DESAIN DAN IMPLEMENTASI KENDALI PID ADAPTIF PADA KECEPATAN MOTOR DC

DESIGN AND IMPLEMENTATION OF ADAPTIVE PID CONTROL TO DC MOTOR SPEED

Hardy Purnama Nurba[1], Erwin Susanto, S.T.,M.T.,PhD.[2], Agung Surya Wibowo, S.T.,M.T.[3]

[1,2,3]Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

[1] hardypurnamanurba@gmail.com, [2] erwinelektro@telkomuniversity.ac.id,

[3] agungsw@telkomuniversity.ac.id

Abstrak

Sistem kendali dengan PID tunggal telah banyak digunakan dalam sistem kendali kecepatan putar motor DC. Sistem kendali ini mampu mengendalikan kecepatan putaran motor DC hingga mencapai setpoint yang diberikan. Dalam kenyataannya, Sistem kendali PID tunggal hanya mampu bekerja untuk kondisi yang bersifat linier. Sedangkan Motor DC merupakan sebuah *plant* yang memiliki efek non-linieritas. Perbedaan sifat ini membuat sistem kendali PID tunggal tidak mampu menghasilkan respon keluaran yang memiliki karakteristik yang sama jika diberikan nilai *setpoint* yang berbeda-beda. suatu kendali PID yang bersifat adaptif yang mampu mengendalikan kecepatan putaran motor DC dalam setiap setpoint yang diberikan sesuai batasan yang ditentukan dengan cara menghilangkan karakteristik non-liniernya. Kendali Adaptif PID dirancang dengan menggunakan metode kendali adaptif *Gain Scheduling*. Metode ini diimplementasikan pada kendali PID dengan melakukan pemetaan pada daerah kerja motor DC. Penggunaan adaptif PID dengan *Gain scheduling* ini menghasilkan waktu naik (*rise time*) sebesar 0,9 detik. Overshoot dan error steady-state dari respon trayektori tidak ada, namun masih terdapat sedikit osilasi sebesar 1 rpm.

Kata Kunci: PID Tunggal, Adaptif PID, non-linieritas, linier, *gain scheduling*

Abstract

A single control system with PID control has been widely used in DC motor speed control system. In fact, a single PID control system is only able to work at linear conditions . But the DC motor is a plant that has non-linearity effect. Effect of non-linearity create a single PID control system is not able to produce output response that has the same characteristics if the given setpoint values are different. an adaptive PID control capable of controlling the rotation speed of the DC motor in any given setpoint corresponding limits specified by removing the non-linear characteristics. Adaptive PID control is designed using an adaptive control method Gain Scheduling. This method is implemented in PID control by mapping the work area DC motor. The use of adaptive PID using Gain scheduling produces the rise time (rise time) of 0.9 sec. Overshoot and error steady-state trajectory of response does not exist, but there is still a slight oscillation at 1 rpm.

Keywords: Single PID, PID Adaptive, non-linearity, linear, gain scheduling

1. Pendahuluan

Saat ini telah banyak sistem kendali yang diaplikasikan untuk mengendalikan kecepatan putar motor DC, salah satu sistem kendali yang telah diaplikasikan yaitu sistem kendali kecepatan dengan PID tunggal. Sistem kendali ini bekerja dengan cara memproses perhitungan berdasarkan variabel kendali Kp, Ki, dan Kd untuk mencapai kondisi sesuai *setpoint* yang diharapkan. Sistem kendali ini mampu menghasilkan respon keluaran dari kecepatan putar motor DC yang cukup baik. Namun dalam kenyataannya, sistem kendali dengan PID tunggal ini belum mampu menghasilkan suatu respon keluaran yang baik sesuai kondisi yang diinginkan saat terjadi perubahan *setpoint*.

Sistem kendali PID tunggal hanya mampu bekerja untuk kondisi yang bersifat linier. Namun

motor DC merupakan sebuah *plant* yang memiliki efek *non-linieritas*. Sehingga kerena perbedaan sifat inilah yang membuat sistem kendali PID tunggal tidak mampu menghasilkan respon keluaran yang memiliki karakteristik yang sama jika diberikan nilai *setpoint* yang berbeda-beda. Untuk menghasilkan suatu respon keluaran dengan karakteristik yang sama dari setiap *setpoint* yang berbeda-beda, maka harus diterapkan suatu metode yang mampu menghilangkan efek non- linieritas ini.

Salah satu cara untuk menghilangkan efek non-linieritas ini, maka dirancanglah kendali PID yang bersifat adaptif dalam mengendalikan kecepatan putar motor DC. *Gain scheduling* merupakan salah satu metode yang mampu menghilangkan efek non-linieritas. Metode ini akan me-linieritaskan kondisi non-linier pada plant yang berupa motor DC dengan cara mencari invers dari fungsi yang menyebabkan efek non-linieritas. Adanya proses penambahan *gain scheduling* pada bagian kendali ini, maka akan menyebabkan respon keluaran sistem memiliki karakteristik yang sama untuk setiap setpoint yang berbeda-beda, karena suatu sistem hanya akan menghasilkan respon keluaran dengan karakteristik sama bila sistemnya linier.

2. Dasar Teori dan Perancangan

2. 1. Kendali PID

Kendali PID merupakan sebuah sistem kendali yang tersusun oleh pengendali Proportional (P), Integral (I), dan Derivative (D). Kendali PID dapat direpresentasikan dalam bentuk simbolis, operator domain waktu, transformasi Laplace dan transformasi-z. Berikut bentuk transformasi- Z[3] :

$$u(t) = K_p e(t) \quad (1)$$

$$U(z) = \frac{K_i}{1-z^{-1}} E(z) \quad (2)$$

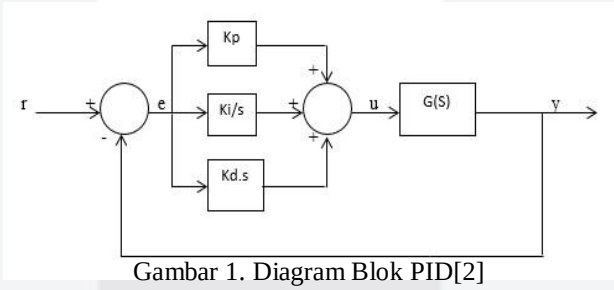
$$E(z) = K_d(1 - z^{-1})E(z) \quad (3)$$

Hasil representasi dapat dibentuk dalam persamaan beda[7], sebagai berikut :

$$u(n) = K_p e(n) \quad (4)$$

$$U(n) = K_i e(n) + U(n-1) \quad (5)$$

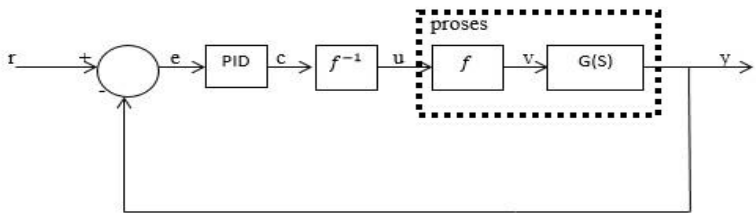
$$E(n) = K_d(e(n) - e(n-1)) \quad (6)$$



Gambar 1. Diagram Blok PID[2]

2. 2. *Gain Scheduling*

Gain Scheduling merupakan sebuah metode kendali adaptif yang bekerja dengan cara membuat batasan linieritas pada suatu daerah kerja yang memiliki efek non-linieritas. Metode ini bekerja dengan mempengaruhi regulator dengan menambahkan sebuah fungsi invers untuk menghilangkan efek non-linieritas. Penambahan fungsi invers terhadap non-linieritas ini bertujuan agar terjadi perubahan linieritas terhadap dinamika sistem. Blok diagram kerja Gain scheduling sebagai berikut :



Gambar 2. Diagram Blok *Gain Scheduling[1]*

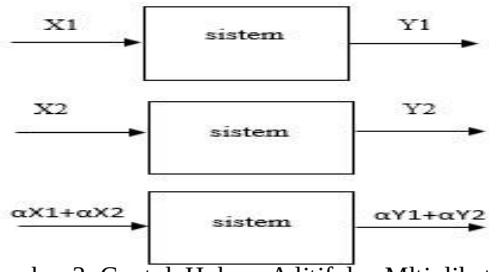
Dari model diagram blok diatas bahwa terdapat fungsi invers dan fungsi non-linearitas yang mempengaruhi kinerja sistem. Fungsi invers dan fungsi non-linearitas tersebut berfungsi agar masukan setpoint tetap dalam kondisi yang sama. Berikut persamaannya [1]:

V= ◆◆◆=◆◆$^{-1}$(◆◆◆) (7)

Dalam melakukan proses gain scheduling terdapat beberapa prosedur, yaitu :
Pertama, Grafik yang melengkung didekati dengan garis-garis lurus
Kedua, Lakukan invers pada grafik fungsi non-linearitas
Ketiga, Cari persamaan fungsi dari setiap garis lurus sebagian (daerah kerja)

2. 3. **Sistem Linier Dan Non-Linier**

Klasifikasi suatu sistem dapat dibagi menjadi 2 tipe sistem, yaitu sistem linier dan sistem non-linier. Sistem dikatakan linier jika suatu sistem memenuhi teorema superposisi, yang merupakan kombinasi dari sifat hukum aditif dan multiplikatif.



Gambar 3. Contoh Hukum Aditif dan Mltiplikatif

Sistem ini akan menghasilkan sebuah keluaran yang sifatnya pasti dengan masukan yang ada dalam suatu proses. Suatu sistem yang linear akan menghasilkan suatu hubungan antara masukan dan keluaran yang jika diplot dalam sebuah grafik maka akan membentuk suatu garis yang lurus tanpa lengkungan.

Sedangkan sistem non-linear merupakan sebuah sistem yang tidak memenuhi teorema superposisi. Sistem ini tidak akan menghasilkan keadaan yang serupa saat diberi masukan yang berbeda. Sistem ini tidak memiliki sifat hukum aditif dan multiplikatif.

2. 4. **Arduino Uno**

Arduino Uno merupakan sebuah kit yang berbasis mikrokontroler Atmega328 yang open source, dimana kit mikrokontroler ini bisa dikembangkan dan diimplementasikan oleh banyak kalangan khususnya dalam bidang elektronika[4]. Arduino Uno dapat bekerja jika telah ditanam sebuah program berbahasa C pada chip Mikrokontroler. Bahasa pemograman Arduino dapat dikatakan lebih sederhana karena memilki kumpulan library yang lengkap. Dalam proses *upload* program tidak memerlukan media downloader secara eksternal karena sudah terdapat bootloader. Komunikasi serial yang digunakan sudah dalam bentuk USB.

2. 5. **Motor DC**

Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan adanya beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Motor DC terdiri dari stator dan rotor. Stator merupakan salah satu bagian yang tidak berputar berupa kumparan medan, sedangkan rotor merupakan salah satu bagian yang bergerak berupa kumparan jangkar. Untuk menggerakkan suatu benda dengan beban tertentu, maka pada motor DC terdapat *gearbox*. *Gearbox* berfungsi sebagai media motor DC untuk menghasilkan tenaga dalam menggerakkan benda tertentu yang memiliki massa. Dengan adanya *gearbox* maka motor DC akan memiliki torsi.

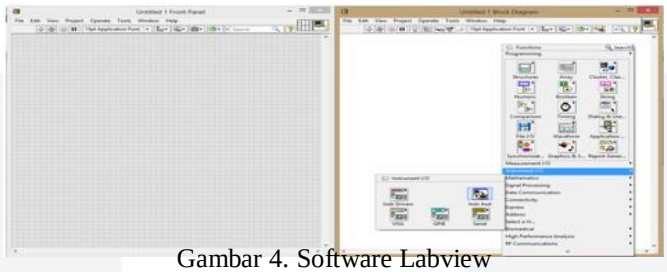
2. 6. **Driver Motor Dual H-Bridge L298**

Driver motor merupakan sebuah pengendali pergerakan motor DC yang dapat mengendalikan arah dan kecepatan putaran motor DC. Pengendali arah dan kecepatan putaran motor DC ini dilakukan dengan representasi grafis rangkaian H-Bridge. Rangkaian ini disebut H-bridge karena menyerupai huruf H yang memiliki 4 switch. Berikut spesifikasi driver motornya :

- Menggunakan rangkaian pada IC L298 dual H-bridge
- Masukan motor 7 hingga 24 V DC
- Menggunakan standar level logic TTL sebagai logic pengendali

2. 7. **Software Labview**

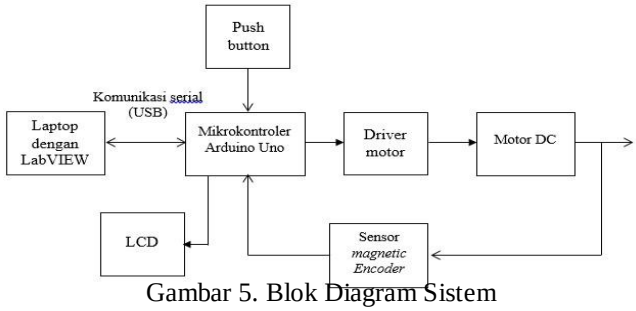
LabVIEW merupakan sebuah software yang dikhususkan untuk pengontrolan sistem elektronika maupun sistem kendali. Labview memiliki keunggulan dalam mempermudah pembuatan HMI (Human Machine Interface) dengan menggunakan G-code (Graphical Code Programming) yang hanya memerlukan wiring dalam pengkodeannya[5]. Dalam penggunaannya, Labview lebih fleksibel dalam memberikan sebuah perintah (tidak tergantung dalam penggunaaan bahasa pengkodean tertentu).



Gambar 4. Software Labview

2. 8. **Perancangan Sistem secara Umum**

Perancangan sistem direalisasikan dalam beberapa tahap blok sistem yang saling berintegrasi untuk mencapai sebuah sistem kendali adaptif PID. Blok sistem terdiri dari bagian masukan, bagian proses, dan bagian keluaran. Pada bagian masukan terdapat dua tahap masukan, yaitu berupa nilai setpoint RPM yang diinginkan dan masukan berupa pulsa-pulsa dari umpan balik sensor *magnetic encoder* pada PIN interupsi di mikrokontroler Arduino Uno. Pemberian nilai setpoint dilakukan melalui masukan eksternal dari pushbutton. Pada bagian proses terdiri dari tiga tahap yang terjadi, yaitu tahap pemilihan daerah kerja yang sesuai, tahap perhitungan oleh kendali PID , dan tahap pemetaan hasil perhitungan kendali PID terhadap daerah kerja yang sesuai batasannya. Pada bagian keluaran terdiri dari proses perputaran motor DC, menampilkan hasil putaran kedalam bentuk trayektori pada sofware LabVIEW secara serial.

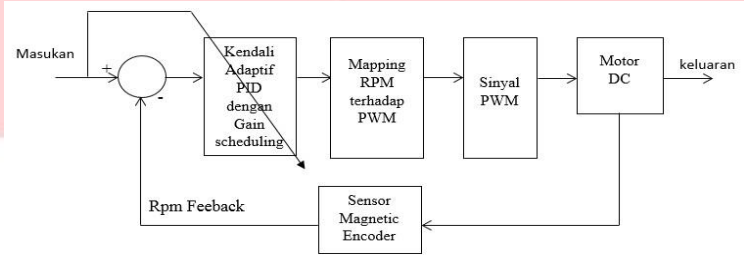


Gambar 5. Blok Diagram Sistem

2. 9. **Kinerja Sistem**

Dalam proses kinerja sistem bahwa setpoint yang diberikan berupa nilai Rpm dari kecepatan putaran motor DC encoder. Pada saat awal kinerja sistem, ditetapkan semua kondisi awal bernilai nol. Ketika diberikan perubahan nilai setpoint, maka sistem akan memilih jenis PID

yang akan digunakan berdasarkan batasan linearitas dari *gain scheduling*. Untuk menghasilkan sinyal PWM yang tepat akan dilakukan proses pemetaan terhadap batasan daerah kerja yang linear. Keluaran berupa sinyal PWM yang akan dikirimkan ke driver motor untuk menggerakkan motor DC. Saat motor DC mulai berputar maka sensor magnetic encoder akan memberikan sinyal pulsa yang disampling setiap 0,1 detik dan hasil penjumlahan pulsa akan diproses secara perhitungan yang akan menghasilkan nilai dalam satuan Rpm sebagai umpan balik.



Gambar 6. Blok Diagram Kinerja Sistem

2. 10. **Kendali Adaptif PID**

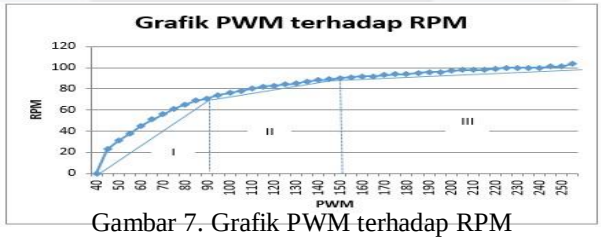
Dalam kendali adaptif yang dirancang ini dibuat dua jenis PID, yaitu PID 1, PID 2 , dan PID 3 dimana dalam setiap jenis PID ini nilai variabel kendali Kp, Ki, dan Kd nya akan berbeda tergantung dari parameter batasan daerah kerja linearitas dari motor DC. penentuan variabel kendali ini dilakukan secara *offline* dengan cara membuat semua variabel kendali dalam keadaan awal nol, lalu secara perlahan diberikan nilai Kp hingga respon mencapai overshoot yang kemudian diberikan nilai Kd dan Ki untuk meredam overshoot dan mencapai steady-state[6]. Daerah kerja ini akan membatasi nilai maksimum dan minimum untuk setiap pemberian nilai *setpoint* pada masing-masing daerah kerja. Batasan daerah kerja yang diinisialisasikan berdasarkan uji coba pada pergerakan motor DC dengan nilai PWM yang berbeda dan kemudian data hasil uji coba di plot kedalam bentuk dinamika yang menghasilkan efek non-linearitas. Kendali adaptif PID ini dirancang berdasarkan beberapa kemungkinan dari tiga daerah kerja yang ditetapkan. Berikut tabel kemungkinan dalam penentuan jenis PID :

Tabel 1 kemungkinan kondisi yang dicapai

Daerah kerja PWM terhadap RPM		Jenis Pid yang digunakan
Rentang RPM	Rentang PWM	
0 hingga 75	40 hingga 90	PID 1
71 hingga 95	91 hingga 150	PID 2
91 hingga 104	151 hingga 255	PID 3

2. 11. **Penentuan Persamaan Linieritas Daerah Kerja**

Dalam menentukan persamaan linearitas pada daerah kerja ini terdapat pada dua hubungan, yaitu pada grafik PWM terhadap RPM dan pada grafik RPM terhadap PWM. Persamaan ini didapatkan dari hasil perhitungan melalui persamaan garis miring yang linearitas pada masing-masing daerah kerja.



Gambar 7. Grafik PWM terhadap RPM

Pada daerah kerja I : $0 < \text{RPM} \leq 70 \rightarrow RPM = \frac{75}{50}(PWM - 40)$

Pada daerah kerja II : 70 < RPM ≤ 90→ $RPM = \frac{20}{60}(PWM - 90) + 75$

Pada daerah kerja II : 90 < RPM ≤ 104→ $RPM = \frac{9}{105}(PWM - 150) + 95$

Maka invers dari grafik PWM terhadap RPM diatas adalah :

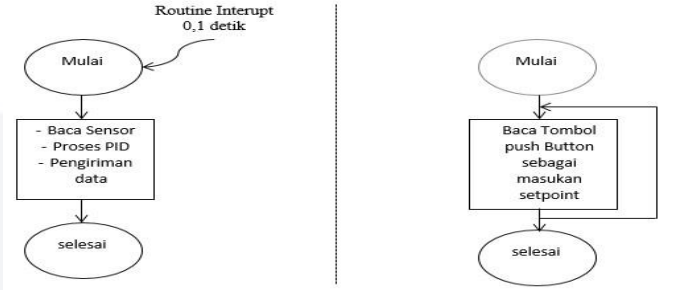
Pada daerah kerja I : 0 < RPM ≤ 70 → $PWM = \frac{50}{75}RPM + 40$

Pada daerah kerja II : 70 < RPM ≤ 90→ $PWM = \frac{60}{20}(RPM - 75) + 90$

Pada daerah kerja II : 90 < RPM ≤ 104→ $PWM = \frac{105}{9}(RPM - 95) + 150$

2. 12. **Perancangan Program Arduino**

Terdapat dua bagian program yang di proses dalam mengendalikan sistem. Bagian pertama merupakan program yang bekerja pada bagian utama pengendali. Pada bagian ini terdapat tiga proses yang dilakukan selama waktu sampling 0,1 detik, yaitu proses pembacaan data dari sensor, proses perhitungan kendali PID, proses pengiriman data secara serial ke komputer. Bagian kedua merupakan program yang selalu menunggu masukan dari pushbutton secara terus-menerus.



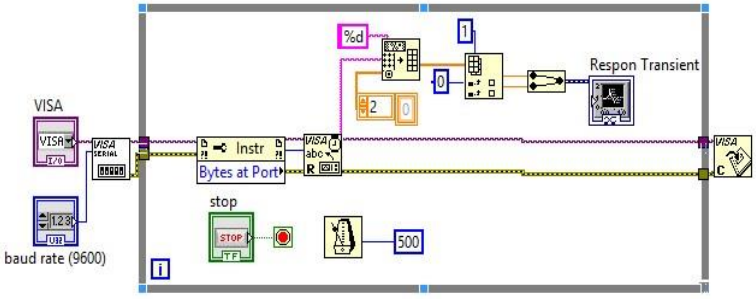
Gambar 8. Perancangan Program Arduino

2. 13. **Perancangan Program LabVIEW**

Pada perancangan Labview terdapat beberapa proses sebagai berikut:

Gambar 9. Perancangan Program LabVIEW

Berikut model grafis program serial LabVIEW yang dirancang :



Gambar 10. Model grafis program LabVIEW

3. Pembahasan

3. 1. Pengujian Dan Analisa

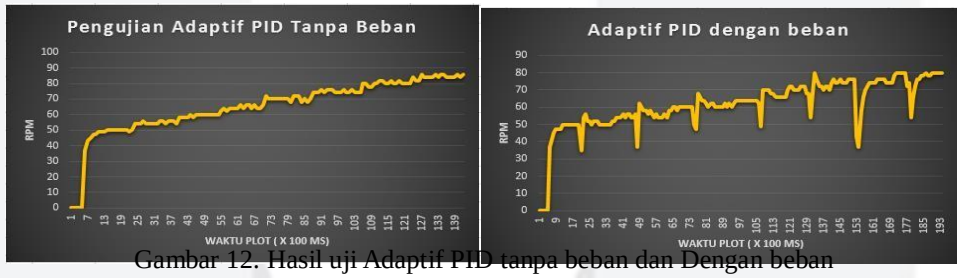
Pada bagian pengujian penulis melakukan pengujian dengan 2 tahap, yaitu tahap pengujian PID tunggal dan tahap pengujian Adaptif PID. Pengujian ini dilakukan dengan memberikan nilai setpoint awal sebesar 50 rpm dan menikan kecepatan hingga 90 rpm. Berikut hasil pengujiannya :

- Pengujian dengan PID tunggal

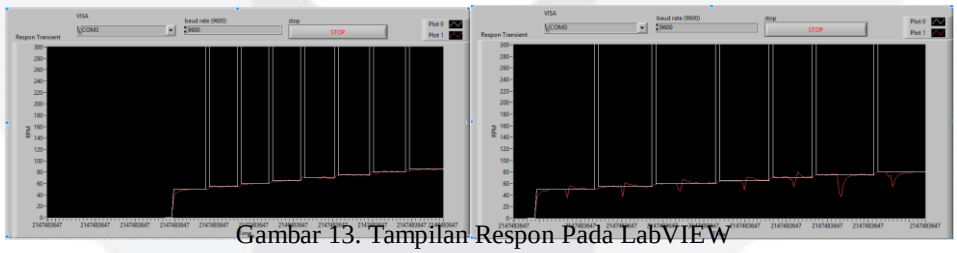
Gambar 11. Hasil uji PID Tunggal

Pada saat pengujian menggunakan PID tunggal tanpa beban maka dilakukan dengan memberikan setpoint awal sebesar 50 RPM, lalu RPM dinaikan dalam kelipatan 10 hingga 100 RPM. Dari hasil tersebut dapat terlihat bahwa PID tunggal belum mampu mengendalikan setpoint yang berubah tanpa batasan yang ditentukan. Overshoot sangat besar mencapai setengah dari setpoint awal.

- Pengujian dengan Adaptif PID



Gambar 12. Hasil uji Adaptif PID tanpa beban dan Dengan beban



Gambar 13. Tampilan Respon Pada LabVIEW

Putaran motor DC menghasilkan respon keluaran yang cenderung lebih baik dan menuju setpoint yang ditentukan. Namun karena saat penentuan daerah linieritas terdapat selisih beberapa rpm dari garis lurus yang diambil, maka saat kenaikan rpm terdapat sedikit osilasi. Namun hal tersebut tidak terlalu bermasalah karena tujuan utama dari penelitian ini yaitu agar dalam setiap kenaikan setpoint respon yang dihasilkan cenderung sama dalam menuju setpoint yang diberikan. Dari respon yang dihasilkan didapat bahwa waktu naik *(rise time*) sebesar 0.9 detik. Saat diberi beban pada setiap tingkatan RPM maka kendali PID dapat menormalkan kembali keadaan sesaat setelah terkena gangguan.

4. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan pada perancangan sistem kendali adaptif PID pada kecepatan motor DC, dapat diambil kesimpulan sebagai berikut :

1. Kendali PID tunggal sudah mampu mengendalikan perputaran motor DC untuk setpoint tertentu, namun saat diinginkan untuk mengendalikan setpoint yang berbeda maka kendali PID tunggal belum memberikan respon trayektori yang sama. Hal tersebut disebabkan karena terdapat karakteristik yang non-linier pada motor DC sebagai *plant* yang dikendalikan.
2. Motor DC merupakan sebuah *plant* yang memiliki karakteristik yang non-linier. Sifat non-linier ini dapat membuat kinerja motor DC tidak selalu sama dalam setiap kondisi masukan yang berbeda. Sehingga ketika diinginkan kinerja motor DC yang selalu sama dalam setiap kondisi setpoint yang berbeda, maka diperlukan sebuah metode untuk mengatasi non-linieritas tersebut. Metode kendali adaptif *Gain Scheduling* dapat menghilangkan non-linearitas dengan pendekatan garis lurus yang terpotong-potong untuk mendekati linier. Dengan adanya pendekatan linieritas ini maka kendali PID diharapkan mampu menghasilkan respon trayektori yang sama dalam setiap waktu.
3. Setalah dilakukan pengujian maka dihasilkan waktu naik (*rise time*) sebesar 0,9 detik. Overshoot dan error stedady-state dari respon trayektori tidak ada, namun masih terdapat sedikit osilasi sebesar 1 rpm yang disebabkan *tuning* yang belum tepat. Disisi lain waktu sampling sangat berpengaruh terhadap respon yang dihasilkan, saat waktu sampling terlalu besar maka respon dari motor DC menjadi lama dan kurang baik. Penggunaan waktu sampling sebesar 0,1 detik mampu menghasilkan respon yang baik.
4. Saat dilakukan pengujian terhadap sistem dengan diberikan gangguan sesaat pada setiap setpoint, maka kendali Adaptif PID mampu mengembalikan keadaan seperti saat belum diberikan gangguan. Hal ini sesuai tujuan sistem kendali yaitu mampu mengembalikan keadaan sesaat setelah terkena gangguan.

Daftar Pustaka

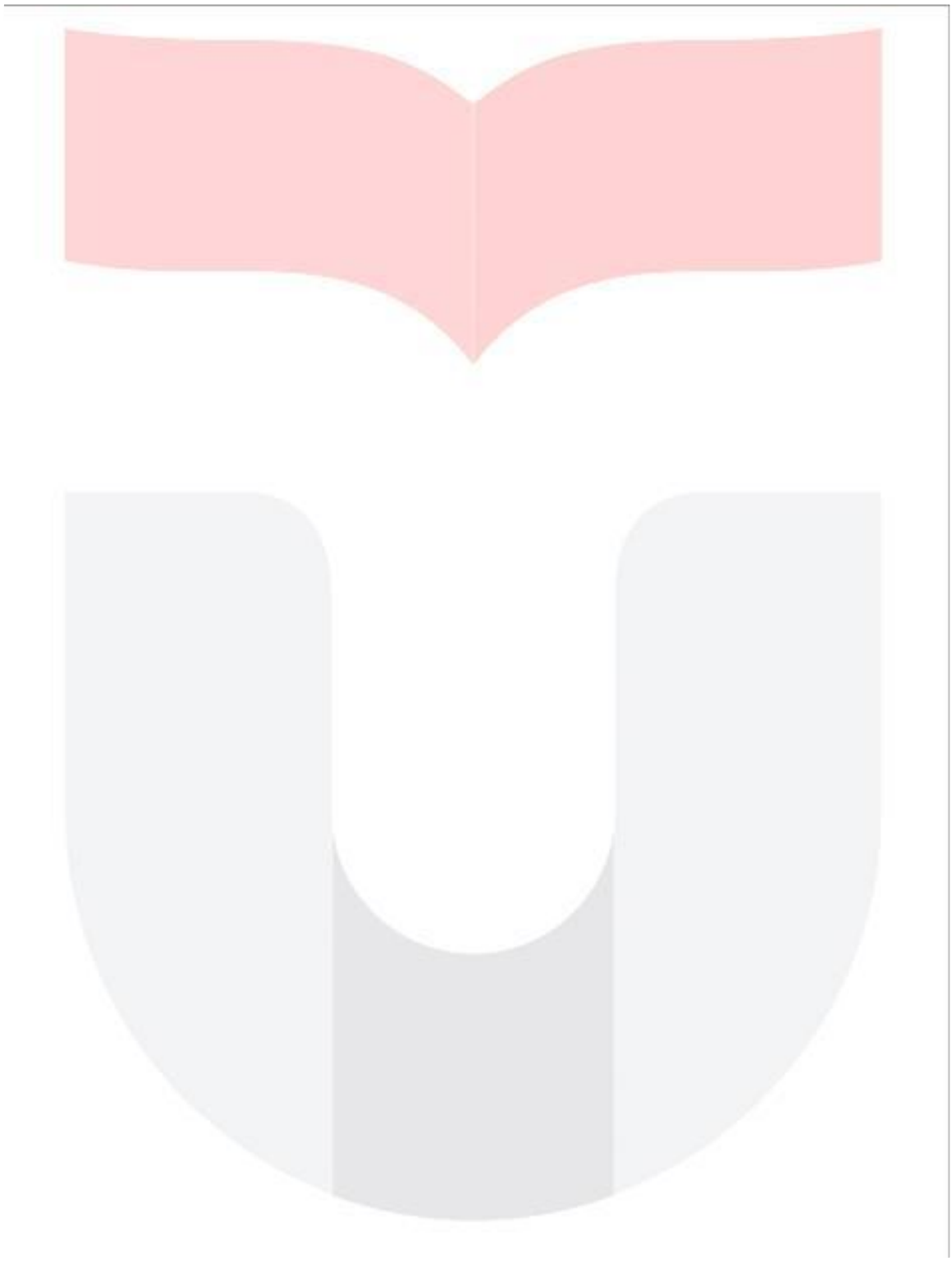
[1] Astrom,J.K. dan Wittenmark,Bjorn. 1989. *Adaptive Control*. Canada : Addison Wesley Publishing Company.

[2] Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 1 Edisi Kedua*. Jakarta : Erlangga.

[3] Ogata, Katsuhiko. 1995. *Discrete-Time Control Systems Second Edition*. New Jersey : Prentice Hall.

[4] Syahwil, Muhammad. 2013. *Panduan Mudah Simulasi & Praktek Mikrokontroler Arduino*. Yogyakarta : Andi.

[5] Artanto, Dian. 2012. *Interaksi Arduino Dan Labview,* Jakarta : Elex Media Computindo.

[6] Vahid,Frank. dan Givargi, Tony. 2002. *Embedded System Design A Unified Hardware/Software Introduction*. United State Of America. John Wiley & Sons, Inc.

[7] Oppenheim,A.V., Willsky.A.S. dengan Nawab, S.H. 2000. *Sinyal & Sistem Jilid 1 Edisi Kedua*. Jakarta : Erlangga.