

Sistem Kendali Kecepatan Motor pada *Mobile Robot* Menggunakan PID dan Analisis *Disturbance* Berbasis *Disturbance Observer*

Bayu Sandi Marta^{1*}, Indra Ferdiansyah², Fernando Ardila³

^{1,2,3}Politeknik Elektronika Negeri Surabaya

*bayu@pens.ac.id

Abstract

Mobile robot's ability to maintain its speed is very important. When the robot is running and there is disturbance from outside system that affect the robot, causing the speed of the robot becomes unstable. For it is necessary to add a mechanism that is able to make the robot keep pace, one of them by giving control. This research will be built a speed control system on a mobile robot. The speed control system aims to improve performance and stability robot speed by minimizing the effect of unmeasured disturbances from outside the robot eg mechanical changes such as granting a load that can cause changes in the behavior of the system so that the necessary arrangements back to the control system. In this control system used control Proportional, Integral, Derivative (PID) and the analysis of disturbance based Disturbance Observer (DOB). PID control functions to improve motor performance robot. While DOB functions to analyze the disturbance from outside robot. The results of the implementation of the Disturbance Observer to PID control produce average root mean square error value of the reference value for the speed of 19.59 rps for right motor and left motor is 16.66 rps. While the average root mean square error of the reference value with PID control without Disturbance Observer on the right motor is 24.47 rps and the left motor is 18.89 rps.

Keywords : *mobile robot, Proportional Integral Derivative (PID), disturbance, Disturbance Observer (DOB), robust.*

Abstrak

Kemampuan *mobile robot* dalam menjaga kecepatannya sangatlah penting. Ketika robot berjalan dan terdapat gangguan yang tak terukur (*disturbance*) dari luar sistem yang mempengaruhi robot, menyebabkan kecepatan robot menjadi tidak stabil. Untuk itu perlu ditambahkan suatu mekanisme yang mampu membuat robot menjaga kecepatannya, salah satunya dengan memberikan kontrol. Pada penelitian ini akan dibangun sebuah sistem kendali kecepatan pada *mobile robot*. Sistem kendali kecepatan ini bertujuan untuk meningkatkan performa robot dan stabilitas kecepatan dengan meminimalkan pengaruh gangguan yang tidak terukur dari luar robot misalnya perubahan mekanis seperti pemberian beban yang dapat menyebabkan perubahan perilaku sistem sehingga diperlukan pengaturan kembali terhadap sistem kontrol. Dalam sistem kendali ini digunakan kontrol *Proportional, Integral, Derivative* (PID) dan analisis gangguan (*disturbance*) berbasis *Disturbance Observer* (DOB). Kontrol PID berfungsi untuk meningkatkan performa motor robot. Sedangkan DOB berfungsi untuk analisis gangguan dari luar robot. Sehingga diharapkan dari penelitian ini dapat dihasilkan sebuah sistem kontrol yang mampu membuat robot *robust* terhadap *disturbance* dan performa robot tetap terjaga tanpa merubah pengaturan sistem kontrol kembali. Hasil implementasi *Disturbance Observer* ke dalam kontrol PID menghasilkan nilai *root mean square error* rata-rata terhadap nilai acuan kecepatan sebesar 19,59 rps untuk motor kanan dan 16,66 rps untuk motor kiri. Sedangkan rata-rata *root mean square error* terhadap nilai acuan kecepatan dengan kontrol PID tanpa *Disturbance Observer* pada motor kanan sebesar 24,47 rps dan pada motor kiri sebesar 18,89 rps.

Kata kunci : mobil robot, PID, gangguan, observasi gangguan, kehandalan sistem

1. Pendahuluan

Robot adalah suatu alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol

manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu. Berdasarkan konstruksinya robot terdiri atas berbagai macam jenis, contohnya *mobile*

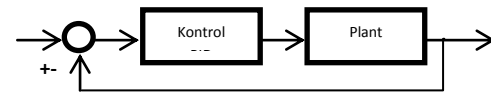
robot, robot manipulator, robot berkaki, dan lain-lain [1]. Dalam dunia robotika khususnya pada *mobilerobot*, sistem kontrol sangatlah penting. Salah satu kegunaan kontrol adalah untuk meningkatkan performa kecepatan *mobile robot* dan menjaga ketahanan performa robot atau *robustness*. Banyak metode yang digunakan dalam meningkatkan dan mempertahankan performa suatu *mobile robot*. Tujuannya tidak lain adalah membuat robot menjadi lebih baik, handal, dan tahan terhadap gangguan yang mempengaruhi performa robot. Gangguan seperti kondisi jalan dan beban dinamis yang diterima robot dapat mengakibatkan ketidakpastian model *plant*, *noise* dan gangguan [2], [3]. Akibat dari ketidakpastian model tersebut akan menyebabkan performa robot menurun. Pada *Disturbance Observer* yang sangat diperhatikan adalah penentuan *low pass filter* dan model *inverse nominal plant*.

Pada penelitian ini, metode yang digunakan adalah kontrol *Proportional Integral Derivative* (PID) yang berbasis *Disturbance Observer* (DOB). Kontrol PID berfungsi untuk mengoptimalkan respon motor penggerak *mobile robot*. Pada kontrol PID yang sangat diperhatikan adalah menentukan nilai konstantanya, karena konstanta akan berpengaruh terhadap performa robot. Sedangkan *Disturbance Observer* berfungsi sebagai analisis gangguan yang diterima robot untuk mempertahankan respon motor yang dikontrol oleh PID.

2. Metoda Penelitian

2.1. Kontrol PID

Proportional-integral-derivative controller (PID controller) adalah sebuah generik kontrol *loop* dengan mekanisme umpan balik. Secara luas digunakan dalam industri sistem kontrol. *PID control* digunakan untuk memperbaiki kesalahan antara variabel proses dan *set point* yang diinginkan dengan menghitung kemudian melakukan koreksi yang dapat menyesuaikan proses dengan cepat dan tepat [1].



Gambar 1. Struktur Kontrol PID

Perhitungan *PID control* melibatkan tiga parameter yang terpisah yaitu nilai proporsional, nilai integral dan nilai derivatif. Nilai proporsional menentukan reaksi terhadap kesalahan saat ini, nilai integral menentukan reaksi berdasarkan jumlah kesalahan baru-baru ini dan nilai derivatif ditentukan dari reaksi yang didasarkan pada tingkat dimana kesalahan telah berubah [4]. Rumus umum *PID* adalah sebagai berikut :

$$U = -K_p e + K_i \int e dt + K_d \frac{de}{dt} \quad (1)$$

dimana :

K_p = Konstanta *Proportional*

K_i = Konstanta *Integral*

K_d = Konstanta *derivative*

Secara umum kontrol *PID* dapat dipisah dimana kontrol P dapat berdiri sendiri ataupun dipasangkan dengan kontrol I saja ataupun kontrol D saja. Dari ketiga parameter tersebut dapat diketahui pengaruh parameter PID terhadap respon sistem yang dapat ditunjukkan pada tabel 1.

Tabel 1. Tanggapan sistem terhadap perubahan parameter PID.

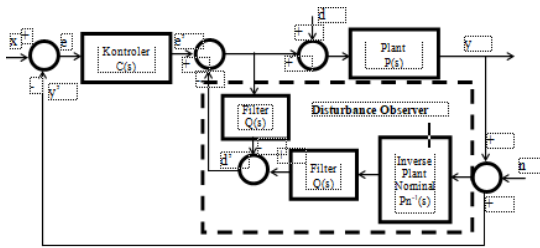
Gain	Rise Time	Overshoot	Settling Time	Steady State error
P	Menurun	Meningkat	Perubahan Kecil	Menurun
I	Menurun	Meningkat	Meningkat	Hilang
D	Perubahan Kecil	Menurun	Menurun	Perubahan Kecil

2.2. Disturbance Observer

Disturbance Observer atau yang biasa disebut dengan DOB merupakan suatu mekanisme pengontrolan yang berada pada loop internal dari sebuah *plant*. Pengendalian bersifat *robust* digunakan untuk mengendalikan *plant* di lingkungan dimana ketidakpastian model dan gangguan berada. Pengendalian berbasis *disturbance observer*

yang menghasilkan sinyal akselerasi ke *plant* adalah pengendali yang bersifat *robust*. Sistem kendali ini membentuk perilaku *input-output* bersifat *robust* dengan menghilangkan gangguan(*d*) terhadap *plant* atau *noise* (*e*) yang mempengaruhi sensor [3], [5].

Metode ini biasanya menggunakan struktur dual loop, yaitu loop internal dan loop eksternal. Loop internal didesain untuk membuat sistem mampu meredam *disturbance* dan ketidakpastian parameter, sedangkan loop eksternal didesain untuk meningkatkan performa sistem secara keseluruhan.



Gambar 2. Struktur *Disturbance Observer*

Berdasarkan arsitektur di atas, kontroler loop internal menghasilkan sinyal kontrol korektif untuk meredam *disturbance* semaksimal mungkin, untuk membuat *plant* aktual menjadi model nominal. *Disturbance* didefinisikan sebagai jumlahan dari sinyal *external disturbance*, seperti gesekan dan semua sinyal yang muncul akibat dari perbedaan antara *plant actual* dengan model nominal, seperti ketidakpastian pemodelan dan variasi parameter [6].

Idealnya, DOB akan melemahkan *disturbance* pada frekuensi rendah dan membuat *plant* yang dikontrol oleh loop eksternal mendekati model nominal. Kemampuan ini dapat ditunjukkan dengan memperhatikan fungsi alih-fungsi alih berikut [7]:

$$G_{uy}(s) = \frac{P(s)P_n(s)}{P_n(s) + Q(s)(P(s) - P_n(s))} \quad (2)$$

$$G_{dy}(s) = \frac{P(s)P_n(s)(1 - Q(s))}{P_n(s) + Q(s)(P(s) - P_n(s))} \quad (3)$$

$$G_{ey}(s) = \frac{P(s)Q(s)}{P_n(s) + Q(s)(P(s) - P_n(s))} \quad (4)$$

Dari ketiga fungsi alih diatas dapat dilihat bahwa desain DOB untuk menentukan performa penolakan *disturbance* tergantung dari pemilihan filter *Q*. Perilaku ketiga fungsi alih sistem diatas ketika nilai filter *Q* mendekati 1 pada *low frequency* akan menunjukkan efek DOB pada sistem sebagai berikut [4]:

1. $G_{uy}(s) \approx P_n(s)$, hal ini berarti dinamika sistem dari *u* ke *y* akan menjadi sama seperti model nominal.
2. $G_{dy}(s) \approx 0$, hal ini menunjukkan bahwa pada frekuensi rendah *disturbance* yang ada akan diredam.
3. $G_{ey}(s) \approx P_n(s)$, hal ini menunjukkan bahwa sensor *noise* akan tetap diteruskan.

Untuk penentuan filter *Q(s)* dalam perancangan DOB digunakan filter *low pass*. Struktur filter *Q(s)* yang banyak digunakan dalam DOB adalah filter binomial dinyatakan pada persamaan 5.

$$Q(s) = \frac{1 + \sum_{k=1}^{N-r} a_k (\tau s)^k}{1 + \sum_{k=1}^N a_k (\tau s)^k} \quad (5)$$

dimana :

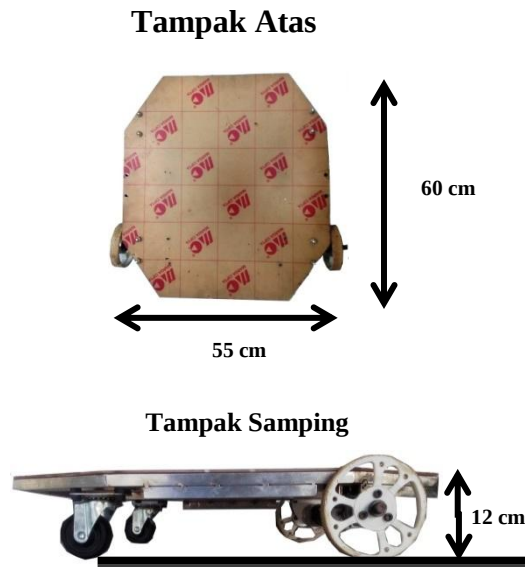
- N = orde dari $Q(s)$
- r = derajat relatif dari $Q(s)$
- $w_c = 1/\tau$ = frekuensi *cut-off* $Q(s)$
- τ = *Time Constant*

Dengan adanya blok *Disturbance Observer*, model *plant actual* secara tidak langsung dipaksa oleh *Disturbance Observer* untuk menjadi seperti model nominal *plant*. Sehingga apabila terdapat gangguan yang menyebabkan perubahan struktur model *plant*, tidak berpengaruh terhadap struktur dari kontrol karena *Disturbance Observer* akan menghasilkan nilai estimasi gangguan untuk mengurangi efek gangguan pada *plant*.

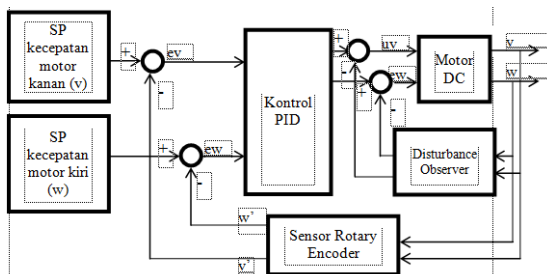
2.3. Perancangan Sistem

Secara garis besar, rancangan “Sistem Kontrol Kecepatan *Mobile Robot* Menggunakan PID dan Analisis *Disturbance* Berbasis *Disturbance Observer*” adalah seperti pada Gbr 3. Dimana ada dua motor yang

berfungsi sebagai penggerak robot secara diferensial dengan mekanik seperti yang ditunjukkan pada Gbr 4. Masing – masing motor dikontrol oleh PID secara terpisah karena setiap motor memiliki karakteristik yang berbeda. Dalam perancangan sistem ini seperti terdapat suatu mekanisme pengendalian yang berasal dari internal motor yang pada penelitian ini disebut sebagai *Disturbance Observer*.



Gambar 3. Mekanik Mobil Robot



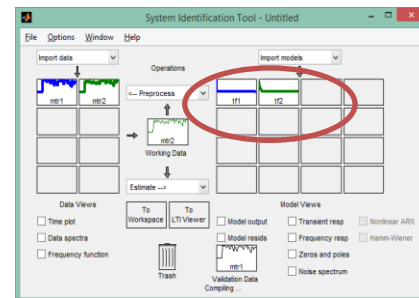
Gambar 4. Blok diagram rancangan umum sistem

Penjelasan dari blok diagram di atas adalah sebagai berikut: Dimulai dengan pembacaan data dari sensor. Sensor *rotaryencoder* akan mendeteksi kecepatan putaran motor. Pendeteksian sensor menghasilkan pulsa kotak yang akan diproses oleh mikrokontroler. Di dalam mikrokontroler ini akan dilakukan kontrol kecepatan oleh kontrol PID. Selain itu di dalamnya juga terdapat proses analisis

gangguan menggunakan *Disturbance Observer*. *Disturbance Observer* ini akan bekerja pada bagian *plant* atau motor yaitu dengan mengoreksi sinyal dari sensor dan sinyal *input* motor sehingga menghasilkan sinyal koreksi pada *input* motor. Sinyal *input* ini digunakan untuk pengaturan kecepatan motor.

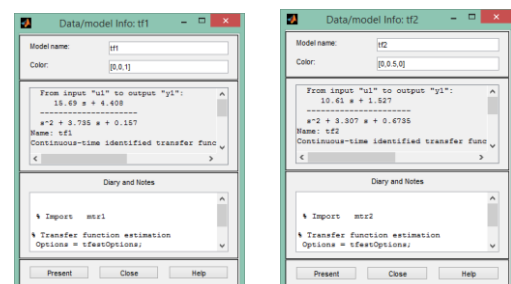
2.3.1. Pemodelan Motor Menggunakan MATLAB

Pada tahapan ini adalah melakukan identifikasi sistem motor yang dilakukan dengan menggunakan program Matlab melalui fungsi *ident*. Terlebih dahulu nilai kecepatan dan masukan tegangan disimpan kedalam suatu variabel. Estimasi yang akan dilakukan adalah mencari model fungsi alih dari data variabel yang dimasukkan pada *Data Views*.



Gambar 8. Grafik Hasil Estimasi Data

Untuk mengetahui persamaan matematika dari kedua model bisa dilakukan klik dua kali pada masing-masing grafik pada bagian model. Sehingga akan diperoleh persamaan matematika seperti pada Gambar 9.



Gambar 9. Fungsi Alih Motor Kanan dan Kiri

Setelah didapatkan model persamaan matematika nominal motor kanan (P_{n1}) dan

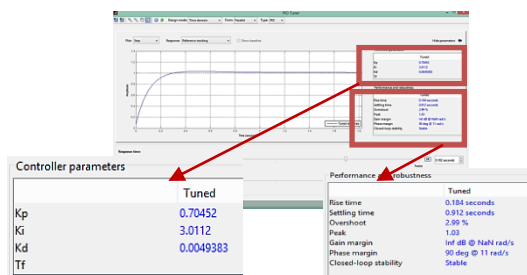
nominal motor kiri (P_{n2}) sebagai berikut :

$$Pn1(s) = \frac{15,69s+4.408}{s^2+3.735s+0.157} \quad (6)$$

$$Pn2(s) = \frac{10.61s+1.527}{s^2+3.307s+0.6735} \quad (7)$$

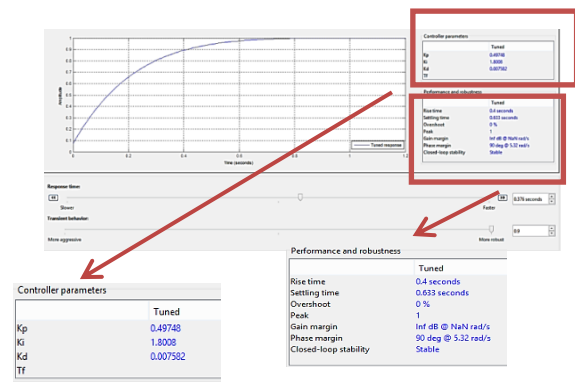
Dari persamaan 6 dan persamaan 7 selanjutnya akan digunakan untuk merancang pengendalian pada *loop* eksternal yang dilakukan oleh kontrol PID. Model matematik dari masing-masing motor diekspor ke *Workspace* Matlab. Untuk dilakukan proses *tunning* konstanta dari kontrol PID.

Pada penentuan konstanta kontrol PID, terdapat berbagai macam cara antara lain dengan metode *Direct Synthesis*, *Ziegler-Nichols*, *Cohen-Coen*. Selain beberapa metode tersebut terdapat cara lain untuk menentukan konstanta PID, yaitu dengan cara menggunakan fasilitas dari program Matlab yaitu fungsi *pidtool*. Dari fungsi tersebut dapat dilakukan *tuning* dan dapat secara langsung perubahan respon motor saat mengubah beberapa parameter respon motor. Proses *tuning* PID motor kanan dan kiri dilakukan seperti pada Gambar 10 dan Gambar 11.



Gambar 10. *Tuning* Konstanta PID Motor Kanan dengan *PIDTOOL*

Hasil *tuning* konstanta PID motor kanan dengan *PIDTOOL* didapatkan nilai konstanta - konstanta PID $K_p = 0.70452$, $K_i = 3.0112$, $K_d = 0.0049$. Hasil *tuning* ini didapatkan respon motor yang memiliki *rise time* = 184 ms, *settling time* = 915 ms, presentase *overshoot* = 2.99%, dan kesalahan kondisi *steady* = 0.03s.



Gambar 11. *Tuning* Konstanta PID Motor Kiri dengan *PIDTOOL*

Hasil *tuning* konstanta PID motor kiri dengan *PIDTOOL* didapatkan nilai konstanta - konstanta PID $K_p = 0.4974$, $K_i = 1.8008$, $K_d = 0.0075$. Hasil *tuning* ini didapatkan respon motor yang memiliki *rise time* = 400 ms, *settling time* = 633 ms, presentase *overshoot* = 0 %, dan kesalahan kondisi *steady* = 0. Konstanta - konstanta yang didapatkan dari proses *tuning* selanjutnya digunakan sebagai parameter perancangan kontrol PID. Persamaan 8 dan persamaan 9 merupakan persamaan dalam kawasan waktu kontinyu.

$$C1(s) = \frac{0.005s^2+0.7045s+3.0112}{s} \quad (8)$$

$$C2(s) = \frac{0.0076s^2+0.4974s+1.8008}{s} \quad (9)$$

Untuk dapat merealisasikan ke dalam bentuk pemrograman, maka persamaan waktu kontinyu tersebut harus diubah terlebih dahulu ke dalam bentuk diskrit sehingga dapat di implementasikan pada perangkat digital seperti mikrokontroler atau komputer untuk proses komputasi.

Proses diskritisasi kontrol proporsional adalah dapat ditunjukkan pada persamaan 10.

$$u(k) = K_p \times e(k) \quad (10)$$

Dimana :

$U(k)$ = keluaran kontrol proporsional waktu sekarang

K_p = konstanta proporsional

$E(k)$ = *error* waktu sekarang

Proses diskritisasi kontrol integral ditunjukkan pada persamaan 11 sampai persamaan 13.

$$u(k) = Ki \sum_{i=0}^k e(i) \times Tc \quad (11)$$

$$u(k) = KiTc \sum_{i=0}^k e(i) = KiTc[e(0) + e(1) + \dots + e(k-1) + e(k)] \quad (12)$$

$$u(k) = KiTc[e(k-1) + e(k)] \quad (13)$$

Dimana :

$U(k)$ = keluaran kontrol integral waktu sekarang

Ki = konstanta integral

Tc = waktu sampel

$E(k)$ = *error* waktu sekarang

$E(k-1)$ = *error* satu detik sebelum waktu sekarang

Proses diskritisasi kontrol derivatif ditunjukkan pada persamaan 14.

$$u(k) = Kd \frac{e(k) - e(k-1)}{Tc} \quad (14)$$

Dimana :

$U(k)$ = keluaran kontrol derivatif waktu sekarang

Kd = konstanta derivatif

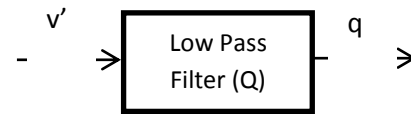
Tc = waktu sampel

$E(k)$ = *error* waktu sekarang

$E(k-1)$ = *error* satu detik sebelum waktu sekarang

2.3.3. Low Pass Filter pada Perancangan DOB

Dalam perancangan sistem kendali dengan *Disturbance Observer* salah satu komponen penting adalah *low pass filter*. Filter ini memiliki 2 fungsi yang berbeda, antara lain filter untuk keluaran kontrol PID yang berfungsi untuk mencegah terjadinya *aljabar loop* yang dapat mengakibatkan kesalahan sistem, sedangkan filter lainnya digunakan untuk keluaran sensor motor yang berfungsi untuk melemahkan sinyal gangguan dari pembacaan sensor motor.



Gambar 12. Blok Diagram Filter *Low Pass*

Dari gambar blok diagram di atas v' adalah masukan filter *low pass* dan q adalah keluaran filter *low pass*. Pada penelitian ini filter yang dirancang memiliki frekuensi *cut-off*(f_c) sebesar 5 Hz dengan frekuensi *sampling*(f_s) sebesar 25 Hz. Ordo filter yang digunakan adalah filter berorde 2 karena sebaiknya minimal berorde sama dengan *plant nominal* yang berorde 2. Sehingga didapatkan persamaan filter sebagai berikut :

$$\tau = \frac{1}{2\pi f_c} \quad (15)$$

$$\tau = \frac{1}{31.4} = 0.03 \quad (16)$$

$$Q(s) = \frac{1}{1 + 2\tau s + (\tau s)^2} \quad (17)$$

$$Q(s) = \frac{1}{1 + 0.06s + 0.0009s^2} \quad (18)$$

$$Q(s) = \frac{10000}{10000 + 600s + 9s^2} \quad (19)$$

Persamaan 19 harus terlebih dahulu ditransformasikan ke dalam domain- z menggunakan Matlab yang mentransformasikan persamaan dalam waktu kontinyu ke persamaan waktu diskrit. Proses transformasi dapat dilihat pada Gambar 13.

```

Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

>> num=[10000];
>> den=[9 600 10000];
>> sys=tf(num,den);
>> syz_d=c2d(sys,0.04,'tustin')

syz_d =

    0.16 z^2 + 0.32 z + 0.16
    -----
    z^2 - 0.4 z + 0.04

Sample time: 0.04 seconds
Discrete-time transfer function.

```

Gambar 13. Proses Transformasi-Z Filter *Low Pass*

Hasil transformasi- z persamaan 19 dengan menggunakan Matlab diperoleh persamaan 20 sebagai berikut.

$$Q(z) = \frac{0.16z^2 + 0.32z + 0.16}{z^2 - 0.4z + 0.04} \quad (20)$$

Untuk diubah kedalam bentuk persamaan beda, masing-masing pembilang dan penyebut pada persamaan 20 dibagi dengan (z^2) sehingga didapatkan persamaan 21.

$$Q(z) = \frac{0.16 + 0.32z^{-1} + 0.16z^{-2}}{1 - 0.4z^{-1} + 0.04z^{-2}} \quad (21)$$

Berdasarkan hubungan masukan dan keluaran dari blok filter *low pass* adalah $q=v' \times Q$.

$$\frac{q(z)}{v'(z)} = \frac{0.16 + 0.32z^{-1} + 0.16z^{-2}}{1 - 0.4z^{-1} + 0.04z^{-2}} \quad (22)$$

Dari persamaan 22 dilakukan perkalian silang dan didapatkan persamaan 23.

$$q(z) = 0.16v'(z) + 0.32v'(z)z^{-1} + 0.16v'(z)z^{-2} + 0.4q(z)z^{-1} - 0.04q(z)z^{-2} \quad (23)$$

Maka persamaan beda dari persamaan 23 adalah sebagai berikut.

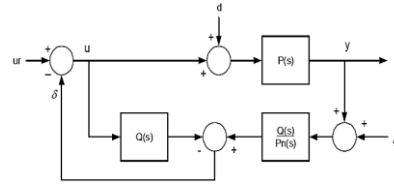
$$q(k) = 0.16v'(k) + 0.32v'(k-1) + 0.16v'(k-2) + 0.4q(k-1) - 0.04q(k-2) \quad (24)$$

Dimana :

$q(k)$ = keluaran filter
 $v'(k)$ = masukan filter sekarang
 $v'(k-1)$ = masukan filter satu detik sebelum
 $v'(k-2)$ = masukan filter dua detik sebelum
 $q(k-1)$ = keluaran filter satu detik sebelum
 $q(k-2)$ = keluaran filter dua detik sebelum

Sebagai analisis *disturbance* epada robot digunakan *Disturbance Observer* yang melakukan koreksi sinyal terhadap gangguan. DOB bekerja pada loop internal yang melakukan koreksi pada bagian *plant*. DOB memproses sinyal dari sensor *rotary encoder* dan sinyal keluaran kontrol PID. Hasil pengolahan kedua sinyal tersebut selanjutnya

diakumulasi sehingga didapatkan sinyal baru yaitu estimasi gangguan. Blok diagram *Disturbance Observer* ditunjukkan pada Gbr 14.



Gambar 14. Struktur Dasar *Disturbance Observer*

Sinyal u , d , ξ dan y berturut-turut adalah sinyal kontrol, gangguan, *noise* sensor, dan sinyal keluaran motor. Sinyal δ merupakan sinyal hasil estimasi. Sinyal ur adalah sinyal kontrol dari kontrol eksternal. $Q(s)$ adalah filter *low pass* dan $Pn(s)$ adalah model nominal *plant*. *Disturbance Observer* akan melemahkan sinyal gangguan pada motor yang berfrekuensi rendah sehingga membuat *plant* aktual motor yang dikontrol oleh kontrol eksternal mendekati model nominal.

$$\frac{Q(s)}{Pn1(s)} = \frac{10000s^2 + 37350s + 1570}{141.21s^3 + 9453.672s^2 + 159544.8s + 44080} \quad (25)$$

$$\frac{Q(s)}{Pn1(s)} = \frac{10000s^2 + 33070s + 6735}{95.49s^3 + 6379.743s^2 + 107016.2s + 15270} \quad (26)$$

Persamaan 25 dan 26 digunakan sebagai perancangan kontrol internal motor kanan dan kiri. Kedua persamaan tersebut harus ditransformasikan kedalam domain z dengan menggunakan Matlab.

```

Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
>> num=[10000 37350 1570];
>> den=[141.21 9453.672 159544.8 44080];
>> Qn1=tf(num,den);
>> Qn1d=c2d(Qn1,0.04,'tustin')

Qn1d =

    0.5449 z^3 - 0.4691 z^2 - 0.5448 z + 0.4692
-----
    z^3 - 1.389 z^2 + 0.4355 z - 0.03955

Sample time: 0.04 seconds
Discrete-time transfer function.
  
```

(a)

```

Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
>> num=[10000 33070 6735];
>> den=[95.49 6379.743 107016.2 15270];
>> Qn2=tf(num,den);
>> Qn2d=c2d(Qn2,0.04,'tustin')

Qn2d =

    0.8018 z^3 - 0.7015 z^2 - 0.801 z + 0.7023
-----
    z^3 - 1.394 z^2 + 0.4377 z - 0.03977

Sample time: 0.04 seconds
Discrete-time transfer function.
  
```

(b)

Gambar 15. (a) Diskritisasi Q_{Pn}^{-1} Motor Kanan (b) Diskritisasi Q_{Pn}^{-1} Motor Kiri

Sesuai dengan transformasi (Q/Pn) motor kanan dan motor kiri yang didapatkan dari Matlab didapatkan transformasi domain z sebagai berikut:

$$\frac{y'(z)}{y(z)} = \frac{0.5449z^3 - 0.4691z^2 - 0.5448z + 0.4692}{z^3 - 1.389z^2 + 0.4355z + 0.03955} \quad (27)$$

$$\frac{y'(z)}{y(z)} = \frac{0.8018z^3 - 0.7015z^2 - 0.801z + 0.7023}{z^3 - 1.394z^2 + 0.4377z - 0.0397} \quad (28)$$

Selanjutnya kedua persamaan di atas masing-masing dibagi dengan z^{-2} . Sehingga diperoleh persamaan berikut.

$$\frac{y'(z)}{y(z)} = \frac{0.5449 - 0.4691z^{-1} - 0.5448z^{-2} + 0.4692z^{-3}}{1 - 1.389z^{-1} + 0.4355z^{-2} - 0.03955} \quad (29)$$

$$\frac{y'(z)}{y(z)} = \frac{0.8018 - 0.7015z^{-1} - 0.801z^{-2} + 0.7023z^{-3}}{1 - 1.394z^{-1} + 0.4377z^{-2} - 0.0397z^{-3}} \quad (30)$$

Masing-masing persamaan dikalikan silang, sehingga diperoleh persamaan 31 dan 32.

$$y'(z) = 0.5445y(z) - 0.4691y(z)z^{-1} - 0.5448y(z)z^{-2} + 0.4692y(z)z^{-3} + 1.389y'(z)z^{-1} - 0.4355y'(z)z^{-2} + 0.03955y'(z)z^{-3} \quad (31)$$

$$y'(z) = 0.8018y(z) - 0.7015y(z)z^{-1} - 0.801y(z)z^{-2} + 0.7023y(z)z^{-3} + 1.394y'(z)z^{-1} - 0.4377y'(z)z^{-2} + 0.0397y'(z)z^{-3} \quad (32)$$

Persamaan dalam domain-z dari persamaan 31 dan 32 diubah ke dalam bentuk persamaan beda agar dapat diimplementasikan pada pemrograman mikrokontroler untuk motor kanan dan kiri.

$$y'(k) = 0.5445y(k) - 0.4691y(k-1) - 0.5448y(k-2) + 0.4692y(k-3) + 1.389y'(k-1) - 0.4355y'(k-2) + 0.03955y'(k-3) \quad (33)$$

$$y'(k) = 0.8018y(k) - 0.7015y(k-1) - 0.801y(k-2) + 0.7023y(k-3) + 1.394y'(k-1) - 0.4377y'(k-2) + 0.0397y'(k-3) \quad (34)$$

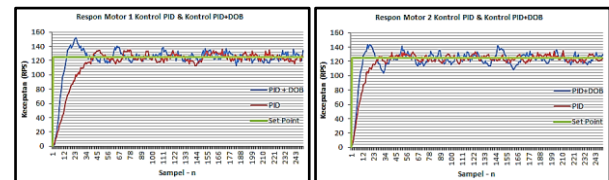
Hasil kedua proses di atas selanjutnya diakumulasi sehingga didapatkan nilai estimasi gangguan. Nilai estimasi gangguan ini selanjutnya diakumulasikan dengan keluaran kontrol PID yang kemudian hasilnya digunakan untuk membangkitkan sinyal PWM untuk masing-masing motor.

3. Hasil Penelitian

3.1. Pengujian Dengan Beban Berbeda

Dari uji coba baik menggunakan kontrol PID maupun kontrol PID berbasis DOB dengan pemberian beban pada robot sebesar 12 Kg. Perbandingan kedua kontrol antara kontrol PID dan Kontrol PID berbasis DOB terhadap

nilai acuan ditunjukkan pada Gambar 16. Dari perbandingan respon yang dihasilkan respon motor kanan dengan kontrol PID dan pembebanan seberat 12 Kg memiliki nilai RMSE sebesar 24,65rps. Sedangkan respon motor kanan dengan kontrol PID berbasis DOB memiliki nilai RMSE sebesar 19,39rps. Kemudian respon motor kiri dengan kontrol PID dan pembebanan seberat 12 Kg memiliki nilai RMSE sebesar 18,74 rps. Sedangkan respon motor kiri dengan kontrol PID berbasis DOB memiliki nilai RMSE sebesar 16,92 rps.



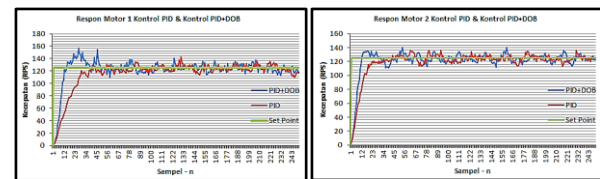
(a)

(b)

Gambar 16. a) Respon Motor Kanan Kontrol PID & PID+DOB

b) Respon Motor Kanan Kontrol PID & PID+DOB

Pengujian selanjutnya dengan beban 36 Kg didapatkan hasil perbandingan dari kedua kontrol antara kontrol PID dan Kontrol PID berbasis DOB terhadap nilai acuan ditunjukkan pada Gambar 17.



(a)

(b)

Gambar 17. a) Respon Motor Kanan Kontrol PID & PID+DOB

b) Respon Motor Kiri Kontrol PID & PID+DOB

Dari perbandingan respon yang dihasilkan respon motor kanan dengan kontrol PID dan pembebanan seberat 36 Kg memiliki nilai RMSE sebesar 24,57. Untuk PID berbasis DOB memiliki nilai RMSE sebesar 19,46 rps. Kemudian respon motor kiri dengan kontrol PID dan pembebanan seberat 36 Kg memiliki

nilai RMSE sebesar 19,15rps. Untuk PID berbasis DOB dan pembebanan 36 Kg memiliki nilai RMSE sebesar 16,51rps.

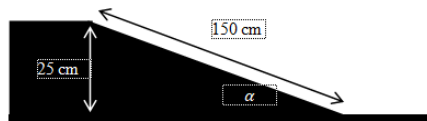
Untuk lebih jelasnya rangkuman data perbandingan kedua kontrol di atas dapat dilihat pada Tabel 2.

Tabel 2. Perbandingan kontrol PID dengan PID DOB

Gangguan	Motor	RMSE (rps)	
		PID	PID + DOB
Beban 12 kg	Kanan	24.65	19.39
	Kiri	18.74	16.92
Beban 36 kg	Kanan	24.57	19.46
	Kiri	19.15	16.51

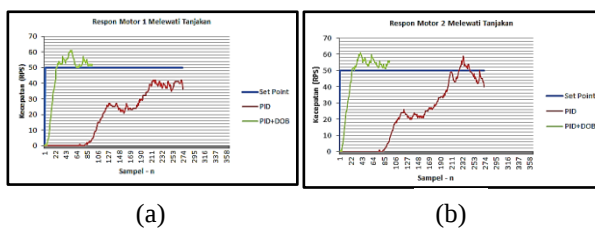
3.2. Pengujian Respon Motor Melewati Tanjakan

Pada uji coba kali ini robot dijalankan dengan melewati rintangan berupa tanjakan. Spesifikasi tanjakan yang akan dilewati adalah sisi miring segitiga dengan panjang lintasan 150 cm, tinggi segitiga 25 cm, dan sudut kemiringan lintasan sebesar $9,6^\circ$. Pengujian dilakukan dengan menganalisa perilaku motor saat robot melewati patahan lintasan.



Gambar 18. Lintasan Tanjakan Pengujian Robot

Perbandingan respon motor kanan baik menggunakan kontrol PID dan kontrol PID dengan DOB terhadap *set point* kecepatan 50 RPS dapat ditunjukkan pada Gambar 19 dengan waktu sampling 40 ms.



(a)

(b)

Gambar 21. a) Respon Motor Kanan Melewati Tanjakan

b) Respon Motor Kiri Melewati Tanjakan

Respon motor saat melintasi tanjakan memiliki nilai RMSE untuk kontrol PID sebesar 1177 rps. Saat robot mulai *start* respon motor tidak langsung naik, namun membutuhkan waktu yang cukup lama, motor mulai berjalan pada saat sampel ke-81 x 40 ms = 3240 ms. Pada saat melewati patahan lintasan, respon motor tidak bisa meningkatkan kecepatannya untuk mencapai nilai *set point*. Sedangkan nilai RMSE untuk kontrol PID yang digabungkan dengan DOB sebesar 313 rps. Motor dapat langsung bergerak pada sampel ke-3 x 40 = 120 ms. Saat melewati patahan lintasan kecepatan motor masih dapat meningkat sampai nilai *set point*. Selanjutnya pengujian pada respon motor kiri yaitu saat melintasi tanjakan memiliki nilai RMSE untuk kontrol PID sebesar 1126 rps. Saat robot mulai *start* respon motor tidak langsung naik, namun membutuhkan waktu yang cukup lama, motor mulai berjalan pada saat sampel ke-83 x 40 ms = 3320 ms. Pada saat melewati patahan lintasan, kecepatan motor mulai melambat dan perlahan meningkatkan kecepatannya menuju nilai *set point*. Sedangkan nilai RMSE untuk kontrol PID yang digabungkan dengan DOB sebesar 292 rps. Motor dapat langsung bergerak pada sampel ke-3 x 40 = 120 ms. Saat melewati patahan lintasan kecepatan motor masih dapat meningkat sampai nilai *set point*.

Detail perbandingan dari kedua kontrol yang digunakan pada masing-masing motor dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan kontrol PID dengan PID DOB melewati tanjakan

Motor	RMSE (rps)	
	PID	PID + DOB
Kanan	1177	313
Kiri	1126	292

Berdasarkan hasil yang didapatkan dapat diketahui pada kondisi ini dengan menggunakan kontrol PID+DOB nilai RMSE dari kedua respon motor memiliki nilai yang sangat kecil sekali yaitu 1/3 dari nilai RMSE PID kontrol tanpa DOB.

4. Kesimpulan

Dengan bertambahnya beban yang membebani motor, maka nilai keluaran kontrol PID akan bertambah besar, hal ini terjadi karena saat pembebanan, seharusnya motor mengalami penurunan kecepatan. Akibat dari penurunan kecepatan ini adalah semakin besarnya *error* antara nilai acuan kecepatan dan nilai kecepatan motor aktual. Sehingga dengan adanya kontrol *Disturbance Observer* yang menghasilkan nilai estimasi gangguan akibat pembebanan pada robot, nilai keluaran PID akan turun hingga mendekati 0. Sehingga kontrol PID seolah-olah tidak berperan besar terhadap penjagaan performa motor. Hal ini disebabkan adanya penambahan nilai estimasi gangguan yang dihasilkan *Disturbance Observer*. Berdasarkan beberapa pengujian yang dilakukan didapatkan rata-rata nilai *RMSE* kontrol PID maupun PID berbasis DOB untuk motor 1 masing-masing 24,47 dan 19,59 dan untuk motor 2 masing-masing 18,89 dan 16,66. Sehingga pemberian kontrol pada loop internal dengan *Disturbance Observer* dapat meningkatkan dan menjaga performa motor terhadap ketidak pastian gangguan yang diterima oleh robot terutama pada motor sebagai penggeraknya.

5. Saran

Dari sistem yang telah dibuat disarankan menggunakan sensor *rotaryencoder* dengan resolusi yang lebih besar untuk mendapatkan nilai putaran dan kecepatan yang lebih akurat.

6. Daftar Pustaka

- [1] Bayu Sandi Marta, "Path Tracking Pada Mobile Robot Dengan Umpan Balik Odometry," Politeknik Elektronika Negeri Surabaya, Surabaya, B.Sc Thesis 2011.
- [2] Nita Rachmadyanti, "Kontrol PID Untuk Pengaturan Kecepatan Motor Pada Prototype Ayunan Bayi Otomatis," Politeknik Elektronika Negeri Surabaya, Surabaya, B.Sc Thesis 2011.
- [3] Muhammad Ibnu Bahrurrahim, "Rancang Bangun Sistem Kontrol dan Analisa Pengendalian Robot Manual Pada Kasus KRI," Politeknik Elektronika Negeri Surabaya, Surabaya, B.Sc Thesis 2007.
- [4] Firdaus, "Rancang Bangun Sistem Kendali Kecepatan Kursi Roda Listrik Berbasis *Disturbance Observer*," Institut Teknologi Sepuluh Nopember Surabaya, Surabaya, B.Sc Thesis 2011.
- [5] Utis Zaenurrohman, "Perancangan Sistem Kontrol Wireless Pada Mobile Robot Manipulator Berbasis Mikrokontroler AT Mega 8," *JNTETI Universitas Gadjah Mada*, vol. 3, no. 1, pp. 69-75, 2014.
- [6] Adelina Kartini, "Sistem Pengaturan Gerakan Tool Pada Prototipe Mesin CNC dengan Kontroler *Disturbance Observer*," *Jurnal Teknik POMITS*, vol. I, no. 1, pp. 1-6, Juni 2012.
- [7] Pramudijanto, Yonatan Ahmad Fatoni, "Perancangan Dan Implementasi *Disturbance Observer* Untuk Pengaturan Kontur Pada Simulator Mesin Freis," *Jurnal Teknik Elektro-FTI*, vol. I, no. 1, pp. 1-7, MAret 2005.