Path Following Controller for Gyroscopically Stabilized Single-Wheeled Robot

ABDULLAH Al-MAMUN (*SM-IEEE*), ZHEN ZHU, PRAHLAD VADAKKEPAT (*SM-IEEE*) and TONG HENG LEE, (*Fellow*, *IEEE*)

Abstract — Path following controller, designed according to the principle of following a virtual robot, for a gyroscopically stabilized single-wheeled robot (gyrobot) is presented in this paper. Linear and angular velocities of the virtual robot are pre-computed such that it follows the desired path. These velocities also constitute the feed-forward component of the reference velocities for the gyrobot. The feedback components of the reference velocities are computed based on the error between the pose of virtual robot and that of the gyrobot. Besides the reference velocity generator, a controller is designed to make the single-wheeled robot follow the reference velocities. Conditions for stability of the single-wheeled robot are taken into consideration while designing the reference velocities. Effectiveness of the design is verified using different types of reference paths.

Index Terms—Gyroscopic precession, Single-wheeled robot, Path following controller, Virtual robot

1. INTRODUCTION

Design of the path following controller for *gyrobot*, a gyroscopically stabilized single-wheeled robot, is presented here. The *gyrobot*, built following the design of the *gyrover* developed in Carnegie Mellon University [1], is a wheel that has the shape of a flying saucer and rolls on its edge (Fig.1).

Stability of the *gyrobot* in its vertical position and its ability to steer can be explained using the principle of gyroscopic precession seen in a rolling wheel. The stability is enhanced significantly with the help of a fast spinning flywheel placed inside the *gyrobot* shell. Large angular momentum of the flywheel provides good dynamic stability and insensitivity to attitude disturbance. Moreover, due to the effect of gyroscopic precession, it shows high maneuverability. All these factors give this special structure advantage over its conventional multi-wheeled counterparts. Since electronic components are enclosed inside the shell, the *gyrobot* is especially suitable for deployment in marshy lands. Readers may refer to [1]-[3] and many other references cited there for details on the operation of a gyroscopically stabilized single-wheeled robot. This paper presents the path following controller for the *gyrobot*.

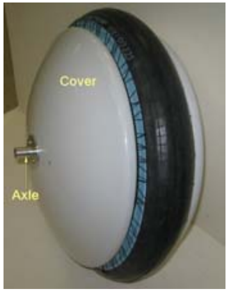

Fig. 1: Gyrobot in its vertical position

Path following control of mobile robots under nonholonomic constraints has been studied by many researchers. Back-stepping control [4], curvature based control [5], use of smooth time-varying feedback [6], piecewise feedback law for exponentially stabilizing and path tracking [7] etc are few examples of these proposed methods. The curvature based method was used to design path following controller for a single wheel robot [8].

A nonholonomic system has limitations in motion emerging from its kinematic model. Some directions of motion are not possible, e.g., the *gyrobot* can not move in the direction parallel to the wheel axle. Feedforward control is often used for nonholonomic systems where the inputs are computed for a known trajectory and all kinematic constraints are implicitly considered while planning the trajectory [9]-[10]. However, being an open loop system, such scheme is not robust to either errors in initial conditions or disturbances. This problem can be overcome by augmenting feedforward control with a feedback loop.

Path following controller presented in this paper uses a reference velocity generator designed according to the

Authors are with the Electrical and Computer Engineering Department, National University of Singapore, Singapore 117576.

Corresponding author: Abdullah Al Mamun, Tel: +65 6516 2251, Fax: +65 6779 1103; e-mail: eleaam@nus.edu.sg.

Emails of other authors – Zhen Zhu (g0305980@nus.edu.sg), Prahlad Vadakkepat (elepv@nus.edu.sg) and Tong Heng Lee (eleleeth@nus.edu.sg)

principle of following a virtual vehicle. This is briefly explained in Section III. Interested readers may refer to [11] for more details on this design concept. We also design a controller using approximate linear model of the *gyrobot* to make it follow the reference velocities.

2. Dynamic Model of the Gyrobot

The dynamic model and non-holonomic constraints of *gyrobot* are derived using constrained Lagrangian method, and matrix partition is used to eliminate the Lagrangian multipliers. The wheel is assumed to be a rigid, homogeneous disk that rolls without slipping on flat surfaces. The inner mechanism of the *gyrobot* is modeled as a two-link manipulator with the flywheel attached at the end of the second link and the first link hanging from the axle of the wheel. Moreover, it is assumed that the inner mechanism hangs just beneath the center of the axle. The dynamic model is

$$\overline{M}(q_0)\ddot{q}_0 = \overline{F}(q_0,\dot{q}_0) + \overline{B}u_0\,, \tag{1}$$

with $q_0 = [\alpha \quad \beta \quad \gamma \quad \beta_a]^T$. The angles $\alpha, \beta,$ and γ represents the precession angle, lean angle and the roll angle of the wheel, respectively, and the angle β_a is the tilt angle of the flywheel. Two torque inputs, tilt motor torque and the drive motor torque, form the input u_0. This model is simplified further by decoupling the tilt variable (β_a) to obtain,

$$M(q)\ddot{q} = F(q,\dot{q}) + Bu\,, \tag{2}$$

with $q = [\alpha \quad \beta \quad \gamma]^T$.

Let,

I_{xw}, I_{yw}, I_{zw}	:	Moment of inertia of the whole wheel,
I_{xf}, I_{yf}, I_{zf}	:	Moment of inertia of the flywheel,
m	:	Mass of the *gyrobot*,
R	:	Radius of the *gyrobot* shell,
g	:	Acceleration due to gravity, and
μ	:	Friction coefficient.

Then the matrices used in the model (equation 2) are,

$$M = \begin{bmatrix} M_{11} & 0 & M_{13} \\ 0 & M_{22} & 0 \\ M_{13} & 0 & M_{33} \end{bmatrix} \text{ with}$$

$$M_{11} = I_{xf}\left(1+\cos^2(\beta+\beta_a)\right) + I_{xw}\left(1+\cos^2\beta\right) + mR^2\cos^2\beta\,,$$

$$M_{13} = 2I_{xw}\cos\beta + mR^2\cos^2\beta\,,$$

$$M_{22} = I_{xf} + I_{xw} + mR^2\,,$$

$$M_{33} = 2I_{xw} + mR^2\,,$$

$$F = [F_1 \quad F_2 \quad F_3]^T \text{ with}$$

$$F_1 = \left(I_{xw}+mR^2\right)(\sin 2\beta)\dot{\alpha}\dot{\beta} + I_{xf}(\sin 2(\beta+\beta_a))\dot{\alpha}\dot{\beta} - \mu\dot{\alpha} - 2I_{xw}(\sin\beta)\dot{\beta}\dot{\gamma} + 2I_{xf}(\sin(\beta+\beta_a))\dot{\beta}\dot{\gamma}_a$$

$$F_2 = -gmR\cos\beta - \left(I_{xw}+mR^2\right)(\cos\beta\sin\beta)\dot{\alpha}^2 - \left(2I_{xw}+mR^2\right)(\sin\beta)\dot{\alpha}\dot{\beta} - I_{xf}\cos(\beta+\beta_a)\sin(\beta+\beta_a)\dot{\alpha}^2 - 2I_{xf}(\sin(\beta+\beta_a))\dot{\alpha}\dot{\gamma}_a$$

$F_3 = 2\left(I_{xw}+mR^2\right)(\sin\beta)\dot{\alpha}\dot{\beta}$, and

$$B = \begin{bmatrix} 0 & 0 & 1 \\ B_{12} & 0 & 0 \end{bmatrix}^T \text{ with}$$

$$B_{12} = I_{xf}\sin(2(\beta+\beta_a))\dot{\alpha} + 2I_{xf}(\sin(\beta+\beta_a))\dot{\gamma}_a\,.$$

The trajectory of the point of contact between the *gyrobot* and the ground is described by the following pair of equations,

$$\begin{aligned} \dot{x}_R &= R\left(\dot{\gamma}\cos\alpha + \dot{\alpha}\cos\alpha\cos\beta - \dot{\beta}\sin\alpha\sin\beta\right), \\ \dot{y}_R &= R\left(\dot{\gamma}\sin\alpha + \dot{\alpha}\sin\alpha\cos\beta + \dot{\beta}\cos\alpha\sin\beta\right). \end{aligned} \tag{3}$$

For the model of equation (2), two input variables are torque for the drive motor and tilt velocity ($\dot{\beta}_a$) of the flywheel. We assume that an inner loop controller controls the tilt velocity for the flywheel faster than the response for changes in wheel lean angle (β). Rotational velocity of flywheel is assumed constant. We omit the detailed derivation of the dynamic model; interested readers may refer to [12].

3. Design of Controller

Design of the path following controller involves two separate steps. In step 1, a feedforward plus feedback loop is used to generate reference velocities required for the *gyrobot* to follow the given path. Concept of following a virtual robot is used and kinematic constraints are taken into consideration. Then in step 2, a feedback controller is designed using a linearized model such that the *gyrobot* follows the reference velocities.

3.1 Reference Velocity Generator

The concept of following a virtual vehicle is illustrated in Fig. 2, where the vector $[x_v\ y_v\ \theta_v]^T$ represents the pose of the virtual robot while the vector $[x_R\ y_R\ \theta_R]^T$ represents that of the actual robot. The virtual robot travels along the desired path (shown as broken line), i.e., the path to be followed by the actual robot.

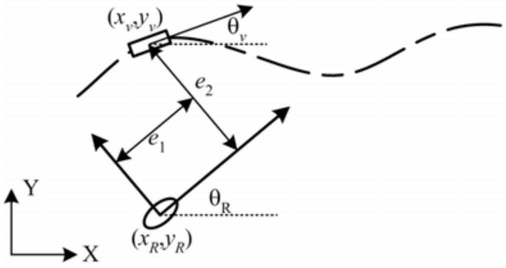

Fig. 2: Robot following a virtual robot

The tangential velocity v_v and the angular velocity ω_v of the virtual robot can be pre-computed according to the inverse kinematics of the virtual robot,

$$\dot{x}_v = v_v \cos\theta_v\ , \quad \dot{y}_v = v_v \sin\theta_v\ , \quad \dot{\theta}_v = \omega_v . \tag{4}$$

The pose vector of the virtual robot $[x_v\ y_v\ \theta_v]^T$ can be computed at every sample. If v_v and ω_v are set as the reference velocities for the *gyrobot* and is controlled to follow these references, then both will trace the same path provided the initial poses of the two are same. However, such assumption on the initial condition is not realistic as the initial pose of the *gyrobot* may not at all coincide with the trajectory it is expected to follow. We use the error between pose vectors of the real robot and the virtual robot to generate a feedback component of the reference such that the error converges asymptotically to zero. Convergence of actual robot's path to that of the virtual robot can be guaranteed by ensuring stability of the feedback loop.

We assume that the tangential velocity v_v and angular velocity ω_v necessary to keep the virtual robot on the desired path is known, and we use these as the feedforward references for tangential velocity v_R and angular velocity ω_R of the gyrobot. We also assume that the kinematic constraints are expressed by,

$$\dot{x}_R = v_R \cos\theta_R\ , \quad \dot{y}_R = v_R \sin\theta_R\ , \quad \dot{\theta}_R = \omega_R . \tag{5}$$

Here, x_R and y_R are the coordinates of the gyrobot's point of contact on the ground and θ_R is its angle of orientation. The difficulty in navigation of the robot under these constraints lies in the fact that there are three degrees of freedom but only two control variables v_R and ω_R. Moreover, the robot constraints for the *gyrobot* are different from those of the other multi-wheeled mobile robots. We need to control not only the attitude of the robot $[x_R\ \ y_R\ \theta_R]^T$ but also to control its lean angle within a stable region.

Let us define an error vector $\mathbf{e} = [e_1\ e_2\ e_3]^T$ that defines the difference between the pose vectors of the virtual robot and the *gyrobot* as illustrated in Fig. 2,

$$e_1 = (x_v - x_R)\cos\theta_R + (y_v - y_R)\sin\theta_R\,, \tag{6.a}$$

$$e_2 = -(x_v - x_R)\sin\theta_R + (y_v - y_R)\cos\theta_R\,, \tag{6.b}$$

$$e_3 = \theta_v - \theta_R\,. \tag{6.c}$$

The errors e_1 and e_2 are shown in Fig. 2. The objective of the reference generator is to select the reference velocities such that the error dynamics (equations 6.a-6.c) is stabilized and the error vector asymptotically approaches zero. We select these reference velocities v_R and ω_R as a combination of feedforward components (v_v, ω_v) and feedback components. Before we derive the expressions for feedback components, let's examine the error dynamics by differentiating the equations above. It can be easily shown that,

$$\dot{e}_1 = v_v \cos e_3 - v_R + \omega_R e_2\,, \quad \dot{e}_2 = v_v \sin e_3 - \omega_R e_1\,, \quad \dot{e}_3 = \omega_v - \omega_R\,. \tag{7}$$

The tangential velocity v_v and the angular velocity ω_v are pre-computed to make the virtual robot follow the desired path. We compute v_R and ω_R as functions of error such that the error dynamics is stabilized. When the error between the orientations of the two robots is small, $\cos(e_3) \cong 1$ and $\sin(e_3) \cong 0$. Then,

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix} = \begin{bmatrix}0 & \omega_R & 0\\ -\omega_R & 0 & v_v\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}e_1\\ e_2\\ e_3\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}f_1\\ f_2\end{bmatrix}. \tag{8}$$

When the error e_1 is positive, the real robot is trailing the virtual robot and tangential velocity of the robot should be increased. Similarly, the orientation error e_3 is reduced by manipulating the angular velocity of the *gyrobot*. The error e_2 can be affected by appropriate changes in angular velocity but taking the direction of motion into consideration. So, we select the feedback components as,

$$f_1 = -h_1 e_1\,, \quad f_2 = -\operatorname{sgn}(v_v) h_2 e_2 - h_3 e_3\,. \tag{9}$$

The gains should be chosen properly to stabilize equation 8. The method proposed by Klancar *et al* in [13] is followed here. The schematic of the velocity reference generator is shown in Fig. 3.

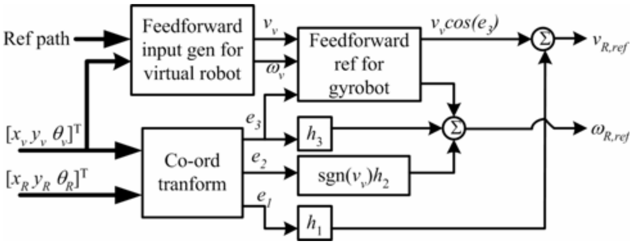

Fig. 3: Schematic of the reference velocity generator

3.2 Controlling ω_R and v_R of gyrobot

The virtual robot approach is used to generate the reference velocities v_R and ω_R of the *gyrobot*. The linear velocity v_R is controlled by changing the rolling speed of the *gyrobot* whereas the precession rate ω_R is controlled by manipulating the lean angle of the *gyrobot*. However, the lean angle β of the *gyrobot* can not be controlled directly. Instead, we control the tilt angle β_a of the flywheel.

In order to design the controller for both rolling speed and lean angle, we linearize the dynamic model of the *gyrobot*. Let's assume that the wheel is balanced near its vertical position, i.e., lean angle is $\beta = (\pi/2)\pm\delta_\beta$ with small perturbation δ_β to achieve precession or steering. We also assume that the rolling speed of the *gyrobot* is controlled around the nominal speed Ω_0, i.e., $\dot{\gamma} = \Omega_0 + \Omega$. The linear model of the *gyrobot* is

$$(I_{xf} + I_{xw})\ddot{\alpha} = 2(I_{xw}\Omega_0 + I_{xf}\dot{\gamma}_a)\dot{\delta}_\beta - \mu_s + 2I_{xf}\dot{\gamma}_a u_2 \qquad (10.a)$$

$$(I_{xf} + mR^2)\ddot{\delta}_\beta = gmR\delta_\beta - 2(I_{xw} + mR^2)\Omega_0\dot{\alpha} - 2I_{xf}\dot{\gamma}_a\dot{\alpha} \qquad (10.b)$$

$$2(I_{xw} + mR^2)\dot{\Omega} = -\mu_g\Omega + u_1. \qquad (10.c)$$

Rolling of the *gyrobot* (equation 10.c) is independent of the lean and yaw dynamics described by equations 10.a and 10.b. We can design a controller for the rolling speed using the first order linear model (equation 10.c) independent of the other two dynamics. Equations (10.a) and (10.b) can be written in state space form as,

$$\begin{bmatrix} \dot{\delta}_\beta \\ \dot{\omega} \\ \ddot{\delta}_\beta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_{22} & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \begin{bmatrix} \delta_\beta \\ \omega \\ \dot{\delta}_\beta \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \end{bmatrix} u_2 . \qquad (11)$$

The precession rate $\dot{\alpha}$ is the same as the angular velocity ω used in the kinematic equation of the robot. We replaced $\dot{\alpha}$ with the angular velocity ω in the above state equation. This 3rd-order system can be stabilized using linear state feedback

$$u_2 = k_1(\delta_{\beta,ref} - \delta_\beta) + k_2(\omega_{ref} - \omega) - k_3\dot{\delta}_\beta . \qquad (12)$$

***Remark* 1**: If we consider the steering rate ω (or $\dot{\alpha}$) as an output of the system described by equation 10, then the corresponding transfer function has a zero on the RHS plane [8]. In order to avoid the non-minimum phase behaviour, we exclude $k_2(\omega_{ref}-\omega)$ from the feedback law (equation 12). It may be appropriate to point out here that the gyrobot does not have any sensor to measure the angular velocity ω. We do not control the angular velocity of the gyrobot directly, but control the lean angle instead. A tilt sensor is used to measure the lean angle β and a potentiometer measures the tilt angle of the flywheel β_a.

Since the gyroscopic torque of the flywheel is very large, the stability of the wheel in the vertical position is primarily maintained by the flywheel and not by the rolling speed of the wheel. Besides, the lean angle β tends to remain unchanged. Assuming approximately constant rolling speed of the wheel, little variation in the lean angle, and low yaw rate variation, we can evaluate the dynamic model for $\dot{\beta} = 0$, $u_1 = u_2 = 0$, and $\ddot{\alpha} = \ddot{\beta} = \ddot{\gamma} = 0$ to obtain,

$$0 = gmR\cos\beta + (2I_{xw} + mR^2)\sin\beta\dot{\alpha}\dot{\gamma} + 2I_{xf}\sin(\beta + \beta_a)\dot{\alpha}\dot{\gamma}_a . \qquad (13)$$

Under these assumptions, we relate the equilibrium angular velocity to a specific lean angle as

$$\omega = \dot{\alpha} = \frac{gmR\delta_\beta}{(2I_{xw} + mR^2)\Omega_0 + 2I_{xf}\dot{\gamma}_a} = \frac{1}{C}\delta_\beta . \qquad (14)$$

Given the reference for ω by the reference velocity generator, the controller computes the reference for change in tilt angle, i.e., $\delta_{\beta,\text{ref}} = C\omega_{\text{ref}}$ according to equation 14.

Control of the linear velocity of the wheel can be accomplished by applying a close loop control around the drive motor. The first-order system of equation 10.c is controlled using a proportional feedback $u_1 = k_\Omega (\Omega_{\text{ref}}-\Omega)$. The reference angular velocity for the drive motor is computed as $\Omega_{\text{ref}} = v_{\text{R,ref}}/\text{R}$, where R is the radius of the *gyrobot* wheel. The reference for the linear velocity $v_{\text{R,ref}}$ is provided by the reference velocity generator discussed earlier in this section.

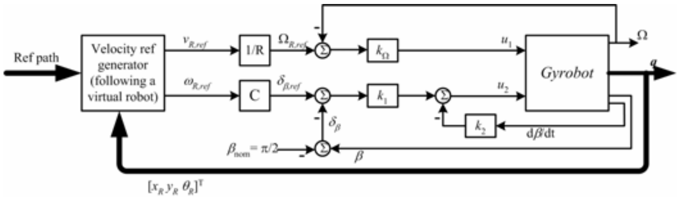

Fig. 4: Block diagram of the path-following controller

4. RESULTS

Effectiveness of the controller is verified using simulation for few different types of paths followed by the *gyrobot*. The response of the *gyrobot* is simulated using the nonlinear dynamic model presented in Section 2.

4.1 Effect of Changing Tilt Angle of Flywheel

As explained earlier, the angular velocity of the *gyrobot* can be controlled by controlling the tilt angle of the flywheel and thus the lean angle of the wheel. To see this effect, we subject the closed loop system to a variation in the reference for δ_β. However, for this simulation, this variation is not caused by the reference velocity generator, rather it is changed directly. Corresponding results illustrating tilt angle of the flywheel β_a, change in the lean angle of the wheel δ_β, and the path traversed by the *gyrobot* are shown in Fig. 5 - Fig. 7. It is evident from these plots that the wheel leans in the direction opposite to the tilt of the flywheel, and leaning the wheel makes it steer in the direction of the lean.

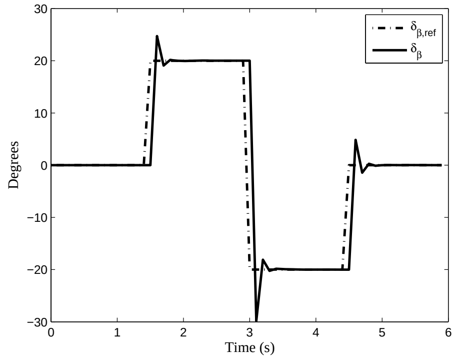

Fig. 5: Variation in lean angle (β) in response to $\delta_{\beta,ref}$

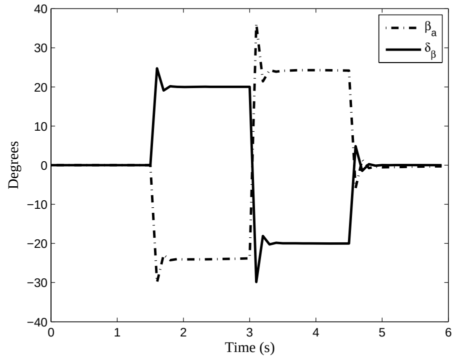

Fig. 6: Wheel leans in the direction opposite to the tilt of flywheel

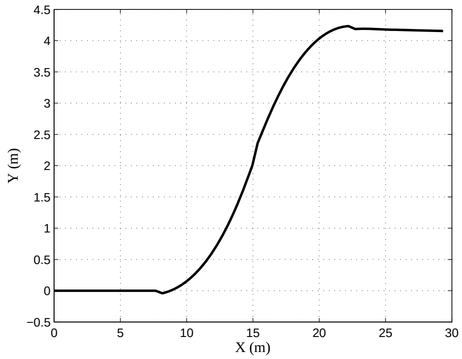

Fig. 7: Trace of path traversed by the gyrobot

4.2 Following a Straight Path

For this simulation, the reference path is described by the equation of a straight line $y = mx + c$. In order to have the virtual robot follow such straight-line path, its angular speed ω_v is set to 0. The linear speed of the virtual robot v_v is set to the nominal speed of the *gyrobot* moving in its vertical posture. Since the nominal speed of the drive motor is 240 RPM, the linear speed of the virtual robot is set to 2π(240/60)R, which is equal to 4.27 m/s. Assuming the initial position of the robot at (0,0) of the X-Y plane, we set the initial position and orientation of the virtual robot at $(0, c)$ and $\tan^{-1}(m)$, respectively. Simulation result for this case is shown in Fig.8.

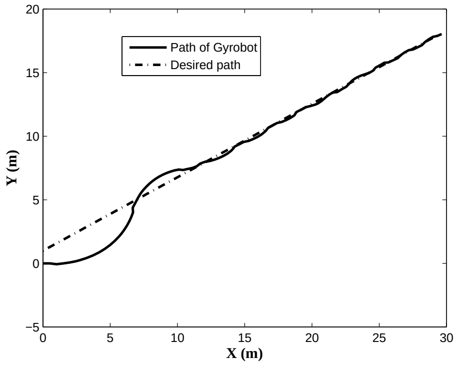

Fig. 8: Gyrobot follows a straight line path

4.3 Following a Circular Path

A circular path of radius ρ and with center at (X_0,Y_0) is described by the equation $(x-X_0)^2 + (y-Y_0)^2 = \rho^2$. We set the tangential speed v_v of the virtual robot to the nominal speed of the *gyrobot* which is 4.27 m/s. Then the angular speed ω_v should be v_v/ρ, i.e., $4.27/\rho$ rad/s. Setting the initial pose of the virtual robot is an important issue. It is observed that the path of the *gyrobot* converges to that of the virtual robot starting from any arbitrary initial pose. However, initial pose of the virtual robot affects the time it takes to converge. Methods for selecting an optimal initial pose will be published in a sequel of this work. Results for following a circular path are shown in Fig. 9.

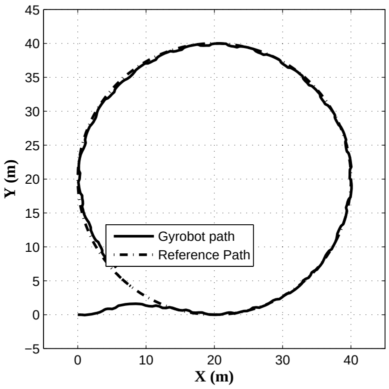

Fig. 9: Gyrobot follows a circular path

4.4 Following a Sinusoidal Path

For a virtual robot following a sinusoidal path, its angular velocity should be changed as function of time. The following set of equations can be used to generate a sinusoidal path,

$$\dot{x}_v = v_v \cos\theta_v, \quad \dot{y}_v = v_v \sin\theta_v, \quad \dot{\theta}_v = \frac{\varphi\pi}{2}\cos(\varphi t). \tag{15}$$

The parameter φ can be adjusted to change the length of one full cycle of the sinusoidal path. Since the angular velocity of the *gyrobot* is controlled by controlling its lean angle, there exists an upper limit to the angular velocity that can be achieved. Tilting the *gyrobot* beyond a limit causes it to fall on its side. Let the maximum allowable angular velocity of the *gyrobot* be $\omega_{R,max}$, which depends on the physical parameters of the robot as well as on the lean angle. Then, according to equation (15), the maximum angular velocity is $\varphi = 2\omega_{R,max}/\pi$. Simulation results are shown in Fig. 10.

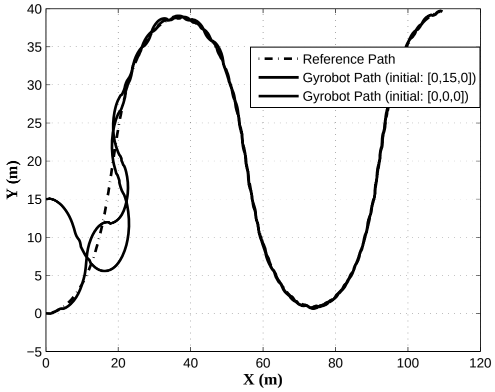

Fig. 10: Gyrobot follows a sinusoidal path

5 CONCLUSION

We investigated a strategy for controlling a gyroscopically stabilized single-wheeled robot, the *gyrobot*, to follow a pre-defined path. The feedback system consists of a reference velocity generator and a velocity controller for the *gyrobot*. The reference velocity generator uses both feedforward and feedback of error between the positions of the actual robot and a virtual robot.

The wheel is made to follow the linear velocity using a closed loop control of the speed of drive motor. Controlling the angular velocity is a more difficult task. Any change in the lean angle of the *gyrobot* causes its angular velocity to change. However, the lean angle can not be changed directly and is controlled by changing the tilt angle of the flywheel. A decoupled, linear model of the *gyrobot* is used to design the controller for lean angle.

References:

[1] H.B. Brown and Y. Xu, ‘‘A single wheel gyroscopically stabilized robot", *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 4, pp. 3658-3663, 1996.

[2] T. Saleh, H.H. Yap, Z. Zhu, A. Al-Mamun, and P. Vadakkepat, ‘‘Design of a gyroscopically stabilized single-wheeled robot", *Proc. of 2004 IEEE Conf. on Robotics, Automation and Mechatronics*, 1-3 Dec, 2004, vol. 2, pp. 904-908, 2004.

[3] A. Al-Mamun, Z. Zhu, P. Vadakkepat and T.H. Lee, ‘‘Tracking control of the gyrobot - a gyroscopically stabilized single-wheeled robot", *Proc. of the 32nd Annual Conf. of the IEEE Industrial Electronics Society*, IECON'05, 6-10 Nov. 2005.

[4] I. Kolmanovsky and N.H. McClamroch, ‘‘Application of integrator back-stepping to nonholonomic control problems", *Proc. IFAC Nonlinear Control System Design Symposium*, pp. 747-752, 1995.

[5] Y.J. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, ‘‘A stable tracking control method for nonholonomic mobile robots", *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, Vol. 3, pp. 1236-1241, 1991

[6] C. Samson, ‘‘Time-varying feedback stabilization of car like wheeled mobile robot", *Int. Journal of Robotics Research*, vol. 12, no. 1, pp. 55-64, 1993

[7] O.J. Sordalen and C. Canudas de Witt, ‘‘Exponential control law for a mobile robot: extension to path following", *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2158-2163, 1992.

[8] K.W. Au and Y. Xu, ‘‘Path following of a single wheel robot", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2925-2930. 2000.

[9] N. Sarkar, X. Yun and V. Kumar, ‘‘Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot", *Int. Journal of Robotic Research*, vol. 13, no. 1, pp. 55-69, 1994.

[10] G. Oriolo, A. Luca and M. Vandettelli, ‘‘WMR control via dynamic feedback linearization: design, implementation, and experimental validation", *IEEE Trans. on Control Systems Technology*, vol. 10, no. 6, pp. 835-852, 2002.

[11] M. Egerstedt, X. Hu and A. Stotsky, ‘‘Control of mobile platforms using a virtual vehicle approach", *IEEE Trans. On Automatic Control*, vol. 46, no. 11, pp. 1777-1782, Nov. 2001

[12] G.C. Nandy and Y. Xu, ‘‘Dynamic model of a gyroscopic wheel", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, pp. 2683-2688, 1998.

[13] G. Klancar, D. Matko and S. Blazic, ‘‘Mobile robot control on a reference path", *Proc. of the 13th Mediterranean Conf. on Control and Automation*, Cyprus, pp. 1343-1348, June 27-29, 2005