EE3060

Control Engineering

Gyroscopic Stabilization of Bicycle using CMG

INDIAN INSTITUTE
OF TECHNOLOGY
PALAKKAD

Submitted by:

Joel Joy	**121901018**
Antony Jerald	**121901004**

Abstract

In this work the self balancing of a bicycle is achieved using a Control Moment Gyro (CMG) and a PID controller. The CMG is made up of a spinning rotor with a large, constant angular momentum which can be rotated to vary the angular momentum vector direction for a bicycle. The CMG acts as a torque amplifier. Here we derive the Equations of Motion of the bicycle system and linearize it to a second order system. The transfer fuction of the system is derived from state space representation of the system. The sytability of the system is then analysed and a PID controller was used to stabilise this unstable system.

1 Objective

The goal of this project is to deduce the equations of motion for the dynamics of a bicycle and use a feedback controller with a Control Moment Gyro (CMG) as an actuator to stabilise a bicycle.

2 Motivation

People all across the world utilise bicycles as one of the cheapest and most environmental friendly modes of transportation. It has been widely utilised for transportation, leisure, travel, and pleasure over the years. Increased use of bicycles for short-distance transport can help to alleviate traffic congestion and air pollution, which are already prevalent in most Indian cities. Cycling is also an excellent kind of exercise for people.

If this two-wheeled bicycle is made to self balance then it can be ridden by individuals of all ages, including children and the elderly. Bicycle is a nonlinear system that is inherently unstable. It behaves like an inverted pendulum in terms of dynamic behaviour. When guided to keep its centre of mass over its wheels, a bicycle may be made to stay upright. Control engineering aids in the development of a wide range of dynamic systems and controllers that enable the systems to behave as intended. Sensors that detect the bicycle's lean angle can be attached to a bicycle and actuators can be used to return it back to a balanced position anytime it deviates from it.

3 State of Art

Various ways for making bicycles self-balance have been proposed in the past. For balancing a bicycle, the gyroscopic stabilisation method uses two connected gyroscopes rotating in opposing directions. The gyroscopic torque generated by the precision of the gyroscope is employed to destabilise the torque caused by gravity forces. The load mass balance system proposed for self-balancing bicycles works by changing the bicycle's centre of mass right or left to turn the bicycle right or left. Another method employs steering control to balance the bicycle, with the steering angle controlled by a servo motor and the forward speed maintained by an electric motor. The Murata Boy, a well-known self-balancing bicycle, balances itself by using a reaction wheel as an actuator and torque generator. In the reaction wheel, the rotating rotor has a nominal spin rate of zero. Its spin axis is attached to the bicycle, and its speed is adjusted to create response torque around the axis. This technology is low-cost and simple, but it cannot create significant quantities of torque and requires more energy. In another method self-balancing of a bicycle is achieved by regulating the torque applied on the steering handlebar. In this approach the amount of torque supplied to the handlebar is controlled by a controller depending on the amount of roll. This method has the advantages of being low-energy and low-mass, but it is not resistant against big roll perturbations.

4 Our Approach

4.1 CMG-Controlled Bicycle

Here we use a single-axis Control Moment Gyroscope (CMG) for balancing the bicycle. It consists of a spinning rotor and motorised gimbals that tilt the rotor's angular momentum. The roll angle is monitored using an inertia measuring unit (IMU sensor). This roll data is given to a controller when the bicycle gets inclined to any-side. The controller then commands the gimbal motor on the CMG to rotate in order to create gyroscopic precession torque. This gyroscopic precession torque aids in maintaining the upright position of the bicycle. The components and vectors of a single gimbal CMG are shown in Figure 1.

The system we're looking at includes two rigid body connections: the first is the bicycle frame, which has one DOF (degree of freedom) along the Z-axis, and the second is the flywheel, as indicated

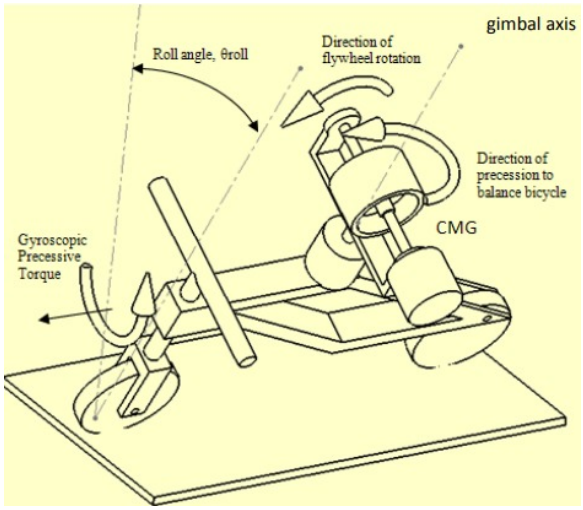

Figure 1: Balancing of bicycle using CMG-generated gyroscopic precession torque

in Figure 2. The flywheel's COG (Center of Gravity) is fixed in relation to the bicycle frame, and the flywheel's speed ω is assumed to be constant. F_{cg} and B_{cg} represent the Flywheel's and the bicycle's centre of mass, respectively.

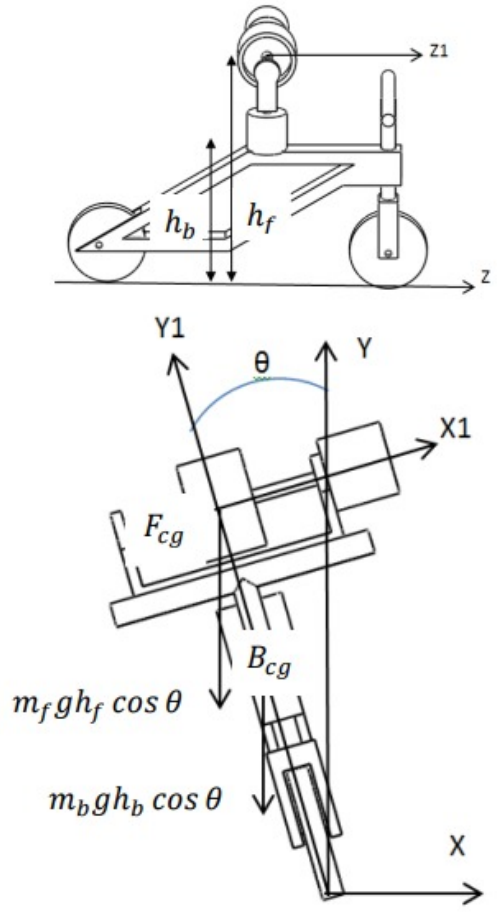

Figure 2: Reference coordinates for the bicycle.

4.2 EOM for CMG-Controlled Bicycle

Here θ is taken as the roll angle around the Z-axis and $\dot{\theta}$ represents the bicycle's angular velocity around the Z-axis. δ represents angular position of flywheel around gimbal axis and $\dot{\delta}$ represents the flywheel's angular velocity around the gimbal axis. Since the flywheel's centre of gravity does not move in relation to the bicycle's centre of gravity, absolute velocities of F_{cg} and B_{cg} remain same.

We can have absolute value of velocities of bicycle and flywheel as:

$$|V_b| = \dot{\theta} h_B \tag{1}$$
$$|V_f| = \dot{\theta} h_f \tag{2}$$

where h_b is the height of the centre of gravity of the bicycle and h_f is the height of the centre of gravity of flywheel.

We can develop the dynamic model of the system using the Lagrange equation as follows:

$$\frac{d}{dt}\left\{\frac{\partial T}{\partial q_i}\right\} - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{3}$$

Here V represents the system's total potential energy, T represents the system's total kinetic energy, Q_i is the total external force and q_i is the generalized coordinate system.

Since there are two links in our system (bicycle and CMG), the total potential energy will be sum of the potential energy of both. So from Fig 2, we can calculate V as:

$$V = m_b g h_b \cos\theta + m_f g h_f \cos\theta \tag{4}$$

Also, the total kinetic energy of the system (T) can be calculated as:

$$T = \frac{1}{2} m_b \left(|v_b|\right)^2 + \frac{1}{2} m_f \left(|v_f|\right)^2 + \frac{1}{2} I_b \dot{\theta}^2 \\ \frac{1}{2}\left[I_r \dot{\delta}^2 + I_p (\dot{\theta}\sin\delta)^2 + I_r (\dot{\theta}\cos\delta)^2\right] \tag{5}$$

In the above equation m_b and m_f are the mass of the bicycle and flywheel respectively. I_r represents the flywheel radial moment of inertia around the centre of gravity, I_p represents the flywheel's polar moment of inertia around the centre of gravity and I_b represents the bicycle moment of inertia around the ground contact line.

On substituting (1) & (2) in (5) we get :

$$T = \frac{1}{2} m_b \left(\dot{\theta}^2 h_b^2\right) + \frac{1}{2} m_f \left(\dot{\theta}^2 h_f^2\right) + \frac{1}{2} I_b \dot{\theta}^2 \\ \frac{1}{2}\left[I_r \dot{\delta}^2 + I_p (\dot{\theta}\sin\delta)^2 + I_r (\dot{\theta}\cos\delta)^2\right] \tag{6}$$

Substituting q_i as the the roll angle θ in (3), Lagrange equation we get

$$\frac{d}{dt}\left\{\frac{\partial T}{\partial \dot{\theta}}\right\} - \frac{\partial T}{\partial \theta} + \frac{\partial V}{\partial \theta} = Q_\theta \tag{7}$$

Taking partial derivatives of V & T given in equation (4) & (6) respectively and substituting in (7) we get :

$$\begin{aligned}\ddot{\theta}\left[m_b h_b^2 + m_f h_f^2 + I_b + I_p \sin^2\delta + I_r \cos^2\delta\right] \\ +2\sin\delta\cos\delta\left(I_p - I_r\right)\dot{\theta}\dot{\delta} \\ -g\left(m_b h_b + m_f h_f\right)\sin\theta = I_p\omega\delta\cos\delta\end{aligned} \tag{8}$$

Taking q_i as δ, the angular position of the flywheel about its gimbal axis and substituting in (3) Lagrange equation we get :

$$\frac{d}{dt}\left\{\frac{\partial T}{\partial \dot{\delta}}\right\} - \frac{\partial T}{\partial \delta} + \frac{\partial V}{\partial \delta} = Q_\delta \tag{9}$$

Taking partial derivatives of V & T given in equation (4) & (6) and substituting in (9) we get :

$$\begin{aligned}\ddot{\delta} I_r - \dot{\theta}^2\left(I_p - I_r\right)\sin\delta\cos\delta \\ = T_m - I_p\omega\dot{\theta}\cos\delta - B_m\dot{\delta}\end{aligned} \tag{10}$$

Here B_m denotes the viscosity coefficient of a DC motor.

We also assume that the DC motor is connected to the Flywheel's gimbal through a final 65:1 ratio combining a beltdrive and a planetary gear head. So we get the equation for torque generated by the motor as:

$$T_m = 65 K_m i \tag{11}$$

where K_m is the torque constant and i is the input current. Substituting (11) in (10) gives :

$$\begin{aligned}\ddot{\delta} I_r - \dot{\theta}^2\left(I_p - I_r\right)\sin\delta\cos\delta \\ = 65 K_m i - I_p\omega\dot{\theta}\cos\delta - B_m\dot{\delta}\end{aligned} \tag{12}$$

We also get the expression for voltage applied to the motor, (U) as:

$$U = L\frac{di}{dt} + R_i + K_e\dot{\delta} \tag{13}$$

where L represents inductance of the motor, K_e presents back EMF constant of the motor and R_i represents resistance of the motor.

From the above derivation we can summarise the equations of motion for the bicycle model as:

$$\begin{aligned}\ddot{\theta}\left[m_b h_b^2 + m_f h_f^2 + I_b + I_p \sin^2\delta + I_r \cos^2\delta\right] \\ +2\sin\delta\cos\delta\left(I_p - I_r\right)\dot{\theta}\dot{\delta} \\ -g\left(m_b h_b + m_f h_f\right)\sin\theta = I_p\omega\delta\cos\delta\end{aligned} \tag{14}$$

$$\ddot{\delta} = \frac{T_m - I_p\omega\dot{\theta}\cos\delta - B_m\dot{\delta} + \dot{\theta}^2\left(I_p - I_r\right)\sin\delta\cos\delta}{I_r} \tag{15}$$

The input to the system is,
$U = L\frac{di}{dt} + R_i + K_e\dot{\delta}$

The output of the system is,
Rate of precession of the CMG = $\ddot{\delta}$

This model is only concerned with the bicycle's balance. The additional inputs that allow the bicycle to be translated are unaffected by these dynamics.

4.3 Linearisation of EOM

Linearization facilitates the application of classical control theory to the development of effective real-time algorithms. The bicycle is designed to work within a fixed balancing range that does not alter significantly in order to keep the bicycle upright (equilibrium). A small-signal analysis method can be used to linearise the equations of motion of a bicycle around the equilibrium point ($\theta = \delta = 0$)

The non linear terms present in the equations of motion of a bicycle are: $\sin\delta$, $\cos\delta$ and $\sin\theta$
We can approximate these non-linear terms as
$\sin\delta \approx 0$, $\cos\delta \approx 1$ and $\sin\theta \approx 0$
Substituting these approximations in equations of motion of a bicycle (14) & (15) gives:

$$\begin{aligned}\theta\left[m_b h_b^2 + m_f h_f^2 + I_b + I_r\right] - g\left(m_b h_b + m_f h_f\right)\theta \\ -I_p\omega\dot{\delta} = 0\end{aligned} \tag{16}$$

$$\ddot{\delta}\mathrm{I_r} - \mathrm{I_p}\omega\dot{\theta} + \mathrm{B_m}\dot{\delta} - 65\ \mathrm{K_m i} = 0 \tag{17}$$

$$\text{Taking, x} = \begin{bmatrix}\theta \\ \dot{\theta} \\ \delta \\ \dot{\delta}\end{bmatrix}, u = U \text{ and } y = \theta$$

We can get state space representation of the dynamic model of the system by combining equations (13), (16) and (17) as:

$$\dot{x} = Ax + Bu \qquad (18)$$
$$y = Cx + Du \qquad (19)$$

where A =

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g(m_b h_b + m_f h_f)}{m_b h_b^2 + m_f h_f^2 + I_b + I_r} & 0 & \frac{I_p \omega}{m_b h_b^2 + m_f h_f^2 + I_b + I_r} & 0 \\ 0 & -\frac{I_p \omega}{I_r} & -\frac{B_m}{I_r} & \frac{65 K_m}{I_r} \\ 0 & 0 & -\frac{K_e}{L} & -\frac{R}{L} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

and $D = [0]$

Calculating the matrix A, B, C and D using bycle parameters from below table gives

Parameters	Value	Unit	Description
m_f	2.02	kg	Mass of flywheel
m_b	20.6	kg	Mass of bicycle
h_f	0.58	m	Flywheel COG upright height
h_b	0.49	m	Bicycle COG upright height
I_b	2.1	kg.m^2	Bicycle moment of inertia around ground contact line
I_p	0.0088	kg.m^2	Flywheel polar moment of inertia around COG
I_r	0.0224	kg.m^2	Flywheel radial moment of inertia around COG
ω	469	rad/s	Flywheel angular velocity
L	0.000119	H	Motor Inductance
R	0.61	Ω	Motor Resistance
B_m	0.003	kg.m^2/s	Motor viscosity coefficient
K_m	0.0259	Nm/A	Motor torque constant
K_e	0.0027	V.s	Motor back emf constant
g	9.81	m/s^2	Gravitational acceleration

Figure 3: Bicycle Parameters

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 14.26 & 0 & 0.53 & 0 \\ 0 & -184.56 & -0.14 & 75.03 \\ 0 & 0 & -22.69 & -5126 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 8403 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

and $D = [0]$

Using these state variables we can calculate transfer function for the system as

$$\frac{\theta(s)}{U(s)} = \frac{336422}{s^4 + 5126s^3 + 2475s^2 + 429984s - 34115}$$

5 Results and Discussion

5.1 Uncompensated System

The self-balancing bicycle system is represented by the transfer function,

$$G(s) = \frac{336422}{s^4 + 5126s^3 + 2475s^2 + 429984s - 34115}$$

Pole-Zero plot of the system is shown in the Figure 4. A zoomed version is also shown in Figure 5 to get a closer view of the poles near imaginary axis. The poles are located at (-5125.7, -0.27256±9.1408i,0.079413). The presence of a pole at RHP makes the overall system unstable.

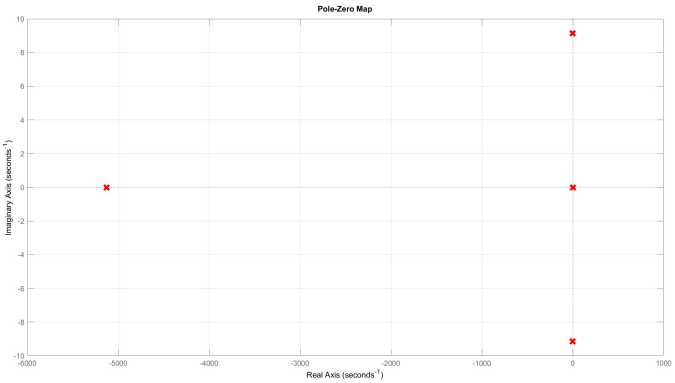

Figure 4: Pole-Zero plot of the system

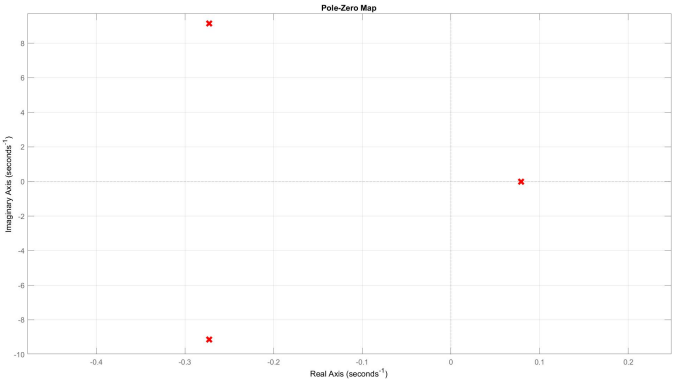

Figure 5: Zoomed Pole-Zero

The bode plot for the system is shown in Figure 6.

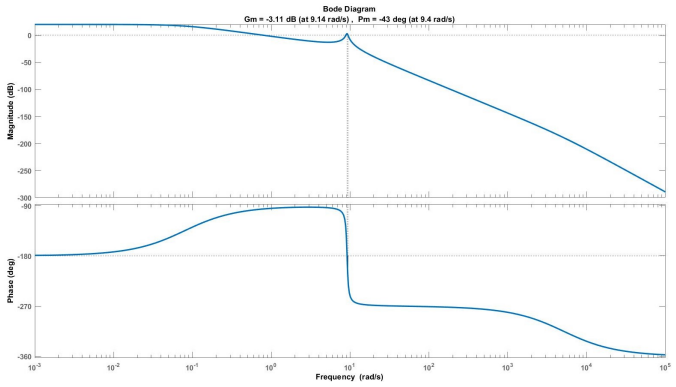

Figure 6: Bode Plot of system

From the bode plot we can see that the Gain Margin (Gm) is -3.11dB and Phase Margin (Pm) is -43°. The negative values indicate that the system is unstable. A small disturbance will cause the system output (Roll Angle) to go unbounded. This can be seen from the step response of the system shown in Figure 7.

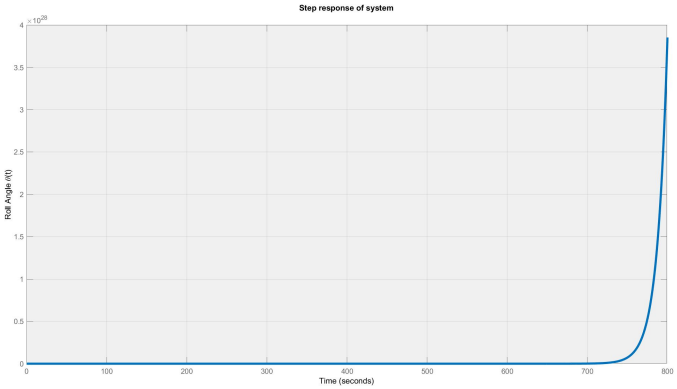

Figure 7: Step response of system

We can thus conclude that a controller is needed to make the system stable.

5.2 Compensation using Proportional Controller

Let us first analyze the root locus of the system shown in Figure 8.

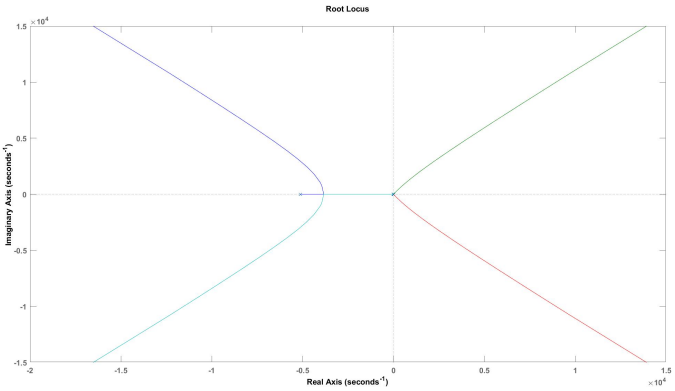

Figure 8: Root locus of system

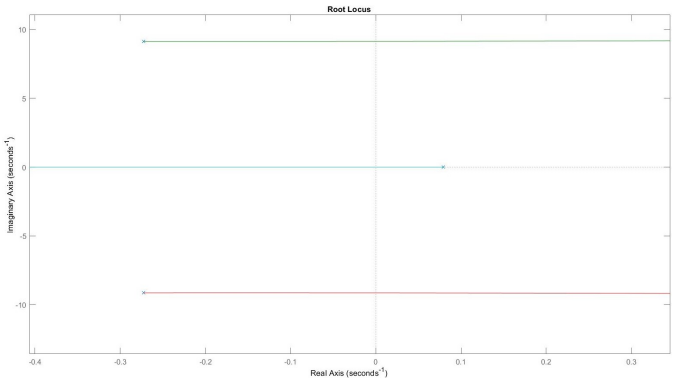

Figure 9: Zoomed version Root locus

From the root locus we can see that for some value of proportional gain K, all the poles would be in the LHP. After increasing the value of K slowly from 0, the value chosen was K=0.62. The transfer function of the closed loop becomes,

$$P(s) = \frac{KG(s)}{1 + KG(s)}$$

Pole Zero for closed loop system using Proportional Controller is shown in Figure 10.

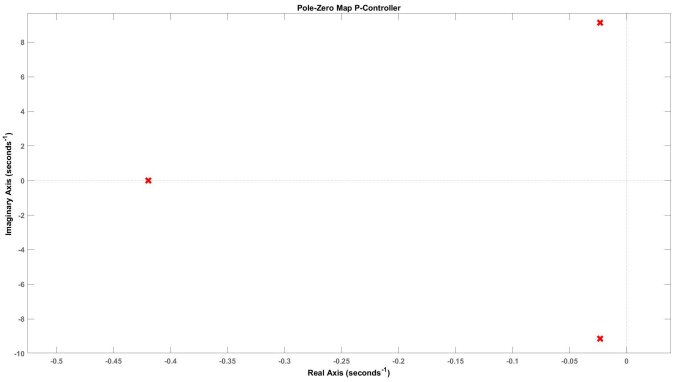

Figure 10: Pole-Zero for system with P controller

It can be seen that all the poles are now in the LHP. Bode plot is shown in Figure 11. The positive values for Gain Margin and Phase margin shows that the system has become stable when a Proportional Controller with K=0.62 was used.

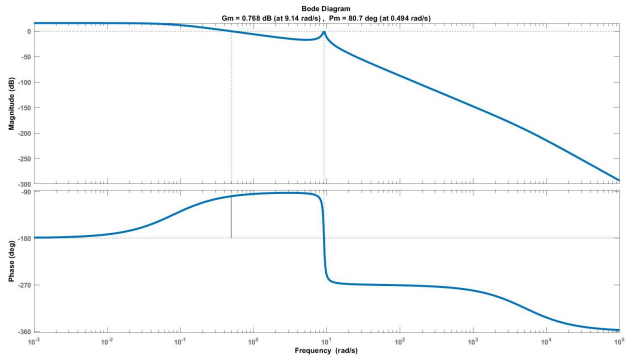

Figure 11: Bode Plot for P-Controller

Step response is shown in Figure 12. Although the Roll Angle output reaches steady state after some time, we can see that there exists a steady state error whose value is around 0.2 deg. We want the steady to be equal to the given input and hence the Proportional controller isn't enough.

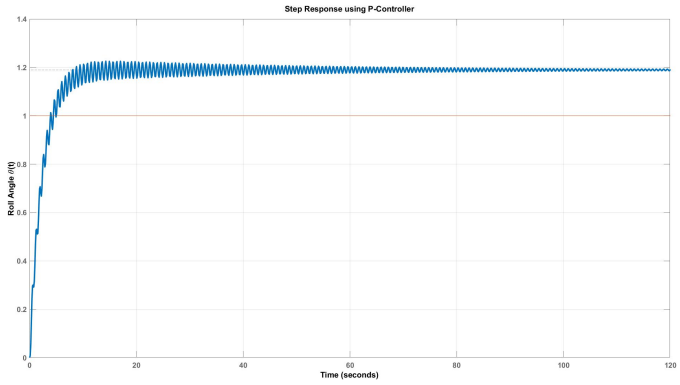

Figure 12: Step response of system using P-Controller

5.3 Compensation using Proportional Derivative (PD) Controller

Along with the proportional term K_P, a derivative term is also added (K_D) in the controller whose transfer function becomes C(s) = $K_D s + K_P$. We need to choose values for proportional gain K_P and Derivative gain K_D to complete the design of the controller.

Tuning PD controller using Ziegler Nichols

To tune the PD controller, we use a heuristic tuning method known as the Ziegler-Nichols method developed by John G. Ziegler and Nathaniel B. Nichols. The method works by initially setting K_D to 0. The proportional gain is slowly increased from 0 until it reaches a maximum gain K_U where we get a stable and sustained oscillation of the output θ_b. We then need to find the time period of oscillation T_U. The maximum gain K_U was found to be **0.64**. From the plot shown in Figure 13, the value of T_U was found to be **685ms**.

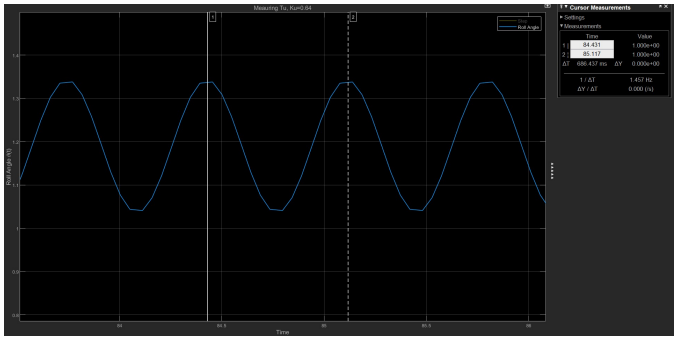

Figure 13: T_U

Since we have the values of K_U and T_U, K_P and K_D can be obtained as ,

$$K_P = 0.8K_U = 0.512$$

$$K_D = 0.1K_U T_u = 0.04384$$

Now we are able to design the PD controller using the values obtained. The pole-zero location for the compensated system using PD controller is shown in Figure 14.

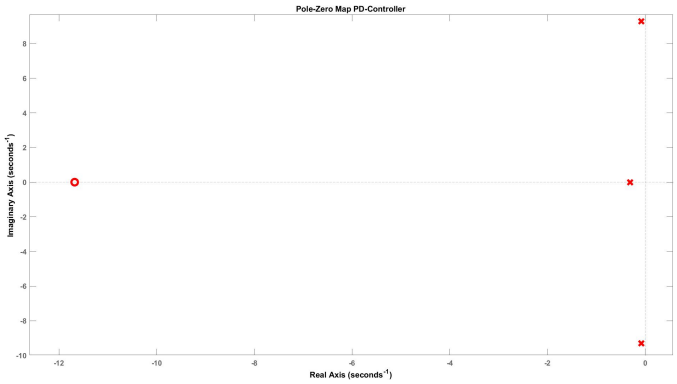

Figure 14: PD Pole-Zero

All the poles are in the LHP and hence the compensated system is stable. We can also see an additional zero on the real axis. The bode plot generated is shown in Figure 15. The new gain margin (G_m) is **3.05dB** which is more than the uncompensated system. The phase margin was found to be 80.3°.

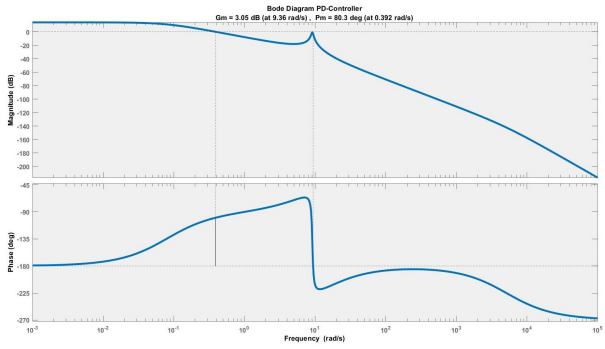

Figure 15: PD Bode Plot

Now a step input is given to the compensated system. The response is shown in Figure 16 We can see that the oscillations have reduced and settling time as well as the overshoot has also decreased. But still it has a steady state error around 0.2deg. We can see that the PD controller does not have any influence on the steady state error. Hence it is also not ideal for our purpose.

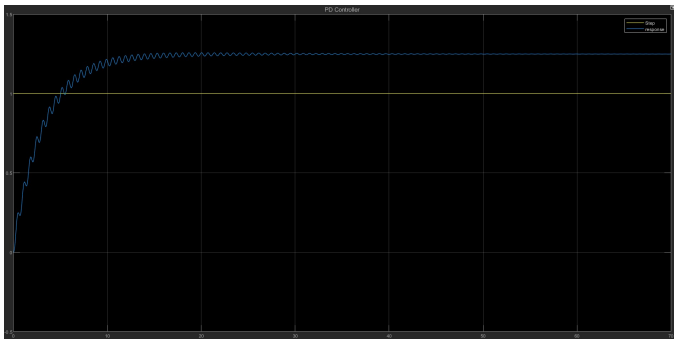

Figure 16: Step response of system using PD controller

$T_{rise}(s)$	6.80
$T_{peak}(s)$	19.36
$T_{set}(s)$	15.67

Table 1: Critical Parameters PD-controller

5.4 Compensation using Proportional Integral Derivative (PID) Controller

PID controller is the most popular feedback controller used in control system. It combines the power of the proportional, integral and derivative controllers. PID controller has a transfer function $C(s) = K_P + \frac{K_I}{s} + K_D s$. Along with K_P and K_D, we also need to find the integral gain K_I.

We will again use the Ziegler-Nichols method which we had used earlier. K_U and T_U were obtained as 0.64 and 0.685 respectively. With these 2 values, the proportional, integral and derivative gains can be obtained as,

$K_P = 0.6K_U$	$K_I = 1.2\frac{K_U}{T_U}$	$K_D = 0.075K_U T_U$
0.384	1.1217	0.03288

The pole-zero plot for the compensated system using PID is shown in Figure 17. All the poles are in the LHP. PID controller addionally gives 2 zeros and 1 pole. We can find the 2 zeros at -5.8394.

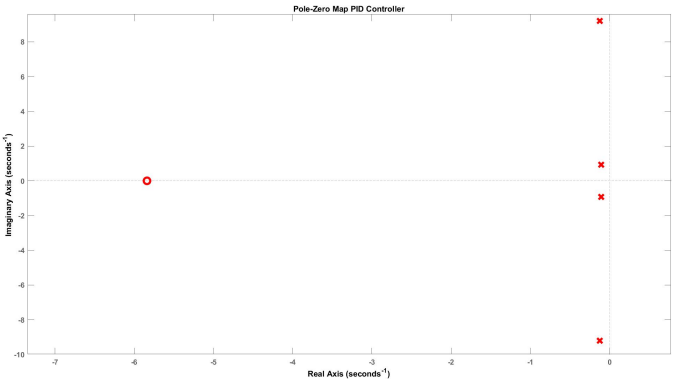

Figure 17: Pole-Zero PID

Bode plot for the open loop system is shown in Figure 18. We can see that the gain margin has increased further to 5.41 dB. Phase margin (P_m) is 13.3 deg.

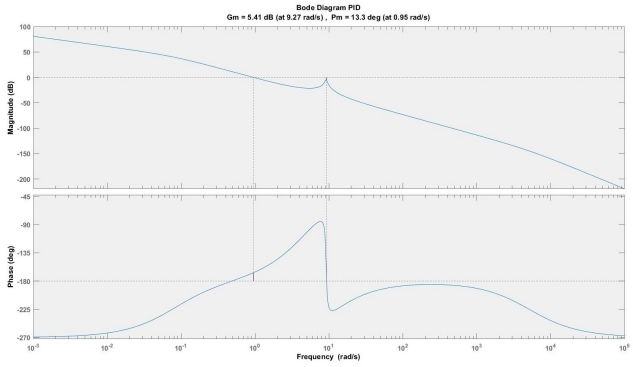

Figure 18: Bode Plot PID

The step respone of the system is now analyzed (shown in Figure 19). We can see that the Roll Angle $\theta(t)$ attains stable value after some time.

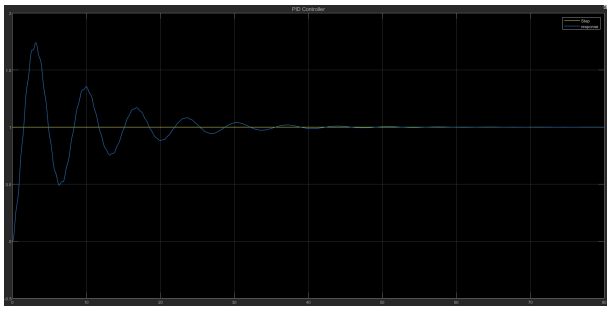

Figure 19: Step response of system using PID

$T_{rise}(s)$	1.13
$T_{peak}(s)$	1.74
$T_{set}(s)$	34.51
$\%OS$	73

Table 2: Critical Parameters PID-controller

While keeping K_D and K_P constant, K_I is varied. The responses obtained are shown in Figure 20.

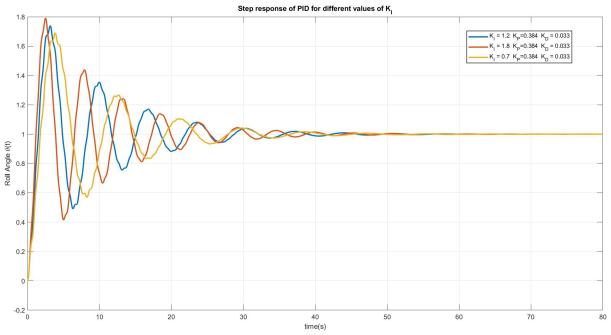

Figure 20: Output for different values of K_I

$K_I = 0.7$	$T_{rise}(s)$	1.39
	$T_{peak}(s)$	3.84
	$T_{set}(s)$	35.10
	%OS	69
$K_I = 1.2$	$T_{rise}(s)$	1.13
	$T_{peak}(s)$	3.16
	$T_{set}(s)$	34.52
	%OS	73.95
$K_I = 1.8$	$T_{rise}(s)$	0.83
	$T_{peak}(s)$	2.51
	$T_{set}(s)$	35.09
	%OS	79.16

From the responses obtained we can infer that as we increase K_I the response becomes faster but the overshoot as well as oscillations also increases. Hence depending on our requirement we can adjust the value of K_I.

Similarly, K_D also has its effect on different parameters. As we increase the value of K_D, the overshoot and settling time decreases but it does not affect the steady state error.

The proportional gain term K_P increases the overshoot but it can decreases the steady state error and rise time.

When we use the Ziegler Nichols method to tune the PID, we typically get an aggressive gain and overshoot. However, the values obtained can be used as an initial point for tuning. Since we know the effects of K_P, K_I and K_D independently, we can tune it as per our requirements.

5.5 Automated PID Tuning using SIMULINK

Earlier, we used the Ziegler-Nichols method to manually find a suitable point and then tune accordingly. SIMULINK has an auto tune option which would find an point for us with suitable K_P, K_I and K_D values. Rather than a normal derivative controller, a filtered derivative controller is used. In the laplace domain, it can be expressed as $\frac{K_D s}{s+\frac{1}{N}}$.

After the auto-tune process, values obtained are,

K_P	K_I	K_D	N
0.78057	0.04562	-0.286	2.1887

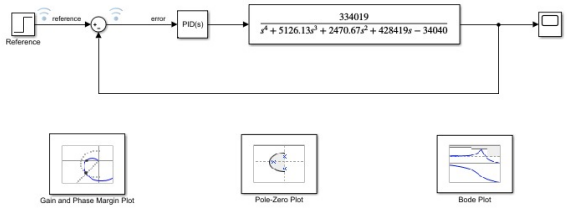

Figure 21

The step response of the system is shown below.

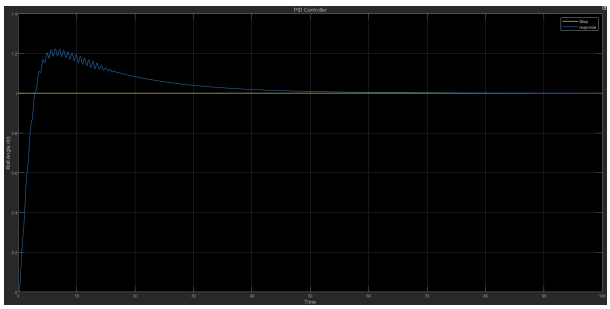

Figure 22: Step response of auto-tuned PID

Similar to what we had done earlier, the variation in response was analyzed by changing the values of K_I.

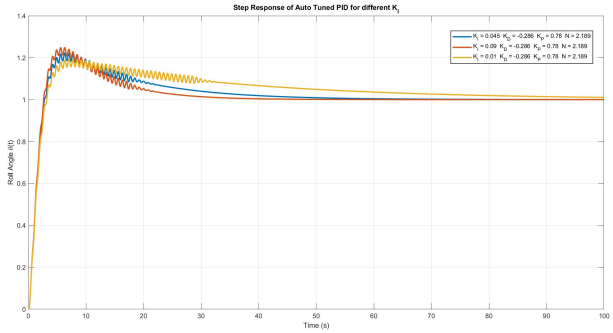

Figure 23: Output for different values of K_I

The variation in Settling time, Overshoot, Peak time, Rise time are shown. The behaviour of critical parameters are same as we had seen earlier. As the value of K_I increases, Rise time decreases, Peak time also decreases. However, the percentage of overshoot increases.

$K_I = 0.01$	$T_{rise}(s)$	2.19
	$T_{peak}(s)$	7.77
	$T_{set}(s)$	65.193
	%OS	17.98
$K_I = 0.045$	$T_{rise}(s)$	2.07
	$T_{peak}(s)$	6.34
	$T_{set}(s)$	39.19
	%OS	22.22
$K_I = 0.09$	$T_{rise}(s)$	1.97
	$T_{peak}(s)$	6.32
	$T_{set}(s)$	27.17
	%OS	24.84

5.6 Analyzing Disturbance

Using the PID controller we designed, the response of system to a disturbance is analyzed. The Auto-tuned PID model was taken for analysis. A SIMULINK model (Figure 24) of the system was developed. Since we want the bicycle to stay upright (Roll Angle $\theta = 0$), the reference signal given to the model is 0. The system is stable if the bicycle becomes upright after some time.

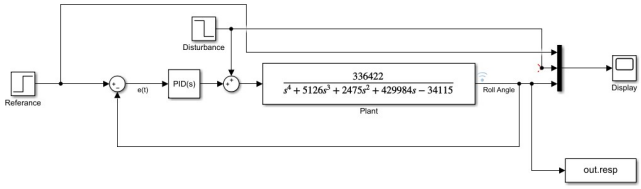

Figure 24: SIMULINK Model with Disturbance

The response of the system is shown in Figures 25 to 27 for different disturbances.

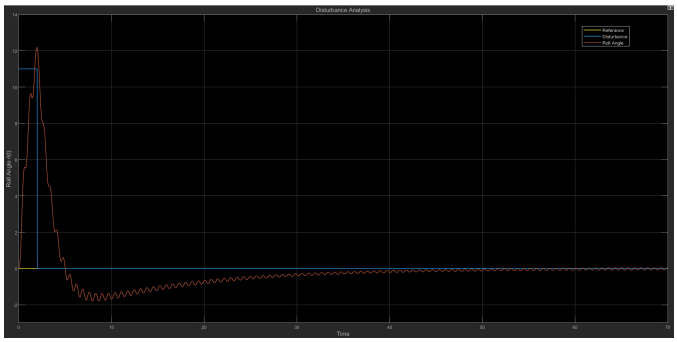

Figure 25

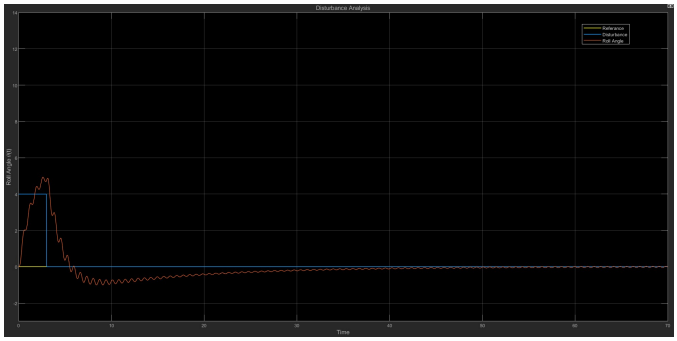

Figure 26

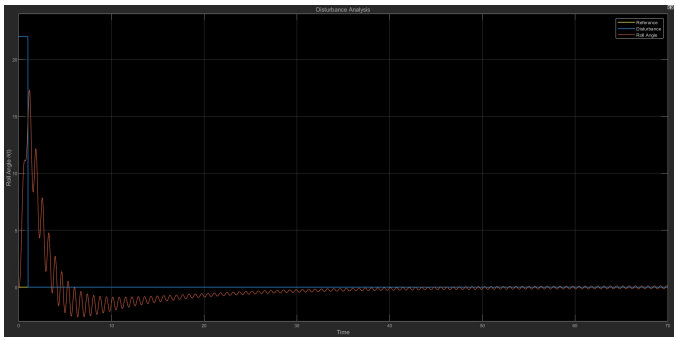

Figure 27

Hence we can see that the PID controller makes the overall system stable. The Bicycle comes back to its upright position after some time.

6 Conclusion

In this work, to balance a bicycle we utilise a Control Moment Gyro (CMG) and a PID controller. The CMG is a device with a quick response time and excellent torque amplification property and here it was employed as a momentum exchange actuator. The linearized equations of motion of the system was derived and the transfer function formulated from the state space representation of the system was found to be unstable due to a pole in RHP. The uncompensated system had negative phase and gain margins. Then

we analyze the effect of using a proportional (P) , proportional-derivative (PD) and Proportional-Integral-Derivative (PID) controllers for stabilising the system. PID controller was found have the best performance in stabilising the bicycle. The effect of the integral gain term K_I on the critical parameters was also found out comparing the responses. The above model was simulated in Matlab and Simulink. Also the response of system to a disturbance is analyzed using the PID controller we designed.

7 Future Perspective

This CMG based self balancing method for self balancing bicycle is incapable of adapting to changes such as an increase in cargo, which will have an impact on the Center of Gravity. In addition, this method consumes more power and is quite heavier. By integrating sensors such as a Light Detection and Ranging sensors, Accelerometer, Gyroscope and equipping it with RGBD depth camera it can detect and map the surroundings, the idea may be evolved further into an autonomous self-balancing bicycle.

8 Code and SIMULINK Model

Mathlab Code and SIMULINK Model can be found Here

References

[1] Pom Yuan Lam and Tan Kian Sin. Gyroscopic stabilization of a self-balancing robot bicycle. *International Journal of Automation Technology*, 5(6):916–923, 2011.

[2] Harun Yetkin, Simon Kalouche, Michael Vernier, Gregory Colvin, Keith Redmill, and Umit Ozguner. Gyroscopic stabilization of an unmanned bicycle. pages 4549–4554, 06 2014.