

# IMPLEMENTASI KALMAN FILTER DAN KONTROLER PID UNTUK ROBOT PENDULUM TERBALIK

## IMPLEMENTATION OF KALMAN FILTER AND PID CONTROLLER FOR INVERTED PENDULUM ROBOT

Muhammad Royyan, Angga Rusdinar, Ph.D. , Erwin Susanto, Ph.D.

Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom  
royzah@gmail.com, anggarusdinar@telkomuniversity.ac.id , abumuflih2005@gmail.com

---

### Abstrak

Penelitian tentang robot pendulum terbalik meningkat pesat dikarenakan manfaat dan kemampuan untuk menyelesaikan tugas tertentu. Maka dari itu variasi teknik dalam pergerakan robot pada lingkungan yang dinamik pun semakin banyak. Pada jurnal, penulis akan mempresentasikan tentang teknik pendulum terbalik tersebut untuk menyeimbangkan robot dengan dua roda, agar dapat berdiri tegak. Robot keseimbangan akan dirancang dengan implementasi Kalman filter dan PID.

Pada tugas akhir ini penulis akan merancang sistem kontrol dengan kalman filter dan PID kemudian penulis akan menganalisa performanya. Dimana parameter input Kalman Filter dan PID langsung mempengaruhi performa sistem. Hasil menunjukkan bahwa pemberian parameter pada Kalman Filter dan PID berefek pada sistem secara keseluruhan. Ditunjukkan bahwa kalman filter diperlukan untuk mengurangi noise dari sensor. Pembacaan data yang baik meningkatkan performa PID untuk membuat robot tetap pada titik imbangnya. Didapat parameter optimal Kalman filter  $R_{pengukuran} = 0.03$  ,  $Q_{accelerometer} = 0.001$  ,  $Q_{gyroscope} = 0.003$ . Sedangkan untuk PID  $K_p = 9.0$  ,  $K_I = 2.0$ ,  $K_D = 3.0$ .

**Kata kunci :** Kalman Filter, PID , Pendulum Terbalik

---

### Abstract

The research on inverted pendulum robot has gained momentum due to their functionality and reliability when completing certain tasks. Therefore variation of technique in dynamic environments movement is increasing. This paper presents about inverted pendulum for balancing robot with two wheels, hence it will be able to stand upright with two wheels. The implementation of Kalman filter and PID will be used.

In this paper, authors design control system with Kalman filter and PID which will be implemented and analyze its performance. The control system performance is directly dependent on Kalman filter and PID controller input parameters. The results clearly show how the adjustable parameters on the control system directly affected the overall system performance. The results also demonstrate the performance and the need of the Kalman filter to remove sensor noise. The almost reliable sensor data increases PID controller performance to drive the robotic platform to vertical equilibrium. The gathered results for the Kalman filter were compared against the raw noisy sensor data. The plots for such comparison are shown on the Kalman filter results section. PID controller output response data was also collected and plotted. From the experimental result obtained the optimal parameters either Kalman filter and PID. The optimal parameters of Kalman filter are  $R_{pengukuran} = 0.03$  ,  $Q_{accelerometer} = 0.001$  ,  $Q_{gyroscope} = 0.003$ . Whereas the optimal parameter of PID are  $K_p = 9.0$  ,  $K_I = 2.0$  ,  $K_D = 3.0$ .

---

### 1. Pendahuluan

Penelitian tentang robot penyeimbang sudah dilakukan puluhan tahun yang lalu. Penelitian ini pun telah menginspirasi banyak penggiat robot termasuk penulis untuk mengembangkan pengetahuannya. Robot yang akan dirancang adalah robot yang dapat menyeimbangkan dirinya hanya dengan dua roda. Kenyataan, sudah ada kendaraan unik yang bernama Segway Personal Transporter , dimana sistem dinamikanya sangat mirip dengan pendulum terbalik. Alat ini memiliki dua roda di kanan dan kiri pengendara yang berdiri disebuah papan yang terhubung dengan roda tersebut. Alat dapat berdiri tegak lurus terhadap permukaan bumi jika pengendara berdiri tegak dan alat ini akan maju jika pengendara bersandar ke depan, akan mundur jika pengendara bersandar ke belakang.

Pada jurnal ini digunakan mikrokontroler AVR ATmega1284P dengan kecepatan kerja 10MHz , 128KB flash, 16KB RAM , dan 4KB EEPROM. Digunakan juga IMU (Inertial Measurement Unit) MPU-6050, yang terhubung menggunakan komunikasi I2C dengan mikrokontroler. IMU ini terdiri dari 3-axis accelerometer dan 3-axis gyroscope. Ketika dua sensor ini digabungkan menggunakan Kalman filter maka akan dihasilkan pembacaan sensor yang stabil. Proportional Integral Derivative (PID) digunakan sebagai metoda pengendalinya yang mana akan menentukan besarnya

kecepatan dan arah putaran motor DC berdasarkan sudut kemiringan badan robot, percepatan sudut robot yang didapat dari sensor IMU yang telah digabung dan posisi dan kecepatan motor DC yang diketahui dari rotary encoder yang terpasang pada motor DC. Sehingga robot ini dapat mempertahankan posisinya tegak lurus dengan seimbang terhadap permukaan bumi pada bidang datar.

## 2. Dasar Teori

### 2.1. Segway PT

Segway PT adalah transportasi pribadi populer yang sudah tersedia untuk umum. Diciptakan oleh Dean Kamen. Dinamika gerak dari Segway PT sendiri identic dengan pendulum terbalik. Untuk meningkatkan mobilitasnya, Segway PT didisain dengan dua roda [1]. Robot yang dipresentasikan untuk jurnal ini dapat mengimplemntasikan algoritma yang serupa dengan yang digunakan Segway PT.



Gambar 1. Segway PT [1]

Robot pendulum terbalik adalah alat peraga yang baik untuk mempelajari kontrol dan penelitian [2] [3]. Penelitian tentang robot pendulum terbalik ini sudah banyak dilakukan, sebagai contoh JOE[4] dan nBot[5]. Dan masih banyak riset tentang desain kontrol dengan pendulum terbalik ini, ada yang mengunakan metoda kontrol klasik dan kontrol linear multivariabel [4],[6],[7],[8], dengan metoda nonlinear [9],[10], anfis [11], dan gabungan [12].

### 2.2. Kalman Filter

R.E Kalman mempublikasikan tulisannya pada tahun 1960 berjudul "A New Approach to Linear Filtering and Prediction Problems." R.E Kalman menjelaskan solusinya atas masalah filter linear pada data diskrit. Sekarang Kalman filter itu sendiri sudah sering digunakan di kontrol sistem embedded dan sistem navigasi.

Penulis ingin menunjukan kelebihan Kalman filter untuk memberi prediksi yang baik untuk sudut robot terhadap permukaan bumi dan menjaga robot agar tetap seimbang. Karena Kalman Filter digunakan untuk menghilangkan segala noise pengukuran pada accelerometer dan gyroscope. Kelemahan dari Filter ini sendiri adalah tidak adanya standar dalam metode mendapatkan persamaan matematis yang akan digunakan di Kalman Filter [13].

Pada paper *An Introduction to the Kalman Filter* [14]. Dijelaskan bahwa kalman filter dapat digunakan untuk mengilangkan noise dari sensor. Proses atau sistem harus di modelkan dalam sistem linear. Sistem linear dapat direpresentasikan menjadi dua persamaan

$$\vec{x}_k = A\vec{x}_{k-1} + B\vec{u}_{k-1} + \vec{w}_{k-1} \quad (1)$$

$$\vec{z}_k = H\vec{x}_k + \vec{v}_k \quad (2)$$

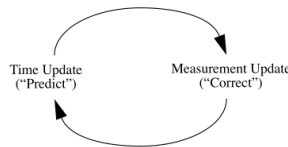
Persamaan (1) merepresentasikan persamaan process state.  $A$ ,  $B$ , dan  $H$  menyatakan state matrices,  $\vec{w}_{k-1}$  menyatakan process noise. Persamaan (2) merepresentasikan persamaan output state.  $\vec{z}_k$  adalah nilai yang terukur dari sensor, dan  $\vec{v}_k$  menyatakan noise pengukuran.  $\vec{w}_{k-1}$  dan  $\vec{v}_k$  adalah variable acak yang independen. Tetapi dapat dimisalkan dengan adanya matrik kovarian  $Q_w$  dan  $R_v$ . Sehingga dapat asumsikan

$$Q_w = E[\vec{w}_k \vec{w}_k^T] \quad (3)$$

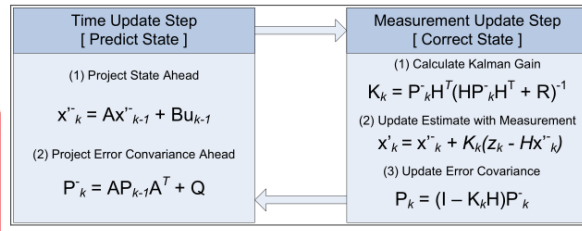
$$R_v = E[\vec{v}_k \vec{v}_k^T] \quad (4)$$

Process noise covariance  $Q_w$  matrix dan measurement noise covariance  $R_v$  merupakan komponen penting dalam kualitas hasil dari keluaran Kalman filter. Kedua covariance matrix ini dapat dijadikan adaptif. Akan tetapi untuk kasus ini agar mempercepat proses, kita jadikan dia konstan, dan diatur nilainya secara manual.

Pertama-tama filter akan mengestimasi keadaan sistem pada suatu waktu lalu mendapatkan pengukuran yang ber-noise untuk dijadikan feedback. Setelah itu, persamaannya terpecah menjadi dua, yaitu persamaan time update dan measurement update. Time update bisa juga disebut persamaan prediksi dan measurement update disebut persamaan pengoreksi.



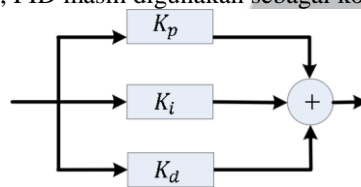
Gambar 2. Skema algoritma Kalman filter[14]



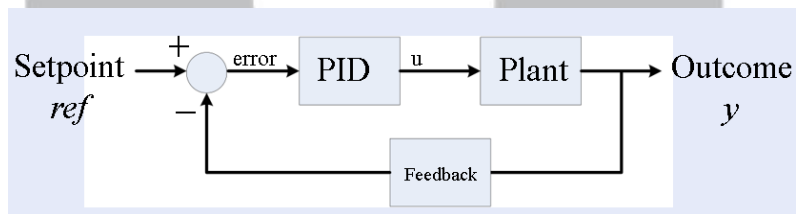
Gambar 3. Algoritma persamaan Kalman filter[14]

### 2.3. PID

Metode yang akan di gunakan pada jurnal ini adalah kontroler linear. Kontroler linear yang digunakan adalah Proportional, Integral, Derivative yang biasa disingkat dengan PID. PID sudah teruji oleh banyak komunitas engineer. Lebih dari 70 tahun setelah ditemukannya PID, PID masih digunakan sebagai kontroler utama pada industri



Gambar 4. PID kontroler



Gambar 5. Diagram Blok PID

Berikut ini merupakan penjelasan singkat dari parameter-parameter yang ada pada kontrol PID :

1. Kontrol Proportional  
Kontrol proportional umumnya dinyatakan dengan sebuah gain tanpa memberikan efek dinamik kepada kinerja kontroler. Adanya kontroler proportional ini mengakibatkan sinyal error akan semakin besar sehingga memperbesar kemungkinan terjadinya overshoot. Namun kontrol proportional dapat mengakibatkan keluaran sistem lebih cepat mencapai set point. Kontrol proportional memperbaiki rise time dan settling time dari sebuah sistem.
2. Kontrol Integral  
Kontrol integral digunakan untuk menghilangkan nilai offset yang biasanya dihasilkan oleh kontrol proportional. Namun pemilihan nilai  $K_i$  yang tidak tepat dapat menyebabkan ketidak stabilan sistem. Jika pemilihan nilai  $K_i$  terlalu tinggi maka akan meyebabkan nilai keluaran akan berosilasi .

Penambahan kontrol integral ini juga akan membuat respon dari sistem menjadi lambat. Oleh sebab itu maka dalam implementasinya kontrol integral dikombinasikan dengan kontrol proporsional.

### 3. Kontrol Derivatif

Keuntungan dari menggunakan kontrol derivative yaitu dapat merespon perubahan error aktuator dan dapat menghasilkan koreksi yang signifikan sebelum magnitude dari error aktuatornya menjadi sangat besar. Seakan-akan kontrol derivative ini mampu memprediksi error yang akan terjadi sebagai efek dari perhitungan error yang sebelumnya. Karena kontrol derivative ini bekerja berdasarkan laju perubahan error aktuatornya dan bukan pada error aktuator itu sendiri maka kontrol derivative ini tidak dapat berdiri sendiri. Kontrol derivative biasanya dikombinasikan dengan kontrol proporsional atau dengan kontrol proporsional plus integral.

Gambar 5 dapat dipresentasikan dalam persamaan :

$$error = ref - y \tag{5}$$

$$u(t) = K_p error(t) + K_i \int_0^t error(t) \Delta t + K_D \frac{\Delta error(t)}{\Delta t} \tag{6}$$

Dan bentuk diskritnya sehingga dapat diimplementasikan pada mikrokontroler adalah :

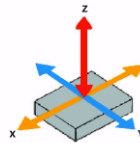
$$\int_0^t error(t) \Delta t \approx T \sum error(k) \tag{6}$$

$$\frac{\Delta error(t)}{\Delta t} \approx \frac{error(k) - error(k-1)}{T} \tag{7}$$

$$u(k) = K_p error(k) + K_i T \sum error(k) + K_D \frac{error(k) - error(k-1)}{T} \tag{8}$$

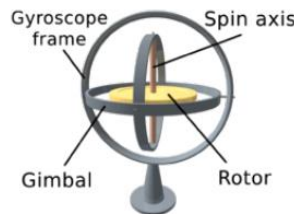
## 2.4. Accelerometer

Accelerometer adalah perangkat yang berfungsi untuk mengukur akselerasi. Akselerasi yang diukur dengan accelerometer belum tentu memiliki laju perubahan velositas. Sebaliknya, accelerometer mendapatkan akselerasi yang dimaksud dengan fenomena berat yang dialami oleh uji massa pada kerangka acuan perangkat accelerometer.



Gambar 6. Gambaran sensor accelerometer

## 2.5. Gyroscope



Gambar 7. Konstruksi dalam sensor gyroscope [15]

Gyroscope adalah roda berat yang berputar pada jari-jarinya. Sebuah giroskop mekanis terdiri dari sebuah roda yang diletakkan pada sebuah bingkai. Roda ini berada di sebuah batang besi yang disebut dengan poros roda. Ketika

gyroscope digerakkan, maka ia akan bergerak mengitari poros tersebut. Poros tersebut terhubung dengan lingkaran-lingkaran yang disebut gimbal. Gimbal tersebut juga terhubung dengan gimbal lainnya pada dasar lempengan. Jadi saat piringan itu berputar, unit gyroscope itu akan tetap menjaga posisinya saat pertama kali dia diputar.

## 2.6. Motor DC

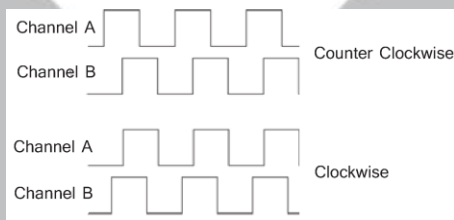
Agar robot dapat berdiri tegak dengan stabil maka pemilihan motor DC sangatlah penting. Motor DC dengan torsi dan RPM yang tinggi menjadi sangat krusial untuk kestabilan robot. Motor DC yang dipakai adalah GM320051.



Gambar 8. Motor DC GM320051

## 2.7. Rotary Encoder

Rotary Encoder adalah sebuah generator kecil yang membangkitkan gelombang tegangan DC yang berfungsi sebagai sensor posisi motor yang menyatu dengan motor dc sekaligus sebagai pengukur kecepatan motor dc. Data output yang dikirim diolah oleh mikrokontroler pada port interrupt. Signal gelombang putaran yang kirim memperlihatkan posisi sudut motor DC. Dari informasi posisi ini juga bisa didapatkan informasi kecepatan dan percepatan. Dengan pulsa yang dihasilkan encoder akan mempermudah menentukan posisi akhir motor DC. Manfaat adanya encoder di sini adalah sebagai feedback dari sistem yang kita buat.

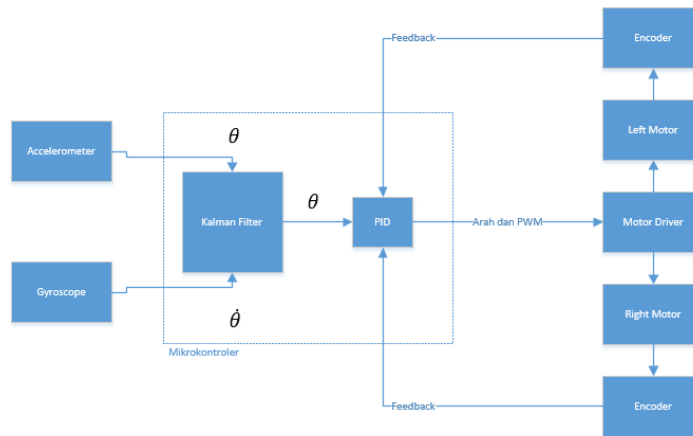


Gambar 9. Gelombang dua channel rotary encoder denga fasa 90°

## 3. Pembahasan

### 3.1. Gambaran Umum Sistem

Perancangan sistem kontrol jurnal ini dapat digambarkan sebagai berikut :

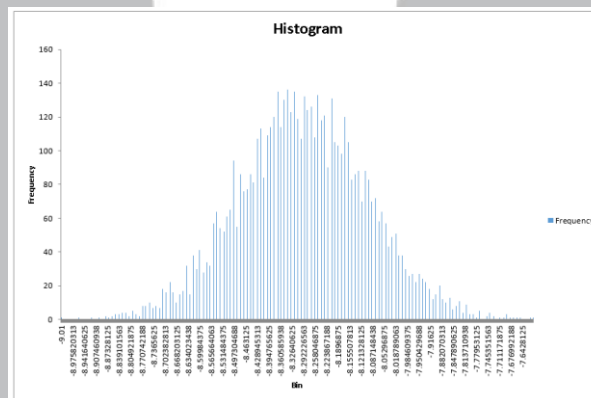


Gambar 10. Blok diagram sistem

Sistem yang telah direalisasikan merupakan sistem yang dapat mempertahankan keseimbangan robot agar tidak jatuh (berdiri tegak). Hal ini dapat dilakukan dengan cara mendeteksi kemiringan dari tubuh robot kemudian dari data kemiringan ini dikirim ke motor dc untuk kembali menyeimbangkan robot.

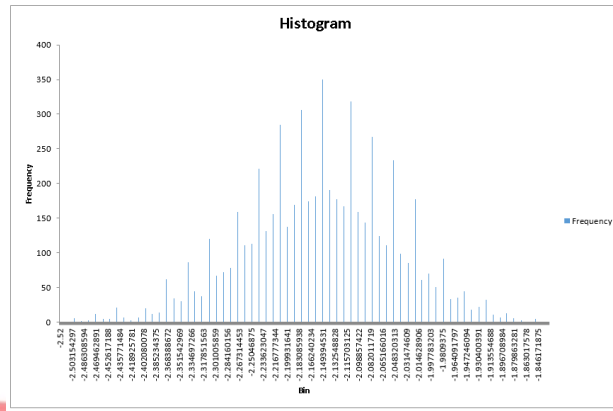
### 3.2. Sensor Fusion Kalman Filter

Accelerometer dapat memberikan pengukuran sudut kemiringan yang akurat ketika sistem sedang diam. Bila sistem sedang berotasi atau bergerak, accelerometer tidak bisa mengikuti pergerakan yang cepat dikarenakan responnya yang lambat dan memiliki noise seperti yang terlihat pada Gambar 11.



Gambar 11. Probability Density Function pembacaan Accelerometer diam

Gyroscope dapat membaca kecepatan sudut (angular rate) yang dinamis. Setelah melakukan komputasi menggunakan integral data dari waktu ke waktu, maka kemiringan sudut dapat dihitung. Tetapi sudut ini akan menjadi tidak akurat dalam jangka panjang karena efek bias dan noise yang tampak pada Gambar 12.



Gambar 12. Probability Density Function pembacaan Gyroscope diam

Untuk mengatasi masalah itu dibutuhkan sensor fusion. Dengan memanfaatkan masing-masing sensor dan mengambil nilai terbaik dari keduanya. Pada kasus ini kita menggunakan hanya accelerometer untuk keadaan diam, dan menggunakan gabungan nilai accelerometer dan gyroscope ketika benda berjalan. Sensor fusion yang digunakan di jural ini adalah Kalman filter. Kalman filter dapat digunakan karena noise dari accelerometer dan gyroscope berbentuk gaussian (Gambar 11. dan Gambar 12.). Kalman filter tidak hanya membaca nilai dan mengabungkan dua sensor tetapi juga menghilangkan noise yang akan mengganggu kestabilan sistem. [13]

### 3.3. Pemodelan Kalman Filter

Kalman Filter bisa dapat diimplementasikan pada robot, jika dan hanya jika ita dapat menemukan model yang akurat. Pemodelan pun harus didekati dengan metoda linear agar mudah diimplementasi. Kalman filter diharapkan dapat memanfaatkan data accelerometer untuk menghilangkan drift dari keluaran gyroscope. Dalam proses, noise dari accelerometer juga akan diminimalkan bahkan dihilangkan.

$$\vec{x}_k = \begin{pmatrix} \theta \\ \dot{\theta}_b \end{pmatrix}_k \quad (9)$$

Dimana  $\theta$  keluaran dari accelerometer pada Gambar 10 dan  $\dot{\theta}_b$  merupakan besarnya kemelencengan bias dari gyroscope (drift). Jadi untuk dapat mendapatkan sudut yang tepat kita harus mengurangi nilai yang kita dapat dengan nilai bias. Maka dari itu dipilih:

$$A = \begin{pmatrix} 1 & -\Delta t \\ 0 & 1 \end{pmatrix} \quad (10)$$

$$B = \begin{pmatrix} \Delta t \\ 0 \end{pmatrix} \quad (11)$$

$$\vec{u}_k = \dot{\theta}_k \quad (12)$$

Nilai  $\dot{\theta}_k$  merupakan keluaran gyroscope pada Gambar 11.  $\Delta t$  merupakan waktu sampling.

Dari persamaan (9) didapat

$$\vec{x}_k = \begin{pmatrix} \theta \\ \dot{\theta}_b \end{pmatrix}_k = \begin{pmatrix} \theta + (\dot{\theta} - \dot{\theta}_b)\Delta t \\ \dot{\theta}_b \end{pmatrix}_{k-1} + \vec{w}_{k-1} \quad (13)$$

Bagan pada Gambar 3 menjelaskan bahwa kalman memiliki beberapa step. Dari Time Update hingga Measurement update.

- Step 1

$$\vec{x}_{k|k-1} = A\vec{x}_{k-1|k-1} + Bu_{k-1} \quad (14)$$

$$\begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k-1|k-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \dot{\theta}_{k-1} \quad (15)$$

$$\begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} = \begin{bmatrix} \theta + \Delta t(\dot{\theta} - \dot{\theta}_b) \\ \dot{\theta}_b \end{bmatrix}_{k-1|k-1} \quad (16)$$

- Step 2

$$\vec{P}_{k|k-1} = A\vec{P}_{k-1|k-1}A^T + Q_k \quad (17)$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k-1|k-1} \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix} + \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b} \end{bmatrix} \Delta t \quad (18)$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} = \begin{bmatrix} P_{00} + \Delta t (\Delta t P_{11} - P_{01} - P_{10} + Q_\theta) & P_{01} - \Delta t P_{11} \\ P_{10} - \Delta t P_{11} & P_{11} + Q_{\dot{\theta}_b} \Delta t \end{bmatrix}_{k-1|k-1} \quad (19)$$

- Step 3

$$\vec{S}_k = H\vec{P}_{k|k-1}H^T + R \quad (20)$$

$$\vec{S}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R \quad (21)$$

$$\vec{S}_k = P_{00|k-1} + R \quad (22)$$

- Step 4

$$\vec{K}_k = \vec{P}_{k|k-1}H^T\vec{S}_k^{-1} \quad (23)$$

$$\begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k = \begin{bmatrix} P_{00} \\ P_{10} \end{bmatrix}_{k|k-1} / \vec{S}_k \quad (24)$$

- Step 5

$$\vec{y}_k = \vec{z}_k - H\vec{x}_{k|k-1} \quad (25)$$

$$\vec{y}_k = z_k - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} \quad (26)$$

$$\vec{y}_k = z_k - \theta_{k|k-1} \quad (27)$$

- Step 6

$$\vec{x}_{k|k} = \vec{x}_{k|k-1} + \vec{K}_k\vec{y}_k \quad (28)$$

$$\begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k} = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \vec{y}_k \quad (29)$$

$$\begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k} = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0\vec{y}_k \\ K_1\vec{y}_k \end{bmatrix}_k \quad (30)$$

- Step 7

$$\vec{P}_{k|k} = (I - \vec{K}_kH)\vec{P}_{k|k-1} \quad (28)$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k} = \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \quad (28)$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k} = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} - \begin{bmatrix} K_0P_{00} & K_0P_{01} \\ K_1P_{00} & K_1P_{01} \end{bmatrix}_{k|k-1} \quad (28)$$

### 3.3. Pengujian Kalman Filter

Kalman filter memberikan prediksi sudut yang lebih baik dari data asli dari accelerometer. Agar filter dapat berjalan dengan lancar, maka perlu dilakukan tuning manual. Parameter yang perlu dituning adalah  $Q_w$  dan  $R_v$ . Bentuk darimatris adalah sebagai berikut,

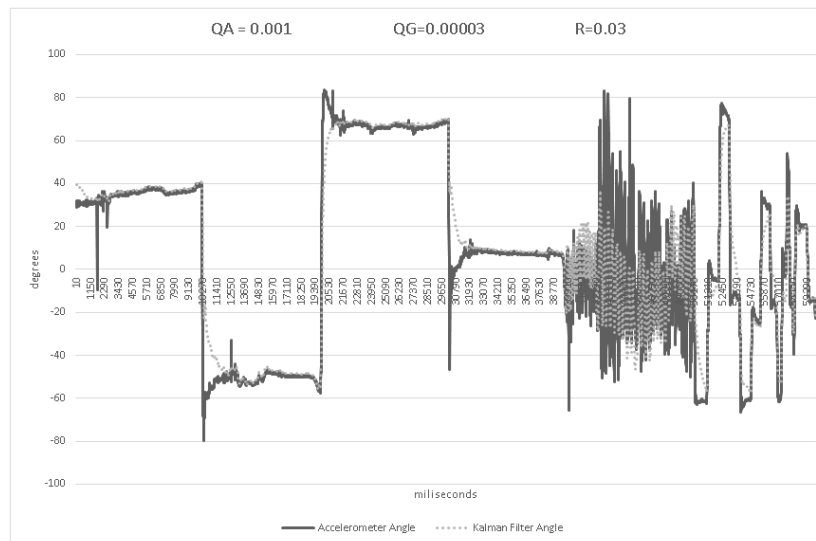
$$Q = \begin{pmatrix} Q_{accelerometer} & 0 \\ 0 & Q_{gyroscope} \end{pmatrix} \quad (5)$$

$$R = (R_{pengukuran}) \quad (6)$$

Metode tuning dilakukan secara trial and error. Ketika kovarian matrik  $Q_w$  dan matrik pengukuran  $R_v$  yang didapatkan memenuhi respon keluaran yang diinginkan, maka nilai itulah yang digunakan. Pengujian dilakukan

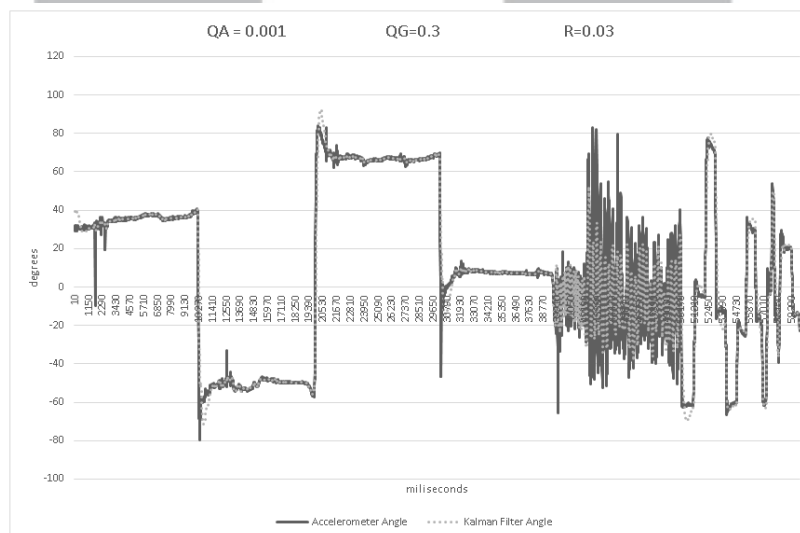


dengan cara IMU diputarputar. Periode sampling dari Kalman filter adalah 10 ms. Pada percobaan di bawah, yang dirubah hanya  $Q_w$  sedangkan nilai  $R_v$  tidak dirubah.



Gambar 13. Grafik pengujian dengan tuning (R=0.03 QG=0.00003 QA=0.001)

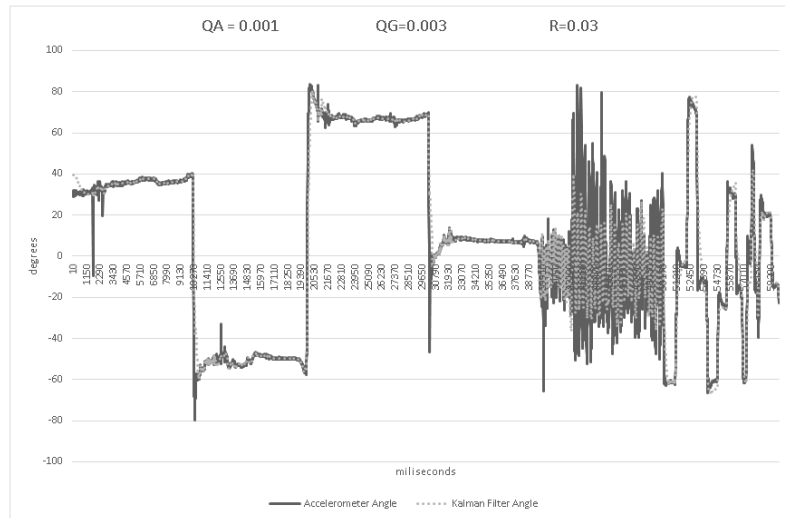
Dari Gambar 13 terlihat jelas bahwa keluaran dari filter tidak memuaskan. Keluaran dari filter tidak mengikuti sudut accelerometer secara real time. Respon keluaran dari Kalman filter sangatlah lambat. Respon yang sangat lambat ini akan menjadi masalah jika dijadikan input dari kontroler PID. Dari Gambar 13 dapat dilihat bahwa  $Q_{gyroscope}$  memiliki nilai yang lebih kecil dari  $Q_{accelerometer}$ . Yang berarti data dari gyroscope lebih dipercaya daripada data dari accelerometer. Maka dari itu kita melakukan pengujian dengan nilai  $Q_{gyroscope}$  lebih besar daripada nilai  $Q_{accelerometer}$ . Yang berarti keluaran accelerometer lebih dipercaya daripada keluaran gyroscope. Pengujian itu diplot seperti yang terlihat pada Gambar 14.



Gambar 14. Grafik pengujian dengan tuning (R=0.03 QG=0.3 QA=0.001)

Dari Gambar 14, dapat diketahui bahwa respon keluaran filter menjadi lebih baik. Keluaran mengikuti data accelerometer yang real time. Akan tetapi, terlihat bahwa terdapat masalah yang muncul. Yaitu ketika filter sedang di inisialisasi. Keluar filter cenderung memiliki overshoot yang besar. Setiap kali sudut berubah, muncul overshoot baru. Overshoot dapat menyebabkan masalah jika menjadi input kontroler PID.

Solusinya adalah dengan coba mengatur nilai  $Q_{gyroscope}$  . Setelah beberapa kali percobaan didapat nilai optimal, yaitu untuk  $Q_{gyroscope} = 0.003$  dan  $Q_{gyroscope} = 0.001$ . Ini adalah hasil terbaik yang penulis temukan. Hasil ditunjukkan pada Gambar 15

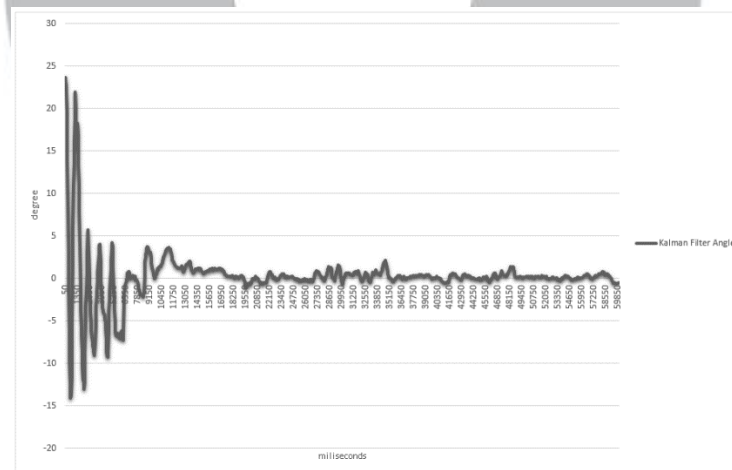


Gambar 15. Grafik pengujian dengan tuning (R=0.03 QG=0. 003 QA=0.001)

### 3.4. Pengujian Kontroler PID

Tujuan dari kontroler PID adalah kestabilan. PID terdiri dari tiga bagian yang mempengaruhi respon keluaran dari sistem.  $K_P$  dapat memperbaiki rise time dari suatu sistem.  $K_I$  dapat memperbaiki error saat steady state.  $K_D$  dapat memperbaiki overshoot dan kestabilan sistem.

Hasil dari gambar 16 menunjukkan respon robot pada sistem loop tertutup PID. Terlihat robot menjaga kestabilannya di sudut  $0^\circ$  . Pada gambar terlihat bahwa pada awal inisialisasi terdapat sebuah overshoot yang disebabkan inisialisasi prediksi sudut kalman, akan tetapi setelah itu overshoot itu hilang. Parameter PID yang digunakan adalah  $K_P = 9.0$   $K_I = 2.0$   $K_D = 3.0$  .



Gambar 16. Pengujian pada tendensi robot menjaga keseimbangannya

#### 4. Kesimpulan

Pada kasus ini accelerometer dan gyroscope yang digunakan murah. Karena itu kedua sensor ini susah sekali digunakan. Dikarenakan pada accelerometer terdapat banyak sekali noise yang tidak diinginkan. Pada gyroscope adanya drift terhadap waktu. Maka dari itu dibutuhkan sensor fusion yang dilakukan dengan Kalman Filter.

Setelah kita melakukan pengujian kita mengerti apa pengaruh dari masing-masing parameter, semakin kecil nilai  $R_{pengukuran}$ ,  $Q_{accelerometer}$ ,  $Q_{gyroscope}$ . Semakin percaya kita dengan sistem yang sudah ada, ini membuat respon semakin cepat akan tetapi banyak akan noise. Semakin tinggi nilainya semakin tidak percaya sehingga diproses lebih jauh. Ini yang mengakibatkan respon sedikit terlambat. Maka dari itu diperlukan tuning yang optimal untuk menyelesaikan masalah ini.

Jurnal menunjukkan bahwa sebuah robot pendulum terbalik dapat dibuat dari komponen yang murah. Karena harganya yang murah, jurnal ini cocok untuk menjadi demonstrasi tentang kontrol di kelas atau perkuliahan.

#### Daftar Pustaka:

- [1] Segway (2014, Nov).<http://www.segway.com>.
- [2] R. Fierro, F. Lewis, and A. Lowe, *Hybrid control for a class of underactuated mechanical systems*, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 29, no. 6, pp. 649–4, nov 1999.
- [3] K. Xu and X.-D. Duan, *Comparative study of control methods of single-rotational inverted pendulum*, in Proceedings of the First International Conference on Machine Learning and Cybernetics, vol. 2, 2002, pp. 776–8.
- [4] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "JOE: A mobile, inverted pendulum," IEEE Transactions on Industrial Electronics, vol. 49, no. 1, pp. 107–14, 2002.
- [5] D. P. Anderson. (2003, Aug.) nBot balancing robot. Online. [Online]. Available: <http://www.geology.smu.edu/dpa-www/robo/nbot>
- [6] R. C. Ooi, "Balancing a two-wheeled autonomous robot," Final Year Thesis, The University of Western Australia, School of Mechanical Engineering, 2003.
- [7] N. M. A. Ghani, F. Naim, and P. Y. Tan, "Two wheels balancing robot with line following capability," World Academy of Science, Engineering and Technology, vol. 55, pp. 634–8, 2011.
- [8] X. Ruan, J. Liu, H. Di, and X. Li, "Design and LQ control of a two-wheeled self-balancing robot," in Control Conference, 2008. CCC 2008. 27th Chinese, july 2008, pp. 275 –279.
- [9] T. Nomura, Y. Kitsuka, H. Suemistu, and T. Matsuo, "Adaptive backstepping control for a two-wheeled autonomous robot," in ICROSSICE International Joint Conference, Aug. 2009, pp. 4687–92.
- [10] G. M. T. Nguyen, H. N. Duong, and H. P. Ngyuen, "A PID backstepping controller for two-wheeled self-balancing robot," in Proceedings fo the 2010 IFOST, 2010.
- [11] K.-H. Su and Y.-Y. Chen, "Balance control for two-wheeled robot via neural-fuzzy technique," in SICE Annual Conference 2010, Proceedings of, aug. 2010, pp. 2838 –2842.
- [12] X. Ruan and J. Cai, "Fuzzy backstepping controller for two-wheeled self-balancing robot," in International Asia Conference on Informatics in Control, Automation and Robotics, 2009, pp. 166–9.
- [13] Miranda, J. L. C., 2009. *APPLICATION OF KALMAN FILTERING AND PID CONTROL FOR DIRECT INVERTED PENDULUM CONTROL*, California State University, Chico.

- [14] Welch,G. & Bishop,G.,2010. *An Introduction to the Kalman Filter*.
- [15] Wikipedia (2014, Nov). Gyroscope. [Online]. Available: <http://en.wikipedia.org/wiki/Gyroscope>
- [16] Abreu, V. V., 2009. *BALANCE-BOT* ,Funchal ,Universidade da Madeira.
- [17] Ahmad, N. B., 2013. *LINEAR QUADRATIC GAUSSIAN (LQG) CONTROLLER FOR INVERTED PENDULUM*, University Tun Hussein Onn Malaysia.
- [18] Colton, S., 2007. *The Balance Filter, A Simple Solution for Integrating Accelerometer and Gyroscope Measurement for a Balancing Platform*, Massachusetts Institute of Technology.
- [19] Lyon, A. *Why are Normal Distributions Normal?*, University of Maryland, College Park.
- [20] Orback, B.C.P., 2009. *Mobile Inverted Pendulum Control Of An Unstable Process Using Open Source Real-Time Operating System*, Gothenburg, Sweden, CHALMERS UNIVERSITY OF TECHNOLOGY.
- [21] Pedley, M., 2013. *Tilt Sensing Using a Three-Axis Accelerometer*, Freescale Semiconductor ,Application Note.
- [22] Permadi, T. A.,2014. *Balancing System of Tray on Waiter Robot using Complementary Filter and Fuzzy Logic*, Bali, IAICT 2014.
- [23] Simon, D.,2001. *Embedded Systems Programming*.

