Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A 3-D Simulator using ADAMS for Design of an Autonomous Gyroscopically Stabilized Single Wheel Robot

Zhen Zhu *(Member, IEEE),* Myint Phone Naing and Abdullah Al-Mamun (*SM-IEEE*)
National University of Singapore
Singapore

***Abstract* —Development of a 3-dimensional simulator for gyroscopically stabilized single wheel robot (gyrobot) is reported in this paper. This virtual environment of simulating 3D motion of the robot, under open-loop control and closed-loop control, can be used to expedite the design and optimization of the robot. Mechanical drawings of the gyrobot are first created using a CAD software, e.g., Solidworks and imported into the ADAMS, where the material type, density are defined. It is underscored using simulation results that the virtual prototype reflects the operation of the gyrobot. The 3D simulator has also been used to design and simulate a controller for autonomous operation of the gyrobot. Although this paper addresses the issues related to the gyrobot, the virtual environment can be easily used for design of other mechatronic systems including robots.**

***Keywords*—Single wheel robot, Virtual prototype, ADAMS, PD Controller**

I. INTRODUCTION

Simulation is important for engineering design of any mechatronics system involving complex operations. Traditionally, most simulations are computer based and require mathematical models is used to describe system dynamics and to find analytical solutions.

Automatic Dynamic Analysis of Mechanical Systems (ADAMS) started as a general purpose program that can analyze systems undergoing large non-linear displacements while under the effect of non-linear force and motion input. The methodology developed by Nicolae Orlandea was the basis of his PhD dissertation at the University of Michigan [1]. ADAMS can be categorized as a general purpose numeric code utilizing a non-minimal set of co-ordinates to develop the equations of motion. It uses stiff integrators to solve these equations and sparse matrix algebra to solve the linear algebraic equations in its innermost computation loop [2]. Since its inception, significant development investments resulted in sophisticated virtual prototyping tools for a wide range of industrial applications.

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 [3]-[6] and many other references cited there for details on the operation of a gyroscopically stabilized single-wheeled robot.

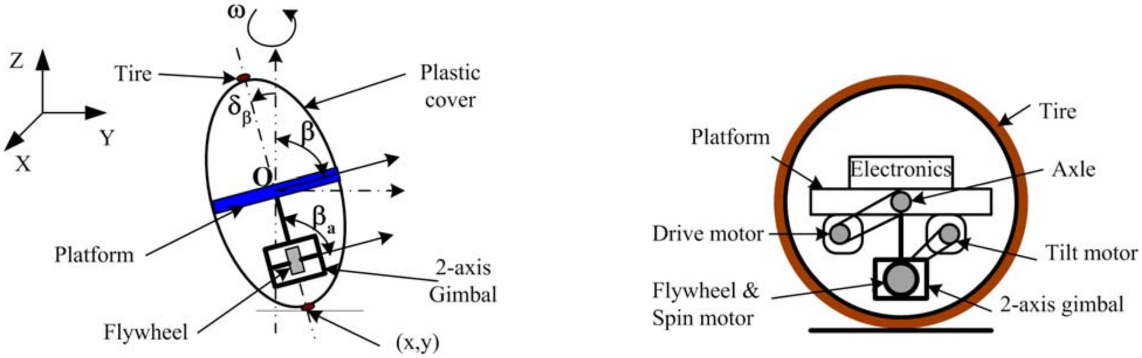
This paper presents the design, definition and simulation of *gyrobot* in ADAMS. The actual system and its drawings are briefly described in section II. System design process is then given with details in Section III on settings required for ADAMS environment. Simulation results in Section IV show that a good representation of the actual *gyrobot* system is created in ADAMS environment.

II. VIRTUAL GYROBOT

Principle of operation of the *gyrobot* can be explained using the schematic drawing shown in Fig. 1. The flywheel is suspended from the axle of the *gyrobot* using a gimbal assembly that allows tilting of the spinning flywheel. The flywheel, attached to the inner gimbal, is spun by the spin motor. The tilt motor, attached to the outer gimbal, can tilt the inner gimbal plus fast spinning flywheel to either side. The gimbal and flywheel structure is suspended from a platform through which the axle of *gyrobot* is passed through using ball-bearing. The axle as well as the outer shell attached to it is driven through a belt mechanism by another motor (drive motor).

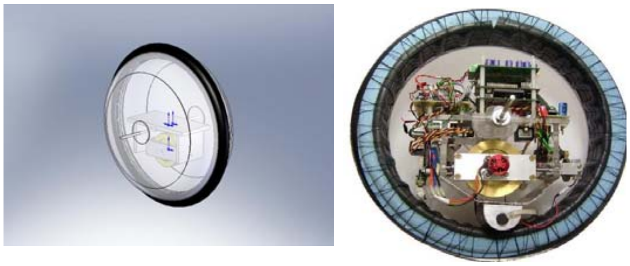
The model of *gyrobot* created using SolidWorks and the internal hardware of the *gyrobot* are shown in Fig. 2. The actual robot is equipped with PC104 CPU and DAQ board as central control module. The robot can be operated in either manual remote-controlled mode or autonomous mode. Smooth switching between these modes is implemented using a HBC101 safety switch. The electrical system enclosed inside the shell also includes power supply module and various sensors to measure tilt angle, lean angle, different velocities and other variables of interest for autonomous operation. However, in order to reduce the complexity, only the most essential parts are included in the model created in ADAMS. These are the wheel, the inner gimbal, outer gimbal, the flywheel and the structure that integrates all. Each part is created according to the shape and dimensions of the actual part used in the *gyrobot*. This structure sufficiently reflects the physical properties of *gyrobot*, which will be verified later. Various mechanical parameters, e.g., mass, density and frictions, are defined in ADAMS/View.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE



(a) Schematic as viewed from the back (b) Schematic as viewed from the side

Fig.1: Internal mechanism of *gyrobot*



SolidWorks model Internal hardware

Fig. 2: *Gyrobot* model in SolidWorks and real hardware

III. DYNAMIC SIMULATION IN ADAMS

After the mechanical system is defined using the SolidWorks model, it is imported into ADAMS and mechanical parameters are defined. The model is now ready for simulating 3D motion of the mechanical parts.

A. Gyrobot in ADAMS

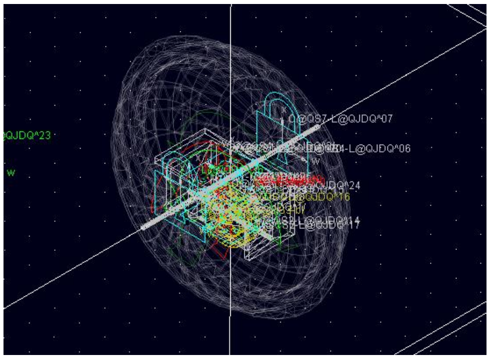


Fig 3. *Gyrobot* Model in ADAMS

The 3D drawings of different parts of the *gyrobot* are imported into ADAMS as parasolids file, shown in Fig 3. Material, mass, density and friction are then defined. Mass and inertial matrices are automatically calculated. Main parameters of different parts used to define the model of the *gyrobot* are listed in Table 1.

Parts	Mass (kg)	I_{xx} (kg m^2)	I_{yy} (kg m^2)	I_{zz} (kg m^2)
Wheel	0.85	2.78×10^{-2}	1.582×10^{-2}	1.58×10^{-2}
Flywheel	1.02	1.21×10^{-3}	0.65×10^{-3}	0.65×10^{-3}
Shaft	0.15	1.55×10^{-3}	1.55×10^{-3}	0.69×10^{-6}
Inner gimbal	0.75	1.54×10^{-3}	1.38×10^{-3}	3.63×10^{-4}
Outer gimbal	1.44	7.53×10^{-3}	7.43×10^{-3}	2.13×10^{-3}
Total	4.22	39.63×10^{-3}	11.17×10^{-3}	3.3×10^{-3}

Table 1. Main Inertial data for *gyrobot* Modules

Gyrobot model in ADAMS takes several aspects into account, such as gravity, contact constraints, friction, inertial properties and reference makers. All these definitions and settings must be defined properly for a good approximation of real robot behavior in the virtual environment. Table 2 shows some of the main joints and motions that have been defined for *gyrobot* model in ADAMS. Each joint is defined with a particular motion with specific purpose. For example, the driving joint is specified as revolute joint and so the definition of the rotation along with it is the flywheel spinning velocity.

B. Settings for Adams Simulation

Accurate representation of the contact between ground and wheel is required for proper dynamic simulation. As the rim of the actual *gyrobot* is wrapped in aluminum foil and tested on an aluminum plate, the coefficients chosen for friction are 0.95 for static condition and 0.8 for dynamic condition, in agreement with the suggestion in [7].

In Adams software, solver is the one that does all the motion calculations behind the simulation. In solver, different types of Integrator can be employed to meet the need of particular simulation problem. Key parameters for the solver are listed below. Detailed explanation on each integrator can be found in Adams/solver documentation.

- Integrator, formulation and corrector
- Simulation frequency and internal frequency
- End time and Steps Vs Duration and Step size
- Optimal step size
- H_{max}, Interpolate and Error

Joint	Motion	Action
Driving_joint	Motion3 (Revolute joint)	Drive the wheel
Spinning_Joint	Motion1(Revolute joint)	Spin the flywheel
Tilting_Joint	SForce (Revolute joint)	Tilt the inner gimbal
Joint_2	Fixed joint	Lock the wheel and shaft
Dummy_1		
Joint_dummy1	Spherical joint	To create reference part to get lean angle
JPR_orient	Orientation primitive joint	To create reference part to get lean angle

Table 2. Some of the Joints and Motions defined for *gyrobot* model in ADAMS

GSTIFF is the recommended integrator for most of the mechanical systems. For *gyrobot* simulation, we use default integrator, formulation and corrector. One may use other integrator settings, based on the requirements of the simulation (for models of other mechanical structures) with clear understanding of integrator setting documentation of Adams/Solver. Simulation frequency defines the rate at which graphic display is to be updated.

Internal frequency is closely related to the system we want to study. It defines how fast the system components are changing with respect to time. This internal frequency is very important for simulation because the solver setting is to be set based on this internal frequency.

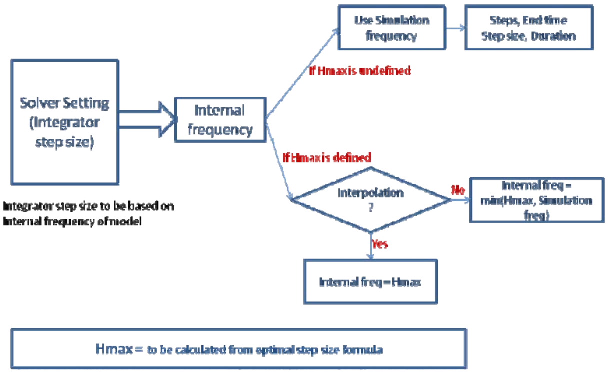


Fig. 4: Integrator Setting Path

Fig 4 shows the roadmap of defining ADAMS solver settings. For *gyrobot* simulation, optimal step size is obtained using the following formula.

Y (deg/sec) x (sec/step) = 5 (deg/step)
Internal freq Optimal step size Each step corresponds to
(based on rpm of flywheel) every 5deg of flywheel rotation

This means the simulation frequency is highly dependant on the speed of the *gyrobot* flywheel. For example, when the flywheel speed is set to the 7000 rpm, this equals to 42000 deg per second. According to the above formula, the H_{max} should be set to 5/42000. For each simulation, H_{max} will be updated based on the flywheel rpm the simulation requires.

IV. OPEN LOOP SIMULATION

Gyrobot operates on two physical laws: law of conservation of angular momentum and gyroscopic precession. Angular momentum of the spinning flywheel acts as a stabilizing factor. According to conservation of angular momentum, the higher the spin velocity, the longer the robot should remain upright, given all other conditions remaining unchanged.

A. Stability Test

Relation between stability and spinning speed of flywheel is tested in this simulation by monitoring the time for which the *gyrobot* remains upright for different flywheel speeds. Initially, both flywheel and wheel are positioned upright. Then with flywheel spinning, the *gyrobot* is allowed to fall with no stabilizing controller in use. Time elapsed before the *gyrobot* falls on the ground is monitored. This simulation has been carried out with flywheel speed increasing from 1000 rpm to 7000rpm, which is the speed range the real *gyrobot* flywheel supports. Table 3 shows the test result. The time for which *gyrobot* remains upright increases with increasing speed of the flywheel. This verifies the stabilizing effect of the flywheel owing to the conservation of angular momentum.

Tilt test is conducted for both stationary *gyrobot* and rolling at 30 rpm. When the direction of wheel rotation is same as the direction of flywheel's spin, the angular momentum of the rolling wheel adds to the angular momentum of the *gyrobot* increasing overall system stability. As result in Table 3 shows, the *gyrobot* remains upright longer when the wheel rolls. This also asserts the law of conservation of angular momentum.

Flywheel speed (rpm)	Time elapsed before the fall of *gyrobot* (s)	
	Stationary *gyrobot*	*Gyrobot* rolling at 30 rpm
1000	2.7	4.5
2000	2.9	13
3000	3.4	20
4000	4.5	26
5000	7.6	35
6000	13	50
7000	13.9	50

Table 3. Stabilization Test Result

B. Tilt Test

Steering of the *gyrobot* exploiting the effect of gyroscopic precession is verified next by tilting the flywheel while simulating the motion of the system. Principle of gyroscopic precession states that when torque is applied to change the spin axis of a spinning wheel, the entire wheel rotates about the third axis, called the precession axis. If we tilt the *gyrobot* while it is rolling, it would precess instead of falling. Gyrobot has no actuator to tilt it directly. Tilting is effectuated indirectly by tilting the flywheel with large angular momentum. As the change of wheel lean angle is coupled with flywheel tilt motion, such tilting motion makes the *gyrobot* tilt to its side. This, in turn, causes the robot to steer. Equation (1) shows the relationship between the wheel lean angle and procession rate. Interested reader may refer to [8] for more details.

$$\delta_{\beta_a} = \frac{\dot{\alpha}_{max}[(2I_{xw} + mR^2)\dot{\gamma}_{max} + 2I_{xf}\dot{\gamma}_a]}{gmR} \qquad (1)$$

δ_{β_a} : Change in flywheel tilt angle

$\dot{\alpha}$: *Gyrobot* precession rate

$\dot{\gamma}$: *Gyrobot* rolling speed

$\dot{\gamma}_a$: Flywheel speed

m: Mass of the *gyrobot*

I_{xw}: Moment of inertia of wheel

I_{xf}: Moment of inertia of flywheel

R: Radius of *gyrobot*

In this test, the *gyrobot* stands on a flat surface with the flywheel spinning at 7000 rpm. The faceting tolerance is set to 3000 to simulate the bumpiness and fineness of the robot contact point with ground.

Test 1:

In this test, the robot is initially standing upright, i.e., both wheel and flywheel are perpendicular to the ground. Flywheel speed is set to 7000 rpm. At 0.5second, the flywheel starts to tilt at a speed of 46° per second and continues to move at constant rate for 0.5 s when the flywheel tilt angle reaches 23°. Results of this simulation are shown in Fig. 5.

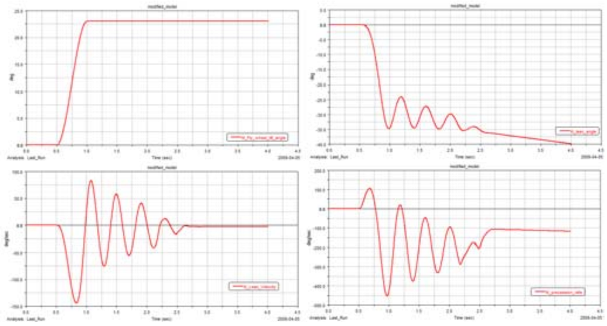


Fig. 5: Simulation result for Tilt Test 1
(from upper left clockwise: tilt angle, lean angle, precession rate, lean velocity)

Test 2

The initial condition for this simulation is same as in Test 1. However, in this test the flywheel tilts to the opposite direction and the rolling speed of the *gyrobot* is set to 30 rpm. Results are illustrated in Fig. 6.

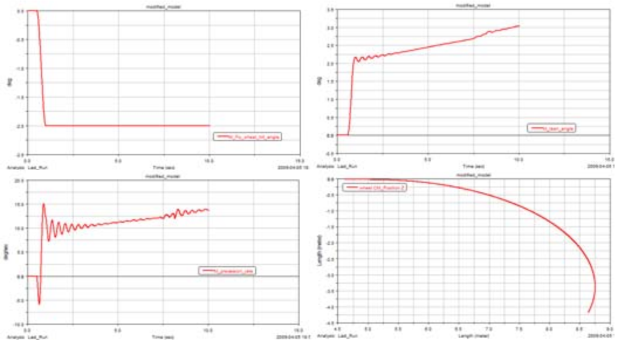


Fig. 6: Simulation result for Tilt Test 2
(from upper left clockwise: tilt angle, lean angle, contact point trajectory, precession rate)

It is also observed that if the flywheel tilts angle changes in positive direction, the wheel lean angle rotates in the other direction. In both simulations, after 1 s the tilt angle remains unchanged, the lean angle will converge to a steady-state value. This is in accordance with the equation [1].

V. SIMULATION WITH CONTROLLER DEFINED IN ADAMS

A closed loop controller is implemented in the ADAMS, using its internal template. Fig. 7 shows the control template available in ADAMS which can be used to realize closed loop controller for the mechanical system.

Design of the controller used here was reported in [5]. The controller is designed using a linearized model of the *gyrobot*. As a result, the controller may not work well when the lean angle is too large. Simulations show that the range of lean angle for which controller performs satisfactorily is [-20°, +20°]. The control law is given by equation (2), where the $\delta_{\beta,ref}$ is the incremental reference for the tilt angle of the flywheel [10]. It is interdependent with the reference precession rate. Gains of the controller are chosen as $k_1 = 3$ and $k_2 = 0.8$.

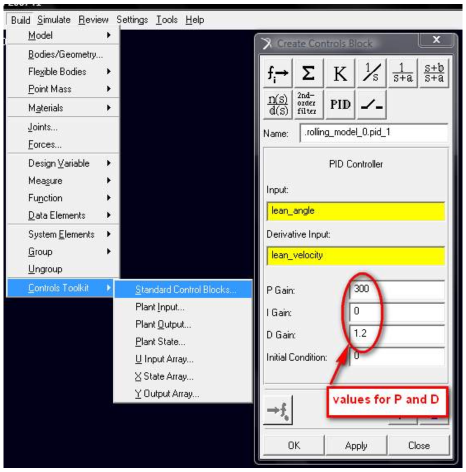
$$u_2 = k_1(\delta_{\beta,ref} - \delta_\beta) + k_2\dot{\delta}_\beta \qquad (2)$$

Case I: Lean the wheel from 0° to 10°

Initial conditions:

- Wheel lean angle: 0 deg
- Flywheel lean angle: 0 deg
- Wheel lean & Flywheel tilt velocity: 0 deg/s
- Wheel velocity: 0 rpm
- Reference wheel lean angle: 10°

The results of this simulation are shown in Fig. 8.



Name: .rolling_model_0.pid_1

Input: lean_angle

Derivative Input: lean_velocity

P Gain: 300

I Gain: 0

D Gain: 1.2

Initial Condition: 0

Fig. 7: Control template to create a PD controller

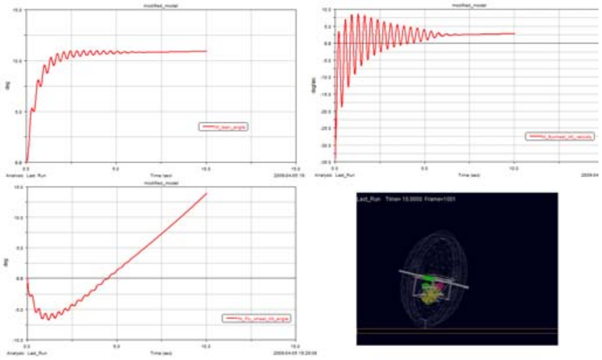


Fig. 8: Simulation Result with $\delta_{\beta,ref} = 10°$
(clockwise from upper left: lean angle, tilt velocity, wheel 3D view, tilt angle)

Case II: Lean the wheel from 0° to 10° while rotating at 30 rpm

Initial conditions:

- Wheel lean angle: 0°
- Flywheel lean angle: 0°
- Wheel lean & Flywheel tilt velocity: 0 deg/s
- Wheel velocity: 30 rpm
- Reference wheel lean angle: 10°

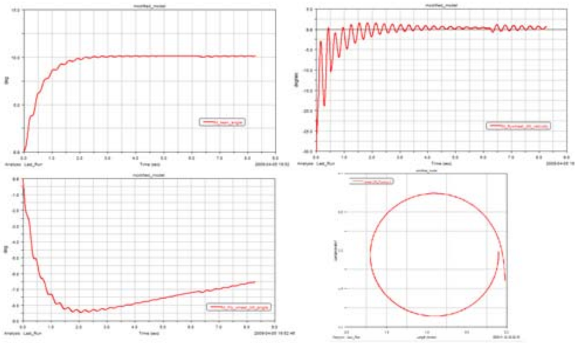


Fig. 9: Simulation Result with $\delta_{\beta,ref} = 10°$ and wheel rolling at 30 rpm
(clockwise from upper left: lean angle, tilt velocity, robot trajectory, tilt angle)

Case III: Lean the wheel from -5° to 5° with wheel rolling at the rate of 30 rpm

Initial conditions:

- Wheel lean angle: 0 deg
- Flywheel lean angle: 0deg
- Wheel lean & Flywheel tilt velocity: 0 deg/s
- Wheel velocity: 30 rpm
- Reference wheel lean angle: -5° to +5°

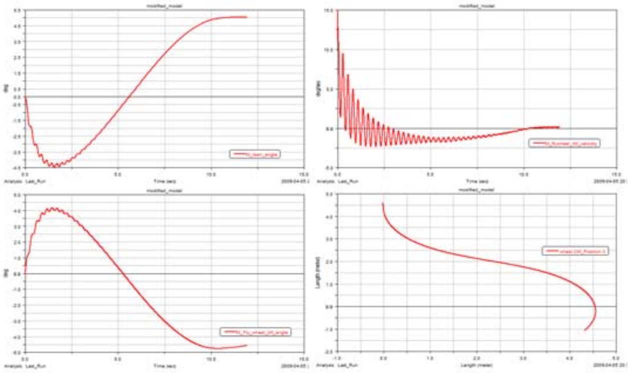


Fig. 10: Simulation Result with $\delta_{\beta,ref}$ changing from 0° to -5° to +5° and wheel running at 30 rpm
(from upper left clockwise: lean angle, tilt velocity, robot trajectory, tilt angle)

VI. CONCLUSION

Dynamic simulation of *Gyrobot* in ADAMS environment has been successfully developed. Several tests have been carried out to verify the *gyrobot* operation principle, namely the law of conservation of angular momentum and gyroscopic procession. Test results show that the virtual robot behavior is in accordance with the two principles. This verifies that the virtual *gyrobot* created in ADAMS represents the actual *gyrobot*.

A controller is also implemented to further justify the effectiveness of the virtual robot. The virtual prototype can now be used for various investigations which are otherwise time-consuming or costly.

Only the internal control template provided by ADAMS can be used to design and implement closed loop operation. This limits the flexibility of this virtual system as a tool for controller design. We are currently working on the co-simulation of ADAMS with MATLAB/SIMULINK which is a powerful tool for control design.

REFERENCES

[1] N. Orlandea, "*Development and application of node-analogous sparsity-oriented methods for simulation of mechanical dynamic systems*", PhD thesis, University of Michigan, Ann Arbor MI, 1973

[2] R. Rampalli, "ADAMS – a Sparse matrix approach to solving multi-body dynamics problems", NASA Workshop on Multi-Body Simulation, California Institute of Technology, 1987

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Coefficient Of Friction Table: *http://www.roymech.co.uk/Useful_Tables/Tribology/co_of_frict.htm*

[8] Y. Xu, H.B. Brown and K.W. Au, "Dynamic mobility with single-wheel configuration", *The International Journal of Robotics Research,* Vol. 18, No. 7, pp. 728-738, July 1999

[9] A. Al-Mamun, Z. Zhu, P. Vadakkepat and T.H. Lee, ''Path following controller for gyroscopically stabilized single-wheel robot", *Proc. of the 9th WSEAS International Conf. on Automatic Control, Modeling and Simulation*, Istanbul, Turkey, 27-20 May. 2007.

[10] Y. Xu and K.W. Au, "Stabilization and path following of a single wheel robot", *IEEE/ASME Transaction on Mechatronics*, vol.9, pp. 407-419, 2004.