

Perancangan Simulasi PID Controller Menggunakan Graphic User Interface dan Simulink

Adri Senen¹; Titi Ratnasari²; Yoakim Simamora³

^{1,2,3} Jurusan Teknik Elektro, Institut Teknologi PLN

¹ adrisenen@itpln.ac.id

ABSTRACT

PID (Proportional – Integral – Derivative) is a controller widely used in the determination of an instrumentation system accuracy, by looking at the response and the feedback characteristics of the system. Nevertheless, to see the response of a system with various combinations of input signals and control action is difficult and tedious. Besides, the process to bring up the system response manually also requires mathematical calculations which is quite complicated, so it is necessary a software capable of overcoming it, in this case the Graphic User Interface (GUI) can be the solution. The Software comes with a variety of toolbox that allows users to perform certain calculations. It has been developed a special toolbox for simulation named Simulink, namely by using a block closed loop system diagram. The description form of the Block diagram is done in the-s domain, so the form transfer function must be in the Laplace domain. The process of PID input can be done quickly and with various combinations even in the process of determining the value still performed by trial and error method, to get the desired response. Its simulations can clearly depict response characteristics of various control actions without having to perform manual calculations anymore.

Keywords: Controller, PID, Simulation, GUI, Simulink

ABSTRAK

Kontroller PID (Proportional–Integral–Derivative) merupakan salah satu kontroller yang banyak dipakai dalam menentukan keakuratan suatu sistem instrumentasi, dengan melihat respon dan karakteristik umpan balik dari sistem tersebut. Namun demikian untuk melihat respon suatu sistem dengan berbagai macam kombinasi sinyal input dan aksi pengontrolan merupakan hal yang sulit dan membosankan. Disamping itu proses untuk memunculkan respon sistem yang dilakukan secara manual pun membutuhkan perhitungan matematis yang lama dan cukup rumit, sehingga diperlukan suatu software yang mampu mengatasi hal tersebut, dalam hal ini Graphic User Interface (GUI) dapat menjadi solusinya. Software ini dilengkapi dengan berbagai toolbox yang memudahkan pemakai untuk melakukan perhitungan- perhitungan tertentu. Bahkan saat ini sudah dikembangkan juga toolbox khusus untuk simulasi yang diberi nama simulink, yakni dengan menggunakan diagram block closed loop system. Bentuk penjabaran diagram block dilakukan dalam domain-s, sehingga bentuk transfer function harus dalam domain laplace. Proses input nilai PID bisa dilakukan dengan cepat dan dengan berbagai kombinasi walaupun dalam proses penentuan nilai masih dilakukann dengan metode trial dan error, untuk mendapatkan respon yang diinginkan. Hasil simulasinya bisa menggambarkan secara jelas karakteristik respon dari berbagai aksi pengontrolan tanpa harus melakukan perhitungan manual lagi.

Kata kunci: Kontroler, PID, Simulasi, Respon, GUI, Simulink

1. PENDAHULUAN

PID Controller merupakan salah satu jenis pengatur yang banyak digunakan. Selain itu sistem ini mudah digabungkan dengan metoda pengaturan yang lain seperti Fuzzy dan Robust [1]. Sehingga akan menjadi suatu sistem pengatur yang semakin baik. PID Controller sebenarnya terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu P (*Proportional*), D (*Derivative*), dan I (*Integral*), masing-masing memiliki parameter tertentu yang harus diset untuk dapat beroperasi dengan baik, yang disebut sebagai konstanta. Masing – masing dari controller tersebut mempunyai respon dan karakteristik yang berbeda – beda. [2]

Dahulu untuk melihat respon suatu sistem dengan berbagai macam kombinasi sinyal input dan aksi pengontrolan merupakan hal yang sulit dan membosankan. Untuk melakukan langkah-langkah di atas diperlukan ketelitian yang tinggi dan hasil penggambarannya sering kali kurang (tidak) akurat. Dikarenakan *transfer function* sebuah system dalam domain s atau dalam fungsi laplace maka untuk menganalisis sebuah respon system perlu dilakukan inverse laplace atau mengubah dari domain s ke domain t [3]. Tentu saja hal tersebut sangat tidak efisien dan memakan waktu yang lama. Maka dari itu dipergunakan software, untuk mempermudah dalam memahai, menganalisis, dan mendapatkan respon system yang diinginkan. Perkembangan teori kontrol juga diikuti oleh software pendukungnya. Mulai dari software untuk pemrograman sistem, sampai dengan software untuk proses simulasinya. Salah satu software yang dapat dipergunakan untuk simulasi tersebut adalah *MatLab* dari *Mathworks, Inc*

Software ini memungkinkan dilakukan perancangan dan analisa PID dalam 2 cara yaitu Perancangan sistem pengendali menggunakan *graphic user interface* (GUI) dan simulink. Kedua rancangan ini bisa untuk mendapatkan gambaran tanggapan sistem dengan sinyal masukan dan aksi pengontrolan. Hasil rancangan ini nanti bisa membantu dalam menganalisis sistem yang dibuat sudah baik atau belum, disamping itu bisa melihat tanggapan suatu sistem dengan berbagai macam kombinasi sinyal masukan dan aksi pengontrolan secara cepat, mudah dan tepat.

Aplikasi *MatLab* dalam bidang pengaturan dilengkapi *Control Toolbox*. Toolbox ini sudah dilengkapi dengan berbagai macam fungsi pendukung yang dipergunakan dalam analisa sistem kontrol [4]. Sehingga perancangan software nantinya akan bisa mengakomodir perubahan sinyal input sehingga dapat melihat respon sistem jika mendapat input sinyal tertentu. Disamping itu kombinasi antara sinyal input dan jenis aksi pengontrolan ini akan menghasilkan dan menampilkan respon yang berbeda-beda, sehingga memudahkan dalam memahami karakteristik dari respon controller itu sendiri. Perancangan yang dihasilkan juga bisa digunakan untuk *closed loop system* atau *open loop system* atau keduanya

2. METODE PENELITIAN

2.1. Metode Konvensional

Dalam merancang sebuah sistem kontrol, hal yang pertama dilakukan adalah membuat sebuah blok diagram sistem. Dimana dalam blok diagram tersebut akan berisi tranfer function sebagai formula yang akan dilihat responnya, dengan memberikan aksi pengontrolan yang berbeda – beda (kombinasi). Parameter – parameter dari PID controller akan memberikan pengaruh yang berbeda pada karakteristik sistem. Secara umum respon parameter setiap aksi pengontrolan dapat dilihat pada tabel 1 dibawah

Tabel 1. Karakteristik Respon Kontroler PID

Tanggapan Loop Tertutup	Waktu Naik	Overshoot	Waktu turun	Kesalahan keadaan tunak
Proportional (K_p)	Menurun	Meningkat	Perubahan Kecil	Menurun
Integral (k_i)	Menurun	Meningkat	Meningkat	Hilang
Derivative (K_d)	Perubahan Kecil	Menurun	Menurun	Perubahan Kecil

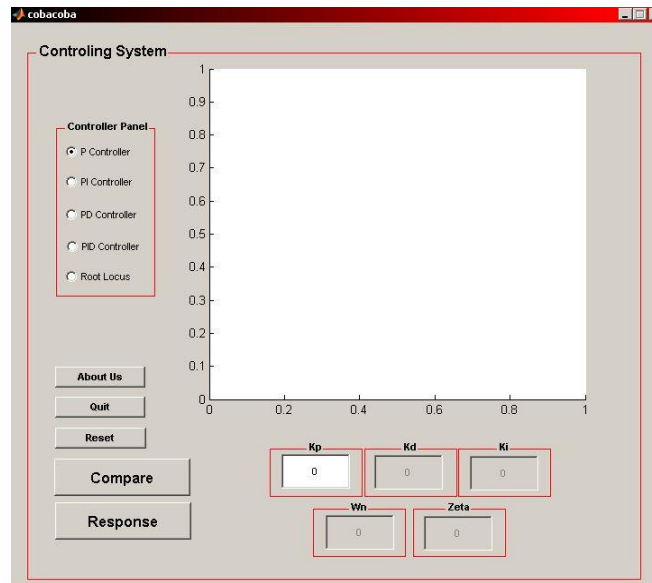
Untuk mendapatkan respon suatu sistem secara konvensional, maka langkah – langkah yang harus dilakukan adalah sebagai berikut [5] :

1. Menentukan transfer function sistem, dengan terlebih dahulu telah di laplace kan (dalam bentuk fungsi s)
2. Memilih jenis aksi pengontrolan proportional – integral atau derivarive serta menentukan masing – masing konstanta nya
3. Menggabungkan transfer function yang sudah didapatkan dengan jenis aksi pengontrolan.
4. Melakukan perhitungan dengan Laplace Transform untuk memperoleh transfer function sistem dalam bentuk s – domain
5. Menentukan jenis aksi pengontrolan dengan menggabungkan transfer function yang telah didapatkan sebelumnya
6. Memberikan sinyal input (biasanya dalam bentuk fungsi step, ramps atau impulse) kedalam tranfer function, sehingga akan memnemtuk transfer function yang baru
7. Menggambarakan fungsi dalam domain t untuk mendapatkan respon sistem, dengan cara melaplace invers kan fungsi yang didapatkan sebelumnya

Untuk melakukan langkah-langkah di atas diperlukan ketelitian yang tinggi dan hasil penggambarannya sering kali kurang (tidak) akurat. Selain itu, jika perancang ingin mengamati respon sistem terhadap sinyal input yang lain, maka proses-proses tersebut sebagian besar akan diulang kembali. Hal ini bertambah kompleks jika perubahan yang dilakukan tidak terbatas pada sinyal input, tetapi juga pada jenis aksi pengontrolannya [6].

2.2. Metode Perancangan Graphic User Interface (GUI)

GUI adalah singkatan dari *Graphical User Interface*, sebuah aplikasi display dari MATLAB yang mengandung tugas, perintah, atau komponen program yang mempermudah user (pengguna) dalam menjalankan sebuah program dalam MATLAB [7]. Pada GUI matlab ini telah disempurnakan sebuah *Graphical User Interface* untuk mempermudah dalam menganalisis PID controller, tampilan dari hasil rancangan GUI dapat dilihat pada gambar 1 dibawah ini.



Gambar 1. Rancangan GUI PID controller

Simulato PID yang ditunjukkan Gambar 1 diatas, inputan controller Kp, Ki dan Kd nya dibuat terpisah agar bisa dilakukan variasi inputan sesuai dengan kombinas controller yang diinginkan.

Sebelum memahami sebuah PID controller, hal yang perlu diketahui adalah diagram block yang digunakan dalam pengontrolan. Pada GUI PID controller diatas menggunakan diagram block closed loop system. Perancangan PID controller pada matlab pada app guide ikuti beberapa langkah berikut untuk mendapatkan respon yang diinginkan [8]:

1. Open Matlab dan App Guide.
2. Tambahkan edit text untuk P,I,D control.
3. Tambahkan Push Button untuk mengetahui output dari program nantinya.
4. Tambahkan exes untuk tampilan output grafik.
5. Lalu running program.
6. Setelah program berjalan (belum bisa di gunakan pada matlab dikarenakan tidak ada program, pada barisan atas terdapat editor program).
7. Masukkan program pada program push button agar dapat menampilkan hasil output yg sesuai.
8. Seimbangkan setiap Kp, Ki, Kd sampai didapatkan keseluruhan respon yang diinginkan. Anda dapat merujuk pada tabel 2.1 sebagai pegangan karakteristik controller.
9. Klik push button.
10. Selesai.

Sekarang perhatikan PID controller, fungsi transfer loop tertutup pada sistem yang dicontohkan dengan PID controller adalah:

$$\frac{X(s)}{F(s)} = \frac{K_p s^2 + K_p s + K_I}{s^2 + (10s + K_p)s^2 + (20 + K_p)s + K_I} \quad (1)$$

Setelah beberapa trail dan error dijalankan, penguatan Kp=350, Ki=300, Kd=50 memberikan respon yang diinginkan. Untuk meyakinkan, masukkan perintah berikut dalam m-file dan jalankan di Command Window. Anda seharusnya mendapatkan plot respon langkah seperti berikut:

Perintah masukan manual

```

handles=guidata(hObject)
n=str2double(get(handles.n,'string'));
d1=str2double(get(handles.d1,'string'));
d2=str2double(get(handles.d2,'string'));
d3=str2double(get(handles.d3,'string'));
Kp=str2double(get(handles.kp,'string'));
Ki=str2double(get(handles.ki,'string'));
Kd=str2double(get(handles.kd,'string'));
num=[0 0 n];
denum=[d1 d2 d3];
Gp=tf(num,denum);
H=(1);
Gc=pid(Kp,Ki,Kd);
Mc=feedback(Gp*Gc,H);
step(Mc);
axes(handles.axes1)
grid on

```

atau bisa juga program langsung

```

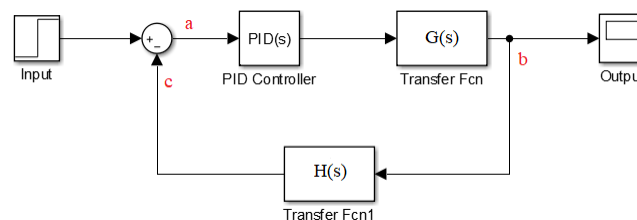
Kp=350;
Ki=300;
Kd=50;
num=[Kd,Kp,Ki];
den=[1 10+Kd 20+Kp];
t=0:0.01:2;
step (num, den, t)

```

Akhirnya, anda tidak perlu menerapkan ketiga controller (P,I,D) dalam sistem tunggal jika tidak perlu. Sebagai contoh, jika PI-controller sudah memberikan respon yang cukup baik (seperti contoh di atas), maka anda tidak perlu menerapkan D-controller pada sistem tersebut. Usahakan sedapat mungkin mudah dijalankan

2.3. Metode Simulink

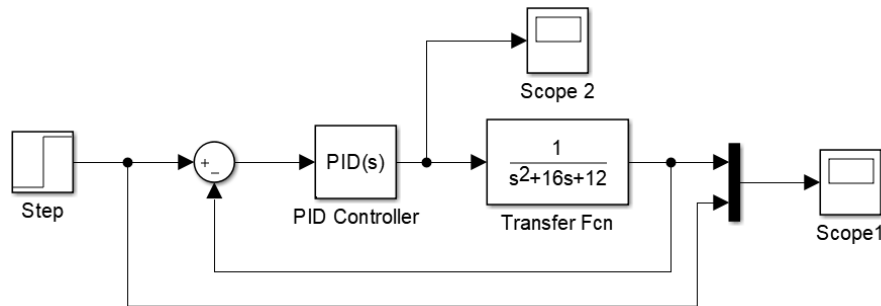
Simulink adalah lingkungan blok diagram untuk pemodelan matematika dan mendukung simulasi. Pada metode simulink ini telah dibuat diagram blok seperti gambar 2 dibawah ini:



Gambar 2. Perancangan diagram block closed loop system GUI PID controller menggunakan simulink

Berdasarkan diagram *block closed loop system* diatas, untuk melihat bentuk gelombang respon output diperlukan penjabaran diagram block. Bentuk penjabaran diagram block dilakukan dalam domain-s atau domain laplace, sehingga bentuk *transfer function* untuk $G(s)$ ataupun $H(s)$ harus

dalam domain-s atau domain laplace. Adapun *transfer function* yang digunakan adalah *transfer function* pertama [9]. Rancangan blok diagram menggunakan simulink bisa dibarakan dalam gambar 3 dibawah

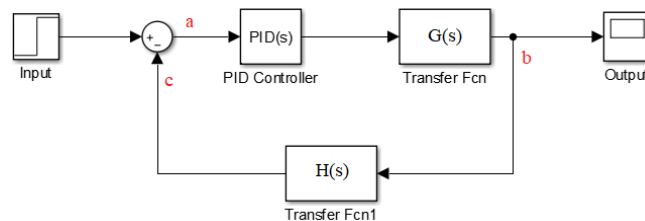


Gambar 3. Blok diagram pada simulink dengan *transfer function*

Diagram blok seperti yang terlihat pda gambar 3 diatas, adalah untuk jenis inputan step. Fungsi Scope 1 adalah untuk menampilkan respon dari transfer function yang telah di control oleh PID controller, sedangkan fungsi scope 2 adalah untuk menampilkan respon output dari inputan PID [10]. Fungsi sum junction adalah untuk menumlahkan respon sinyal inputan dan sinyal feedback seperti terlihat pda diagram blok dibawah

3. HASIL DAN PEMBAHASAN

Untuk melihat bentuk gelombang respon output diperlukan penjabaran diagram blok seperti tampak pada gambar 4.



Gambar 4. Asumsi parameter pada diagram *block closed loop system* GUI PID controller

Bentuk penjabaran diagram block dilakukan dalam domain-s atau domain laplace, sehingga bentuk *transfer function* untuk G(s) ataupun H(s) harus dalam domain-s atau domain laplace. Untuk penjabaran bentuk persamaan output berdasarkan diagram *block closed loop system* diatas dapat dilihat pada penjabaran rumus dibawah ini

$$G(s) = \frac{G_1(s)}{G_2(s)} \tag{1}$$

$$H(s) = \frac{H_1(s)}{H_2(s)} \tag{2}$$

$$PID(s) = \frac{K_d \cdot s^2 + K_p \cdot s + K_i}{s} \tag{3}$$

$$b = G(s) \cdot PID(s) \cdot (input - c)$$

$$b = G(s) \cdot PID(s) \cdot (input - H(s) \cdot b)$$

$$b = G(s) \cdot PID(s) \cdot input - G(s) \cdot PID(s) \cdot H(s) \cdot b$$

$$b + G(s) \cdot PID(s) \cdot H(s) \cdot b = G(s) \cdot PID(s) \cdot input$$

$$b \cdot \{1 + G(s) \cdot PID(s) \cdot H(s)\} = G(s) \cdot PID(s) \cdot input$$

$$b = \frac{G(s) \cdot PID(s) \cdot input}{1 + G(s) \cdot PID(s) \cdot H(s)} \tag{4}$$

$$b = \frac{\frac{G_1(s)}{G_2(s)} \cdot \left(\frac{K_d \cdot s^2 + K_p \cdot s + K_i}{s} \right) \cdot \text{input}}{1 + \left(\frac{G_1(s)}{G_2(s)} \right) \cdot \left(\frac{K_d \cdot s^2 + K_p \cdot s + K_i}{s} \right) \cdot \left(\frac{H_1(s)}{H_2(s)} \right)} \quad (5)$$

$$b = \frac{G_1(s) \cdot H_2(s) \cdot \text{input} \cdot (K_d \cdot s^2 + K_p \cdot s + K_i)}{s \cdot G_2(s) \cdot H_2(s) + G_1(s) \cdot H_1(s) \cdot (K_d \cdot s^2 + K_p \cdot s + K_i)} \quad (6)$$

Sehingga persamaan outputnya adalah

$$\text{output} = \frac{G_1(s) \cdot H_2(s) \cdot \text{input} \cdot (K_d \cdot s^2 + K_p \cdot s + K_i)}{s \cdot G_2(s) \cdot H_2(s) + G_1(s) \cdot H_1(s) \cdot (K_d \cdot s^2 + K_p \cdot s + K_i)} \quad (7)$$

Dimana:

$G_1(s)$ = Pembilang dari persamaan $G(s)$

$G_2(s)$ = Penyebut dari persamaan $G(s)$

$H_1(s)$ = Pembilang dari persamaan $H(s)$

$H_2(s)$ = Penyebut dari persamaan $H(s)$

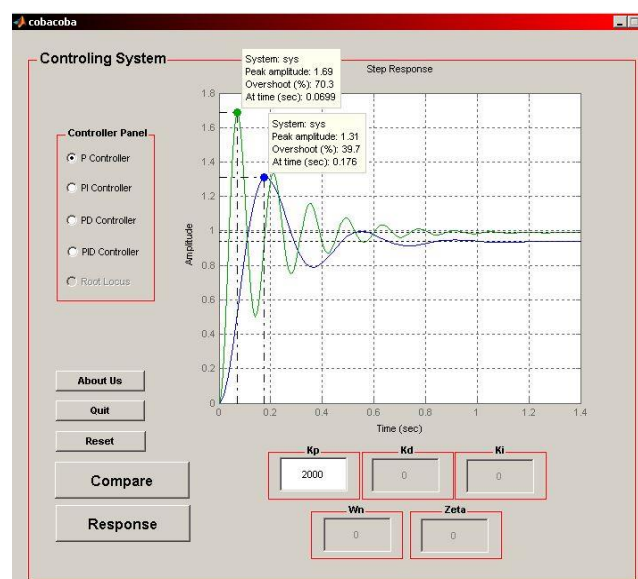
K_p = Nilai Parameter *proportional*

K_i = Nilai Parameter *integral*

K_d = Nilai Parameter *Differential*

3.1. Simulasi Perancangan PID dengan GUI

Simulasi dilakukan dengan hanya memasukkan nilai K_p , K_i , K_d pada masing – masing inputan yang telah disediakan. Setela itu dengan menekan tombol respon maka akan langsung muncul respon sistem. Sedangkan untuk membandingkan hasil respon sistem cukup dengan menekan tombol *compare*. Hasil running program rancangan PID seperti terlihat pada gambar dibawah:

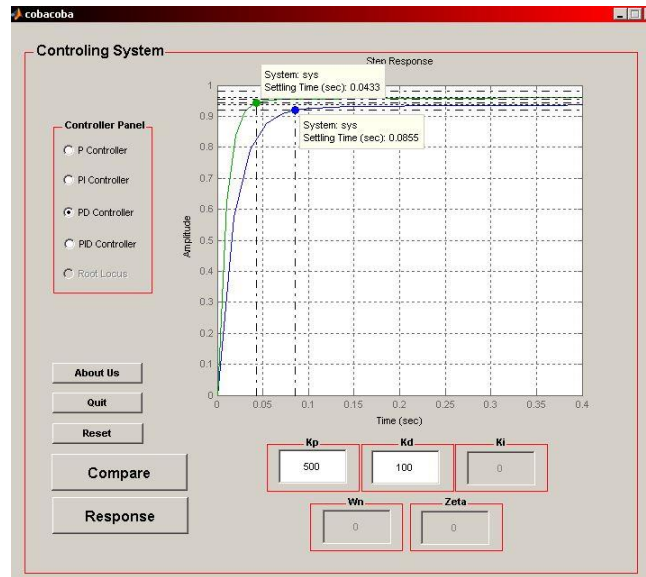


Gambar 5. Hasil Simulasi Kontroller P

Gambar 5 merupakan respon sistem dengan menggunakan proportional controller. Dimana dengan menaikan nilai K_p akan menunjukkan bahwa proporsional controller mereduksi waktu naik dan kesalahan keadaan tunak, meningkatkan overshoot, mengurangi waktu turun dalam skala kecil, selain itu kondisi steady state tidak dipengaruhi oleh nilai K_p karena waktu yg dibutuhkan relatif sama.

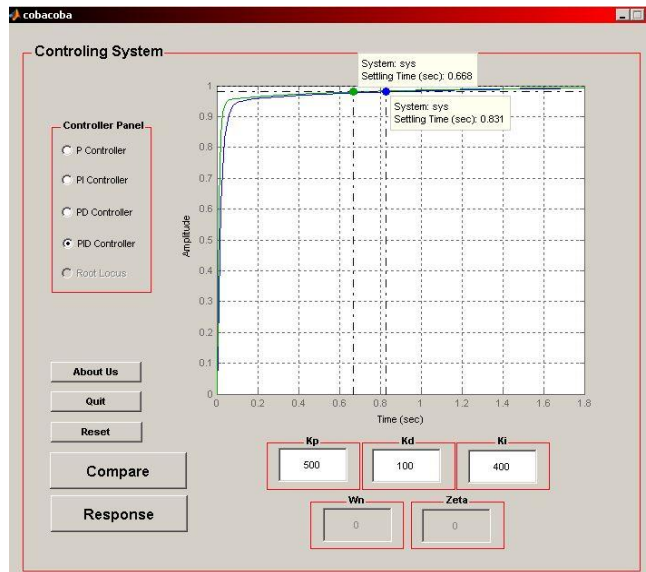
Jika dilakukan kombinasi kontroler proporsional dengan kontroler derivatif dengan memvariasikan nilai inputan K_d dengan K_p dibuat tetap, maka akan hasil simulasi menunjukkan

derivative controller mereduksi overshoot dan waktu turun dan memberikan efek kecil pada waktu naik dan kesalahan keadaan tunak seperti terlihat pada gambar 6 dibawah



Gambar 6. Hasil simulasi kontroler Proporsional – derivatif

Jika dilakukan simulasi lengkap untuk semua kontroler yaitu K_p , K_i dan K_d maka akan terlihat karakteristik respon seperti yang ditunjukkan gambar dibawah :



Gambar 7. Hasil simulasi kontroler Proporsional – Integral - Derivatif

Plot yang muncul pada gambar 7 menunjukkan bahwa sistem dengan waktu naik yang cepat, tanpa overshoot, dan tanpa kesalahan keadaan tunak Setting time PID controller akan makin cepat seiring bertambahnya ketiga nilai komponennya, Nilai overshoot akan ditekan akibat adanya K_d dan akan menghasilkan steady state pada waktu yg relatif konstan.

3.2. Simulasi Perancangan PID dengan Simulink

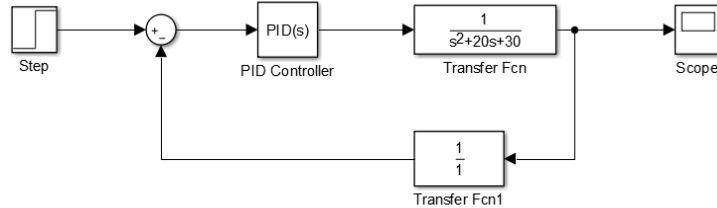
Bentuk penjabaran diagram block dilakukan dalam domain-s atau domain laplace, sehingga bentuk *transfer function* untuk $G(s)$ ataupun $H(s)$ harus dalam domain-s atau domain laplace. Untuk penjabaran bentuk persamaan output berdasarkan diagram block closed loop system diatas dapat dilihat pada penjabaran rumus dibawah ini

$$G(s) = \frac{1}{s^2+20s+30}$$

(8)

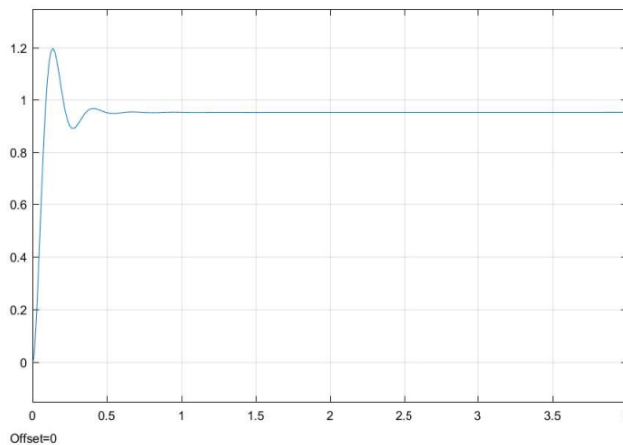
$$H(s) = 1 ; Kp = 600 ; input = 1; Time = 4 seconds$$

Dimana diagram blok nya dapat dilihat pada gambar dibawah



Gambar 8. Blok Diagram untuk Kontroller Proporsional

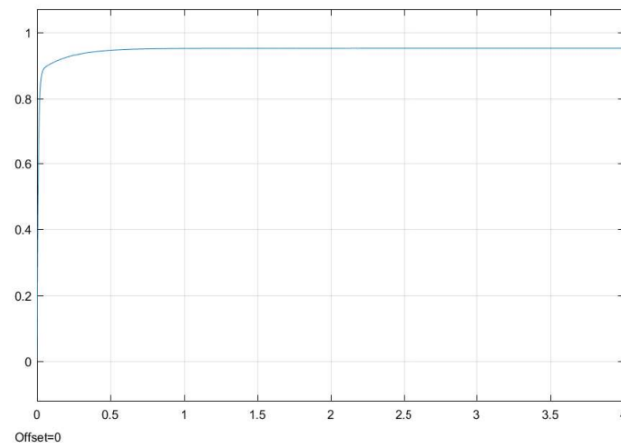
Sedangkan hasil respon dari blok diagram yang ditunjukkan gambar 8 tersebut secara keseluruhan karakteristiknya sama dengan dengan perancangan menggunakan GUI. Hasilnya dapat dilihat pada gambar 9 dibawah ini:



Gambar 9. Keluaran respon dari kontroller proporsional

Untuk memastikan perancangan hasil simulink ini benar maka dilakukan juga pengujian dengan melakukan variasi kontroller yang berbeda, dalam hal ini dipilih kontroller proporsional - derivatif, dengan tetap mengacu kepersamaan sebelumnya cuma ditambahkan nilai inputan sebagai berikut :

$$H(S) = 1; Kp = 600; Kd = 110; input = 1; Time = 4 seconds. Hasil respon seperti gambar 10.$$



Gambar 10. Keluaran respon kontroler proporsional – derivativ

Hasil respon pada gambar 10, memperlihatkan karakteristik dari kontroler proportional dan derivative, dimana terlihat respon system lebih stabil. Penambahan kontroler derivative menjadikan overshoot lebih kecil, mengurangi settling time dengan perubahan settling time relatif kecil.

4. KESIMPULAN DAN SARAN

Hasil perancangan kontroler PID yang dilakukan baik secara GUI maupun simulink ternyata bisa menunjukkan karakteristik respon sistem. Proportional controller memberikan efek mengurangi waktu naik, tapi tidak menghapus kesalahan keadaan tunak. Integral controller memberikan efek menghapus kesalahan keadaan tunak tapi berakibat pada respon transient yang buruk. Sedangkan derivatif controller akan memberikan efek meningkatkan stabilitas sistem, mengurangi overshoot, dan menaikkan respon transient.

Hasil rancangan ini lebih mudah digunakan dan hasil simulasi sangat memudahkan pengguna dalam menganalisa respon sistem secara cepat dan tepat yang didasarkan uji coba beberapa persamaan polinomial. Keakuratan hasil simulasi sudah sesuai dengan yang seharusnya, ini ditunjukkan hasil simulasi sama dengan hasil pemodelan simulink (diagram blok) yang di ujikan. Rancangan ini bisa dijadikan software bagi pengguna mula dalam memahami respon sistem dalam tiap varian inputan, set point, dan jenis kontroler yang digunakan, sehingga informasi dari tiap fungsi yang diberikan utuh kepada pengguna

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Institut Teknologi PLN khususnya LPPM yang telah memberi dukungan yang membantu pelaksanaan penelitian dan atau penulisan artikel.

DAFTAR PUSTAKA

- [1] J. Live, H. B. Wang, X. M. Zhu, Z. H. Shen and J. Y. Chen, "Simulation Research of Fuzzy Auto-Tuning PID Controller Based on Matlab," IEEE International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, 2017, pp. 180-183. 2017
- [2] M. A. Siddiqui, S. H. Laskar, M. N. Anwar and M. Naseem, "A Model-Free PI/PID Controller based on Direct Synthesis Approach to achieve Disturbance Rejection," IECON 2019 - 45th

Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 2019, pp. 207-212

- [3] Ghazanfar Shahgholian, Controller Design for Three-Mass Resonant System Based on Polynomial Method, *International Journal of Science, Technology and Society*. Vol. 5, No. 2, 2017, pp. 13-25
- [4] L. Ying, "The Simulation Design of PID Control Based on LabVIEW and MATLAB," 2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI), Shanghai, 2018, pp. 1-8
- [5] N. A. Patil and G. V. Lakhekar, "Design of PID controller for cascade control process using genetic algorithm," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, 2017, pp. 1089-1095
- [6] Christer Dalen, David Di Ruscio. PD/PID controller tuning based on model approximations: Model reduction of some unstable and higher order nonlinear models. Vol. 38, No. 4, 2017
- [7] Y. Zhang, Y. Jia, T. Chai, D. Wang, W. Dai and J. Fu, "Data-Driven PID Controller and Its Application to Pulp Neutralization Process," in *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 828-841, May 2018
- [8] Hasan, "Improvement of systems response of a PID controller in underdamped condition," *Systems Response*, p. 6, 2015
- [9] V. S. Patel Ankur, "Performance Analysis of PID Controller and its significance for closed loop system," S.E. Hamamci, "A New PIDTuning Method Based on Transient Response Control", *Balkan Journal of Electrical & Computer Engineering*, Vol.2, No.3, 2014
- [10] S.E. Hamamci, "A New PIDTuning Method Based on Transient Response Control", *Balkan Journal of Electrical & Computer Engineering*, Vol.2, No.3, 2014