

Implementasi g-h Filter Pada Sensor Kompas Sebagai Peningkatan Akurasi *Trajectory Tracking* Robot *Differential Drive*

Gillang Al Azhar^{1*}, Totok Winarno², Syarifatul Izza³

^{1,2} Politeknik Negeri Malang, Malang

³ Politeknik Unisma Malang, Malang

email: gillang_al_azhar@polinema.ac.id

Received:10/12/2021

Revised:25/12/2021

Accepted:03/01/2022

Abstract. *Trajectory tracking for differential drive robot's movement depends on how good the sensor performance that in use, where odometry sensor is always been used as the main sensor, and a lot of factors that could affect the sensor's performances. The way to overcome this problem is to add another sensors, such as a compass sensor. A compass sensor has a noisy data output, so a filter is needed for decreasing the value of the noise from the sensor, and the results will be improved by applying the filter to the sensor. One of those filter is g-h filter. This research simulates compass data which already has noise as an input for the g-h filter algorithm, and for the kinematic control, the filter's output will be used as an input for the differential drive mobile robot. As using g-h filter, a proper value of the g and h variable is needed, so the output of the filter will be more optimal. From the results of the simulation, a proper value for the g variable is 0.56, and the h variable is 0.12, which produce an optimal output for filtering the compass data and the results of the trajectory tracking can be improved, as the results shown that the root mean square error value is less than 0.02 meter.*

Keywords: *g-h filter, mobile robot, trajectory tracking, kinematic control*

Abstrak. Proses *trajectory tracking* pergerakan robot *differential drive* sangatlah bergantung pada data sensor yang digunakan, dimana umumnya menggunakan sensor odometri yang sangat rentan akurasi. Dengan menambahkan sensor lain merupakan salah satu upaya dalam meningkatkan akurasi algoritma tersebut, salah satunya adalah dengan sensor kompas. Namun *noise* data yang sangat tinggi yang dimiliki sensor kompas, berakibat diperlukannya filter untuk memperbaiki data sensor tersebut, dengan tujuan agar hasil proses *trajectory tracking* dapat lebih akurat. Salah satunya dengan g-h filter. Penelitian dilakukan dengan mensimulasikan data kompas yang memiliki noise yang sangat tinggi sebagai input dari g-h filter dimana output data filternya nanti dijadikan input data dari kontrol kinematik dari robot *differential drive*. Dalam penggunaan g-h filter dibutuhkan proses tuning pada parameter g dan h yang optimal, sehingga mampu menghasilkan output yang sesuai. Dari hasil simulasi pengujian g-h filter didapatkan nilai konstanta g sebesar 0.56 dan konstanta h sebesar 0.12 yang memiliki output data yang untuk menghilangkan data noise dari sensor kompas sehingga mampu meningkatkan akurasi data dari *trajectory tracking* pergerakan robot beroda, terbukti dari hasil pengujian yang menunjukkan nilai rata – rata error data dibawah 0.02 meter.

Kata Kunci: filter g-h, robot beroda, *trajectory tracking*, kontrol kinematik

I. PENDAHULUAN

Trajectory tracking pada pergerakan robot beroda dilakukan dengan tujuan untuk mengetahui posisi robot tersebut di area tempat robot tersebut diletakkan. Akurasi data pada *trajectory tracking* posisi robot sangatlah penting, karena semakin akurat data posisi dari robot yang dimiliki, semakin mudah untuk mengoperasikan robot tersebut secara otonomus. Proses *trajectory tracking* dapat

dilakukan dengan menggunakan berbagai macam sensor, seperti penggunaan sensor odometri[1][2], sensor laser [3][4], sensor ultrasonic[5][6], serta sensor kamera [7][8]. Dimana masing – masing penggunaan sensor tersebut memiliki kelebihan dan kekurangannya masing – masing.

Pada artikel [9] pembahasan mengenai penggunaan sensor odometry pada robot beroda,

dimana dapat disimpulkan bahwa penggunaan sensor odometry saja memiliki data yang kurang akurat dikarenakan adanya selip pada ban robot, kemudian pada artikel [10] yang membahas kurang akuratnya penggunaan sensor laser pada tracking posisi robot dikarenakan sensor laser yang hanya mampu bekerja secara 2 dimensi saja, kemudian pada artikel [11] yang meneliti penggunaan sensor ultrasonik sebagai sensor tracking posisi robot yang hasilnya kurang akurat karena sensor ultrasonik memiliki banyak titik buta pada pemasangannya. Serta artikel [12] yang membahas mengenai penggunaan sensor kamera yang memiliki data yang bagus, namun memiliki *cost* harga yang sangat tinggi. Dari referensi yang didapatkan pada penelitian – penelitian sebelumnya, sensor yang dapat dioptimalkan penggunaannya adalah sensor odometry, dimana untuk meningkatkan akurasi datanya dapat dilakukan dengan memanfaatkan sensor kompas.

Sensor kompas memiliki kekurangan dan kelebihan yang dapat mempengaruhi peforma penggunaannya. Pada dasarnya sensor kompas ini digunakan untuk menunjukkan arah secara akurat, tetapi beberapa modul sensor yang tersedia dipasaran memiliki kekurangan yaitu terdapat noise berupa data *drifting* yang mampu membuat data dari sensor tersebut menjadi tidak pasti dan tidak akurat. Sehingga dibutuhkan algoritma yang mampu mengurangi noise yang ada pada sensor kompas tersebut, sehingga nantinya dapat diintegrasikan datanya dengan sensor odometry agar dapat meningkatkan akurasi dari data trajectory tracking dari robot beroda. Pada jurnal ini akan dibahas dan disimulasikan penggunaan metode g-h filter untuk memperbaiki data sensor kompas yang nantinya akan digunakan untuk meningkatkan data trajectory tracking dari robot beroda 2 dengan sistem penggerak *differential drive*. Simulasi akan dilakukan untuk membandingkan hasil trajectory tracking dari robot beroda yang hanya menggunakan sensor odometry saja, kemudian dengan menggabungkan data sensor odometry dan sensor kompas, serta data sensor odometry dan sensor kompas yang telah diberi g-h filter pada datanya.

II. METODE YANG DIGUNAKAN

Trajectory tracking dilakukan dengan memanfaatkan data dari kontrol kinematik robot beroda. Selain itu data kompas yang masih memiliki noise yang tinggi serta data kompas yang telah difilter dengan menggunakan g-h filter akan disimulasikan sebagai input dari kontrol kinematik robot beroda. Sehingga pembahasan metode yang digunakan akan dibagi menjadi 4, yaitu pembahasan g-h filter, pemodelan untuk kontrol kinematik, kontrol gerak kinematik pada robot *differential drive*, dan simulasi implementasi g-h filter pada kontrol kinematik robot *differential drive*.

A. Sensor Kompas GY-BNO055

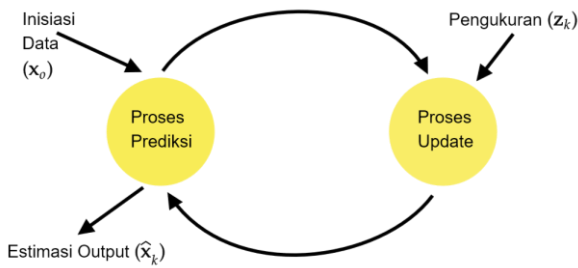
Sensor kompas yang digunakan sebagai acuan data input yang digunakan adalah GY-BNO055, dimana sensor kompas ini telah memiliki internal filter yang bersifat pabrikan yang berguna untuk mengurangi data dari masing – masing sensor yang ada didalamnya, yaitu sensor *gyroscope*, *accelerometer*, dan *magneto*. Data dari sensor GY-BNO055 ini meskipun telah diberikan internal filter, tetapi masih memiliki noise yang cukup besar pada data outputnya. Noise yang dimaksud adalah data *drift* pada output dari sensor tersebut.

B. Anaconda-Spyder sebagai platform simulator pemrograman bahasa python

Data sensor yang diambil akan diberikan algoritma filter dan disimulasikan kedalam kontrol kinematik robot roda dengan menggunakan pemrograman Bahasa python. Aplikasi yang digunakan adalah Anaconda-Spyder, dimana aplikasi ini selain digunakan untuk menuliskan program, telah dilengkapi dengan terminal untuk menjalankan program secara langsung. Sehingga data yang telah didapatkan dari sensor, kemudian disimulasikan kedalam kinematik robot, dan hasilnya akan bisa langsung terlihat.

C. G-H Filter

G–H filter adalah sebuah algoritma filter yang memanfaatkan 2 buah konstanta sebagai parameter filternya [13][14]. Konstanta tersebut adalah konstanta g dan konstanta h. Konstanta g adalah sebuah nilai yang digunakan sebagai konstanta pengali data pengukuran.



Gambar 1. Jalannya algoritma g-h filter

Sedangkan untuk konstanta h merupakan sebuah nilai yang digunakan sebagai konstanta pengali rate perubahan data pengukuran. Sehingga berdasarkan 2 konstanta tersebut dapat dihasilkan output yang dituliskan sebagai notasi berikut

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (1)$$

Seperti yang dapat dilihat pada gambar 1, g-h filter merupakan sebuah algoritma yang menghasilkan sebuah prediksi data yang nilainya didasari dengan nilai pengukuran. Prediksi data yang dihasilkan bergantung dari nilai output prediksi yang telah dihasilkan sebelumnya, sehingga semakin banyak iterasi (pengambilan data) yang dilakukan akan membuat output filter ini semakin bagus, dengan catatan mampu memberikan nilai konstanta g dan h yang tepat. Proses prediksi dilakukan berdasarkan nilai rate perubahan data output yang kemudian dijadikan input atau berdasarkan dari nilai inialisasi yang diberikan pada awal filter dijalankan. Perhitungan dapat dituliskan sebagai berikut

$$\bar{\mathbf{x}} = \mathbf{x}_0 + d\mathbf{x} * \Delta t \quad (2)$$

Dimana $\bar{\mathbf{x}}$ merupakan hasil prediksi berdasarkan input \mathbf{x}_0 yang berasal dari nilai inialisasi atau berdasarkan nilai output terakhir dari filter yang ditambahkan nilai rate perubahan data ($d\mathbf{x}$) per waktu samplingnya (Δt). Kemudian hasil prediksi tersebut akan dikoreksi melalui step update, yaitu dimana nilai prediksi akan dikoreksi datanya berdasarkan nilai pengukuran yang hasilnya akan bergantung dengan nilai konstanta g dan h yang diberikan [15], berikut persamaan dari proses update pada implementasi g-h filter

$$\mathbf{e} = \mathbf{z} - \bar{\mathbf{x}} \quad (3)$$

$$d\mathbf{x} = d\mathbf{x} + \frac{h * \mathbf{e}}{\Delta t} \quad (4)$$

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + g * \mathbf{e} \quad (5)$$

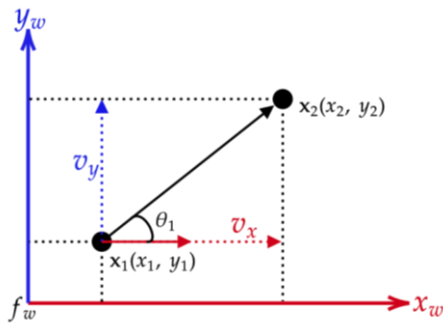
Pada persamaan (3) dihitung nilai residual (\mathbf{e}) atau perbedaan nilai prediksi dan nilai pengukuran. Kemudian dihitung kembali rate perubahan data ($d\mathbf{x}$), kali ini dihitung berdasarkan nilai residual yang telah didapatkan sebelumnya. Perhitungan rate perubahan data ini melibatkan nilai konstanta h . Selanjutnya hitung nilai output $\hat{\mathbf{x}}$, dimana yang perlu diperhatikan adalah nilai output ini disimbolkan memiliki tanda $\hat{}$ yang berarti ini merupakan nilai estimasi. Perhitungan output estimasi ini dihitung dengan melibatkan nilai konstanta g .

D. Kontrol Kinematik

Pada dasarnya kontrol kinematik adalah sebuah kontrol yang mengatur pergerakan dari sebuah benda secara translasi dan rotasi pada sumbu pergerakan. Berikut merupakan notasi dari pergerakan translasi dan rotasi dari benda, yang dapat dituliskan sebagai berikut

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (6)$$

Dengan \mathbf{x} merupakan vektor yang mewakili pergerakan benda translasi dan rotasi yang terdiri dari pergerakan translasi sumbu x, y dan z , serta pergerakan rotasi sumbu $x (\omega_x), y (\omega_y)$ dan $z (\omega_z)$.



Gambar 2. Pergerakan benda pada frame 2 dimensi

Sebuah benda yang bergerak pada bidang 2 dimensi dapat bergerak ke arah sumbu x dan sumbu y . Dimana pada gambar 3 dapat diperhatikan ketika sebuah benda bergerak dari titik $x_1[x_1, y_1]$ ke titik $x_2[x_2, y_2]$. Pada pergerakan ini benda tersebut mampu bergerak dengan kecepatan tertentu ke arah sumbu x (v_x) dan sumbu y (v_y). Serta benda tersebut mampu melakukan rotasi terhadap sumbu z (ω_z) sebesar sudut θ . Robot beroda merupakan salah satu kasus yang dapat dimasukkan kedalam pergerakan pada bidang 2 dimensi, dimana dapat diasumsikan bahwa robot beroda tidak mampu melakukan pergerakan translasi ke arah sumbu z dan robot juga tidak mampu melakukan pergerakan rotasi terhadap sumbu x (ω_x) dan sumbu y (ω_y). Sehingga persamaan (6) dapat diubah menjadi

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 0 \\ 0 \\ 0 \\ \omega_z \end{bmatrix} \quad (7)$$

$$= \begin{bmatrix} x \\ y \\ \omega_z \end{bmatrix} \quad (8)$$

dan dikarenakan pergerakan rotasi terhadap sumbu z memiliki nilai θ , maka persamaan (10) dapat dituliskan sebagai berikut

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (9)$$

Seperti yang telah disebutkan sebelumnya bahwa sebuah kontrol kinematik merupakan sebuah kontrol yang mengontrol pergerakan dari sebuah

benda yang berarti sebuah kontrol yang digunakan untuk dapat mengetahui sebuah pergerakan benda, dimana pergerakan benda dapat direpresentasikan sebagai posisi berdasarkan pergerakan dari aktuatornya. Pengertian tersebut dapat direpresentasikan menjadi persamaan berikut

$$\mathbf{x} = f(\mathbf{q}) \quad (10)$$

dimana q merepresentasikan aktuator. Sebuah aktuator akan lebih mudah untuk dikendalikan kecepatannya dibandingkan dengan posisinya, sehingga persamaan (12) dapat dicari persamaan diferensialnya sebagai berikut

$$\frac{d\mathbf{x}}{dt} = \frac{df(\mathbf{q})}{dt} \quad (11)$$

$$\dot{\mathbf{x}} = \frac{df(\mathbf{q})}{d\mathbf{q}} \cdot \frac{d\mathbf{q}}{dt} \quad (12)$$

$$\dot{\mathbf{x}} = \mathbf{J} \cdot \dot{\mathbf{q}} \quad (13)$$

dengan \mathbf{J} adalah matriks jacobian atau matrix yang berisikan turunan fungsi aktuator terhadap masing – masing aktuatornya. Persamaan (13) dapat disebut dengan *forward velocity kinematics*. Sebaliknya jika digunakan untuk mencari kecepatan dari aktuator robot berdasarkan posisi robot digunakan persamaan kebalikannya, yaitu persamaan *invers velocity kinematics* yang dapat dituliskan persamaannya sebagai berikut

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \cdot \dot{\mathbf{x}} \quad (14)$$

Dengan menggunakan persamaan (13) dan (14) dapat digunakan sebagai kontrol gerak robot (*invers velocity kinematic*) serta dapat digunakan sebagai *trajectory tracking* dari posisi robot saat robot tersebut bergerak (*forward velocity kinematic*). Setelah didapatkan nilai kecepatan perpindahan robot ($\dot{\mathbf{x}}$) pergerakan robot dapat di *tracking* dengan melakukan update pada nilai posisinya, dapat diaplikasikan persamaan berikut

$$\frac{d\mathbf{x}}{dt} = \frac{\Delta \mathbf{x}}{\Delta t} \quad (15)$$

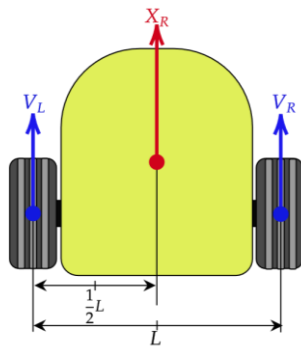
$$\dot{\mathbf{x}} = \frac{(\mathbf{x}_t - \mathbf{x}_{t-1})}{\Delta t} \quad (16)$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \dot{\mathbf{x}} \cdot \Delta t \quad (17)$$

Dimana nilai posisi robot yang baru didapatkan dari posisi terakhir robot berada ditambahkan dengan nilai kecepatan perpindahan posisi robot yang dikalikan dengan waktu sampling.

E. Pergerakan Robot Beroda dengan Sistem Pergerakan Differential Drive

Differential Drive Mobile Robot merupakan salah satu jenis robot beroda yang memiliki 2 buah aktuator yang digunakan untuk bergerak. Agar pergerakan robot dapat terkontrol dengan baik serta dapat di *tracking* dengan baik maka dibutuhkan kontrol kinematik untuk mengatur pergerakannya. Kontrol kinematik sendiri bersifat general untuk berbagai macam jenis robot, termasuk juga untuk mengendalikan robot beroda *Differential Drive*.



Gambar 3. Parameter pada robot dengan sistem penggerak differential drive

Dasar acuan dari persamaan kontrol kinematik adalah persamaan (13) dan (14), yang membedakan dari kontrol kinematik yang digunakan antara jenis robot satu dengan jenis lainnya ada pada jacobianya (J). Desain pemodelan kontrol kinematik yang digunakan pada robot *differential drive* persamaan *forward velocity kinematic*-nya adalah

$$J = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L_B/2 & L_B/2 \\ L_B/r_w & -L_B/r_w \end{bmatrix} \quad (18)$$

Sehingga persamaan (18) dapat disubstitusikan ke persamaan (13) untuk persamaan *forward velocity kinematic* - nya dan persamaan (14) untuk persamaan *invers velocity kinematic* - nya, yang dapat dituliskan sebagai berikut,

Untuk persamaan *forward velocity kinematic* - nya

$$\dot{x} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L_B/2 & L_B/2 \\ L_B/r_w & -L_B/r_w \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (19)$$

$$\dot{q} = \begin{bmatrix} r_w/2 \cos(\theta) & r_w/2 \cos(\theta) \\ r_w/2 \sin(\theta) & r_w/2 \sin(\theta) \\ r_w/L_B & -r_w/L_B \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (20)$$

F. Simulasi implementasi g-h filter pada kontrol kinematik robot Differential Drive

Untuk melakukan pengujian peforma g-h filter yang diterapkan untuk memperbaiki data dari sensor kompas dapat disimulasikan dengan membandingkan hasil dari *trajectory tracking* antara robot yang tidak menggunakan sensor, hanya dengan menggunakan sensor odometry saja, kemudian robot yang menggunakan sensor odometry dan sensor kompas tanpa filter, serta dengan robot yang menggunakan sensor odometry dan sensor kompas dengan g-h filter didalamnya. Simulasi dilakukan dengan memberikan noise pada data sensor odometry serta pada data sensor kompas.

Robot akan disimulasikan untuk bergerak secara lurus, sehingga dapat diasumsikan data sensor kompas jika ideal adalah selalu bernilai 0, tetapi dengan adanya penambahan nilai random dengan standard deviasi tertentu, maka nilai random tersebut dapat diibaratkan sebagai noise. Berikut adalah algoritma program yang digunakan untuk melakukan simulasi robot yang bergerak hanya dengan menggunakan sensor odometry:

1. Inisialisasi posisi awal robot ($x_{t=0}$)
2. Inisialisasi posisi target yang akan dituju robot (x^*)
3. Hitung nilai error posisi (e)
4. Hitung nilai kecepatan aktuator (\dot{q}) dengan menggunakan persamaan *invers velocity kinematic*
5. Hitung kecepatan perpindahan (\dot{x}) dengan menggunakan persamaan *forward velocity kinematic*

6. Update Posisi robot serta dengan menambahkan data random sebagai asumsi adanya noise pada pergerakan robot
7. Ulangi mulai dari step 2 hingga nilai error mendekati 0

Kemudian berikut ini merupakan algoritma untuk melakukan simulasi robot yang bergerak dengan menggunakan sensor odometry dan sensor kompas tanpa menggunakan filter

1. Inisialisasi posisi awal robot ($x_{t=0}$)
2. Inisialisasi posisi target yang akan dituju robot (x^*)
3. Hitung nilai error posisi (e)
4. Hitung nilai kecepatan aktuator (\dot{q}) dengan menggunakan persamaan *invers velocity kinematic*
5. Hitung kecepatan perpindahan (\dot{x}) dengan menggunakan persamaan *forward velocity kinematic*
6. Update Posisi robot serta dengan menambahkan data random sebagai asumsi adanya noise pada pergerakan robot
7. Ganti data arah hadap robot dengan data sensor kompas yang diasumsikan bernilai 0 dengan ditambahkan nilai random sebagai asumsi noise data yang ada pada sensor kompas
8. Ulangi mulai dari step 2 hingga nilai error mendekati 0

Kemudian untuk algoritma simulasi robot yang bergerak dengan menggunakan sensor odometry dan sensor kompas yang menggunakan g-h filter ditambahkan 1 tahapan lagi yaitu masukkan data sensor kompas kedalam perhitungan g – h filter pada step ke 8.

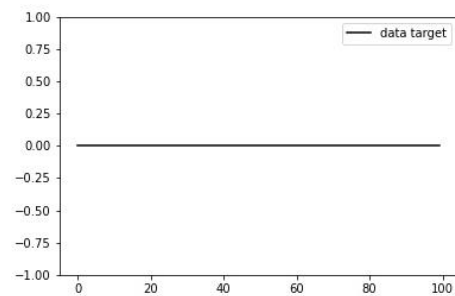
III. HASIL DAN PEMBAHASAN

Hasil simulasi yang didapatkan adalah berupa 3 buah data pergerakan robot yang memanfaatkan 3 kondisi yang berbeda, yaitu *trajectory tracking* pergerakan robot dengan sensor odometry saja, robot dengan sensor odometry dan sensor kompas tanpa filter, serta robot dengan sensor odometry dan sensor kompas yang telah diberikan g-h filter pada datanya. Data pengujian simulasi akan disajikan dalam bentuk grafik, yang menunjukkan data arah

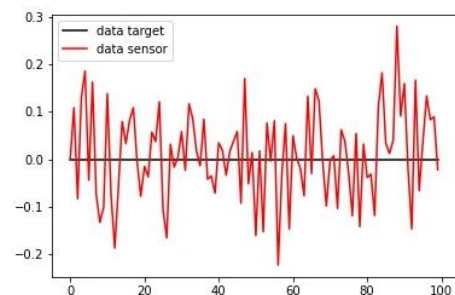
hadap sebenarnya, data sensor kompas dengan noise, dan data sensor kompas yang telah diberikan g-h filter. Kemudian grafik lainnya merepresentasikan *trajectory tracking* dari pergerakan robot ketika diberikan titik koordinat tujuan yang sama namun dengan pengolahan data sensor yang berbeda.

A. Pengujian g-h filter pada data sensor kompas

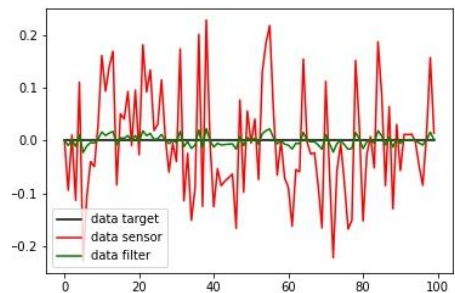
Berikut merupakan hasil penggunaan g-h filter pada data sensor kompas yang memiliki data noise yang cukup tinggi. Besaran konstanta g dan h diambil beberapa sample untuk menguji efek dari perubahan nilai konstanta yang diberikan. Berikut data pengujian yang disajikan dalam bentuk grafik.



(a) Kondisi data ideal



(b) Noise pada data pembacaan sensor (Merah)



(c) Data sensor yang diberi filter (Hijau)

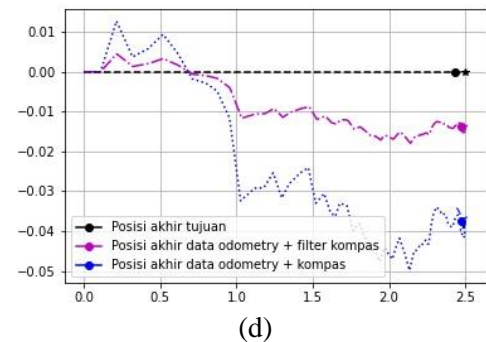
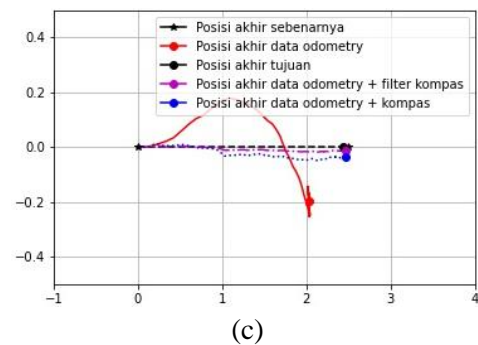
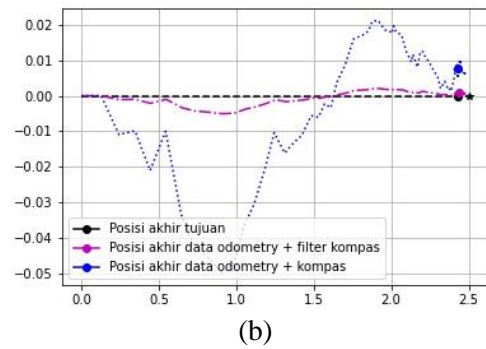
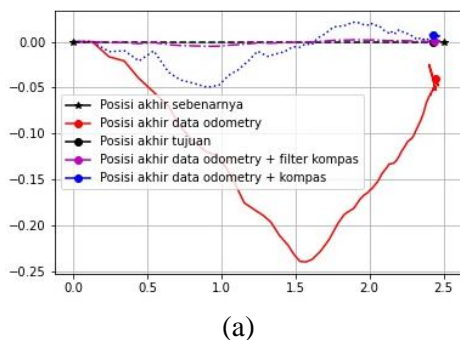
Gambar 4. Hasil perbandingan data pembacaan sensor

Sumbu y menunjukkan data nilai sudut dalam satuan radian, dan sumbu x menunjukkan jumlah

waktu yang digunakan untuk mengambil data dalam satuan detik. Dari hasil yang didapatkan dan ditunjukkan pada gambar 4, dapat dilihat bahwa data sensor (ditunjukkan dengan warna merah pada (b) dan (c)) yang memiliki noise akan memiliki grafik yang dapat dibidang acak dan tidak tentu, serta nilai sudut yang diberikan cukup berbeda jauh dari kondisi ideal yang diinginkan (ditunjukkan dengan warna hitam pada (a), (b), dan (c)). Pada gambar (c) dimana data sensor telah diberikan filter (ditunjukkan dengan warna hijau), dapat dilihat bahwa noise yang ada pada data sensor dapat direduksi dan nilainya hampir mendekati nilai idealnya. Berdasarkan hasil tersebut dapat dilihat bahwa penggunaan g-h filter dapat meningkatkan data sensor dengan noise yang bersifat acak dan tidak pasti, menjadi data sensor yang lebih stabil dan layak untuk dapat digunakan sebagai input dari sistem kontrol kinematik. Secara perbandingan data, dapat diperkirakan bahwa penggunaan data sensor yang telah terfilter akan lebih baik untuk digunakan dibandingkan dengan data sensor yang belum diberikan filter.

B. Hasil Trajectory Tracking dengan 3 metode penggunaan sensor yang berbeda

Berikut ini merupakan hasil simulasi trajectory tracking terhadap robot yang berjalan dengan 3 metode berbeda. Hasil trajectory tracking dari simulasi pergerakan robot ditunjukkan dalam grafik yang menunjukkan pergerakan dari masing – masing robot.



Gambar 5. Perbandingan data trajectory tracking pada robot beroda

Pada gambar 5 menunjukkan data posisi koordinat dalam satuan meter. Pada gambar bagian (a) dan (c) menunjukkan perbandingan dari trajectory tracking pergerakan dari masing – masing robot, dan pada gambar bagian (b) dan (d) menunjukkan pengaruh penggunaan filter untuk data sensor kompas pada trajectory tracking pergerakan robot beroda. Berikut hasil pengujian simulasi dari 5 kali percobaan dengan titik tujuan yang berbeda – beda, dimana R1 menunjukkan data simulasi hanya dengan data odometry saja, R2 menunjukkan data simulasi dengan menggunakan kompas tanpa filter, dan R3 adalah penggunaan g-h filter pada data kompas. Dimana semua data yang ada pada tabel 1 merupakan titik koordinat pada sumbu [x, y] dalam satuan meter.

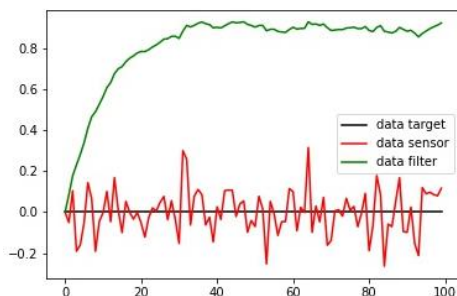
Tabel 1. Data pengujian trajectory tracking pergerakan robot

$x(m)$	$x^*(m)$	Posisi Akhir (m)		
		R1	R2	R3
[0,0]	[2,0]	[1.3, 3]	[1.9, 0.3]	[1.9, 0.1]
	[2,2]	[3, 4.2]	[1.7,-1.9]	[1.9, 2.1]
	[3,0]	[2, -1.2]	[2.8, 0.4]	[2.9, 0.1]
	[4,3]	[3.7,2.1]	[3.9, 0.2]	[4.0, 2.9]
	[4,4]	[2.3,6.4]	[3.9, 0.4]	[3.9, 3.8]

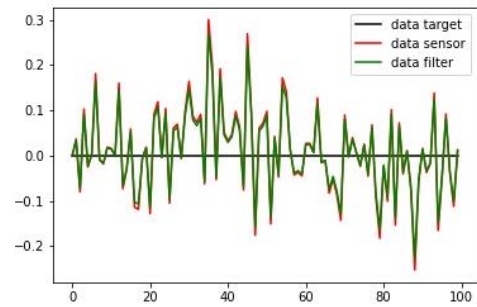
Dari hasil yang didapatkan, dapat diperhatikan peningkatan akurasi data yang dihasilkan dengan penggunaan filter pada data sensor kompas yang digunakan. Dapat diperhatikan bahwa hasil trajectory tracking dari robot yang menggunakan data sensor kompas yang telah difilter memiliki akurasi yang hampir mendekati kondisi ideal dari pergerakan robot yang berjalan secara lurus. Hal ini terbukti dari nilai rata – rata error tiap datanya adalah kurang dari 0.02 meter untuk data yang menggunakan g-h filter.

C. Pengujian perubahan data filterisasi kompas berdasarkan nilai konstanta g-h filter

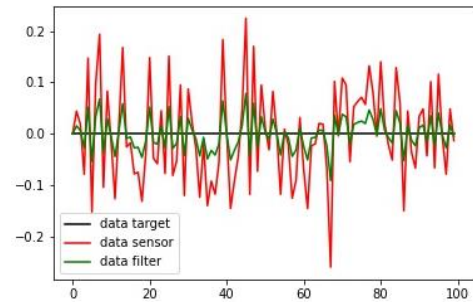
Hasil pengujian berdasarkan perubahan data trajectory tracking robot, disimulasikan berdasarkan pada sensor odometry dan kompas yang telah diberikan g-h filter. Perbedaan nilai konstanta g yang diujikan menunjukkan efek dari perubahan trajectory tracking dari robot.



(a)



(b)



(c)

Gambar 6. Pengujian perubahan output filter berdasarkan nilai konstanta pada g-h filter

Perubahan nilai konstanta g-h pada filter akan berpengaruh terhadap hasil output filter yang dihasilkan, hal ini dapat dilihat pada gambar 6, dimana pada gambar 6(a) nilai konstanta g bernilai 0.2 dan konstanta h bernilai 0.1 berakibat output filter yang semakin tidak akurat. Pada gambar 6(b) nilai konstanta g bernilai 0.9 dengan nilai konstanta h yang tetap. Hal ini mengakibatkan data output filter sama dengan data sensor aslinya, yaitu memiliki banyak noise. Sedangkan pada gambar 6(c) nilai konstanta g bernilai 0.65 dengan konstanta h yang sama, menghasilkan data yang lebih baik dan semakin mendekati nilai ideal yang diinginkan. Kombinasi nilai konstanta yang berhasil menghasilkan output yang mendekati nilai ideal adalah 0.56 untuk konstanta g dan 0.12 untuk konstanta h. Nilai output yang dihasilkan ditunjukkan pada gambar 4(c).

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Hasil Pergerakan robot yang memanfaatkan data sensor kompas yang telah difilter menunjukkan hasil yang lebih baik. Hal ini terbukti bahwa penggunaan g-h filter dalam proses filterisasi data kompas mampu memberikan data trajectory

tracking pergerakan robot *differential drive* yang lebih baik. Meskipun hasilnya masih belum sesuai dengan kondisi ideal, tetapi terbukti penggunaan g-h filter dapat meningkatkan akurasi dari *trajectory tracking* dari pergerakan robot dengan catatan nilai konstanta g dan h pada filter dapat diatur secara optimal. Dari hasil simulasi didapatkan nilai konstanta g sebesar 0.56 dan konstanta h sebesar 0.12 dapat memberikan hasil filterisasi yang optimal.

B. Saran

Berdasarkan hasil pengujian secara simulasi dari penggunaan g-h filter telah menunjukkan hasil yang cukup baik, sehingga penggunaan g-h filter ini dapat diuji cobakan kedalam prototype robot *differential drive* untuk melihat performa yang dihasilkan. Selain itu agar filter yang digunakan dapat lebih adaptif, disarankan menggunakan pengembangan dari g-h filter yaitu dengan menggunakan Kalman filter.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih pada lab kontrol dan robotika Politeknik Negeri Malang yang telah memberikan fasilitas untuk mendukung proses kegiatan penelitian yang telah dilaksanakan, serta kepada group riset ISAR(*intelligent System and Robotics*) Politeknik Negeri Malang yang telah bersedia menjadi wadah dan tempat diskusi dalam penelitian dibidang sistem kendali dan robotika.

REFERENSI

- [1] SIRADJUDDIN, Indrazno et al. Desain dan pemodelan kontrol kinematik pergerakan robot beroda dengan menggunakan 6 roda omni-wheels. **JURNAL ELTEK**, [S.l.], v. 18, n. 1, p. 116-127, apr. 2020. ISSN 2355-0740.
- [2] I. Siradjuddin, T. Winarno, M. Khairuddin, M. N. Achmadiyah, R. P. Wicaksono and G. Al Azhar, "An Exponential Decreased Kinematic and PID Low Level Control Schemes for an Omni-Wheeled Mobile Robot," *2021 International Conference on Electrical and Information Technology (IEIT)*, 2021, pp. 274-280, doi: 10.1109/IEIT53149.2021.9587370
- [3] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.[Online].Available:<https://doi.org/10.1177/1729881419839596>
- [4] I. Siradjuddin *et al.*, "A Non-Iterative Solution for Rigid Body Transformation Estimation," *2018 International Conference on Applied Science and Technology (iCAST)*, 2018, pp. 573-576, doi: 10.1109/iCAST1.2018.8751534
- [5] R. H. Abiyev, N. Akkaya, and I. Günsel, "Control of omnidirectional robot using z-number-based fuzzy system," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 238–252, 2019.
- [6] P. Devid, P. Bambang Eka, W. Indah Uly, "Implementasi Model Robot Edukasi Menggunakan Mikrokontroler Atmega8 Untuk Robot Pemadam Api," in *IPI Indonesian Journal on Networking and Security (IJNS)*, Surakarta, 2013.
- [7] Y.-H. Cheng, P.-J. Chao, and C.-N. Kuo, "Mobile robot path planning using a teaching learning-interactive learning-based optimization," *IAENG International Journal of Computer Science*, vol. 46, no. 2, pp. 199–207, 2019.
- [8] E. Kaneko and N. Umezū, "Rapid construction of coarse indoor map for mobile robots," *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, Nagoya, 2017, pp. 1-3, doi: 10.1109/GCCE.2017.8229367.
- [9] C. H. KAZANCI and A. F. KOCAMAZ, "PID Optimization on Differential Drive Mobile Robot," *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Turkey, 2018, pp. 1-6.
- [10] A. Birari, A. Kharat, P. Joshi, R. Pakhare, U. Datar and V. Khotre, "Velocity control of omni drive robot using pid controller and dual feedback", *2016 IEEE First International Conference on Control Measurement and Instrumentation (CMI)*, pp. 295-299, 2016.

- [11] D. Chwa and J. Boo, "Adaptive fuzzy output feedback simultaneous posture stabilization and tracking control of wheeled mobile robots with kinematic and dynamic disturbances", *IEEE Access*, vol. 8, pp. 228 863-228 878, 2020.
- [12] I. Siradjuddin, S. P. Tundung, A. S. Indah and S. Adhisuwignjo, "A real-time model based visual servoing application for a differential drive mobile robot using beaglebone black embedded system", *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, pp. 186-192, 2015.
- [13] R. R. Labbe, "Kalman and Bayesian Filters in Python", MIT, 2015 [Online]. Available: <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python>
- [14] X. G. Wei, S. Zhang and S. C. Chan, "A novel visual object tracking algorithm using multiple spatial context models and Bayesian Kalman filter," *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 1034-1037, doi: 10.1109/ISCAS.2015.7168813.
- [15] R. Dehghannasiri, M. S. Esfahani, X. Qian and E. R. Dougherty, "Optimal Bayesian Kalman Filtering With Prior Update," in *IEEE Transactions on Signal Processing*, vol. 66, no. 8, pp. 1982-1996, 15 April 2018, doi: 10.1109/TSP.2017.2788419.