

PERANCANGAN DAN IMPLEMENTASI KONTROL KECEPATAN MOTOR DC MENGGUNAKAN LABVIEW UNTUK KESTABILAN LAJU PADA ROBOT TANK DENGAN METODE PID

DESIGN AND IMPELEMNTATION OF DC MOTOR SPEED CONTROL USING LABVIEW FOR STABILITY RATE ON TANK ROBOT WITH PID

Agung Adiprasetya¹, Erwin Susanto.,ST.,MT.,PhD², Agung Surya Wibowo.,ST.,MT³

^{1,2,3}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹agungadiprasetya@gmail.com, ²erwinelektro@telkomuniversity.ac.id, ³agungsw@telkomuniversity.ac.id

Abstrak

Dunia robotika semakin berkembang pesat, terutama pada robot surveillanc. Jenis robot yang biasanya digunakan untuk robot surveillance adalah robot tank karena robot jenis ini dikenal memiliki kemampuan untuk melewati medan yang cukup berat. Namun, kebanyakan orang hanya memanfaatkan torsi motor saja untuk kekuatannya tanpa memperhatikan faktor yang lain. Maka dari itu, dirancanglah sebuah robot tank yang dilengkapi dengan kontrol PID. Kontrol PID ini diharapkan dapat memberikan kestabilan kecepatan rotasi motor dc pada robot, sehingga robot dapat bekerja secara maksimal.

Tujuan dalam perancangan ini adalah bagaimana cara merancang sistem PID untuk robot, sehingga robot dapat bergerak dengan kecepatan yang stabil dalam melewati jalan yang mendatar, menanjak, dan menurun. Selain itu, sistem mekanik dan elektronik pada robot juga sangat diperhatikan untuk menunjang kinerja dari robot..

Kontrol PID ini ditanamkan pada LabView. Hasil dari perhitungan PID dikirim ke mikorkontroler ATmega 328P secara wireless. Nilai tersebut merupakan nilai PWM yang berguna untuk menggerakkan motor DC. Berdasarkan pengujian secara open loop, hasil pembacaan dari sensor masih kurang baik. Terdapat gangguan antara 1 RPM hingga 5 RPM. Namun, hasil dari pengujian close loop, pada medan naik dan turun, dari percobaan pada tujuh sudut (5°-35°), hanya enam sudut (85,71 %) yang berhasil. Satu sudut yaitu sudut 35° robot gagal melewatinya, baik naik maupun turun.

Kata kunci : PID, robot tank, ATmega 328P, LabView

Abstract

The world of robotics is growing rapidly. Especially on the surveillance robot. The type of robot used for surveillance robot is tank robot because this robot have the ability to pass through terrain that is quite heavy. However, most people only use the motor torque only for its strenght regardless of other factors. Therefore, I try to designed a tank robot equipped with PID control. PID control is expected to provide a stable speed rotation of the dc motor on the robot, so the robot can work optimally.

The purpose of this design is how to designed a PID system for the robot, so the robot can move at a steady pace as it passes through the uphill road, flat, and downhill road. In addition, mechanical and electronics systems on the robot is also very concerned to support the performance of the robot.

PID control is embedded in LabView. The result of the calculation is sent to the microcontroller ATmega 328P wirelessly. This value is the PWM value that useful to drive the dc motor. Based on open loop experiment, the readings from sensor is still not good. There are some error between 1 RPM until 5 RPM. However, the result of experiment in close loop system, on the up and down terrain, from experiments on seven angle (5°-35°), only six angle can be managed. One angle (35°), the robot was fail to pass the up and down terrain.

Keywords : PID, robot tank, ATmega 328P, LabView

1. Pendahuluan

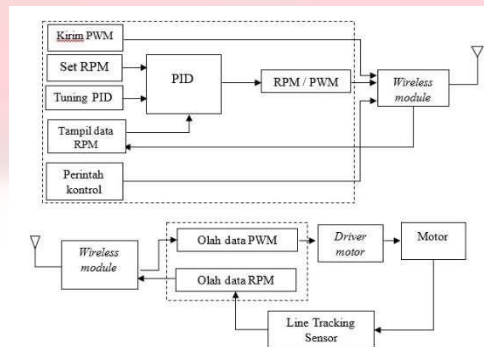
Dewasa ini dunia robotika semakin berkembang pesat, terutama robot surveillance. Robot ini banyak digunakan untuk memudahkan manusia dalam menanggungan bencana alam. Robot berjenis ini banyak menggunakan robot berjenis robot tank. Dengan roda yang dapat mencangkram tanah dan tidak mudah tergelincir, robot ini sangat diandalkan kemampuannya dalam melewati segala medan. Selain itu, robot tank juga memiliki torsi motor yang besar untuk melewati medan yang berat.

Namun, hanya memanfaatkan torsi motor saja tidak cukup tanpa memerhatikan faktor lain. Maka dari itu, penulis merancang sistem PID untuk robot tank. Sistem PID ini ditanamkan pada suatu user interface yang dibuat menggunakan software LabVIEW. Perangkat LabVIEW ini tidak hanya sebagai user interface saja, tetapi juga sebagai pusat kendali PID.

Kendali PID ini akan memanfaatkan data RPM yang didapat dari kecepatan roda pada robot. Data RPM dari robot dikirim secara wireless menggunakan RF Modul ke PC. Di PC, penulis membuat user interface dengan menggunakan software LabVIEW. LabVIEW merupakan suatu software dari National Instrument yang didesain untuk melakukan simulasi kontrol atau membuat user interface untuk pengontrolan dan monitoring. Pada user interface ini, data RPM diolah pada blok PID untuk mengatur kestabilan laju robot. Nilai keluaran dari hasil perhitungan berupa nilai RPM. Nilai RPM ini dikalikan dengan faktor pengali yang kemudian dikirim kembali ke robot secara wireless.

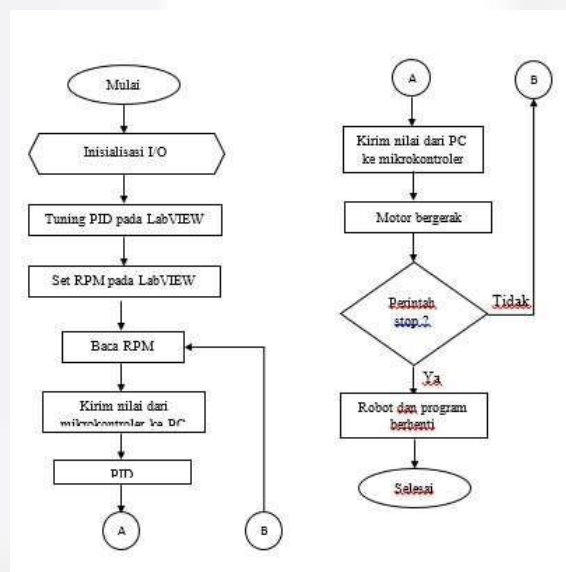
2. Dasar Teori dan Perancangan

Berikut ini merupakan rancangan sistem yang dibangun.



Gambar 2.1 Gambaran Umum Sistem

Gambar 2.1 merupakan rancangan sistem yang dibuat. Dimulai dari pembacaan nilai RPM, pengiriman data ke PC, pengolahan data RPM menggunakan PID, pengonversian dari data RPM menjadi PWM, pengiriman data dari PC ke robot, hingga menggerakkan motor pada robot berdasarkan nilai yang diterima.



Pada sistem yang dirancang, kontrol PID ditanamkan pada user interface yang dibangun menggunakan LabVIEW, sehingga user interface yang digunakan bukan hanya sebagai alat monitoring saja tetapi juga sebagai pengendali kecepatan motor pada robot. Kontrol PID pada LabVIEW ini membaca nilai RPM yang dikirim dari mikrokontroler. Data RPM yang diterima lalu masuk ke blok PID. PID yang digunakan adalah PID yang sudah ada pada LabVIEW untuk mempermudah proses perancangan. Setelah diproses pada blok PID, nilai keluaran dikalikan dengan faktor pengali agar mendapatkan nilai PWM. Nilai PWM inilah yang dikirim ke mikrokontroler. Mikrokontroler yang digunakan adalah ATMEga 328P. Nilai PWM yang diterima mikrokontroler diteruskan ke driver motor 4 Ampere "H-Bridge". Pada driver motor ini, nilai PWM diubah menjadi tegangan keluaran yang digunakan untuk menggerakkan motor pada robot.

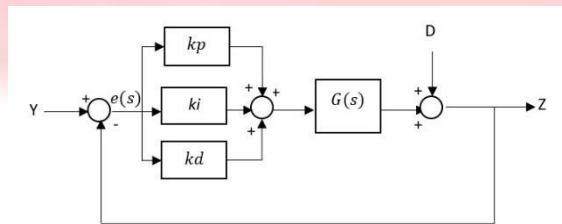
2.1 Pembacaan Nilai RPM

Untuk melakukan pembacaan nilai RPM, penulis menggunakan line tracking sensor. Sensor ini pada dasarnya membaca warna hitam dan putih dan akan bernilai satu apabila membaca putih dan bernilai nol apabila membaca hitam. Pada perancangan ini, sensor akan membaca perpindahan warna dari hitam ke putih maupun sebaliknya. Perpindahan itulah yang akan dihitung oleh mikrokontroler sehingga menghasilkan nilai RPM.

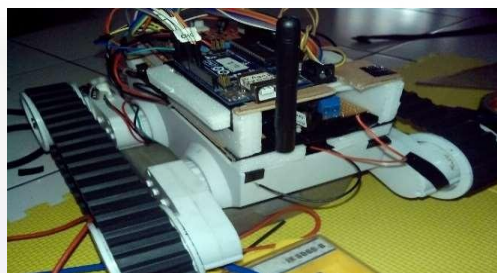
2.2 Kontrol Kecepatan Motor dengan Kontrol PID

Pada tahap ini, kecepatan motor akan distabilkan oleh sistem kontrol PID. Kontrol PID merupakan suatu tipe kontrol yang paling klasik. Kontrol PID terdiri atas kontrol proporsional, integral, dan derivatif dengan masing-masing kontrol memiliki kelebihan dan kekurangan yang berbeda. Kontrol PID ini berpengaruh untuk menutupi semua kekurangan yang dimiliki masing-masing kontrol.

Kontrol PID hanya dapat bekerja pada sistem yang cenderung linier terhadap waktu. Artinya kondisi model tidak boleh berubah dalam rentang waktu pengontrolan, walaupun dalam kenyataannya kondisi yang benar-benar linier tidak ada. Sehingga kontrol PID merupakan kontrol yang paling tepat untuk kondisi model yang ideal dengan mengabaikan kondisi-kondisi non-linier yang dianggap sulit untuk dimodelkan secara matematis.



Gambar 2.3 Diagram Blok Sistem



Gambar 2.4 Perancangan robot rover 5 yang digunakan

2.2.1 Kontrol P (Proportional)

Kontrol proportional merupakan penguatan murni K_p . Persamaan output kontroler adalah sebagai berikut :

$$u_p = K_p e(t) \quad \text{.....(1)}$$

Pertambahan nilai K akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar respon dan mengurangi error steady state. Nilai keluaran bergantung perkalian error, yakni kecepatan referensi dikurangi aktual, dengan konstanta K_p . Jika error positif, maka kecepatan aktual lebih besar daripada referensi. Sebaliknya, jika error negatif, maka kecepatan aktual lebih kecil daripada referensi.

2.2.2 Kontrol I (Integral)

Persamaan output kontroler adalah sebagai berikut :

$$u_i = K_i \int e(t) dt \quad \text{.....(2)}$$

dengan K_i adalah konstanta integral. Jika error mendekati konstanta (bukan nol), maka output akan menjadi sangat besar, sehingga diharapkan dapat memperbaiki error. Jika error mendekati nol, maka efek kontrol integral akan semakin kecil.

2.2.3 Kontrol D (Derivative)

Sinyal kontrol u_d yang dihasilkan oleh kontrol D dapat dinyatakan sebagai berikut :

$$u_d = K_d \frac{de(t)}{dt} \quad \text{.....(3)}$$

Dari persamaan tersebut dapat dilihat bahwa sifat dari kontrol ini berlain dalam konteks “kecepatan” atau rate dari error. Dengan sifat ini dapat digunakan untuk memperbaiki error yang akan terjadi. Umpan

balik yang diberikan adalah sebanding dengan kecepatan perubahan error, sehingga kontroler dapat mengantisipasi error yang akan terjadi.

2.3 Komunikasi Antar Perangkat

Untuk pengiriman data dari robot ke PC maupun PC ke robot, penulis menggunakan RF modul sebagai perangkatnya. Alat ini menggunakan sistem komunikasi serial. RF Modul yang digunakan merupakan RF Modul 3DR yang biasa digunakan pada pesawat nirawak karena kemampuannya dalam menerima dan mengirim data dengan range yang jauh. Itu pun bergantung pada antena yang digunakan.

Frekuensi yang digunakan alat ini adalah 433 MHz. Frekuensi ini merupakan frekuensi gratis yang diperbolehkan oleh pemerintah Indonesia untuk kegiatan radio amatir, riset, dll.

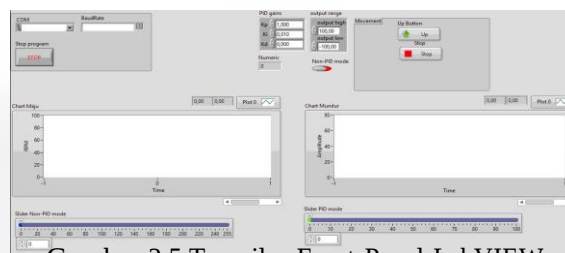
2.4 Perancangan Mekanik

Chasis robot yang digunakan pada percobaan ini adalah chasis robot tank rover 5. Chasis ini memiliki 4 buah motor dengan masing-masing motor memiliki torsi 10 kg/cm.. Selain itu, tegangan yang dibutuhkan oleh motor ini sebesar 7,2 V dengan arus maksimum 2,5 A.

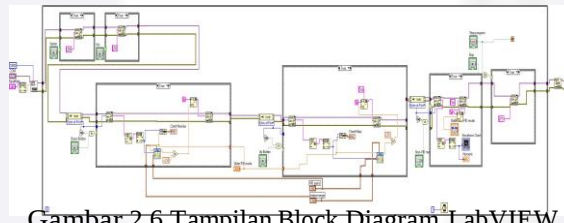
2.5 Perancangan Software LabVIEW

Aplikasi yang dibuat menggunakan LabVIEW adalah untuk membuat aplikasi user interface. Aplikasi yang dibuat ini digunakan untuk mengontrol gerakan robot, mengetahui kecepatan motor DC pada robot, dan untuk sistem pengontrolan PID.

PID yang digunakan pada program LabVIEW ini merupakan library PID yang telah disediakan oleh LabVIEW. Di dalam library ini semua perhitungan PID sudah terdapat didalamnya. Penulis hanya perlu memasukkan nilai setpoint, data masukan dari mikrokontroler, dan memasukkan nilai Kp, Ki, dan Kd.



Gambar 2.5 Tampilan Front Panel LabVIEW



Gambar 2.6 Tampilan Block Diagram LabVIEW

3. Uji Coba, dan Analisa

3.1 Pengujian Aplikasi

Pengujian ini dilakukan untuk mengetahui fungsionalitas dari aplikasi yang telah dibuat. Pengujian ini dilakukan dengan cara mencoba tiap – tiap item secara berulang kali.

Tabel 1. Pengujian Aplikasi

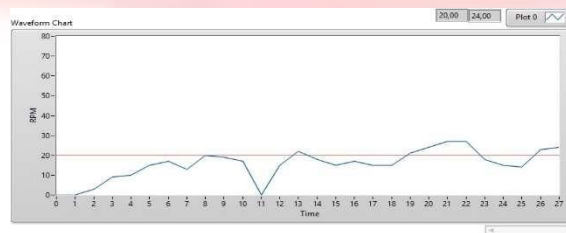
Item	Banyak Pengujian	Berhasil	Gagal	Hasil
Memulai aplikasi dan komunikasi	30	30	0	Berhasil
Slider PID	30	30	0	Berhasil
Slider Non-PID	30	30	0	Berhasil
Input PID Gains	30	30	0	Berhasil
Tombol Stop	30	30	0	Berhasil
Tombol Non-PID Mode	30	30	0	Berhasil
Tombol Maju	30	30	0	Berhasil

3.1 Pengujian Robot pada Jalan Menanjak

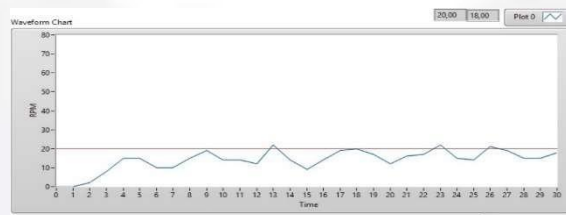
Pengujian ini dilakukan dengan menjalankan robot pada medan menanjak dengan kemiringan tertentu. Pada pengujian ini, kemiringan medan yang digunakan setiap perubahan 5° dengan sudut maksimal 35° . Nilai $k_p = 0,5$, $k_i = 0,03$, $k_d = 0,001$.

Tabel 1. Data Keberhasilan Robot saat Melewati Medan Menanjak

Sudut	Banyak percobaan	Berhasil	Gagal	Keberhasilan	Ket.
5	5 kali	5	0	100%	Berhasil
10	5 kali	5	0	100%	Berhasil
15	5 kali	5	0	100%	Berhasil
20	5 kali	5	0	100%	Berhasil
25	5 kali	5	0	100%	Berhasil
30	5 kali	5	0	100%	Berhasil
35	5 kali	0	5	0%	Gagal



Gambar 3.1 Grafik Saat Melewati Tanjakan 5°

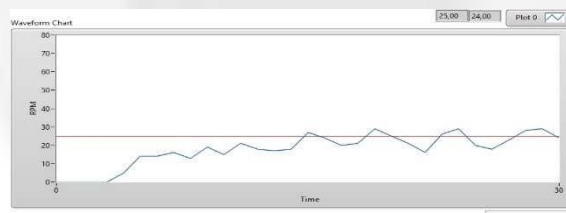


Gambar 3.2 Grafik Saat Melewati Tanjakan 30°

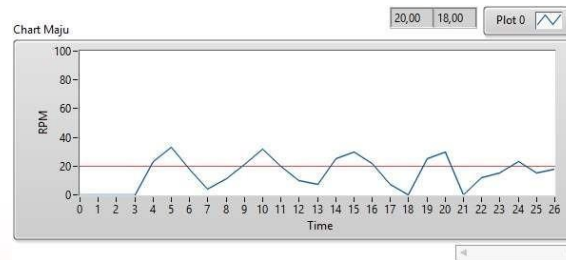
Berdasarkan pengujian, dari sudut tanjakan 5° hingga 30° , robot dapat menanjak dengan kecepatan yang konstan. Namun, pada sudut 35° , robot tidak sanggup melewati tanjakan tersebut. Itu artinya, kontrol PID yang berada pada LabVIEW dan kemampuan robot dapat bekerja dengan baik hingga sudut tanjakan 30° .

3.2 Pengujian Robot pada Jalan Menurun

Pengujian ini dilakukan dengan menjalankan robot pada medan menurun dengan kemiringan tertentu. Pada pengujian ini, kemiringan medan yang digunakan setiap perubahan 5° dengan sudut maksimal 30° . Nilai $k_p = 0,5$, $k_i = 0,05$, dan $d = 0,001$.



Gambar 3.3 Grafik Saat Melewati Turunan 5°



Gambar 3.4 Grafik Saat Melewati Turunan 30°

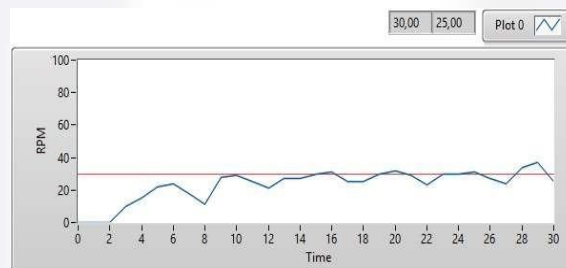
Tabel 2. Data Keberhasilan Robot saat Melewati Medan Menanjak

Sudut	Banyak percobaan	Percobaan berhasil	Percobaan gagal	Keberhasilan	Keterangan
5	5 kali	5	0	100%	Berhasil
10	5 kali	5	0	100%	Berhasil
15	5 kali	5	0	100%	Berhasil
20	5 kali	5	0	100%	Berhasil
25	5 kali	5	0	100%	Berhasil
30	5 kali	5	0	100%	Berhasil
35	5 kali	0	5	0%	Gagal

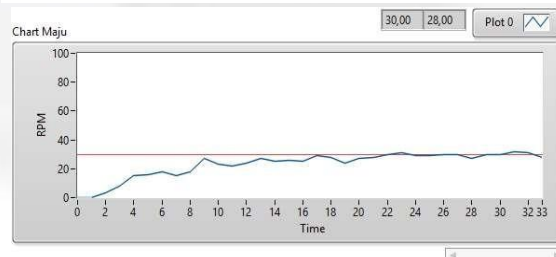
Berdasarkan hasil pengujian, dari sudut turunan 5° hingga 30°, robot dapat menurun dengan kecepatan yang konstan. Namun, pada saat melewati turunan 25° dan 30°, kecepatan motor sempat mengalami overshoot karena robot tertarik gravitasi. Selain itu, pada saat melewati turunan sudut 35°, robot tidak dapat mengontrol kecepatannya hingga akhirnya robot meluncur bebas.

3.3 Pengujian Robot pada Jalan Mendatar dan Free Run

Pengujian ini dilakukan untuk dibandingkan dengan pengujian pada saat robot diuji free run atau tanpa beban. Berdasarkan hasil uji coba, kecepatan robot pada saat free run dan pada saat berjalan pada jalan mendatar hasilnya sama. Kecepatan motor dapat stabil baik keadaan free run ataupun saat berjalan di jalan mendatar.



Gambar 3.5 Grafik Saat Free Run dengan Kontrol PID

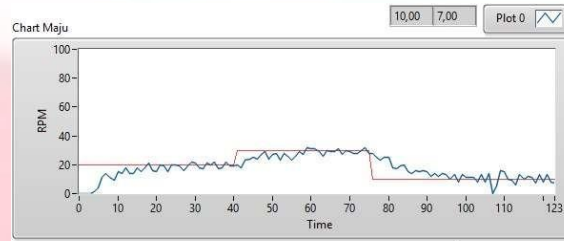


Gambar 3.6 Grafik Saat Melewati Jalan Mendatar

Itu artinya, kontrol PID masih dapat bekerja apabila terdapat perubahan beban yang tidak terlalu besar. Gangguan yang terjadi pada saat berjalan diatas tanah masih sangat kecil, sehingga kontrol PID masih dapat bekerja secara optimal.

3.4 Pengujian dengan Perubahan Setpoint

Pada pengujian ini, nilai setpoint diubah-ubah selama motor bergerak. Hal ini bertujuan untuk mengetahui respon dari robot pada saat terjadi perubahan setpoint.

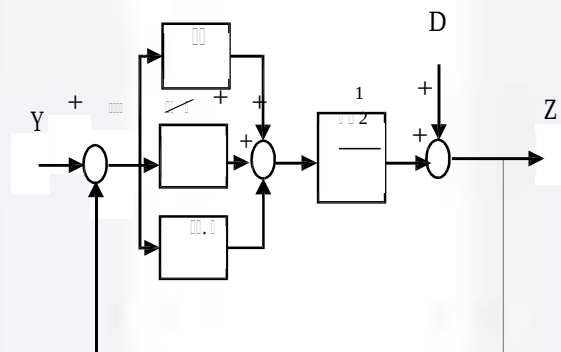


Gambar 3.7 Grafik Saat Setpoint Diubah

Berdasarkan hasil pengujian, kecepatan motor dapat menyesuaikan dengan setpoint walaupun setpoint diubah pada saat motor DC sedang berputar. Hanya saja respon perubahan kecepatan masih terlalu lama tetapi masih bisa menuju setpoint. Itu artinya, kontrol PID pada LabVIEW dapat bekerja dengan baik walaupun terdapat perubahan pengaturan setpoint.

3.5 Analisis Pengaruh Disturbance Terhadap Kecepatan Motor DC

Berdasarkan pengujian yang telah dilakukan, kecepatan motor DC pada robot tidak dapat stabil. Masih ada sedikit ripple yang kecil antara 2 hingga 8 RPM pada kondisi open loop. Pada saat pengujian menggunakan kontrol kecepatan motor juga sangat tidak stabil. Hal ini bisa disebabkan oleh gangguan dari internal, seperti karakteristik motor DC yang digunakan, adanya noise dari pembacaan sensor, dan juga kondisi medan yang dilalui oleh robot. Kondisi sistem yang diberikan gangguan atau memiliki gangguan dari internal maupun eksternal dapat digambarkan seperti pada Gambar 3.8.

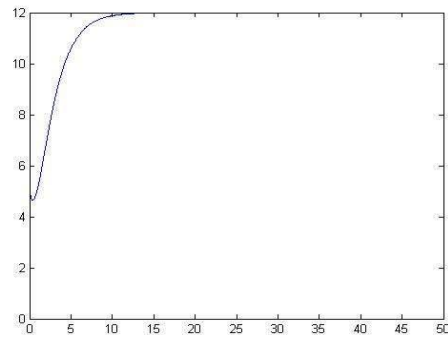


Gambar 3.8 Diagram Blok Sistem Close Loop dengan kontrol P, I, dan D ditambah Disturbance

Dari diagram blok diatas, nilai $\frac{1}{s}$ merupakan fungsi transfer yang digunakan dan nilai 1 merupakan gangguan yang diberikan kepada sistem. Dari diagram blok dan nilai-nilai diatas, bisa didapatkan persamaan fungsi transfer sistem yaitu,

$$\frac{Z}{Y} = \frac{1}{s^3 + 2s^2 + 3s + 1}$$

Berdasarkan fungsi transfer dan diagram blok diatas, maka grafik keluaran yang dihasilkan seperti pada Gambar 3.9.



Gambar 3.9 Grafik Close Loop Menggunakan Kontrol P, I, dan D pada Simulink

Dari hasil diatas, dapat dikatakan bahwa sistem yang terdapat gangguan baik dari dalam maupun dari luar, hasilnya kurang baik dan juga respon sistem masih cukup lambat. Namun, apabila sistem diberi kontrol PID, hasilnya menjadi lebih baik. Sistem menjadi lebih responsif dan minim error.

Hal tersebut juga terjadi pada robot yang digunakan pada tugas akhir ini. Respon dan keluaran yang dihasilkan tidak jauh berbeda dengan yang disimulasikan. Keluaran sistem dari robot masih terdapat error, namun lambat laun bisa mencapai setpoint dengan error yang semakin kecil setelah diberi kontrol PID.

4. Kesimpulan

Berdasarkan hasil uji coba dan analisis dapat disimpulkan:

- Dari beberapa percobaan, ripple masih saja terjadi walaupun sudah diberi kestabilan menggunakan PID. Hal ini kemungkinan dikarenakan karakteristik dari motor dc. Selain itu, kemungkinan terdapat noise pada pembacaan dari sensor, sehingga hasil nilai RPM yang terbaca masih kurang baik.
- Dengan menggunakan kontrol P, I, dan D dengan nilai konstan pada kondisi free run berjalan di jalan datar, menaik, dan menurun kecepatan motor bisa mencapai setpoint. Kecepatan motor juga lebih stabil, namun lambat laun ripple masih muncul walaupun kecil.
- Dengan menggunakan kontrol P, I, dan D pada kondisi free run, kecepatan motor bisa mencapai setpoint. Kecepatan motor juga lebih stabil, namun lambat laun error masih muncul walaupun kecil.

5. Daftar Pustaka

- [1] Alciatore, David G. Dan Michael B. Histan. 2012. Introduction to Mechatronics and Measurement System. New York. The McGraw-Hill Companies.
- [2] Kristiana, Samuel. 2010. Desain dan Sistem Pengendalian Robot Beroda Pemadam Api. [Online] Available at: <http://jurtek.akprind.ac.id/sites/default/files/hal-14-23-samuel-gabung-ok-pdf/> [Accessed 6 April 2015]
- [3] Pakpahan, S. 1988. Kontrol Otomatik. Jakarta: Erlangga.
- [4] Phillips, Charles L. dan Royce D. Harbor. 1996. Sistem Kontrol Dasar-Dasar. Jakarta: PT Prehallindo.
- [5] Pitorwarno, Endra. 2006. Robotika Desain, Kontrol, dan Kecerdasan Buatan. Yogyakarta: Andi.
- [6] Waluyo, Aditya Fitrianyah dan Syahril. 2013. Analisis Penalaan Kontrol PID pada Simulasi Kendali Kecepatan Putaran Motor DC Berbeban Menggunakan Metode Heuristik. [Online] Available at: <http://lib.itenas.ac.id/kti/wp-content/uploads/2014/03/2.-Analisis-Penalaan-Kontrol-PID.pdf/> [Accessed 18 April 2015]