovery board, while the computation of world frame and body frame parameters are carried out in SIMULINK Environment. The Time Discrimination is maintained within the layers as discribe in section III(C).The completed OmniBot is Shown in Fig.16.With the implemented system motors wre individualy controlled to the desired speeds. The Completed system is still under study and further Analysis is to be continued.

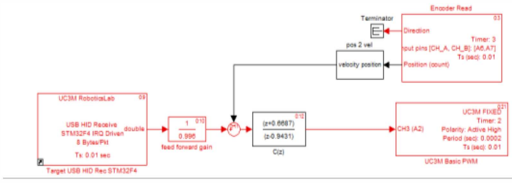


(a) (b)

Fig. 13. Zero Pole map of the Systems (a). The motor transfer function (b). Closed loop system with the controller.



(a)



(b)

Fig. 14. Discreet motor controller models (a). The simulation model (b). Modified controller to be used with DISCOVERY board.

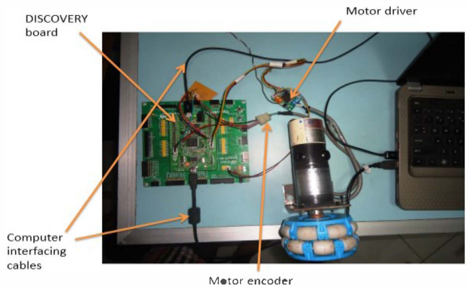


Fig. 15. Hardware interfacing with DISCOVERY board

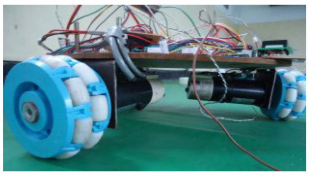


Fig. 16. Completed OmniBot

VI. Results and Discussion

The robot was first simulated on MATLAB® SIMULINK™ platform. Simulations were carried out step-by-step in a bottom-up fashion. First, the motor controller was simulated as shown in Fig. 5. As in [9], the un-modeled dynamics such as slippage and friction were added as a disturbance to the motor controller. A saturation block was added to model the limitation in the voltage supply (24V). The step response of the motor controller is shown in Fig. 6. With the used of inbuilt auto PID tuner, the properly tuned PID controller, the closed loop motor response was made faster as can be seen from Fig. 6.

Then, the next layer, the local frame (i.e. the body frame) of the robot was simulated using the structure shown in Fig. 7. The Step responses for the three basic motions shown in Fig. 3 are shown in Fig.s 8 and 9. These are for the pure rotation and translation in the "forward" and in the "normal" direction respectively. The motor responses match the expected dynamics as theoretically shown in section III.

Finally, the combined systems with all three subsystems were simulated with a reference trajectory of a circle provided. The simulation blocks are shown in Fig. 11 and the resulting path traced by the robot is shown in Fig. 12. The command signals applied to each of the motors and their responses are shown in Fig. 13. Since the robot is not on the desired trajectory at the beginning, as shown in Fig. 12, there is a transient period during which the robot tries to "catch up to" the desired trajectory. After it has done so, the trajectory is followed continuously.

VII. Conclusions

This paper presents the design, simulation and implementation of an omni-directional robot named OmniBot. It is a three wheeled symmetric holonomic omni-directional robot built using omni-wheels. A simplified control algorithm is proposed and tested on the robot. The control scheme involves a layered control architecture. The control of the subsystems is decoupled by virtue of response time. The inner system is made roughly ten times faster than the outer system. This makes the inner system reach the steady state of the previous command when the next command is applied from the outer system. Though this imposes a limitation on the sampling time of the trajectory plan, it works with accuracy as acceptable to the designers. Thus, this can be utilized as a low cost simple and reliable robot platform to carry out testing with minimum requirements.

References

[1] http://robots.net/rcfaq.html, Visited 24th July, 2013

[2] Siegwart, R., Nourbakhsh, I. R., Introduction to Autonomous Mobile Robots,MIT Press, Cambridge, Massachusetts,2004.

[3] Breazeal, C. L., Designing Sociable Robots, MIT Press, Cambridge, Massachusetts, 2002

[4] Oliveira, H. P., Sousa, A. J., Moreira A. P., Costa, P. J., "Modeling and Assessing of Omni-directional Robots with Three and Four Wheels", Contemporary Robotics – Challenges and Solutions, InTech Open Science, December 1st, 2009

[5] Diegel, O., Badve, A., Bright, G., Potgieter, J., Tlale, S., "Improved Mecanum Wheel Design for Omni-directional Robots", Proc. Australasian Conference on Robotics and Automation, Auckland, November, 2002.

[6] http://www.cs.cmu.edu/afs/cs/user/gwp/www/robots/Uranus.html, Visited 03rd May, 2013

[7] http://www.cs.cmu.edu/~reshko/PILOT/, Visited 03rd May, 2013

[8] http://www.vexrobotics.com/omni-wheels.html, Visited 03rd May, 2013

[9] Liu, Y., Zhu. J., Williams II, R. L., Wu, J., "Omni-directional Mobile Robot Controller Based on Trajectory Linearization",Robotics and Autonomous Systems, pp. 461 – 479, Vol. 56, 2008.

[10] Golnaraghi, F., Kuo, B. C., Automatic Control Systems, 9th Ed., Wiley, 2009.

[11] Åström, K., Wittenmark, B., Computer Controlled Systems: Theory and Design, 3rd Ed., Dover, 2011.

[12] Khalil, H., Nonlinear Systems, 3rd Ed., Prentice Hall, 2001.

[13] Wu, J., Dynamic Path Planning of an Omni-directional Robot in a Dynamic Environment, PhD Thesis, Ohio University, March, 2005.

[14] Bullo, F., Lewis, A. D.,Geometric Control of Mechanical Systems: Modelling, Analysis and Design for Simple Mechanical Control Systems, Springer, 2004.

[15] Arnold, V. I., Mathematical Methods of Classical Mechanics, 2nd Ed., Springer-Verlag, 1989.