

## PERANCANGAN DAN IMPLEMENTASI KONTROL PERGERAKAN MAJU PADA ELECTRIC UNICYCLE

Design and Implementation of Forward Movement Control of Electric Unicycle  
Ni Luh Andrea Maurilla Sarasvanya<sup>1</sup>, Angga Rusdinar<sup>2</sup>, Cahyantari Ekaputri<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[andreamaurilla@students.telkomuniversity.ac.id](mailto:andreamaurilla@students.telkomuniversity.ac.id), <sup>2</sup>[anggarusdinar@telkomuniversity.ac.id](mailto:anggarusdinar@telkomuniversity.ac.id),

<sup>3</sup>[cahyantarie@telkomuniversity.ac.id](mailto:cahyantarie@telkomuniversity.ac.id)

### Abstrak

Kendaraan yang memiliki fleksibilitas yang tinggi tanpa mengesampingkan kenyamanan dan keamanan saat dikendarai merupakan solusi dari permasalahan kendaraan bermotor saat ini, khususnya sepeda motor. Dikarenakan beroperasi menggunakan tenaga listrik, electric unicycle memiliki keunggulan yaitu menghemat bahan bakar dan tidak menyebabkan polusi udara. Electric unicycle ini dilengkapi GY521 MPU6050, yaitu modul terintegrasi yang terdiri dari sensor accelerometer untuk mendeteksi sudut kemiringan motor dan sensor gyroscope untuk menjaga keseimbangan motor. Arduino Uno digunakan sebagai controller utama dari keseluruhan sistem kendaraan, mulai dari pengolahan data sensor sampai navigasi kendaraan. Untuk memfilter pembacaan sensor agar bebas dari noise dan mengestimasi sudut kemiringan kendaraan, digunakan metode Kalman Filter. Sedangkan untuk kendali kecepatan motor digunakan metode PD. Pada tugas akhir ini digunakan parameter Kalman filter  $Q_{accelerometer} = 0.001$ ,  $Q_{bias} = 0.003$  dan  $R = 0.03$ . Sementara parameter PD yang digunakan adalah  $K_p = 5$  dan  $K_d = 1.9$ .

**Kata Kunci :** Electric Unicycle, Accelerometer, Gyroscope, Inverted Pendulum, Kalman Filter, PD

### Abstract

*Vehicles that have a high flexibility without neglecting the comfort and safety while driving is a solution to the problem nowadays especially for motorcycles. Operated using electrical power, the electric unicycle has advantages that it saves fuel and it's pollutant-free. Electric unicycle is equipped with GY521 MPU6050, which is an integrated module which consists of an accelerometer sensor for detecting the angle of the motor and gyroscope sensor to maintain the balance of the vehicle. Arduino Uno is used as a main controller to process the sensor data and to navigate the vehicle. To filter sensor readings to be noise-free and to estimate the angle of the vehicle, Kalman Filter method is used. As for the motor speed, control method used is PD. In this final project, parameters used for Kalman filtering process are  $Q_{accelerometer} = 0.001$ ,  $Q_{bias} = 0.003$  dan  $R = 0.03$ . Parameters used for PD controller are  $K_p = 5$  and  $K_d = 1.9$ .*

**Keywords :** Electric Unicycle, Accelerometer, Gyroscope, Inverted Pendulum, Kalman Filter, PD

### I. Pendahuluan

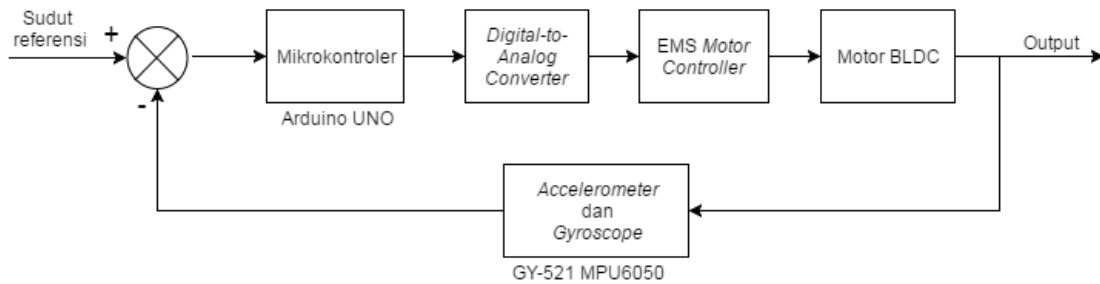
Di negara-negara berkembang seperti Indonesia, sepeda motor menjadi pilihan utama moda transportasi masyarakat. Harga yang relatif terjangkau dan fleksibilitasnya yang tinggi dalam menghadapi kondisi lalu lintas menjadi alasannya. Karena dapat dengan mudah dimiliki, maka penggunaan sepeda motor di Indonesia menjamur.

Penggunaan sepeda motor yang menjamur menyebabkan berbagai masalah. Konsumsi bahan bakar minyak menjadi semakin banyak dan otomatis menyebabkan kelangkaan. Selain itu emisi dari penggunaan bahan bakar yang melimpah menyebabkan polusi udara yang sangat meresahkan masyarakat. Solusi dari permasalahan di atas terjawab dengan munculnya inovasi mobil listrik karya anak bangsa. Namun mobil listrik belum sepenuhnya memenuhi keinginan para pengendara yang membutuhkan fleksibilitas dan mobilitas tinggi dalam berkendara.

Penyusun membuat kendaraan motor satu roda yang menggunakan energi listrik sebagai sumber dayanya. Kendaraan motor satu roda ini dapat dioperasikan di berbagai jalanan maupun di dalam rumah, perkantoran, rumah sakit, bandar udara, dan lainnya yang tidak terjangkau dengan sepeda motor biasanya. Keunggulan dari motor ini yaitu teknologi self-balancing yang memudahkan dalam berkendara karena sistem kendalinya otomatis. Teknologi self-balancing juga dipilih karena memiliki tingkat keamanan yang tinggi sehingga pengendara tidak perlu khawatir terjatuh.

## 2. Dasar Teori dan Perancangan

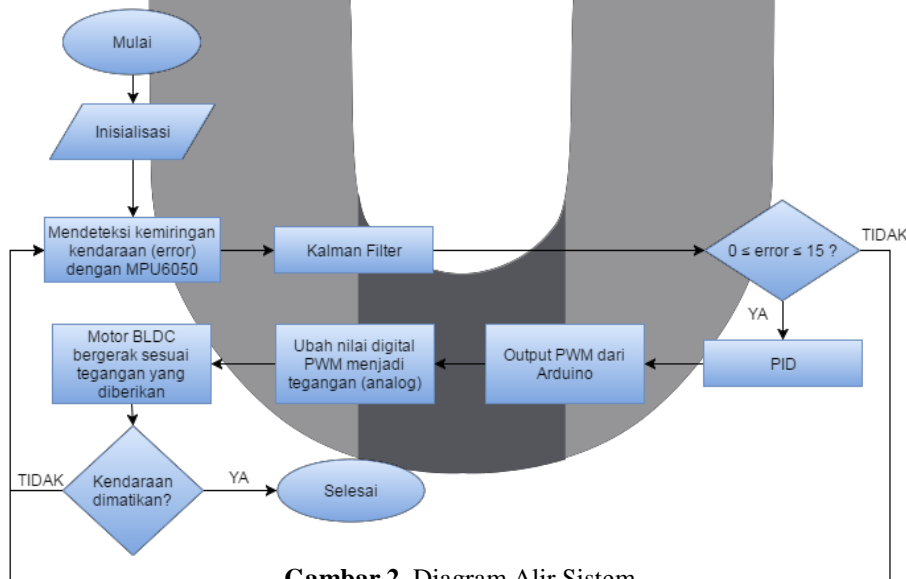
### 2.1. Perancangan Sistem



Gambar 1. Diagram Blok Sistem

*Electric Unicycle* merupakan kendaraan sepeda motor roda satu yang beroperasi dengan menerapkan prinsip *inverted pendulum*. Kendaraan ini menggunakan baterai jenis Lithium Polimer sebagai catu daya dan motor DC jenis *brushless* sebagai aktuator kendaraan. Modul GY521 MPU6050, yang terdapat pada kendaraan, terdiri dari 3 aksis *gyroscope* dan 3 aksis *accelerometer*. Output modul ini berupa informasi perubahan sudut yang akan mempengaruhi kecepatan kendaraan. Perubahan sudut kemiringan kendaraan diukur berdasarkan tolak ukur garis normal gravitasi terhadap bumi.

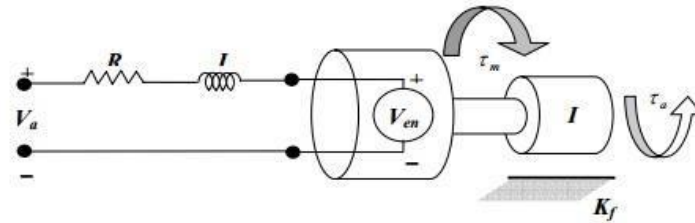
Data output modul GY521 MPU6050 akan dikirim dan diproses oleh Arduino UNO. Data tersebut terlebih dahulu akan mengalami proses filterisasi agar menghasilkan data output yang minim *noise*. Metode yang digunakan untuk filterisasi adalah Kalman Filter. Setelah proses filterisasi, data tersebut akan menjadi nilai *error* pada kontroler PID. Output kendali PID yang berupa nilai PWM (*Pulse Width Modulation*) akan dijadikan sebagai pengatur kecepatan motor BLDC. Nilai PWM yang merupakan sinyal digital harus dikonversi menjadi sinyal analog menggunakan rangkaian DAC (*Digital to Analog Converter*), sebab *controller* BLDC hanya bisa menerima masukan berupa sinyal *analog*. Nilai analog berupa tegangan inilah yang masuk ke *controller* BLDC untuk mengatur kecepatan motor BLDC.



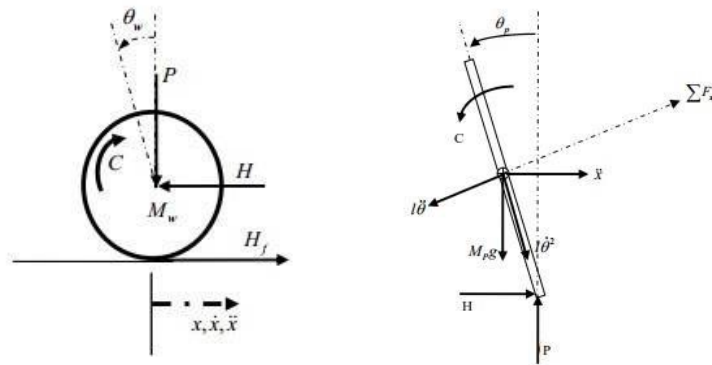
Gambar 2. Diagram Alir Sistem

### 2.2. Pemodelan Sistem

Pada bagian ini pemodelan *state space* dari *electric unicycle* didapatkan. Pemodelan didapatkan dengan memodelkan tiga bagian yang terdapat pada *electric unicycle* yaitu, pemodelan motor DC, pemodelan roda, dan pemodelan chassis.



Gambar 3. Pemodelan Linear Motor DC



Gambar 4. Pemodelan Roda dan Chassis Electric Unicycle

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{g}{l} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{g}{l} \\ 0 \\ 0 \end{bmatrix} \quad (1)$$

Dimana :

$$\begin{aligned}
 \frac{g}{l} &= \frac{m_w g}{m_w l + m_c l} \\
 \frac{g}{l} &= \frac{m_w g}{m_w l + m_c l} \left( \frac{m_w r^2 + I_w}{m_w l + m_c l} + \frac{I_c}{m_w l + m_c l} \right)
 \end{aligned}$$

Pemodelan diimplementasikan pada *electric unicycle* dengan spesifikasi sebagai berikut :

Tabel 1. Spesifikasi Electric Unicycle

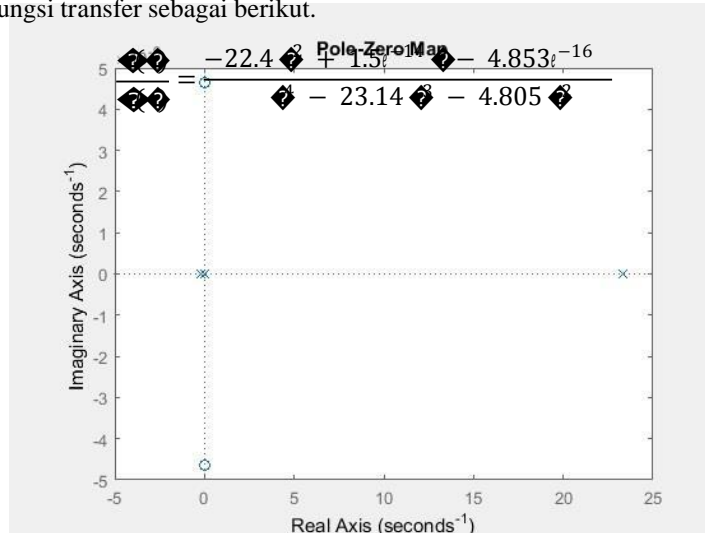
$m_w$ (massa roda)	15 kg
$m_c$ (massa chassis / rangka)	10 kg
$l$ (panjang dari pusat roda ke pusat beban rangka)	0.4 m
$I_w$ (momen inersia roda)	0.9015 kg.m <sup>2</sup>
$I_c$ (momen inersia chassis)	0.533 kg.m <sup>2</sup>
$K_m$ (konstanta torsi motor)	0.464
$K_b$ (konstanta <i>back EMF</i> motor)	0.12
R (resistansi motor)	2.3 Ohm
r (jari-jari roda)	0.29 m

Mengacu pada persamaan (1), parameter-parameter spesifikasi dimasukkan sehingga mendapatkan persamaan *state space* sebagai berikut :

$$\begin{bmatrix} \dot{\phi} \\ \phi \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.1193 & 2.605 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0.7794 & 23.26 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -22.4 \end{bmatrix}$$

Sehingga didapatkan fungsi transfer sebagai berikut.



Gambar 5. Diagram Pole-Zero Sistem

Dapat dilihat dari diagram *pole & zero* yang disimulasikan pada *software* Matlab, sistem belum stabil karena ada *pole* yang terdapat pada *right half plane*, sehingga perlu penstabilan sistem.

### 2.3. MPU6050 dengan Kalman Filter

Modul GY521 MPU6050 adalah perangkat terpadu pertama yang memiliki 6 aksis *motion tracking*. Pada modul ini terdapat kombinasi dari 3 aksis *gyroscope* dan 3 aksis *accelerometer*. Masing-masing *accelerometer* dan *gyroscope* dapat diakses secara mandiri dan semua berbasis di sumbu yang sama, sehingga mencegah terjadinya kesalahan *cross-axis* saat menggunakan perangkat secara terpisah.

Pada GY521 MPU6050, *accelerometer* dapat memberikan pengukuran sudut saat sistem berada pada kondisi diam. Sedangkan pada saat sistem berotasi, *accelerometer* tidak bisa bekerja secara maksimal karena memiliki respon yang lambat. Kelemahan inilah yang dapat diatasi oleh *gyroscope* karena dapat membaca kecepatan sudut yang dinamis. Namun *gyroscope* juga memiliki kelemahan yaitu proses perpindahan kecepatan sudut dalam jangka waktu yang panjang menjadi tidak akurat karena ada efek bias yang dihasilkan oleh *gyroscope*.

Kalman Filter adalah filter digital rekursif yang dapat memprediksi dengan sangat efektif untuk proses apapun. Filter ini bertugas mengurangi *noise* dari sensor sebelum data *sensor* tersebut dikirim ke sebuah sistem kontrol..

Kalman filter dibagi kedalam 2 bagian utama yaitu *time update* dan *measurement update*. *Time update* bisa disebut juga sebagai proses *predict*, yaitu menggunakan *estimation state* dari waktu sebelumnya untuk mendapatkan sebuah *estimation state* saat ini. *Measurement update* bisa disebut juga sebagai proses *correct*, yaitu menggunakan informasi pengukuran saat ini untuk memperbaiki prediksi dengan tujuan untuk mendapatkan *estimation state* yang akurat. Kalman filter akan terus menerus melakukan iterasi *predict – correct*.

#### Time Predict

Dalam dua persamaan pertama kita akan mencoba untuk memprediksi keadaan saat ini dan kovariansi *error* matriks pada waktu *k*. Pertama, filter akan mencoba untuk memperkirakan keadaan saat berdasarkan semua *state* sebelumnya dan hasil pengukuran *gyroscope*.

Dengan  $\hat{x}_{k-1}$  adalah *state process*.

$$\hat{x}_{k-1} = G_{k-1} \hat{x}_{k-2} + w_{k-1}$$

$$\hat{x}_{k-1} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$



Dimana  $\theta$  adalah *input* sudut dari *accelerometer* dan  $\theta_0$  adalah sudut bias dari *gyroscope*.

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \end{bmatrix} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \ddot{\theta}$$

Nilai  $\ddot{\theta}$  adalah nilai dari pembacaan *gyroscope*.

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_k = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \ddot{\theta}_{k-1}$$

Berikutnya, kita akan mencoba untuk memperkirakan matrix kovarian error apriori berdasarkan matrix kovarian error sebelumnya yang didefinisikan sebagai :

$$P_{k|k-1} = F_{k|k-1} P_{k-1|k-1} + Q_k$$

Dengan  $Q_k$  adalah *process noise covariance matrix*, dan  $P$  adalah *P error covariance matrix*. Selanjutnya didefinisikan *process noise covariance matrix* sebagai berikut.

$$Q_k = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_g^2 \end{bmatrix}$$

Dimana  $\sigma_a^2$  adalah *noise proses* dari *accelerometer*, dan  $\sigma_g^2$  *noise proses* dari *gyroscope*.

$$\begin{bmatrix} \sigma_{\theta_0}^2 & \sigma_{\dot{\theta}_0}^2 \\ \sigma_{\dot{\theta}_0}^2 & \sigma_{\ddot{\theta}_0}^2 \end{bmatrix}_{k|k-1} = \begin{bmatrix} \sigma_{\theta_0}^2 + \Delta t^2 \sigma_a^2 & -\sigma_{\dot{\theta}_0}^2 - \Delta t \sigma_a^2 \\ -\sigma_{\dot{\theta}_0}^2 - \Delta t \sigma_a^2 & \sigma_{\dot{\theta}_0}^2 + \Delta t^2 \sigma_g^2 \end{bmatrix}$$

**Measurement Update**

Hal pertama yang akan kita lakukan adalah untuk menghitung perbedaan antara  $z_k$  pengukuran dan state apriori  $\hat{x}_{k|k-1}$ .

$$z_k = y_k - H_{k|k-1} \hat{x}_{k|k-1}$$

Dikarenakan hasil pengukuran yang akan digunakan hanya nilai dari *accelerometer* dan tidak mencakup sudut bias dari *gyroscope*, maka matriks H didefinisikan sebagai berikut.

$$H_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Hal berikutnya adalah menghitung pembaruan kovarian :

$$P_k = F_{k|k-1} P_{k-1|k-1} + Q_k$$

$$P_k = P_{k|k-1} + Q_k$$

Dimana R adalah *measurement noise*. Langkah selanjutnya adalah menghitung Kalman gain. Kalman gain digunakan untuk menunjukkan seberapa banyak kita percaya pembaruan dan didefinisikan sebagai :

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$

$$K_k = \begin{bmatrix} K_{k1} \\ K_{k2} \end{bmatrix}$$

Langkah selanjutnya adalah memperbarui perkiraan kondisi posteriori dari *state* saat ini :

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k z_k$$

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_k = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_{k1} \\ K_{k2} \end{bmatrix} z_k$$

Hal terakhir yang dilakukan adalah memperbarui matriks kovarian error posteriori :

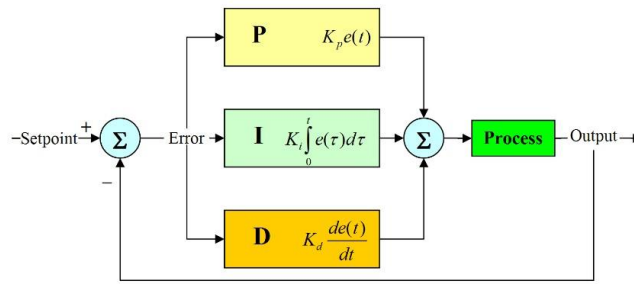
$$P_k = (I - K_k H_k) P_{k|k-1} = \begin{bmatrix} \sigma_{\theta_0}^2 & \sigma_{\dot{\theta}_0}^2 \\ \sigma_{\dot{\theta}_0}^2 & \sigma_{\ddot{\theta}_0}^2 \end{bmatrix}_{k|k-1} - \begin{bmatrix} K_{k1} \sigma_{\theta_0}^2 & K_{k1} \sigma_{\dot{\theta}_0}^2 \\ K_{k2} \sigma_{\theta_0}^2 & K_{k2} \sigma_{\dot{\theta}_0}^2 \end{bmatrix}$$

Dan untuk mendapat nilai parameter kalman filter dilakukan dengan cara *trial and error* nilai matriks Q dan R sampai mendapatkan nilai yang paling optimal.

**2.4. PID**

Pengendali *Proportional Integral Derivative* (PID) merupakan salah satu pengendali yang bertujuan memperbaiki kinerja suatu sistem, termasuk sistem kendali putaran motor DC. Kecepatan respon dan *error steady state* merupakan parameter yang diukur untuk menilai kinerja suatu sistem kendali.

Sesuai dengan penamaannya, sistem kontrol PID merupakan gabungan metode pengaturan yang terdiri dari kendali *Proportional, Integral, dan Derivative* yang masing-masing memiliki kegunaan.



Gambar 6. Diagram Blok PID

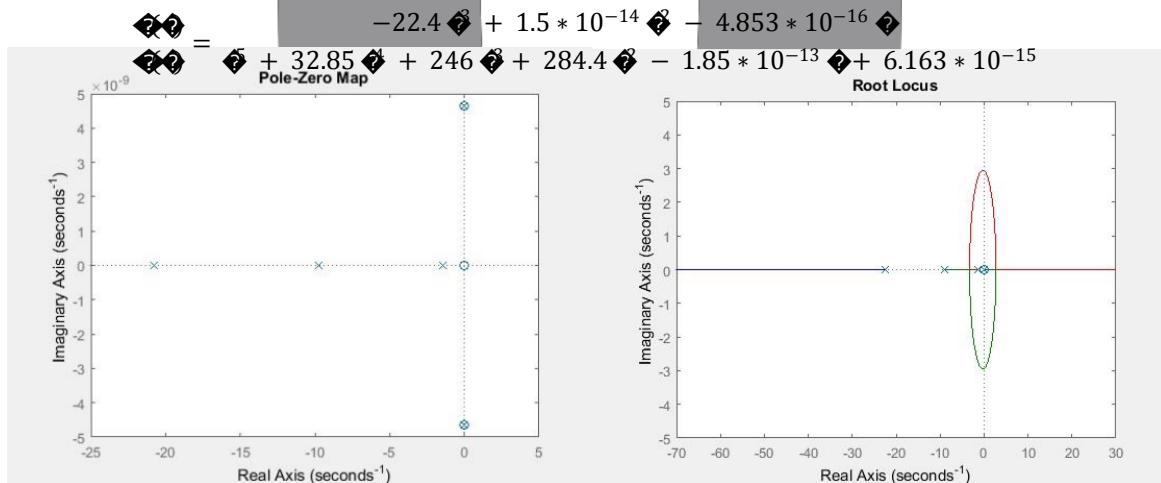
Masing-masing kendali dapat bekerja dan diimplementasikan baik secara terpisah maupun kombinasi secara bersamaan dengan tujuan saling melengkapi diantara masing-masing karakteristik kendali tersebut. Penjelasan untuk masing-masing kendali, berikut ini:

- Pada aksi kendali proporsional, keluaran sistem kendali akan berbanding lurus dengan masukan dan *error*, dan menghasilkan tanggapan yang cepat. Akan tetapi *overshoot* meningkat sehingga sistem cukup bermasalah terutama saat awal beroperasi.
- Pada aksi kendali integral, keluaran sistem berubah dengan cepat sesuai perubahan *error*, sehingga *error steady state* mendekati nol.
- Pada aksi kendali *derivative* bekerja sesuai dengan laju perubahan *error*. Oleh karena itu, kendali ini berfungsi mereduksi laju perubahan *error* sehingga menjaga kestabilan sistem.

Pengendali PID pada Tugas Akhir ini didesain untuk memenuhi spesifikasi sebagai berikut : tidak ada overshoot, rise time 0.7 detik, dan sistem *damped*. Penataan parameter  $K_p$ ,  $K_i$ , dan  $K_d$  dilakukan dengan metode penempatan *pole*. Sebelum menempatkan *pole*, fungsi transfer dari sistem harus diketahui terlebih dahulu.

$$G(s) = \frac{-22.4s^2 + 1.5e^{-14}s - 4.853e^{-16}}{s^2 - 23.14s - 4.805}$$

Dengan simulasi *tuning* PID pada *software* Matlab, berdasarkan fungsi transfer sistem didapatkan  $K_p = 11.2$ ,  $K_i = 12.7$ ,  $K_d = 2.5$ . Berikut fungsi transfer sistem *lup* tertutup setelah diberi kontroler PID.



Gambar 7. Diagram Pole-Zero dan Root Locus dengan Kendali PID

Dapat dilihat dari diagram *pole-zero*, semua *pole* dari sistem berada di *left half plane* yang menandakan didapatkan kestabilan sistem dengan kendali PID.

### 3. Pengujian dan Analisis

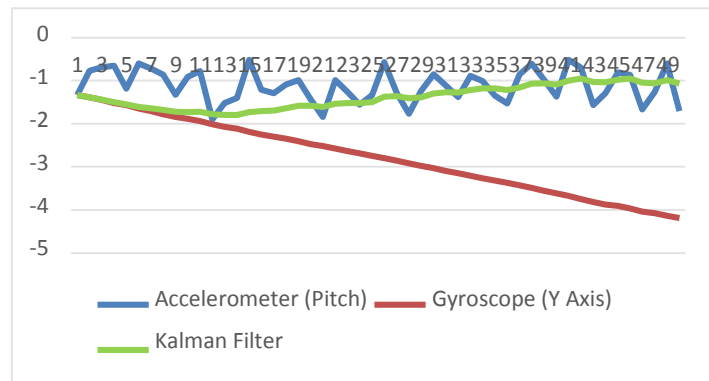
#### 3.1. Pengujian Kalman Filter



Pengujian MPU6050 dengan Kalman Filter bertujuan untuk mengetahui apakah metode Kalman Filter berfungsi dengan baik pada hasil pembacaan MPU6050. Serta membandingkan hasil pembacaan MPU6050 tanpa filter dengan hasil pembacaan MPU6050 yang outputnya sudah difilter dengan Kalman Filter.

*Process noise covariance* ( $\Sigma$ ) *matrix* dan *measurement noise covariance* ( $\Sigma$ ) merupakan komponen penting dalam kualitas hasil dari keluaran Kalman filter. Kedua matriks kovarian ini dapat diperoleh dengan menggunakan persamaan statistik. Namun, pada tugas akhir ini penulis menggunakan metode *tuning*. Yaitu dengan *trial and error* sampai mendapat nilai yang paling baik untuk pembacaan sensor yang digunakan.

Pada pengujian, parameter Kalman filter yang digunakan untuk *filtering* sudut MPU6050 antara lain  $Q_{accelerometer} = 0.001$ ,  $Q_{bias} = 0.003$ , dan  $R = 0.03$ . Data yang diambil sebagai sampel sebanyak 50 data dengan hasil sebagai berikut.

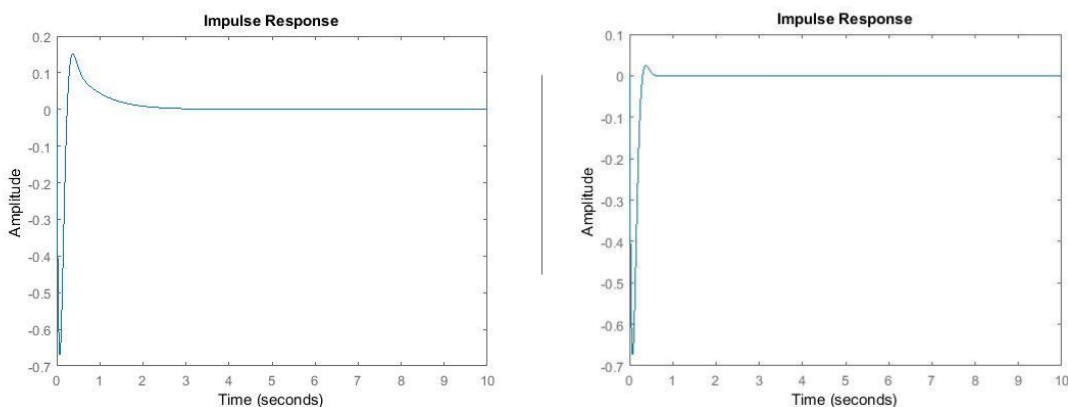


Gambar 8. Kalman Filter dengan  $Q_{accelerometer} = 0.001$ ,  $Q_{bias} = 0.003$ , dan  $R = 0.03$

Dengan  $Q_{accelerometer} = 0.001$ ,  $Q_{bias} = 0.003$ , dan  $R = 0.03$  dicapai keluaran filter Kalman yang dinilai efektif. Dapat dilihat dalam grafik di atas, keluaran hasil filter Kalman menunjukkan bahwa *noise* pembacaan *accelerometer* berhasil diminimalisir dengan *sensor fusion* dengan *gyroscope*, tanpa membuat pembacaan sudut menjadi bias. Berdasarkan hasil pengujian, Kalman Filter sangat efektif diimplementasikan pada sistem yang sudah dirancang. Jika dibandingkan dengan hasil pembacaan MPU6050 tanpa filter, *output* yang dihasilkan Kalman Filter sangat baik, karena pemodelan untuk prediksi sudut yang akan datang sangat membantu untuk meminimalisir *noise* pembacaan yang dihasilkan MPU6050.

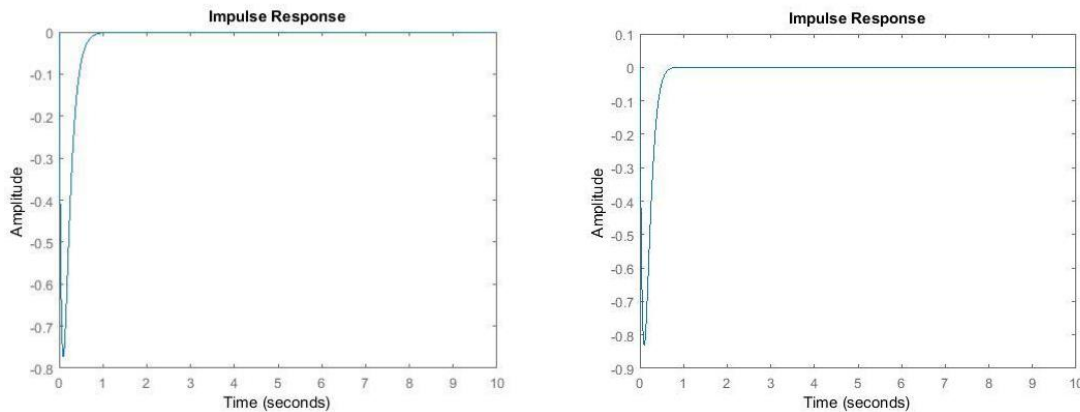
### 3.2. Pengujian PID

Pengujian ini bertujuan untuk mengetahui respon sistem terhadap kontroler PID. Parameter  $K_p$ ,  $K_i$ , dan  $K_d$  yang akan diuji sesuai dengan perancangan yang telah dilakukan yaitu  $K_p = 10.2$ ,  $K_i = 13.7$ , dan  $K_d = 2$ .



Gambar 9. Simulasi respon sistem dengan  $K_p=10.2$ ,  $K_i=13.7$ ,  $K_d=2$  dan  $K_p=10.2$ ,  $K_i=0$ ,  $K_d=2$

Pada simulasi di atas, dapat dilihat bahwa terdapat *overshoot* pada sistem dengan kontroler  $K_p = 10.2$ ,  $K_i = 13.7$ , dan  $K_d = 2$ . *Overshoot* dapat diminimalisir dengan mengurangi parameter P dan I pada kontroler PID. Pengujian selanjutnya dilakukan dengan mengurangi  $K_i$  hingga didapat respon sistem yang tidak memiliki *overshoot*. Berikut adalah hasil pengujian selanjutnya. Dengan pengurangan nilai  $K_i$  sampai dengan 0, *overshoot* masih belum bisa dihilangkan secara sempurna namun sistem juga tidak mengalami *error steady state* yang berarti. Hal ini menunjukkan bahwa parameter I tidak dibutuhkan dalam implementasi pada sistem. Langkah selanjutnya adalah mengurangi parameter  $K_p$  hingga *overshoot* dapat dihilangkan.



**Gambar 10.** Simulasi respon sistem dengan  $K_p=5$ ,  $K_i=0$ ,  $K_d=2$  dan  $K_p=5$ ,  $K_i=0$ ,  $K_d=1.9$

Dengan nilai  $K_p = 5$ , tidak terjadi *overshoot* pada sistem, namun respon sistem masih relatif lambat. Respon sistem dapat dipercepat dengan mengurangi parameter D karena respon sistem terhadap parameter D lebih mulus, namun memperlambat *settling time*. Respon sistem dipercepat tanpa membuat sistem berosilasi dengan penurunan  $K_d$  menjadi 1.9. Dari hasil pengujian dapat diambil parameter PID yang optimal yaitu  $K_p=5$ ,  $K_i=0$ ,  $K_d=1.9$  sehingga sistem hanya menggunakan kontroler PD.

#### 4. Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan pada penelitian ini dapat diambil beberapa kesimpulan sebagai berikut :

1. Pengimplementasian modul MPU6050 pada *electric unicycle* sangat tepat, karena modul ini sudah mencakup *accelerometer* dan *gyroscope*. *Accelerometer* digunakan sebagai informasi sudut dan *gyroscope* digunakan sebagai informasi kecepatan sudut sehingga didapatkan data pengukuran kemiringan yang sempurna.
2. Penggunaan metode Kalman Filter sangat efektif digunakan pada penelitian ini. *Noise* pada hasil pembacaan *accelerometer* MPU6050 sangat banyak dan pembacaannya sangat sensitif terhadap lingkungan. Dengan menggunakan Kalman Filter dengan nilai  $Q_{\text{accelerometer}} = 0.001$ ,  $Q_{\text{bias}} = 0.003$  dan  $R = 0.03$ , didapat output yang baik dan *noise* dapat diminimalisir.
3. Pengimplementasian kendali PD  $K_p = 5$  dan  $K_d = 1.9$  pada sistem dinilai tepat karena menghasilkan respon yang relatif lebih cepat dibandingkan kontroler lainnya. Namun, kendali PD yang diimplementasikan pada *electric unicycle* tidak begitu akurat karena pemodelan linear diimplementasikan sistem yang bersifat nonlinear.

#### Daftar Pustaka :

- Kalman, R. E., 1960. A New Approach to Linear Filtering and Prediction Problems. ASME - Journal Of Basic Engineering, pp. 35-45.
- Ooi, Rich Chi. 2003. Balancing of a Two Wheeled Autonomous Robot.
- Ziyu, Shao. Daliang, Liu. Balancing Control of a Unicycle Riding.
- Syam, Rafiudin. 2015. Rancang Bangun Self-Balancing Scooter Sebagai Moda Transportasi. Jurnal Mekanikal, Vol. 6 No. 1: Januari 2015: 522-531
- Ellul, J. 1964. The Technological Society, Translated by John Wilkinson. Random House, Note to the Reader, p. xxv.
- Hoffman, C., 2014. TAKING A SPIN ON RYNO'S ONE-WHEELED, SELF-BALANCING ELECTRIC 'MICROCYCLE' [Interview] (21 January 2014).
- Lauzuz, Kristian Sloth, 2012. A Pratical Approach to Kalman Filter and How to Implement It. [Online] <http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/>, diakses pada tanggal 17 April 2016.
- Setiawan, W., 2014. Cara kerja Accelerometer. [Online], <https://wirasetiawan29.wordpress.com/2014/04/06/accelerometer/>, diakses pada tanggal 28 November 2015.
- Eliezer, I. P. G., 2013. Accelerometer dan Gyroscope. [Online], <http://www.geyosoft.com/2013/accelerometer-dan-gyroscope>, diakses pada tanggal 29 November 2015.
- Anon. [http://dairyroadsolutions.com/htmlMinSeg/pid/MinSegModel\\_PID.html](http://dairyroadsolutions.com/htmlMinSeg/pid/MinSegModel_PID.html) , diakses pada tanggal 26 Februari 2016.
- Anon. <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling>, diakses pada tanggal 26 Februari 2016.