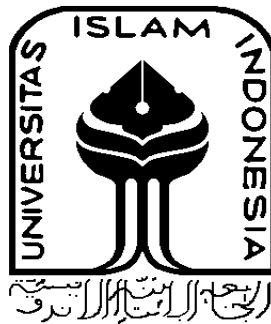


**Perancangan dan Implementasi *Controller* NCTF untuk Operasi  
PTP pada Sistem Posisi Linier Menggunakan *Ball-Screw* :  
Analisis Kinerja *Compensator PI* dan *Compensator Fuzzy***

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



**Disusun Oleh :**

**Nama : Saga Kusuma Wijaya**

**No. Mahasiswa : 12525070**

**NIRM : 2012060205**

**JURUSAN TEKNIK MESIN  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2017**

## PERNYATAAN BEBAS PLAGIARISME

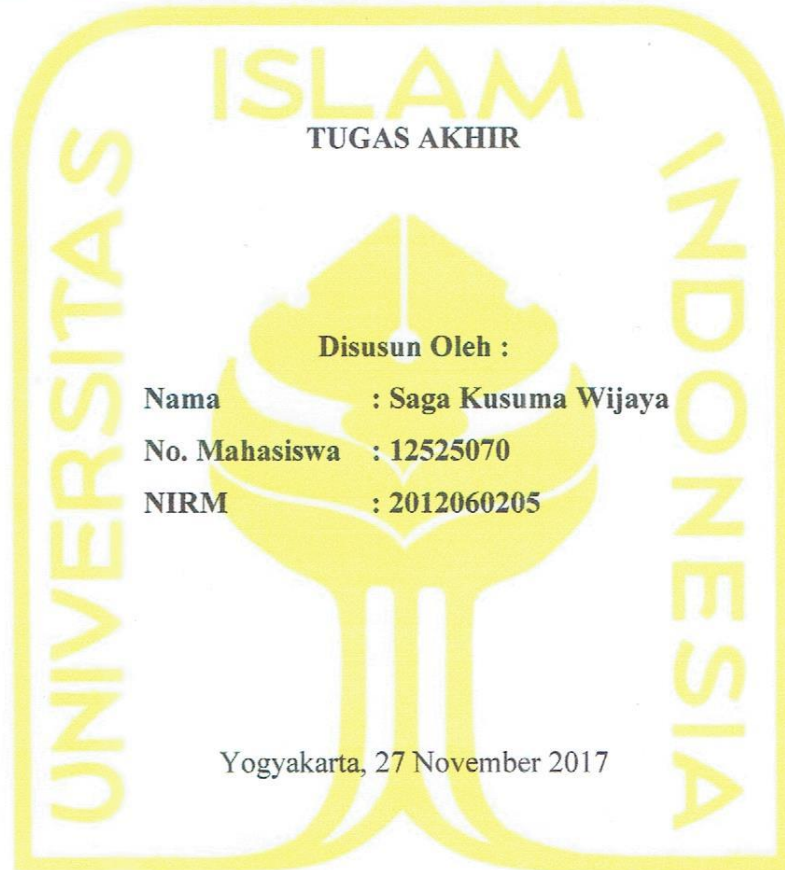
“Dengan ini saya menyatakan dalam penulisan skripsi ini tidak terdapat karya atau pendapat yang telah ditulis dan dipublikasikan orang lain kecuali tertulis dengan jelas sebagai acuan dalam naskah dengan disebutkan nama penulis atau pengarang dan dicantumkan dalam daftar pustaka. Pernyataan ini saya buat dengan sungguh-sungguhnya dan apabila di kemudian hari terdapat ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.”

Yogyakarta, 25 Januari 2018

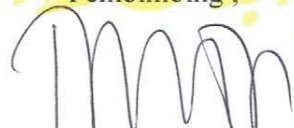


**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

**Perancangan dan Implementasi *Controller* NCTF untuk Operasi  
PTP pada Sistem Posisi Linier Menggunakan *Ball-Screw* :  
Analisis Kinerja *Compensator PI* dan *Compensator Fuzzy***



الإسلامية  
Pembimbing ,

  
Purtojo, S.T., M.Sc.

## LEMBAR PENGESAHAN DOSEN PENGUJI

Perancangan dan Implementasi *Controller* NCTF untuk Operasi  
PTP pada Sistem Posisi Linier Menggunakan *Ball-Screw* :  
Analisis Kinerja *Compensator PI* dan *Compensator Fuzzy*

TUGAS AKHIR

Disusun Oleh :

Nama : Saga Kusuma Wijaya

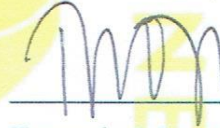
No. Mahasiswa : 12525070

NIRM : 2012060205

Tim Penguji

Purtojo S.T., M.Sc.

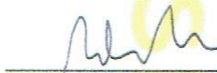
Ketua



Tanggal : 9 Januari 2018

Agung Nugroho Adi S.T., M.T.

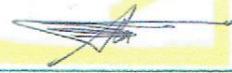
Anggota I



Tanggal : 9 Januari 2018

Faisal Arif Nurgesang S.T., M.Sc.

Anggota II



Tanggal : 9 Januari 2018

Mengetahui

Ketua Jurusan Teknik Mesin



Risdiyono, S.T., M.Eng., Ph.D.

## HALAMAN PERSEMBAHAN

Setiap detak jantung ini selalu ingin mengingat  
Mengingat alasan dan tujuan berdiri di sini  
Banyak jalan yang hanya salah satu dari mereka yang kupilih  
Berdiri di sini, adalah bagian dari perjalanan di jalan yang telah kupilih

Terima kasih kepada semua yang mendukung, menolong, dan bahkan  
yang menghalangi aku untuk berdiri di sini  
Telah banyak pelajaran yang telah kuambil dari tempat aku berdiri ini

Wahai kalian, pendukung, penolong, dan penghalangku, selanjutnya  
kalian akan telah menjadi pembentukku  
Terima Kasih, sekali lagi aku ucapkan

Dengan penuh penghormatan dan terima kasih  
Aku persembahkan karya tulisku ini untuk para pembentukku

Dariku,  
Saga Kusuma Wijaya

## HALAMAN MOTTO

*“Tak akan pernah ada Einstein jika Thales tak memulai.*

*Tak akan pernah ada kesimpulan valid jika tak ada premis.*

*Setidak-mengerti apapun kita terhadap sesuatu, yakinlah bahwa ada penjelasan dibelakangnya, karena tak akan pernah ada akibat jika tak ada sebab.”*

*Anonym*

*“Hidup itu adalah tentang mengambil keputusan untuk suatu tujuan, seperti keputusan menarik pelatuk sebuah pistol untuk suatu titik tujuan.*

*Jika pelatuk sudah ditarik, tak usah ragu, bersikaplah seperti peluru yang mengejar titik tujuan.”*

*Anonym*

*“A person who never made a mistake never tried anything new”*

*Albert Einstein*

## **KATA PENGANTAR**

**Assalamu'alaikum Warahmatullahi Wabarakatuh.**

Puji dan syukur alhamdulillah, penulis panjatkan kehadirat Allah SWT, atas limpahan rahmat dan hidayah-Nya. Akhirnya tugas akhir dengan judul **"Perancangan dan Implementasi *Controller* NCTF untuk Operasi PTP pada Sistem Posisi Linier Menggunakan *Ball-Screw* : Analisis Kinerja *Compensator PI* dan *Compensator Fuzzy*"** ini telah selesai setelah melalui proses belajar yang cukup panjang. Perjuangan melewati masa-masa sulit telah berlalu dengan selesainya laporan tugas akhir ini.

Saya mengucapkan terima kasih yang sebesar-besarnya kepada seluruh dosen Prodi Teknik Mesin FTI UII khususnya kepada Bapak Purtojo S.T., M.Sc. sebagai dosen pembimbing tugas akhir penulis, kepada keluarga yang selalu mendukung, dan kepada teman-teman yang membantu menghilangkan kejenuhan. Sulit rasanya menjabarkan terima kasih yang setulus-tulusnya dalam bentuk tulisan. Sekali lagi, kepada semuanya, saya hanya bisa mengucapkan Terima Kasih.

Semoga tulisan ini bermanfaat bagi para pembaca. Penulis sangat terbuka terhadap kritik dan saran untuk tulisan ini. Penulis harap, tulisan ini dapat berperan sebagai karya ilmiah yang dapat membantu kemajuan ilmu pengetahuan secara umum, dan dapat menjadi referensi bagi mahasiswa dalam pengerjaan karya ilmiah secara khusus. Demikianlah dari penulis, ***Selamat Belajar!***

**Wassalaamu'alaikum Warahmatullaahi Wabarakatuh.**

***Perancangan dan Implementasi Controller NCTF untuk Operasi  
PTP pada Sistem Posisi Linier Menggunakan Ball-Screw : Analisis  
Kinerja Compensator PI dan Compensator Fuzzy***

***Saga Kusuma Wijaya***

**ABSTRAK**

*Operasi point-to-point menggunakan pengendali nominal characteristic trajectory following (NCTF) telah dilakukan sebelumnya. NCTF adalah metode praktis untuk mengendalikan objek kendali motor DC. Controller NCTF terdiri dari nominal characteristic trajectory (NCT) dan compensator, yang mana NCT dibentuk berdasarkan eksperimen open-loop. Pada penelitian ini, compensator NCTF diteliti untuk menganalisis kinerjanya. Jenis compensator yang diteliti adalah compensator proportional-integral (PI) dan Fuzzy, yang mana keduanya telah dirancang dan dianalisis. Arduino UNO sebagai controller-card, Scilab sebagai perancang blok diagram, driver Escon sebagai driver motor DC, linear ball-screw yang dihubungkan dengan motor DC sebagai objek kendali, dan linear encoder sebagai sensor. Compensator PI didesain dengan analisis matematis, yang mana prosedur analisis tersebut sebelumnya sudah dilakukan oleh peneliti lain. Compensator fuzzy didesain dengan tiga variabel fuzzy, yaitu dua input dan satu output. Setiap variabel memiliki tiga membership function, yaitu negatif, zero, dan positif. Metode defuzzification yang digunakan adalah center of area (COA). Penelitian ini menghasilkan analisis kinerja compensator PI dan fuzzy. Compensator PI dapat mengikuti NCT pada eksperimen objek. Compensator fuzzy memiliki settling time lebih kecil daripada compensator PI pada simulasi komputer.*

***Kata kunci : NCTF, PI, Fuzzy.***



***Perancangan dan Implementasi Controller NCTF untuk Operasi  
PTP pada Sistem Posisi Linier Menggunakan Ball-Screw : Analisis  
Kinerja Compensator PI dan Compensator Fuzzy***

***Saga Kusuma Wijaya***

**ABSTRACT**

*Point-to-point operation using nominal characteristic trajectory following (NCTF) controller has been done. NCTF is practical method to controlling DC motor control object. NCTF controller consist of a nominal characteristic trajectory (NCT) and a compensator, which is NCT constructed based on open-loop experiment. In this research, NCTF's compensator researched to analyze its performance. Types of compensator studied were proportional-integral (PI) and fuzzy compensator, which both of them have been designed and analyzed. Arduino UNO as controller-card, Scilab as block diagram designer, escon driver as DC motor driver, linear ball screw connected to DC motor as control object, and linear encoder as sensor. PI compensator designed by way of mathematical analysis, which is its analysis procedure has been done by another researcher. Fuzzy compensator designed with three fuzzy variabls, namely two inputs and an output. Each variabel has three membership functions, namely negative, zero, and positive. Defuzzification method used was center of area (COA). This research was yielding performance analysis of PI and fuzzy compensator. PI compensator can followed NCT at real object simulation. Fuzzy compensator has settling time smaller than PI compensatorat computer simulation.*

***Keywords : NCTF, PI, Fuzzy.***

## DAFTAR ISI

Halaman Judul .....	i
Lembar Pengesahan Dosen Pembimbing .....	ii
Lembar Pengesahan Dosen Penguji .....	iii
Halaman Persembahan .....	iv
Halaman Motto .....	v
Kata Pengantar.....	vi
<i>Abstrak</i> .....	vii
<i>Abstract</i> .....	viii
Daftar Isi .....	ix
Daftar Tabel.....	xi
Daftar Gambar .....	xii
Bab 1 Pendahuluan .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.6 Sistematika Penulisan .....	4
Bab 2 Tinjauan Pustaka .....	6
2.1 Konsep NCTF .....	6
2.1.1 Eksperimen <i>Open-loop</i> .....	7
2.1.2 Konstruksi NCT.....	8
2.2 Desain <i>Compensator PI</i> .....	9
2.3 Desain <i>Compensator fuzzy</i> .....	10
2.3.1 <i>Membership Function</i> .....	10
2.3.2 <i>Fuzzification</i> .....	12
2.3.3 <i>Rule Base</i> .....	13
2.3.4 Mekanisme Inferensi .....	13
2.3.5 <i>Defuzzification</i> .....	14
2.3.6 Konstanta <i>Compensator fuzzy</i> .....	15

Bab 3 Metode Penelitian .....	16
3.1 Alur Penelitian .....	16
3.2 Peralatan.....	17
3.2.1 Perangkat Keras .....	17
3.2.2 Perangkat Lunak.....	21
3.3 Perancangan .....	21
3.3.1 Merangkai <i>Controller, Driver, Encoder, dan Actuator</i> (Motor dan <i>Ball-Screw</i> ).....	21
3.3.2 Identifikasi Hubungan Nilai PWM Arduino dengan Tegangan yang Keluar dari Escon <i>Driver</i> .....	25
3.3.3 Pembuatan Blok Diagram untuk Eksperimen <i>Open-loop</i> .....	27
3.3.4 Konstruksi NCT.....	28
3.3.5 <i>Smoothing</i> NCT .....	29
3.3.6 Fungsi Transfer.....	31
3.3.7 Desain <i>Compensator PI</i> .....	32
3.3.8 Desain <i>Compensator fuzzy</i> .....	34
3.4 Pengujian dengan Simulasi Komputer dan Eksperimen objek.....	39
Bab 4 Hasil dan Pembahasan .....	42
4.1 Analisis terhadap <i>Compensator PI</i> .....	42
4.2 Analisis terhadap <i>Compensator Fuzzy</i> .....	46
Bab 5 Penutup.....	49
5.1 Kesimpulan .....	49
5.2 Saran .....	49
Daftar Pustaka .....	50
Lampiran 1 <i>Datasheet</i> Maxon DC Motor 26 mm .....	52
Lampiran 2 Pengaturan Parameter Escon <i>Driver</i> .....	53

## DAFTAR TABEL

Tabel 2-1 Fuzzy Rule.....	13
Tabel 3-1 Hasil pengukuran tegangan.....	26
Tabel 3-2 Nilai $\omega_n$ dan $\zeta$ .....	33
Tabel 3-3 Nilai $K_p$ dan $K_i$ .....	33
Tabel 3-4 Parameter <i>membership function</i> dalam bentuk variabel.....	35
Tabel 3-5 Parameter <i>membership function</i> dalam bentuk nilai.....	35
Tabel 3-6 Fuzzy Rule Base.....	37
Tabel 4-1 Karakteristik respon objek menggunakan <i>compensator PI</i> .....	46
Tabel 4-2 <i>Steady state error</i> 0% PI.....	46
Tabel 4-3 Karakteristik respon objek menggunakan <i>compensator fuzzy</i> .....	48
Tabel 4-4 <i>Steady state error</i> 0% fuzzy.....	48

## DAFTAR GAMBAR

Gambar 2-1 Struktur sistem kendali NCTF.....	6
Gambar 2-2 NCT dan pergerakan objek.....	7
Gambar 2-3 Input <i>Stepwise</i> dan Respon Objek.....	7
Gambar 2-4 NCT ( <i>Nominal characterisic trajectory</i> ).....	8
Gambar 2-5 <i>Stable region</i> sistem kendali NCTF .....	9
Gambar 2-6 Struktur <i>Compensator fuzzy</i> .....	10
Gambar 2-7 Membership function <b>up</b> .....	11
Gambar 2-8 Perbesaran Membership function <b>up</b> .....	11
Gambar 2-9 <i>Membership function</i> input <b>ui</b> .....	12
Gambar 2-10 <i>Membership function</i> output <b>u</b> .....	12
Gambar 3-1 Diagram Alir Penelitian.....	16
Gambar 3-2 Diagram alir proses desain <i>compensator</i> .....	17
Gambar 3-3 Arduino UNO .....	18
Gambar 3-4 <i>Driver</i> motor ESCON.....	18
Gambar 3-5 <i>Maxon</i> motor .....	19
Gambar 3-6 <i>Linear ball-screw</i> .....	19
Gambar 3-7 <i>Linear encoder</i> .....	19
Gambar 3-8 Posisi <i>encoder</i> pada <i>linear ball screw</i> .....	19
Gambar 3-9 Kabel <i>jumper</i> .....	20
Gambar 3-10 <i>Project board</i> .....	20
Gambar 3-11 <i>Power supply</i> .....	20
Gambar 3-12 Pengaturan pin <i>Power</i> dan Motor pada <i>driver</i> motor Escon 50/5..	22
Gambar 3-13 Pengaturan pin <i>Digital</i> pada <i>driver</i> motor Escon 50/5.....	22
Gambar 3-14 Hasil rangkaian <i>hardware</i> .....	23
Gambar 3-15 Keterangan warna kabel <i>linear encoder</i> RLS LM10IC001AA10K12 .....	23
Gambar 3-16 Rangkaian <i>controller, driver, encoder, dan actuator</i> (motor).....	24
Gambar 3-17 Alur kerja rangkaian perangkat keras.....	24
Gambar 3-18 Perbesaran gambar hubungan <i>work block</i> dengan motor .....	25
Gambar 3-19 Blok diagram pengukuran tegangan dengan variasi PWM.....	25

Gambar 3-20 Grafik hasil pengukuran tegangan.....	27
Gambar 3-21 Blok diagram untuk eksperimen <i>open-loop</i> .....	27
Gambar 3-22 Grafik eksperimen <i>open-loop</i> .....	28
Gambar 3-23 Nominal Characteristic Trajectory – <i>Original</i> .....	29
Gambar 3-24 Aplikasi <i>Curve Fitting</i> di Matlab R2015a.....	30
Gambar 3-25 Nominal Characteristic Trajectory – <i>Smoothing Spline 3</i> .....	31
Gambar 3-26 Parameter NCT – <i>Smoothing Spline 3</i> .....	31
Gambar 3-27 Verifikasi fungsi transfer .....	32
Gambar 3-28 Grafik hasil simulasi verifikasi fungsi transfer .....	32
Gambar 3-29 Deskripsi <i>Compensator fuzzy</i> .....	34
Gambar 3-30 Input <b><i>up</i></b> .....	35
Gambar 3-31 Input <b><i>ui</i></b> .....	36
Gambar 3-32 Output <b><i>u</i></b> .....	36
Gambar 3-33 Memasukkan parameter <b><i>up</i></b> ke “fls Editor” .....	36
Gambar 3-34 Memasukkan parameter <b><i>ui</i></b> ke “fls Editor” .....	37
Gambar 3-35 Memasukkan parameter <b><i>u</i></b> ke “fls Editor” .....	37
Gambar 3-36 <i>Matching</i> berdasarkan <i>Rule Base</i> .....	38
Gambar 3-37 Simulasi Komputer – <i>Compensator PI</i> .....	40
Gambar 3-38 Simulasi Komputer – <i>Compensator fuzzy</i> .....	40
Gambar 3-39 Eksperimen objek – <i>Compensator PI</i> .....	41
Gambar 3-40 Grafik konversi tegangan menjadi PWM.....	41
Gambar 3-41 Eksperimen objek – <i>Compensator fuzzy</i> .....	41
Gambar 4-1 Grafik jarak-waktu simulasi komputer – <i>compensator PI</i> – input 0.2 .....	42
Gambar 4-2 Grafik NCTF simulasi komputer – <i>compensator PI</i> – input 0.2 .....	42
Gambar 4-3 Grafik jarak-waktu simulasi komputer – <i>compensator PI</i> – input 2	43
Gambar 4-4 Grafik NCTF simulasi komputer – <i>compensator PI</i> – input 2 .....	43
Gambar 4-5 Grafik jarak-waktu eksperimen objek – <i>compensator PI</i> – input 0.2 .....	44
Gambar 4-6 Grafik NCTF eksperimen objek – <i>compensator PI</i> – input 0.2 .....	44
Gambar 4-7 Grafik jarak-waktu eksperimen objek – <i>compensator PI</i> – input 2 .	44
Gambar 4-8 Grafik NCTF eksperimen objek – <i>compensator PI</i> – input 2 .....	45

Gambar 4-9 Blok diagram hasil perbaikan.....	47
Gambar 4-10 Grafik jarak-waktu simulasi komputer – <i>compensator fuzzy</i> – input 0.2 .....	47
Gambar 4-11 Grafik jarak-waktu simulasi komputer – <i>compensator fuzzy</i> – input 2 .....	47

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kendali posisi *point-to-point* (PTP) menjadi salah satu bidang yang diteliti lebih lanjut dikarenakan implementasi yang cukup luas, salah satunya adalah mesin berbasis *computer numerical control* (CNC). Perancangan sistem kendali posisi dengan tingkat ketelitian tinggi tergantung pada beberapa parameter, diantaranya adalah metode pengendalian, kualitas *controller-card*, kualitas *actuator*, dan ketelitian *encoder*.

Sistem kendali PTP dirancang untuk mengendalikan *actuator* bergerak dari titik awal ke titik akhir. Hasil yang diharapkan adalah *actuator* dapat mencapai titik akhir dengan cepat dan tepat. Metode pengendalian akan menentukan seberapa cepat dan seberapa tepat *actuator* mencapai titik akhir. Performa metode pengendalian berada di dalam batas kemampuan *controller*, *actuator*, dan *encoder*. Tingkat akurasi *controller* mengirimkan sinyal ke *actuator* sehingga dapat bergerak sesuai sinyal, tingkat akurasi *actuator* menerima sinyal dari *controller*, dan tingkat kepresisian *encoder* membaca pergerakan *actuator* secara *realtime*.

Metode konvensional dalam merancang sistem kendali adalah dengan membuat fungsi transfer<sup>1</sup> dari sebuah objek kendali. Hal tersebut mengharuskan pendesain untuk mengidentifikasi data fisik objek kendali. Misalkan objek kendali adalah motor DC, identifikasi terhadap data fisik motor DC ini hanya dapat dilakukan dengan melihat *datasheet* dari perusahaan pembuat motor DC tersebut. Permasalahan muncul ketika perusahaan pembuat motor DC ini tidak mengeluarkan *datasheet*. Permasalahan lain juga muncul ketika motor DC dalam kondisi telah terpakai untuk waktu yang lama, yang artinya kondisinya sudah berbeda dengan yang tertera di *datasheet*. Perbedaan kondisi ini, pada terminologi sistem kendali dinamakan karakteristik non-linier. *Friction* dan saturasi adalah karakteristik non-linear yang dapat menyebabkan sistem tidak stabil. Sistem PTP

---

<sup>1</sup> Fungsi transfer adalah representasi matematis dari sebuah objek kendali.



juga dapat terpengaruhi oleh ketidak-pastian parameter, salah satunya adalah variasi parameter yang disebabkan oleh variasi inersia. Variasi inersia disebabkan karena variasi beban (Wahyudi, Sato, & Shimokohbe, 2003). Sistem PTP memerlukan sistem kendali yang dapat menyelesaikan permasalahan-permasalahan ini, oleh karena itu persyaratan sistem PTP diantaranya adalah akurasi yang tinggi, respon cepat, *overshoot* yang kecil atau nol, dan *robustness* terhadap ketidak-pastian objek. Dengan demikian, pada penelitian ini sistem kendali *nominal characteristic trajectory following* (NCTF) digunakan untuk mengatasi permasalahan-permasalahan tersebut (Sato Kaiji, Wahyudi, & Shimokohbe Akira, 2001).

Alasan NCTF dapat dijadikan sebagai solusi permasalahan-permasalahan tersebut adalah karena proses konstruksi NCTF. Konstruksi NCTF didasarkan pada prinsip karakteristik aktuator. Karakteristik aktuator didapat saat melakukan eksperimen *open-loop*, yaitu ketika deselerasi aktuator dari performa maksimal menuju performa nol. Hasil yang didapat pada saat deselerasi aktuator digunakan untuk mengkonstruksi *nominal characteristic trajectory* (NCT). Penjelasan lebih rinci mengenai proses ini dapat dilihat pada bagian 2.1.

Sampai saat ini, banyak jenis *controller* yang telah diajukan dan dievaluasi untuk sistem posisi, diantaranya adalah *controller* dengan *disturbance observer* (C. J. Kempf, 1996; Masayoshi Tomizuka, 1996; T. Umeno, T. Kaneko, & Y. Hori, 1993), *time-optimal controllers* (C. J. Kempf & S. Kobayashi, 1999; M. L. Workman, R. L. Kosut, & G. F. Franklin, 1987; Min-Ho Park & Chung-Yeun Won, 1991; WU Shang-Teh & FU Jeng-Young, 1998), dan *sliding mode controllers* (S. Sankaranarayanan & F. Khorrami, 1997; Y. Fujimoto & A. Kawamura, 1995). *Controller-controller* ini dapat menghasilkan kinerja yang bagus dalam sistem posisi, tetapi masih membutuhkan keahlian di bidang kendali pada tingkatan *expert*.

Perancangan sistem kendali mensyaratkan dua jenis perangkat, yakni perangkat keras dan perangkat lunak. Selain objek kendali, perangkat keras lain yang utama adalah *controller-card*. *Controller-card* digunakan untuk mengkonversi “hasil desain sistem kendali” yang telah dirancang pada perangkat lunak, sehingga hasil konversi dapat menggerakkan objek kendali. Maksud dari

“hasil desain sistem kendali” adalah blok diagram yang telah diterjemahkan menjadi bahasa mesin<sup>2</sup>. Perangkat lunak yang digunakan pada perancangan sistem kendali adalah perangkat lunak yang memiliki fasilitas untuk merancang blok diagram<sup>3</sup>.

## 1.2 Rumusan Masalah

Telah dijelaskan pada bagian 1.1 bahwa untuk mendapatkan *controller* yang bagus pada sistem PTP, maka persyaratan yang harus terpenuhi adalah akurasi yang tinggi, respon cepat, *overshoot* yang kecil atau nol, dan *robustness* terhadap ketidak-pastian objek. Oleh karenanya digunakanlah NCTF sebagai *controller*. Secara mendasar, NCTF terdiri dari dua bagian, *nominal characteristic trajectory* (NCT) dan *compensator*. Kinerja NCTF bergantung pada *compensator*, sehingga diperlukan *compensator* yang dapat memenuhi persyaratan sistem PTP. Apa saja *compensator* yang dapat digunakan untuk memenuhi persyaratan tersebut? Jika telah mengetahui *compensator* yang digunakan, selanjutnya adalah bagaimana kinerja *controller* pada setiap *compensator*?

Proses mendesain NCTF *controller* hingga tahap implementasi memerlukan perangkat lunak untuk mendesain blok diagram dan perangkat keras *controller-card*. Pada penelitian ini akan digunakan perangkat lunak yang *open-source*. Kemudian *controller-card* yang digunakan juga yang bersifat *open-source* dan terjangkau secara ekonomi. Bagaimana kinerja *controller* pada sistem PTP menggunakan perangkat lunak dan perangkat keras yang bersifat *open-source* ini?

## 1.3 Batasan Masalah

Jenis *compensator* yang akan digunakan pada penelitian ini adalah *compensator* (proporsional+integral) PI dan *compensator fuzzy*. *Compensator PI* digunakan karena *gain* proporsional dapat mempercepat respon *rise time* dan *settling time*, sedangkan *gain* integral dapat meminimalisir *error steady state*.

---

<sup>2</sup> Bahasa mesin adalah bahasa yang dimengerti oleh komputer, yang terdiri dari bilangan biner (1 dan 0), yang digunakan untuk komunikasi perangkat elektronik.

<sup>3</sup> Blok diagram adalah susunan blok yang merepresentasikan sistem kendali. Contoh blok diagram dapat dilihat pada Gambar 3-21.

*Compensator fuzzy* digunakan karena dapat menggunakan konstanta proporsional dan integral.

Pada bagian 1.1 telah dijelaskan bahwa sistem kendali memerlukan perangkat lunak untuk mendesain blok diagram yang bersifat *open-source* dan perangkat keras *controller-card* yang bersifat *open-source* dan terjangkau secara finansial. Oleh karena itu, pada penelitian ini perangkat lunak untuk mendesain blok diagram yang digunakan adalah Scilab 5.5.1 dan *controller-card* yang digunakan adalah Arduino Uno.

Selain *controller-card*, perangkat keras yang digunakan pada penelitian ini adalah *driver* untuk *actuator*, *actuator*, dan sensor. Perangkat-perangkat keras tersebut tidak bersifat variabel, artinya hanya akan digunakan satu jenis *driver*, satu jenis *actuator*, dan satu jenis sensor. Mengenai jenisnya secara spesifik dapat dilihat pada bagian 3.2.

#### **1.4 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah :

1. Merancang dan mengimplementasikan *controller* NCTF dengan tipe *compensator* yang digunakan adalah *compensator PI* dan *compensator fuzzy*, untuk operasi PTP pada sistem linier menggunakan *ball-screw*.
2. Mengetahui kinerja controller yang menggunakan *compensator PI* dan *compensator fuzzy* dengan menganalisis *settling time*, *rise time*, dan *error steady state* melalui respon objek pada simulasi komputer maupun eksperimen objek.

#### **1.5 Manfaat Penelitian**

Hasil penelitian ini dapat digunakan sebagai acuan untuk menentukan mana *compensator* yang lebih baik antara fuzzy dan PI untuk penerapan yang berada dalam ruang batasan masalah yang telah dijelaskan pada bagian 1.3.

#### **1.6 Sistematika Penulisan**

Laporan penelitian ini terdiri dari lima bab. Bab pertama berisi mengenai mengapa penelitian ini dilakukan, ruang lingkup penelitian, dan tujuan dari

penelitian. Bab kedua berisi mengenai teori dan penelitian terdahulu yang menjadi referensi penelitian ini. Bab ketiga membahas mengenai proses penelitian yang dilakukan, mulai dari alur penelitian, peralatan yang digunakan, proses perancangan, dan pengujian. Bab keempat berisi hasil pengujian dan analisis terhadap hasil pengujian. Terakhir, bab kelima membahas mengenai kesimpulan penelitian dan saran.

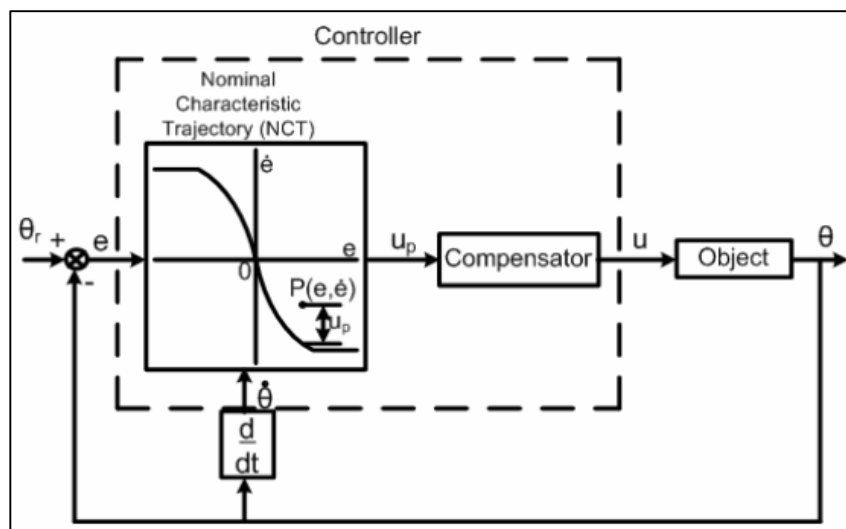
## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Konsep NCTF

Struktur konsep NCTF dapat dilihat pada Gambar 2-1. *Controller* NCTF terdiri dari dua bagian, yakni NCT dan *Compensator* (Wahyudi dkk., 2003). NCT adalah grafik yang dibentuk berdasarkan respon objek dan *compensator* adalah bagian yang berfungsi mengolah data yang berasal dari NCT kemudian dikirimkan ke objek kendali. *Controller* NCTF dapat bekerja dengan dua asumsi berikut:

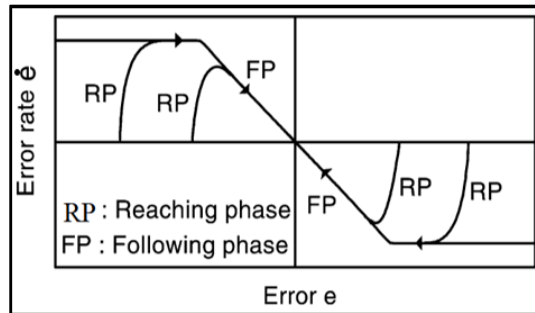
1. *Actuator* sebagai objek kendali adalah motor DC atau AC.
2. Sistem posisi PTP telah dijelaskan pada bagian 1.1 di mana input (titik akhir)  $\theta_r$  adalah konstanta, maka  $\dot{\theta}_r = 0$ .



**Gambar 2-1 Struktur sistem kendali NCTF**

**Sumber :** (Purtojo, Wahyudi, Akmeliawati, & Shafie, 2009, gbr. 1)

Objek bergerak mengikuti lintasan NCT dan berakhir di *origin* NCT seperti ditunjukkan pada Gambar 2-2. Sinyal  $u_p$  adalah selisih antara titik koordinat aktual dan titik koordinat hasil eksperimen *open-loop* pada grafik NCT. Sinyal  $u_p$  akan menjadi nol apabila pergerakan objek secara persis mengikuti lintasan NCT. Gambar 2-2 menunjukkan pergerakan objek. Pada *reaching phase*, objek bergerak mencapai lintasan NCT. Pada *following phase*, objek mengikuti lintasan NCT dan berakhir di *origin*, yang mana *origin* merepresentasikan bahwa objek telah mencapai titik akhir.



**Gambar 2-2 NCT dan pergerakan objek**

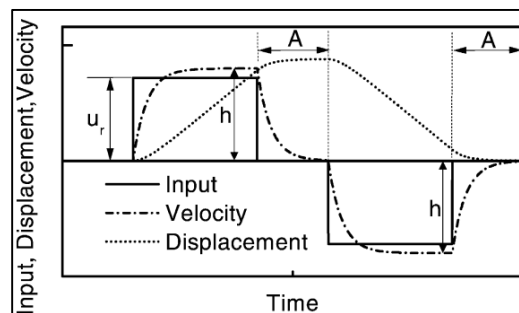
Sumber : (Wahyudi dkk., 2003, gbr. 2)

*Controller* NCTF dikonstruksi berdasarkan eksperimen *open-loop* terhadap objek kendali. Proses konstruksi ini terdiri dari tiga langkah, eksperimen *open-loop*, konstruksi NCT, dan desain *compensator*. *Compensator* yang didesain ada dua, yaitu *compensator PI* dan *compensator fuzzy*.

### 2.1.1 Eksperimen *Open-loop*

Diadaptasi dari Wahyudi dkk.(2003) NCT dikonstruksi dengan melakukan eksperimen *open-loop* terhadap objek kendali. Eksperimen *open-loop* dilakukan dengan memberikan input *stepwise* terhadap objek kendali. Gambar 2-3 menunjukkan input *stepwise* terhadap objek, respon kecepatan objek dan respon jarak objek.

Input  $u_r$  adalah nilai input terhadap objek,  $A$  adalah area batas kurva deselerasi (*deceleration*) objek dari maksimum menuju nol, dan  $h$  adalah kecepatan maksimum objek. Kecepatan dan jarak akan digunakan untuk mengkonstruksi NCT.



**Gambar 2-3 Input *Stepwise* dan Respon Objek**

Sumber : (Wahyudi dkk., 2003, gbr. 3.(a))

## 2.1.2 Konstruksi NCT

NCT dikonstruksi berdasarkan kurva respon kecepatan objek dan respon jarak objek pada area  $A$  seperti pada Gambar 2-4. Parameter  $m$  pada NCT adalah gradien (*slope*) di dekat *origin*. Karakteristik-karakteristik seperti *friction* diabaikan, dan diasumsikan bahwa objek mengikuti sistem orde kedua (Wahyudi dkk., 2003, bag. 3.2) :

$$\frac{\Theta(s)}{U(s)} = K \frac{\alpha}{s(s + \alpha)} \quad \text{Pers. (1)}$$

di mana  $\Theta(s)$  merepresentasikan jarak objek,  $U(s)$  merepresentasikan input terhadap objek, dan,  $K$  dan  $\alpha$  merupakan parameter objek tersimplifikasi.

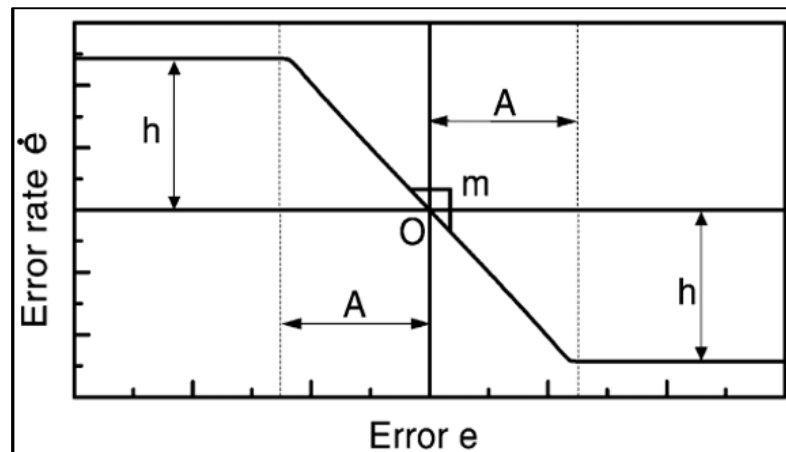
Gradien (*slope*)  $m$  pada NCT berhubungan dengan parameter objek tersimplifikasi  $\alpha$ . NCT pada Gambar 2-4 menunjukkan

$$\frac{d\dot{e}}{de} = m \quad \text{Pers. (2)}$$

$$\frac{d\dot{e}}{de} = -\alpha \quad \text{Pers. (3)}$$

$$m = -\alpha \quad \text{Pers. (4)}$$

$$K = \frac{h}{u_r} \quad \text{Pers. (5)}$$



Gambar 2-4 NCT (*Nominal characteristic trajectory*)

Sumber : (Wahyudi dkk., 2003, gbr. 3.(b))

## 2.2 Desain Compensator PI

Compensator PI digunakan untuk mengontrol pergerakan objek sehingga dapat mengikuti NCT yang dapat dinyatakan dengan

$$G_c(s) = K_p + \frac{K_i}{s} \quad \text{Pers. (6)}$$

$K_p$  merepresentasikan konstanta proporsional dan  $K_i$  merepresentasikan konstanta integral (Wahyudi dkk., 2003, bag. 3.3.1 dan 3.3.2). Sinyal  $u_p$  pada Gambar 2-1 dapat dinyatakan dengan

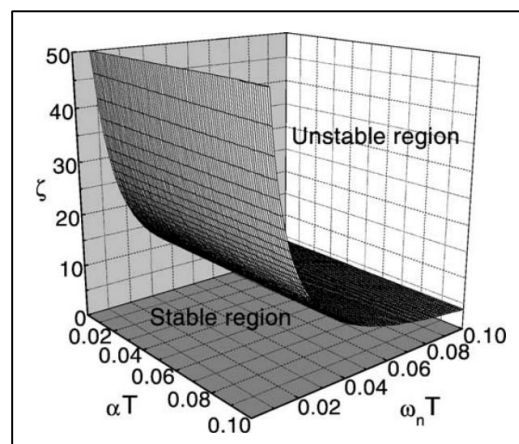
$$u_p = \dot{e} + \alpha e = \alpha e - \dot{\theta} \quad \text{Pers. (7)}$$

Parameter compensator PI ( $K_p$  dan  $K_i$ ) dapat dinyatakan sebagai fungsi dari frekuensi natural  $\omega_n$  dan *damping ratio*  $\zeta$  sebagai berikut

$$K_p = \frac{2\zeta\omega_n}{\alpha K} \quad \text{Pers. (8)}$$

$$K_i = \frac{\omega_n^2}{\alpha K} \quad \text{Pers. (9)}$$

Frekuensi natural  $\omega_n$  dan *damping ratio*  $\zeta$  ditentukan berdasarkan *stable region* pada grafik yang ditunjukkan oleh Gambar 2-5. Parameter  $T$  pada Gambar 2-5 adalah periode *sampling*.



Gambar 2-5 *Stable region* sistem kendali NCTF

Sumber : (Wahyudi dkk., 2003, gbr. 5)

Diadaptasi dari Wahyudi dkk.(2003, bag. 3.4), prosedur desain sistem kendali NCTF menggunakan *compensator PI* dapat diringkas menjadi :

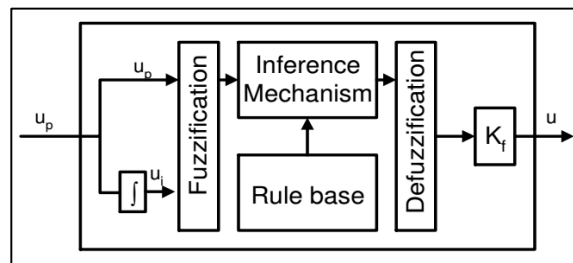
1. Mengkonstruksi input *stepwise* seperti yang ditunjukkan oleh Gambar 2-3 dengan mempertimbangkan *range* nilai input *actuator*.



2. Menjalankan objek dengan input *stepwise* dan mengukur jarak dan kecepatan respon objek.
3. Mengkonstruksi NCT berdasarkan jarak dan kecepatan yang terukur dan menentukan nilai  $h$  dan  $m$ .
4. Memilih parameter *compensator PI* ( $\omega_n$  dan  $\zeta$ ) berdasarkan *stable region* pada Gambar 2-5.
5. Mendapatkan nilai  $K_p$  dan  $K_i$  menggunakan Pers. (8) dan Pers. (9).

### 2.3 Desain *Compensator fuzzy*

Sinyal  $u_p$  yang keluar dari NCT menjadi input *compensator fuzzy*. Sinyal  $u_p$  dibagi menjadi dua, sinyal  $u_p$  itu sendiri dan  $u_i$  yang mana merupakan integral dari  $u_p$ . Seluruh parameter yang akan digunakan pada *compensator fuzzy* berasal dari NCT. Struktur *compensator fuzzy* dapat dilihat pada Gambar 2-6.



**Gambar 2-6 Struktur *Compensator fuzzy***

Sumber : (Purtojo dkk., 2009, gbr. 3)

Dirangkum dari Purtojo dkk.(2009) dan Passino & Yurkovich (1998) proses desain *compensator fuzzy* terdiri dari lima langkah, yaitu :

1. Konstruksi *Membership Function*
2. *Fuzzification (fuzzification)*
3. Konstruksi *Rule Base*
4. Mekanisme inferensi (*inference mechanism*)
5. *Defuzzification (defuzzification)*

#### 2.3.1 *Membership Function*

*Membership function* dikonstruksi berdasarkan informasi NCT. Terdapat tiga variabel *compensator fuzzy*, yaitu  $u_p$  dan  $u_i$  sebagai input dan  $u$  sebagai output. Di setiap variabel *compensator fuzzy* ada tiga *membership function*, yaitu

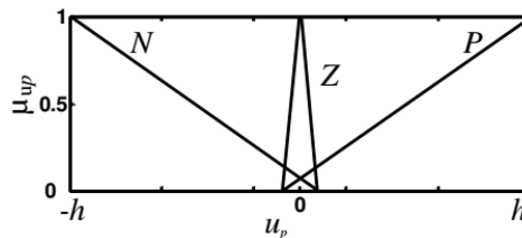
Negatif (N), Zero (Z), dan Positif (P). Setiap *membership function* (N, Z, dan P) pada setiap variabel *compensator fuzzy* mengindikasikan kepastian (*certainty*) terhadap suatu variabel (Purtojo dkk., 2009). Misalkan nilai  $u_p = 0$ , maka nilai  $\mu_Z(u_p) = 1$  yang berarti bahwa  $u_p$  100% dalam kondisi Z (*zero*).

### 2.3.1.1 Membership Function Input $u_p$

Diadopsi dari Purtojo dkk.(2009), *range* input  $u_p$  adalah  $\pm h$ . *Membership function* Z ditentukan berdasarkan resolusi dari sensor, dalam penelitian ini sensornya adalah *linear encoder*. Nilai  $u_p$  dapat dinyatakan seperti pada Pers. (7), maka *membership function* Z dapat dinyatakan dengan

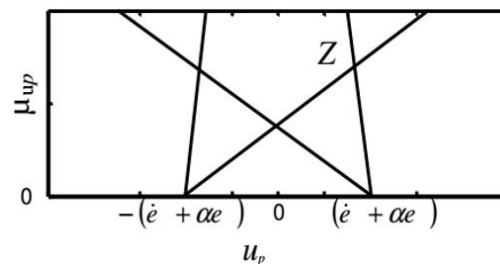
$$\mu_Z(u_p) = \pm(\dot{e} + \alpha e) \quad \text{Pers. (10)}$$

yang mana  $\alpha = -m$ ,  $e$  adalah kemungkinan *error* terkecil dengan batas terkecil ditentukan oleh resolusi sensor, dan  $\dot{e}$  adalah *error rate* pada  $e$ .



**Gambar 2-7** Membership function  $u_p$

Sumber : (Purtojo dkk., 2009, gbr. 4.(a))



**Gambar 2-8** Perbesaran Membership function  $u_p$

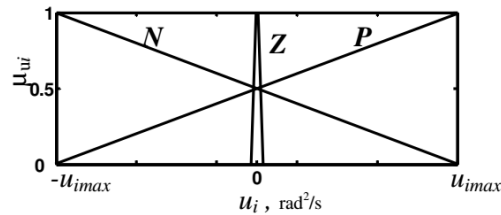
Sumber : (Purtojo dkk., 2009, gbr. 4.(b))

### 2.3.1.2 Membership Function Input $u_i$

*Range* input  $u_i$  adalah  $\pm u_{i_{max}}$  yang mana  $u_{i_{max}}$  merupakan integral dari  $u_p$ . *Range* ini dinyatakan dengan

$$u_{i_{max}} = \int u_p de \cong \frac{1}{2} Ah \quad \text{Pers. (11)}$$

di mana  $A$  dan  $h$  didapat dari NCT pada Gambar 2-4. *Range membership function* N dan P dapat dilihat pada Gambar 2-9, sedangkan *membership function* Z adalah sama dengan *membership function* Z pada input  $u_p$  (Purtojo dkk., 2009).

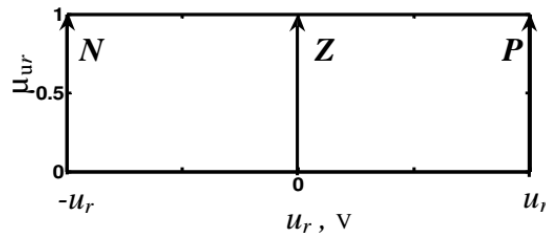


Gambar 2-9 *Membership function* input  $u_i$

Sumber : (Purtojo dkk., 2009, gbr. 5)

### 2.3.1.3 *Membership Function* Output $u$

*Range* output  $u$  adalah  $\pm u_r$ , dengan jenis *membership function* (N, Z, dan P) adalah *singleton* (Purtojo dkk., 2009). Output  $u_r$  adalah input terhadap *actuator*<sup>4</sup> (objek kendali).



Gambar 2-10 *Membership function* output  $u$

Sumber : (Purtojo dkk., 2009, gbr. 6)

### 2.3.2 *Fuzzification*

*Fuzzification* adalah proses untuk memperoleh sebuah nilai dari variabel input dan menemukan nilai numerik dari *membership function* yang didefinisikan untuk variabel tersebut (Passino & Yurkovich, 1998). Misalkan ketika nilai  $u_p$  menempati *membership function* N, Z, dan P, maka

$$\mu_N(u_p) \neq \mu_Z(u_p) \neq \mu_P(u_p)$$

---

<sup>4</sup> *Actuator* dalam penelitian ini adalah motor DC, yang secara keseluruhan merupakan *linear ball screw* yang digerakkan oleh motor DC. Nilai *range* motor DC dapat dilihat di *datasheet* bagian Nominal Voltage pada Lampiran 1.

### 2.3.3 Rule Base

*Compensator fuzzy* didesain untuk memaksa objek agar dapat mencapai NCT dan setelah itu mengikuti NCT. Agar objek dapat mencapai NCT, maka ketika nilai  $u_p$  positif dan nilai  $u_i$  juga positif, maka output terhadap objek juga harus positif. Ketika nilai  $u_p$  negatif dan nilai  $u_i$  juga negatif maka output terhadap objek juga harus negatif.

Kemudian untuk mengendalikan agar objek dapat mengikuti NCT, maka ketika nilai  $u_p$  positif dan nilai  $u_i$  adalah nol, maka output harus positif. Ketika nilai  $u_p$  negatif dan nilai  $u_i$  adalah nol, maka output adalah negatif. Ketika nilai  $u_p$  positif dan nilai  $u_i$  negatif, maka  $u$  harus nol. Begitu juga sebaliknya ketika nilai  $u_p$  negatif dan nilai  $u_i$  positif maka  $u$  juga harus nol.

Terakhir, ketika objek kendali telah mencapai titik akhir, yang berarti bahwa objek berakhir di *origin* NCT, mengindikasikan bahwa ketika nilai  $u_p$  nol, dan  $u_i$  nol, maka output harus nol.

Kriteria dari nilai  $u_p$ ,  $u_i$ , dan  $u$  pada *compensator fuzzy* ini dapat dinyatakan dengan tabel yang ditunjukkan oleh Tabel 2-1.

Tabel 2-1 Fuzzy Rule

Fuzzy Rule				
Object		$u_p$		
		Positive	Zero	Negative
$u_i$	Positive	P	Z	Z
	Zero	P	Z	N
	Negative	Z	Z	N

### 2.3.4 Mekanisme Inferensi

Diadaptasi dari Passino & Yurkovich (1998), mekanisme inferensi memiliki dua langkah dasar, yaitu *matching* dan *inference step*. *Matching* adalah menentukan mana *rule* yang aktif sesuai dengan situasi sekarang (*current situation*). *Inference step* adalah mendapatkan kesimpulan menggunakan input sekarang (*current input*) dan informasi dari *rule base*.

### 2.3.4.1 Matching

Diasumsikan *membership function* N dan Z pada input  $u_p$  aktif, dengan kepastian  $N = 0.1$  dan  $Z = 0.6$ . Dan *membership function* N pada input  $u_i$  aktif, dengan kepastian  $N = 1$ . Maka terdapat dua kombinasi *rule* :

1. **Jika  $u_p = N$  dan  $u_i = N$ , maka  $u = N$**
2. **Jika  $u_p = Z$  dan  $u_i = N$ , maka  $u = Z$**

### 2.3.4.2 Inference Step

*Inference step* adalah pengambilan kesimpulan dari kondisi input sekarang berdasarkan *rule base*. Pertama, langkah yang harus dilakukan sebelum menuju ke kesimpulan adalah mendapatkan nilai *certainty* dari premis setiap *rule* yang telah didapat pada langkah *matching*.

Pada *rule* 1, dengan menggunakan “maksimum” untuk merepresentasikan premis *rule* 1 adalah dengan menggunakan rumus

$$\mu_{\text{premis}(1)} = \max\{0.1, 1\} = 0.1$$

Langkah selanjutnya adalah mendapatkan kesimpulan dari premis 1 dengan menggunakan “minimum”

$$\mu_{(1)}(u) = \min\{0.1, \mu_N(u)\}$$

Pada *rule* 2 sama dengan *rule* 1, menggunakan maksimum untuk premis (disebut juga *aggregation*) dan menggunakan minimum untuk kesimpulan (disebut juga *implication*),

$$\mu_{\text{premis}(2)} = \max\{0.6, 1\} = 0.6$$

$$\mu_{(2)}(u) = \min\{0.6, \mu_Z(u)\}$$

### 2.3.5 Defuzzification

Diadaptasi dari Purtojo dkk. (2009), setelah mendapatkan nilai *certainty* dari semua *rule*, selanjutnya adalah mengkonversi nilai *certainty* yang telah didapatkan tersebut menjadi nilai yang dapat digunakan oleh objek kendali. Langkah ini disebut *defuzzification*. Pada penelitian ini, metode yang digunakan untuk mengkonversi nilai tersebut adalah metode *Center of Area* (COA)

$$u = \frac{\int u\mu(u)du}{\int \mu(u)du} \quad \text{Pers. (12)}$$

dengan nilai  $\mu(u)$  dicari menggunakan “maksimum”

$$\mu(u) = \max\{\mu_1(u), \mu_2(u), \dots, \mu_i(u)\} \quad \text{Pers. (13)}$$

### 2.3.6 Konstanta *Compensator fuzzy*

Komponen terakhir dari *compensator fuzzy* adalah konstanta *compensator fuzzy*  $K_f$ , yang sama dengan nilai  $K$  pada Pers. (5)

$$K_f = \frac{h}{u_r} \quad \text{Pers. (14)}$$

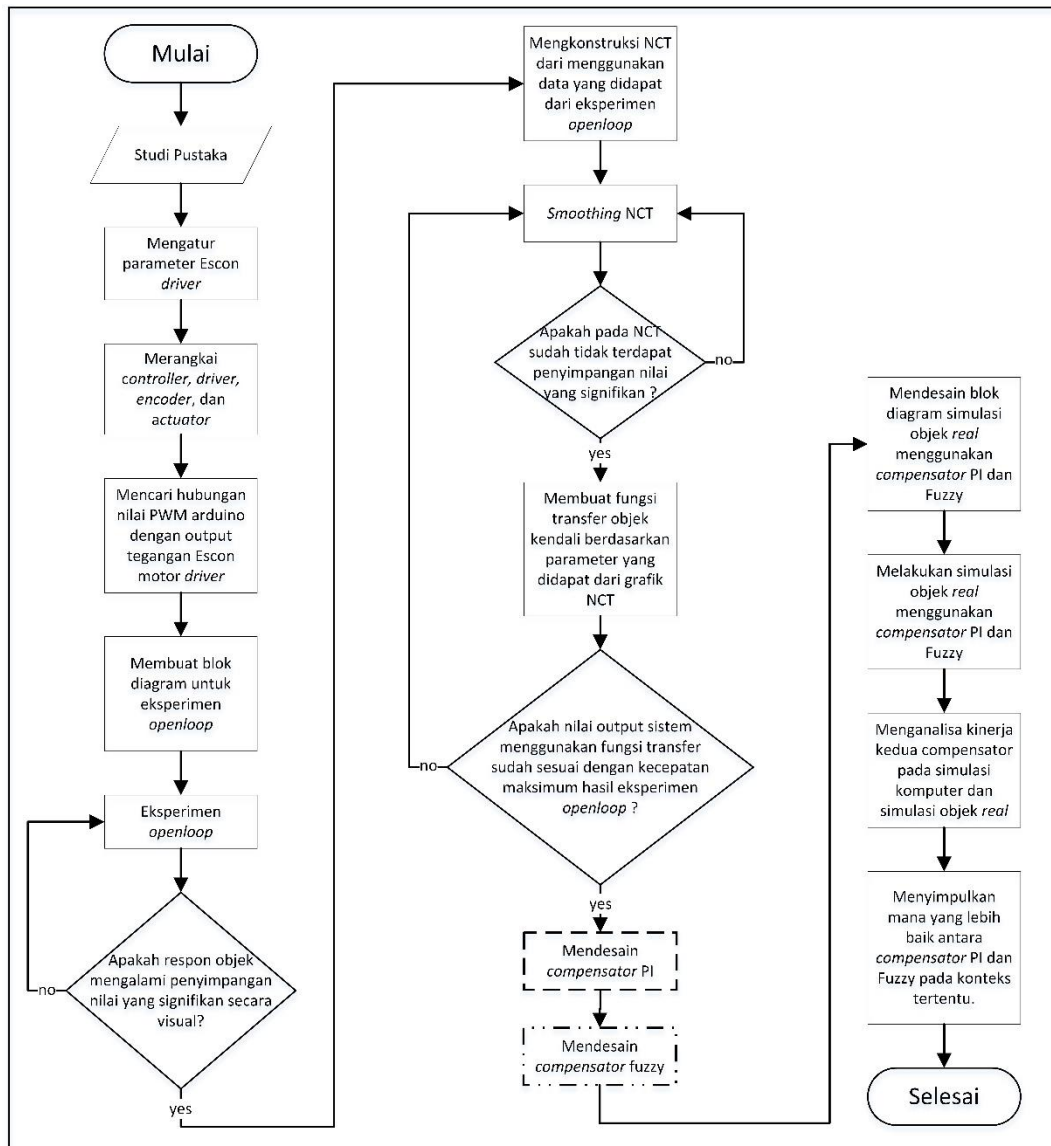
yang mana  $h = -m$  dan  $u_r$  adalah input terhadap objek kendali.

## BAB 3

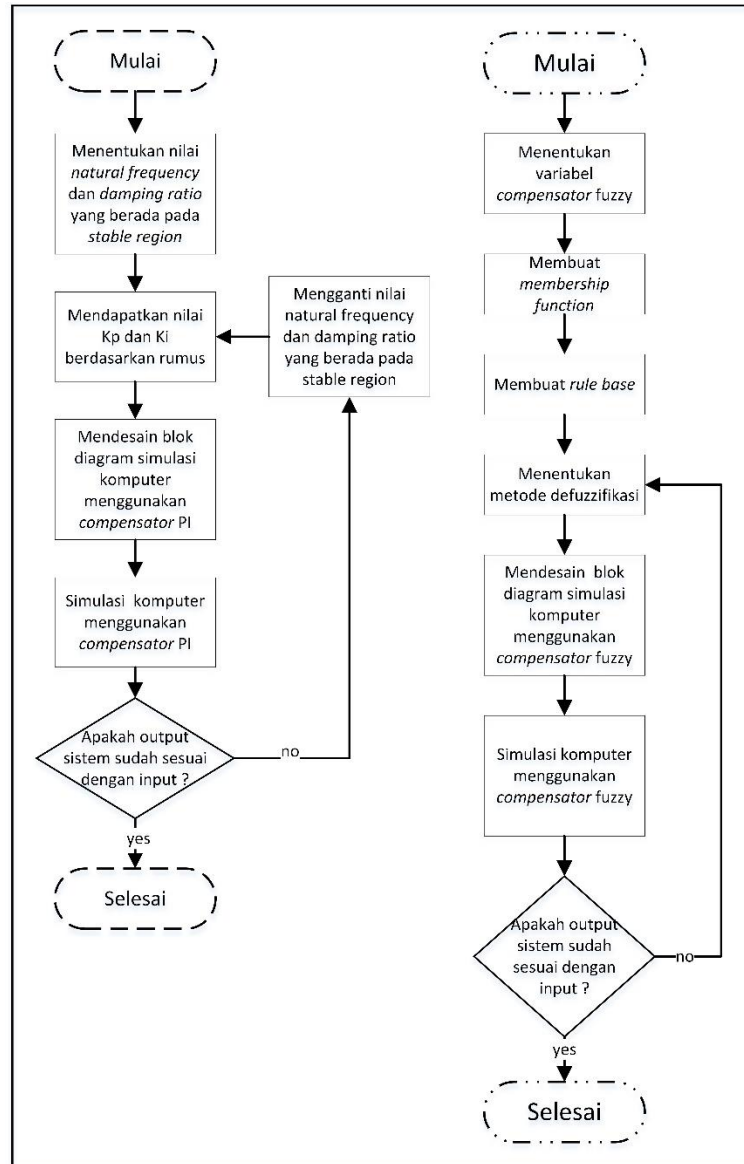
### METODE PENELITIAN

#### 3.1 Alur Penelitian

Gambar 3-1 menggambarkan alur penelitian secara *general*. Gambar 3-2 adalah sub-proses dari Gambar 3-1, yang disesuaikan dengan garis luar blok *flowchart*. Sub-proses pertama adalah mendesain *compensator PI* dengan blok *flowchart* yang menggunakan garis putus-putus. Sub-proses yang kedua adalah mendesain *compensator fuzzy* dengan blok *flowchart* yang menggunakan garis-titik-titik.



Gambar 3-1 Diagram Alir Penelitian



Gambar 3-2 Diagram alir proses desain *compensator*

## 3.2 Peralatan

Peralatan yang digunakan pada penelitian ini terdiri dari dua jenis, yaitu perangkat keras dan perangkat lunak.

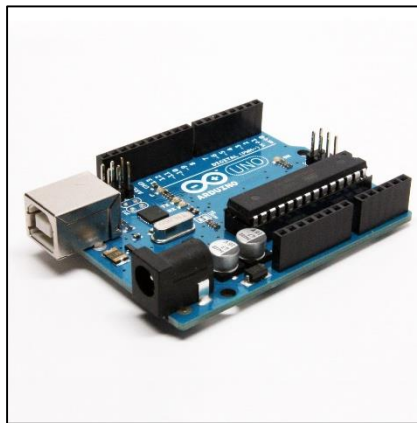
### 3.2.1 Perangkat Keras

Perangkat keras yang digunakan pada penelitian adalah :

1. Arduino UNO sebagai *controller-card*. (Gambar 3-3)
2. Escon 50/5 sebagai *driver* motor. (Gambar 3-4)



3. *Maxon* DC motor tipe 126691 sebagai *actuator*<sup>5</sup> (Gambar 3-5), dengan datasheet yang dapat dilihat pada Lampiran 1.
4. *Linear ball-screw* sebagai bagian dari *actuator*. (Gambar 3-6)
5. RLS LM10IC001AA10K12 sebagai *linear encoder* (sensor). (Gambar 3-7)
6. Kabel *jumper* sebagai penghubung rangkaian. (Gambar 3-9)
7. *Project Board* sebagai *base* penghubung sekaligus pencabang rangkaian. (Gambar 3-10)
8. *Power Supply* (Input 220 VAC, Output 24 VDC) sebagai penyuplai tegangan untuk *actuator* dan *driver* motor. (Gambar 3-11)



**Gambar 3-3 Arduino UNO**

Sumber : <http://www.nexuscyber.com>



**Gambar 3-4 Driver motor ESCON**

Sumber : [maxonmotor.com](http://maxonmotor.com)

---

<sup>5</sup> *Maxon* DC motor 126691 adalah kombinasi antara motor DC dan *drive train gear*. Penjelasan lebih lanjut dapat dilihat pada bagian 3.3.1.



**Gambar 3-5 Maxon motor**



**Gambar 3-6 Linear ball-screw**  
Sumber : <http://www.thk.com/>



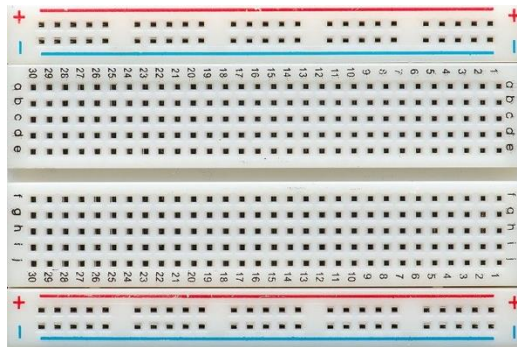
**Gambar 3-7 Linear encoder**  
Sumber : <https://www.rls.si/>



**Gambar 3-8 Posisi encoder pada linear ball screw**



**Gambar 3-9 Kabel jumper**



**Gambar 3-10 Project board**



**Gambar 3-11 Power supply**

### 3.2.2 Perangkat Lunak

1. Arduino IDE (*Integrated Development Environment*) 1.8.2 digunakan untuk meng-upload *source-code* Arduino toolbox<sup>6</sup>.
2. Scilab 5.5.1 digunakan untuk mendesain blok diagram.
3. ESCON Studio 2.2 digunakan untuk mengatur parameter pada *driver* motor Escon.
4. Matlab R2015a digunakan untuk *curve fitting* (*curve smoothing*).

### 3.3 Perancangan

Bagian ini menjelaskan mengenai proses perancangan *controller* menggunakan metode NCTF, dan beberapa perincian yang belum dijelaskan pada bagian 2.1.

#### 3.3.1 Merangkai *Controller, Driver, Encoder, dan Actuator* (Motor dan *Ball-Screw*)

Perhatikan Gambar 3-12 dan Gambar 3-13 untuk melihat nama grup pin pada *driver* motor Escon. Pin negatif (-) dan positif (+) pada grup J1 dihubungkan dengan pin negatif (0 volt) dan pin positif (24 volt) pada *power supply*. Secara berturut-turut pin 1 dan pin 2 pada grup J2 dihubungkan dengan terminal positif dan negatif motor DC. Pin 1 grup J5 (pin penerima sinyal PWM) dihubungkan dengan pin 10 (pin pengirim sinyal PWM) Arduino<sup>7</sup>. Pin 5 grup J5 dihubungkan dengan pin *ground* Arduino.

---

<sup>6</sup> *Source-code* Arduino toolbox v3 adalah kode program yang dibuat untuk mengkoneksikan Arduino dengan scilab. *Source-code* dapat di-download di <https://atoms.scilab.org/toolboxes/Arduino/1.1>.

<sup>7</sup> Pin PWM Arduino adalah 3, 5, 6, 9, 10, dan 11.

J1 Power		
Pin	Signal	Configured Purpose
-	Power_GND	Ground of supply voltage
+	+V <sub>cc</sub>	Power supply voltage (+10...+50 VDC)
J2 Motor		
Pin	Signal	Configured Purpose
1	DC: Motor (+M) EC: Motor winding 1	DC motor: Motor+
2	DC: Motor (-M) EC: Motor winding 2	DC motor: Motor-
3	DC: not connected EC: Motor winding 3	-
4	Motor shield	Cabel shield

**Gambar 3-12** Pengaturan pin *Power* dan *Motor* pada *driver* motor Escon 50/5

J5 Digital I/Os		
Pin	Signal	Configured Purpose
1	DigIN1	PWM - Set Value
2	DigIN2	Enable
3	DigIN/DigOUT3	
4	DigIN/DigOUT4	
5	GND	Ground
6	+5 VDC	Auxiliary output voltage (+5 VDC; <=10 mA)

**Gambar 3-13** Pengaturan pin *Digital* pada *driver* motor Escon 50/5

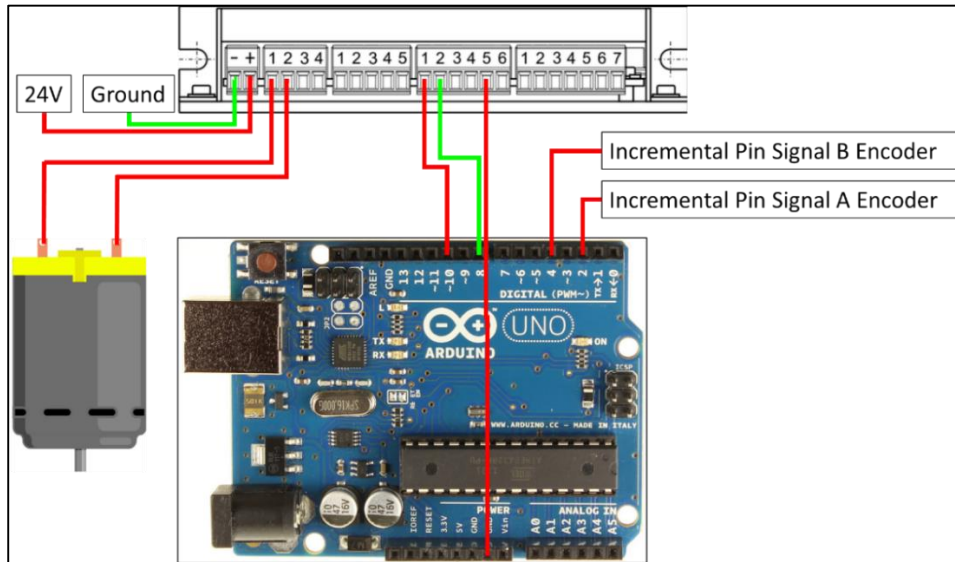
Secara berturut-turut pin 2 dan pin 4 Arduino dihubungkan dengan pin sinyal inkremental A dan B *encoder*. Secara berurut kabel *power* 5 v dan 0 v pada *encoder* dihubungkan dengan pin 5 v dan *ground* pada Arduino. Keterangan warna kabel berdasarkan *datasheet* LM10 *incremental linear magnetic encoder* dengan kode LM10D01\_14. Potongan keterangan untuk tipe *linear encoder* RLS LM10IC001AA10K12 dapat dilihat pada Gambar 3-15. Gambar menyeluruh pada rangkaian *controller*, *driver*, *encoder*, dan *actuator* dapat dilihat pada Gambar 3-14. *Actuator* adalah komponen yang bergerak berdasarkan sinyal yang diterimanya. Pada penelitian ini, *actuator* adalah motor DC yang menerima sinyal dari *driver* Escon. Motor DC berfungsi sebagai penggerak *ball-screw*. Gerak rotasional motor diubah menjadi gerak *linear* oleh *ball-screw*. Gerak linear *ball-screw* ini dapat diketahui melalui *linear encoder* yang terpasang pada *ball-screw* (lihat Gambar 3-18).



Gambar 3-14 Hasil rangkaian *hardware*

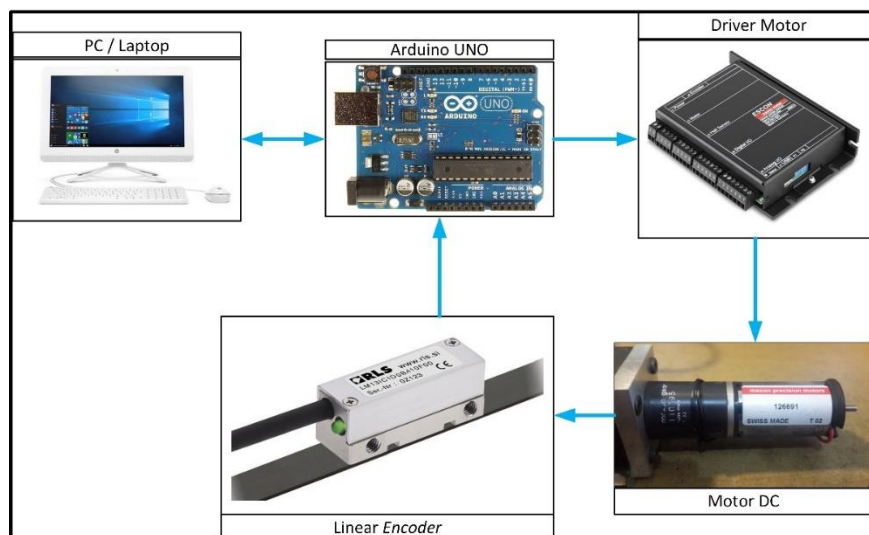
Function	Signal	Colour (option F)
Power	5 V	Brown
	0 V	White
Incremental signals	A	Green
	A-	Yellow
	B	Blue
	B-	Red
Reference signals	Z	Pink
	Z-	Grey
Alarm	E	Violet
	E-	Black

Gambar 3-15 Keterangan warna kabel *linear encoder* RLS LM10IC001AA10K12

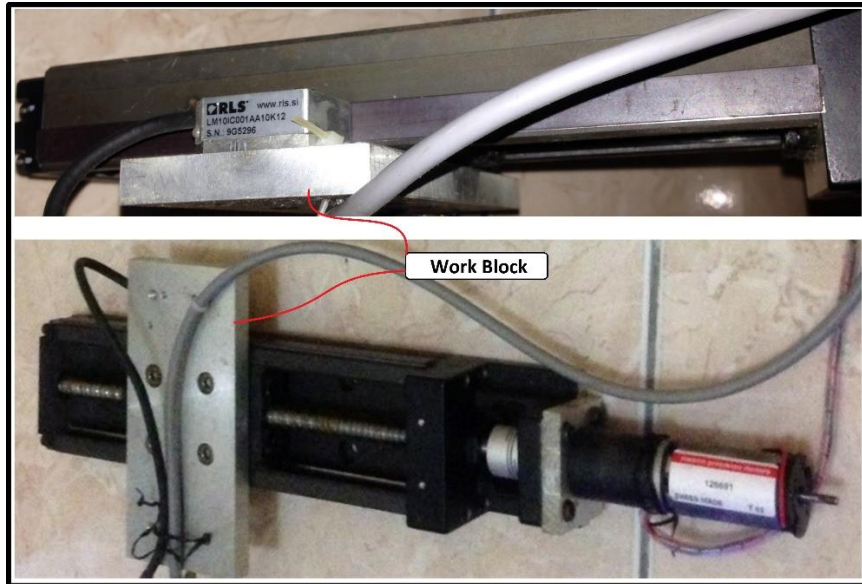


**Gambar 3-16 Rangkaian controller, driver, encoder, dan actuator (motor)**

Koneksi antar pin pada *controller*, *driver*, *encoder*, dan motor dapat dilihat pada Gambar 3-16. Alur kerja rangkaian *controller*, *driver*, *encoder*, dan *actuator* ditunjukkan oleh Gambar 3-17. Laptop mengirimkan sinyal ke arduino, kemudian arduino mengirimkannya ke *driver* motor, setelah itu motor bergerak berdasarkan sinyal yang diterimanya dari *driver*. Akibat pergerakan motor, *work block* juga ikut bergerak sehingga *encoder* membaca pergerakan *linear* dari *work block* (lihat Gambar 3-18). Pergerakan tersebut diubah oleh *encoder* menjadi sinyal inkremental yang kemudian dikirimkan ke arduino UNO. Sinyal yang diterima arduino UNO dari *encoder* dikirimkan kembali ke PC.



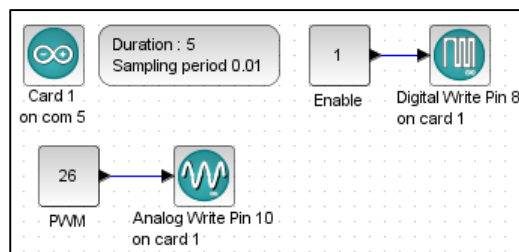
**Gambar 3-17 Alur kerja rangkaian perangkat keras**



Gambar 3-18 Perbesaran gambar hubungan *work block* dengan motor

### 3.3.2 Identifikasi Hubungan Nilai PWM Arduino dengan Tegangan yang Keluar dari Escon *Driver*

Tegangan yang keluar dari *driver* Escon digunakan sebagai input oleh objek kendali motor DC. Pada sistem kendali, untuk mengendalikan objek kendali maka yang perlu dihasilkan adalah nilai yang langsung berhubungan dengan objek kendali. Pada penelitian ini, untuk menggerakkan motor DC dapat dilakukan dengan memberikan tegangan mulai dari -24 volt hingga 24 volt. Tetapi, karena *controller card* Arduino UNO mengirimkan sinyal PWM ke *driver* motor, maka perlu dilakukan konversi dari sinyal PWM dan tegangan yang keluar dari *driver* Escon menuju motor DC.



Gambar 3-19 Blok diagram pengukuran tegangan dengan variasi PWM

Pengaturan parameter *driver* Escon untuk *range* nilai PWM adalah dari 10% sampai 90%. Pengukuran tegangan yang keluar dari *driver* Escon dilakukan dengan mengatur sinyal PWM Arduino dalam *range* 10% sampai 90%, yaitu dari 26 sampai 229. Pengukuran dilakukan dengan simulasi menggunakan blok



diagram di Scilab seperti pada Gambar 3-19, dan hasil pengukuran dapat dilihat pada Tabel 3-1 dan Gambar 3-20.

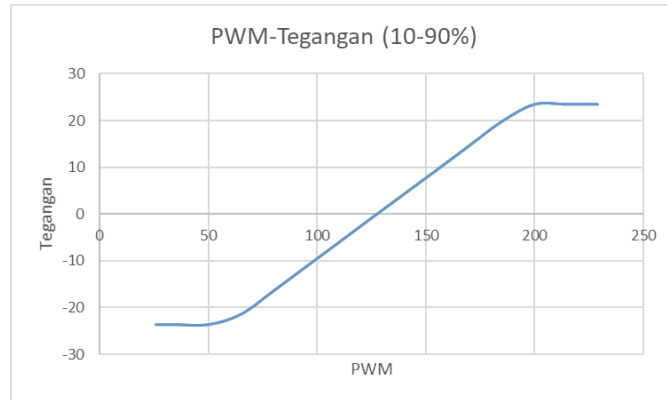
Blok “Card 1 on com 5” pada Gambar 3-19 menunjukkan identitas Arduino yang digunakan pada blok diagram tersebut, yaitu Card 1 sedangkan com 5 adalah nomor port yang telah terkoneksi dengan Arduino melalui USB<sup>8</sup>. Blok “Enable” berisi nilai yang berfungsi untuk mengirimkan sinyal sehingga dapat mengaktifkan *driver* Escon melalui pin digital 8 pada *controller* Arduino. Pada blok PWM Gambar 3-19, nilai yang tertera adalah 26. Pengukuran dilakukan dengan menjalankan (*execute*) blok diagram, dan mengukur tegangan yang keluar dari pin 1 dan pin 2 grup J2 pada *driver* Escon menggunakan multimeter. Pengukuran dilakukan dengan mengatur nilai PWM mulai dari 26 sampai 229 seperti yang tertera pada kolom pertama Tabel 3-1.

**Tabel 3-1 Hasil pengukuran tegangan**

Nilai PWM	Tegangan Terukur
26	-23.6
35	-23.6
50	-23.6
65	-21.4
80	-16.4
95	-11.2
110	-6.04
125	-0.84
140	4.3
155	9.46
170	14.66
185	19.84
200	23.5
215	23.5
229	23.5

---

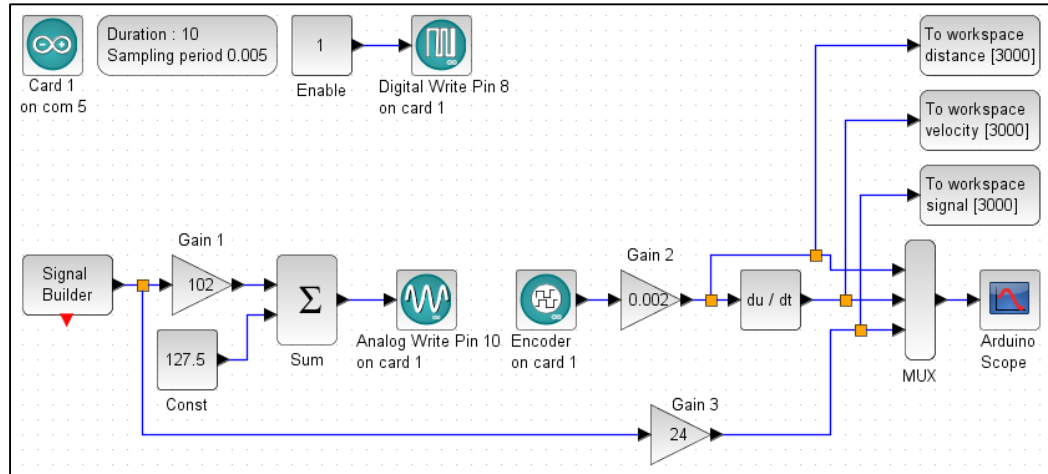
<sup>8</sup> USB adalah singkatan dari *Universal Serial Bus* yang merupakan terminal pada perangkat yang terhubung ke *Arduino Controller*.



Gambar 3-20 Grafik hasil pengukuran tegangan

### 3.3.3 Pembuatan Blok Diagram untuk Eksperimen *Open-loop*

Blok diagram didesain menggunakan Scilab Xcos. Desain blok diagram untuk eksperimen *open-loop* dapat dilihat pada Gambar 3-21. Blok Signal Builder pada Gambar 3-21 berisi sinyal yang secara visual dapat dilihat pada kurva signal berwarna merah (garis putus-putus) Gambar 3-22. Hanya saja pada kurva berwarna merah (garis putus-putus) tersebut, nilai pada sumbu y sudah dikalikan dengan *Gain* 3 pada Gambar 3-21, yaitu 24.



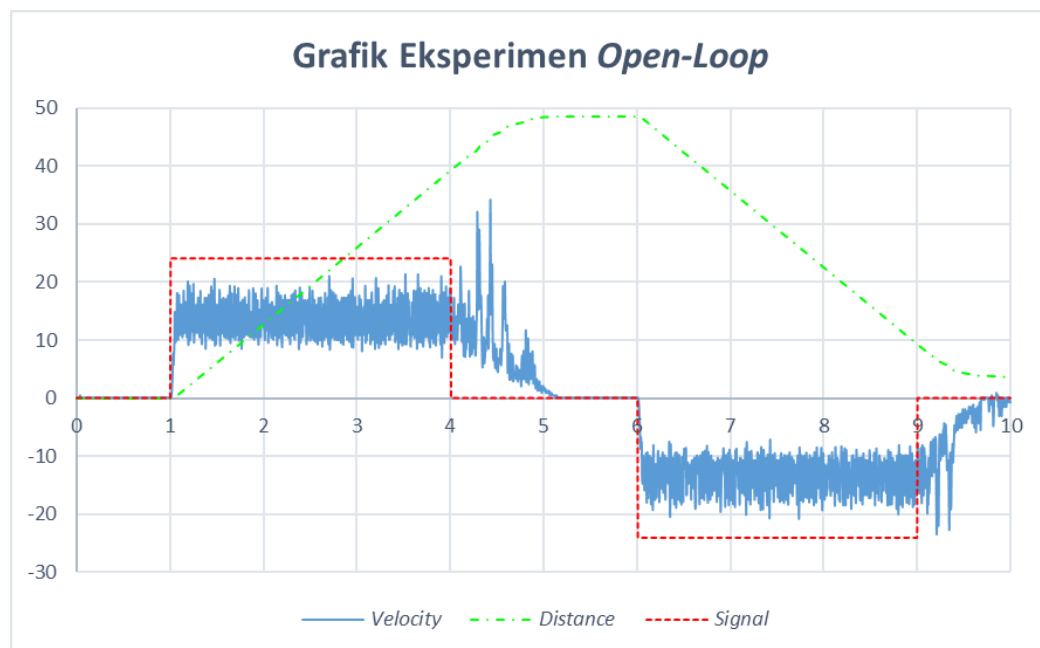
Gambar 3-21 Blok diagram untuk eksperimen *open-loop*

Nilai blok *Gain* 1 pada Gambar 3-21 adalah untuk mengalikan nilai sinyal yang berasal dari blok Signal Builder dan menjumlahkannya dengan nilai pada blok Const. Sehingga ketika sinyal dari Signal Builder bernilai 1, 0, dan -1, maka nilai PWM yang akan dikeluarkan oleh pin 10 Arduino adalah 229.5 (90% dari

nilai 255)<sup>9</sup>, 127.5, dan 25.5 (10% dari nilai 255) secara berurut. Yang mana secara berturut-turut nilai PWM 229.5, 127.5, dan 25.5 mengindikasikan motor DC bergerak dengan kecepatan maksimal searah jarum jam, kecepatan nol, dan kecepatan maksimal berlawanan arah jarum jam.

Nilai blok *Gain 2* pada Gambar 3-21 digunakan untuk mereduksi sinyal dari *encoder* karena resolusi *linear encoder* adalah  $2\ \mu\text{m}$ , sehingga sinyal yang keluar dari blok *Gain 2* memiliki satuan *mm*.

Kurva kecepatan berwarna biru (garis utuh) pada Gambar 3-22 memiliki satuan *mm/s* dan kurva jarak berwarna hijau (garis titik) pada Gambar 3-22 memiliki satuan *mm*. Sedangkan kurva berwarna merah (garis putus-putus) memiliki satuan volt.



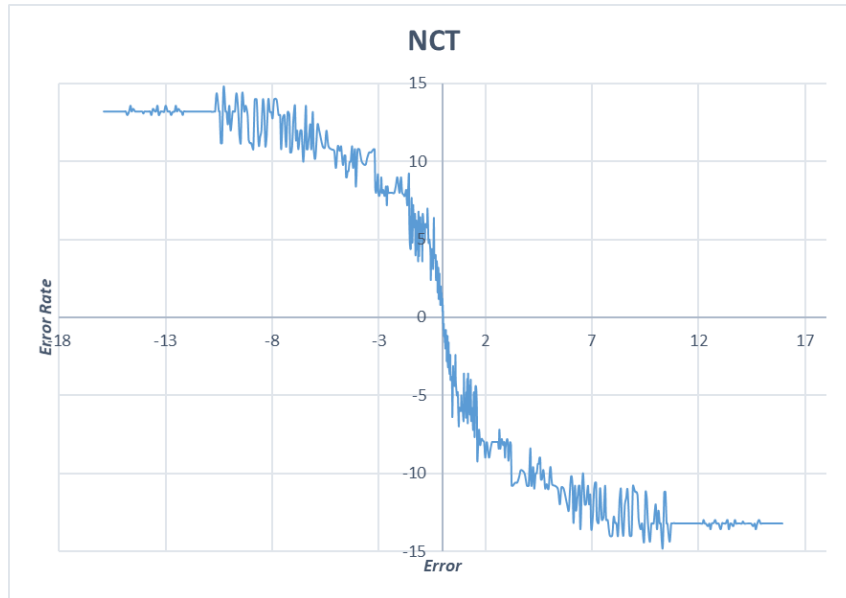
Gambar 3-22 Grafik eksperimen *open-loop*

### 3.3.4 Konstruksi NCT

NCT dibentuk berdasarkan data yang didapat dari eksperimen *open-loop*. NCT dapat dilihat pada Gambar 3-23. Eksperimen *open-loop* dilakukan selama 10 detik dengan *sampling period* adalah 0.005 detik (lihat Gambar 3-21).

---

<sup>9</sup> Nilai maksimal PWM Arduino UNO adalah 8 bit, yaitu 255.



**Gambar 3-23 Nominal Characteristic Trajectory – Original**

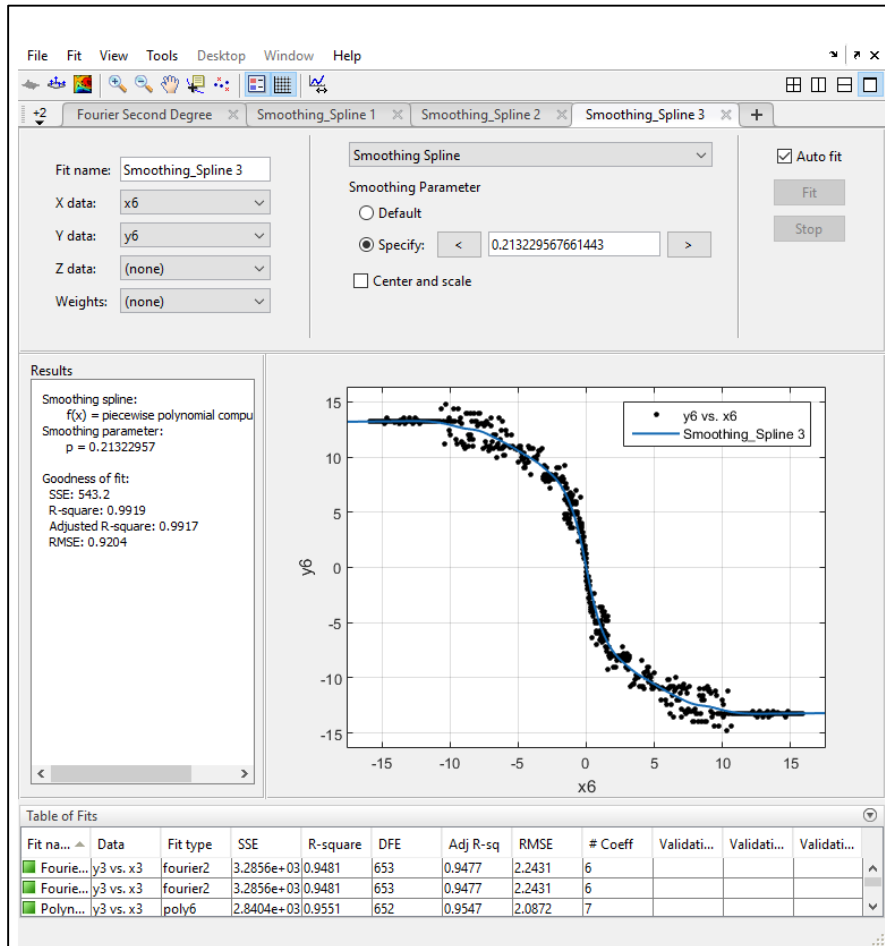
Data NCT diambil data eksperimen open loop mulai detik ke 3.5 sampai detik ke 5.145. Kemudian data tersebut dibalik dari detik ke 5.145 sampai detik ke 3.5.

### 3.3.5 *Smoothing* NCT

NCT yang didapat dari eksperimen *open-loop* masih sangat kasar, maka perlu dilakukan proses *smoothing* grafik. *Smoothing* grafik dilakukan menggunakan fitur *Curve Fitting* pada *software* Matlab R2015a<sup>10</sup>, dapat dilihat pada Gambar 3-24.

---

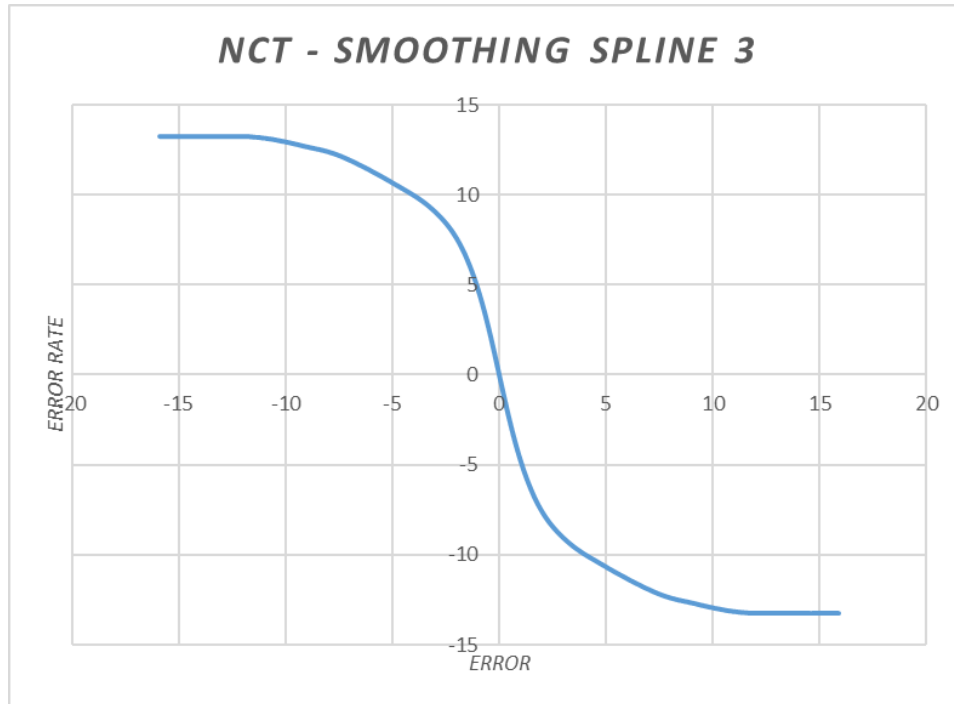
<sup>10</sup> *Smoothing* dapat dilakukan tidak hanya dengan Matlab, masih banyak *software* yang dapat digunakan untuk *smoothing* grafik, diantaranya adalah SimFit, PeakFit, MasterPlex QT, dan MagicPlot



Gambar 3-24 Aplikasi Curve Fitting di Matlab R2015a

*Smoothing* grafik dilakukan beberapa kali sampai mendapat hasil terbaik<sup>11</sup>. Beberapa metode yang digunakan sebelum akhirnya menggunakan metode *Smoothing Spline* adalah metode *Rational*, metode *Polynomial 6<sup>th</sup> Degree*, dan metode *Fourier Second Degree*. Percobaan menggunakan metode *Smoothing Spline* dilakukan sebanyak tiga kali hingga didapat hasil terbaik. Percobaan pertama dengan menspesifikasi nilai *smoothing* parameter **0.0102**, **0.2455** untuk percobaan kedua, dan **0.2132** untuk percobaan ketiga. Hasil terbaik dari *smoothing* grafik adalah menggunakan metode *Smoothing Spline* ketiga dengan parameter *smoothing* adalah **0.2132** yang dapat dilihat pada Gambar 3-25.

<sup>11</sup> Hasil terbaik yang dimaksud di sini adalah grafik hasil *smoothing* dapat mengikuti *trendline* NCT *original*. Tidak hanya itu, justifikasi hasil terbaik juga dilihat dari percobaan menggunakan blok diagram, lihat bagian 3.3.6.



Gambar 3-25 Nominal Characteristic Trajectory – *Smoothing Spline 3*

Parameter NCT yang telah didiskusikan pada bagian 2.1.2 dapat dilihat pada Gambar 3-26. Parameter NCT ini dibentuk berdasarkan data pada NCT yang di-*smoothing* menggunakan metode *Smoothing Spline 3*.

m	h (mm/s)	$\alpha$	Ur (v)	Kf	"Kf* $\alpha$ "	T	"T* $\alpha$ "	A
-5.417	13.23209	5.417	24	0.551337	2.986592	0.005	0.027085	9.174

Gambar 3-26 Parameter NCT – *Smoothing Spline 3*

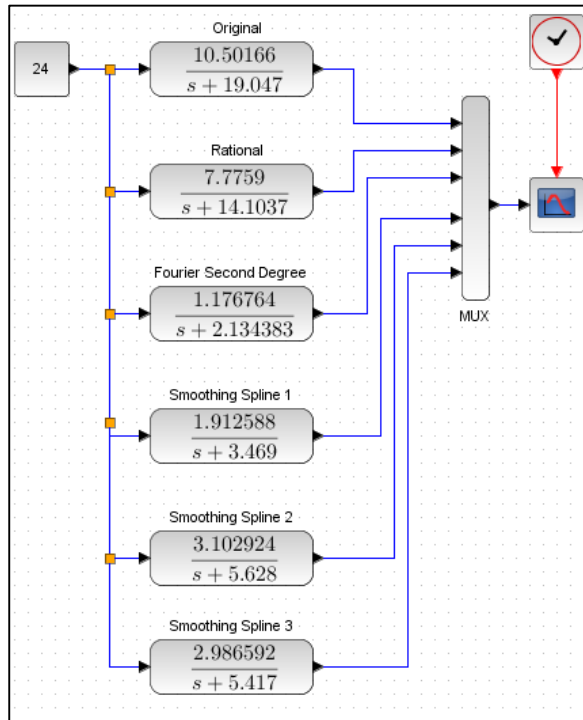
### 3.3.6 Fungsi Transfer

Sampai di sini, fungsi transfer objek kendali seperti telah didiskusikan pada bagian 2.1.2 Pers. (1) sudah dapat dibentuk. Dengan menggunakan parameter pada Gambar 3-26, maka fungsi transfer dari objek kendali dapat dinyatakan dengan<sup>12</sup>

$$\frac{\Theta(s)}{U(s)} = \frac{2.986592}{s(s + 5.417)} \quad \text{Pers. (15)}$$

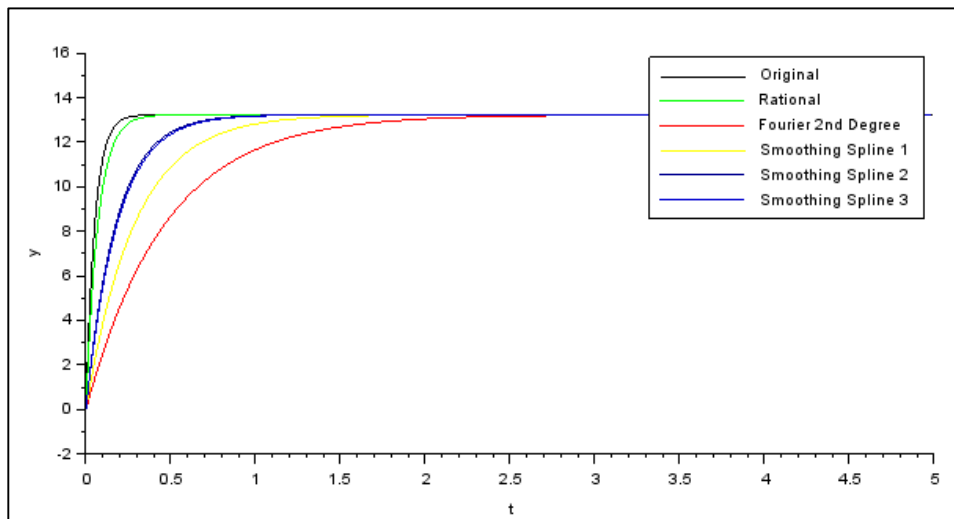
Metode *Smoothing Spline 3* terpilih dari semua metode yang telah dicoba dikarenakan telah melewati verifikasi simulasi menggunakan blok diagram seperti pada Gambar 3-27.

<sup>12</sup> Variabel  $K$  pada Pers. (1) adalah sama dengan variabel  $K_f$  pada Gambar 3-26. Hanya saja variabel  $K_f$  pada penelitian ini digunakan sebagai *identifier* konstanta *compensator fuzzy* (lihat bagian 3.3.8.5).



Gambar 3-27 Verifikasi fungsi transfer

Proses verifikasi adalah dengan melihat apakah hasil simulasi sudah sama dengan atau mendekati nilai  $h = 13.23209$ . Hasil simulasi verifikasi fungsi transfer dapat dilihat pada Gambar 3-28.



Gambar 3-28 Grafik hasil simulasi verifikasi fungsi transfer

### 3.3.7 Desain *Compensator PI*

Proses mendesain *compensator PI* dilakukan dengan cara mencari nilai  $K_p$  dan  $K_i$  dengan acuan *stable region* pada Gambar 2-5.

### 3.3.7.1 Menentukan Nilai *Natural Frequency* $\omega_n$ dan *Damping Ratio* $\zeta$

Nilai natural frequency  $\omega_n$  dan *damping ratio*  $\zeta$  ditentukan berdasarkan *stable region* pada Gambar 2-5. Nilai  $\omega_n$  dan  $\zeta$  yang telah disesuaikan dengan penelitian ini dapat dilihat pada

Tabel 3-2 Nilai  $\omega_n$  dan  $\zeta$

$\zeta$	" $\omega_n * T$ "					" $\omega_n * T$ " sesuai diagram di jurnal					$\omega_n$				
	0.1	0.2	0.3	0.4	0.5	0.02	0.04	0.06	0.08	0.1	20	40	60	80	100
2	0.1	0.2	0.3	0.4	0.5	0.02	0.04	0.06	0.08	0.1	20	40	60	80	100
4	0.1	0.2	0.3	0.4	0.5	0.02	0.04	0.06	0.08	0.1	20	40	60	80	100
6	0.1	0.2	0.3	0.4		0.02	0.04	0.06	0.08		20	40	60	80	
8	0.1	0.2	0.3			0.02	0.04	0.06			20	40	60		
10	0.1	0.2				0.02	0.04				20	40			
12	0.1					0.02					20				
14	0.1					0.02					20				
16	0.1					0.02					20				
18	0.1					0.02					20				
20	0.1					0.02					20				

### 3.3.7.2 Penentuan Nilai $K_p$ dan $K_i$

Berdasarkan Pers. (8) dan Pers. (9) maka nilai  $K_p$  dan  $K_i$  dapat diperoleh seperti yang ditunjukkan pada Tabel 3-3. Setelah melakukan *trial-error* menggunakan nilai  $K_p$  dan  $K_i$  pada Tabel 3-3, didapatkan hasil terbaik dengan respon tercepat dan *error steady state* terkecil. Nilai  $K_p$  dan  $K_i$  terpilih adalah 321.43661 dan 1205.3873. Pembahasan mengenai nilai spesifik hasilnya dapat dilihat pada bagian 4.1.

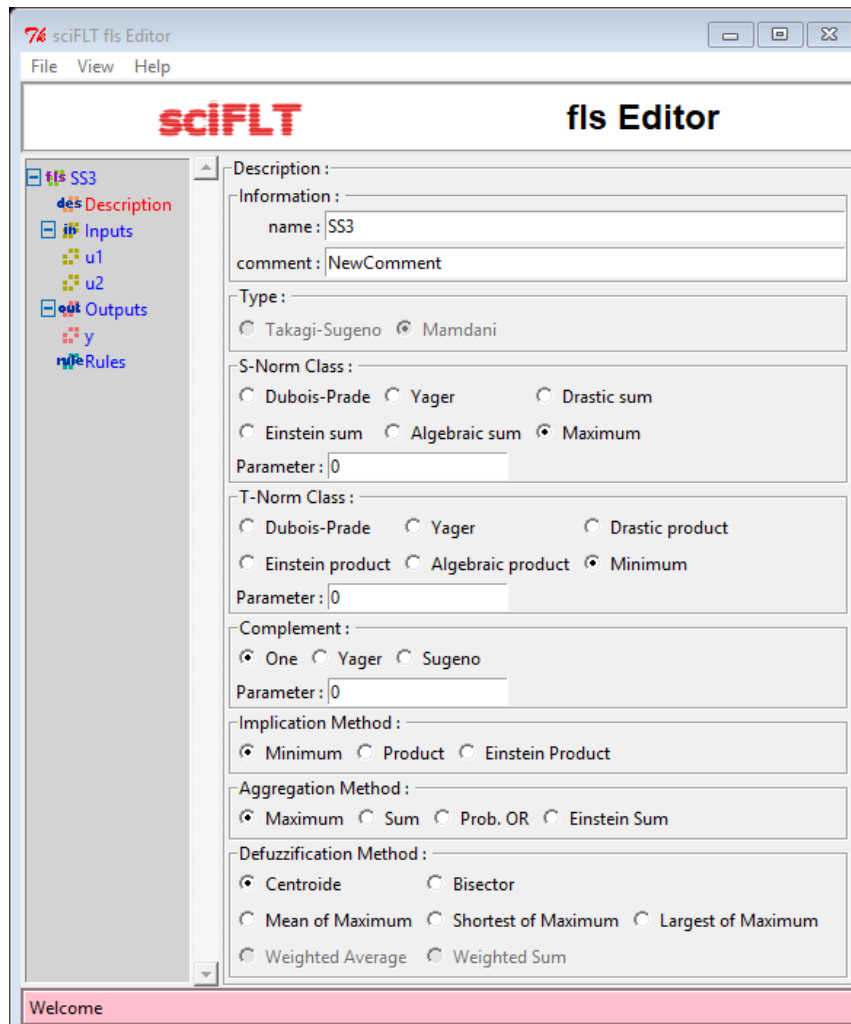
Tabel 3-3 Nilai  $K_p$  dan  $K_i$

$K_p$					$K_i$				
26.7864	53.5728	80.3592	107.146	133.932	133.932	535.728	1205.39	2142.91	3348.3
53.5728	107.146	160.718	214.291	267.864	133.932	535.728	1205.39	2142.91	3348.3
80.3592	160.718	241.077	321.437		133.932	535.728	1205.39	2142.91	
107.146	214.291	321.437			133.932	535.728	1205.39		
133.932	267.864				133.932	535.728			
160.718					133.932				
187.505					133.932				
214.291					133.932				
241.077					133.932				
267.864					133.932				



### 3.3.8 Desain *Compensator fuzzy*

*Compensator fuzzy* didesain menggunakan *atoms module* di Scilab. Caranya adalah dengan mengetikkan *command* “editfls” pada Scilab Console. Deskripsi dari hasil desain *compensator fuzzy* dapat dilihat pada “fls Editor” seperti yang terlihat pada Gambar 3-29. Langkah yang lebih rinci pada proses ini akan dibahas pada sub bagian 3.3.8 ini.



Gambar 3-29 Deskripsi *Compensator fuzzy*

#### 3.3.8.1 *Membership Function*

Penjelasan pada bagian 2.3.1 dapat direpresentasikan dengan Tabel 3-4. Tabel 3-5 merepresentasikan hal yang sama dengan Tabel 3-4, tetapi sudah dalam bentuk nilai. Pada Tabel 3-5 juga terdapat tambahan keterangan pada tiga baris terbawah, yaitu *certainty range of value*.

Tabel 3-4 Parameter *membership function* dalam bentuk variabel

Fuzzy Compensator's Parameters		Input						Output		
		$u_p$			$u_i$			$u$		
Linguistic Values		N	Z	P	N	Z	P	N	Z	P
Min	Universe of Discourse	$-h$			$-u_{imax}$			$-u_r$		
Max		$h$			$u_{imax}$			$u_r$		
Min	Range of Value	$-h$	$-(\dot{e} + \alpha e)$	$-(\dot{e} + \alpha e)$	$-u_{imax}$		$-u_{imax}$	$-u_r$	0	$u_r$
Mid		--	0	--	--	0	--			
Max		$(\dot{e} + \alpha e)$	$(\dot{e} + \alpha e)$	$h$	$u_{imax}$		$u_{imax}$			

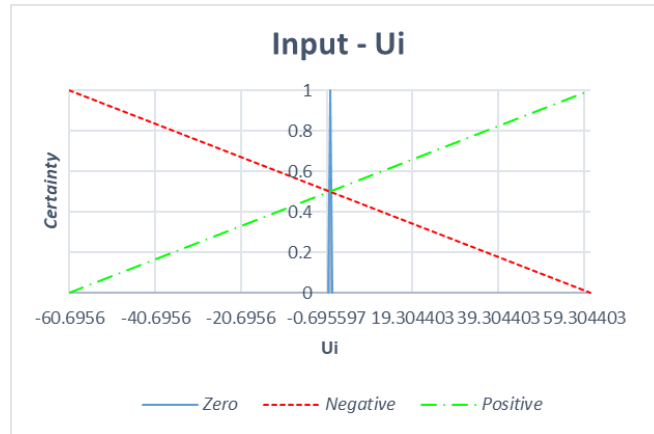
Representasi grafik *membership function* dapat dilihat pada Gambar 3-30, Gambar 3-31, dan Gambar 3-32.

Tabel 3-5 Parameter *membership function* dalam bentuk nilai

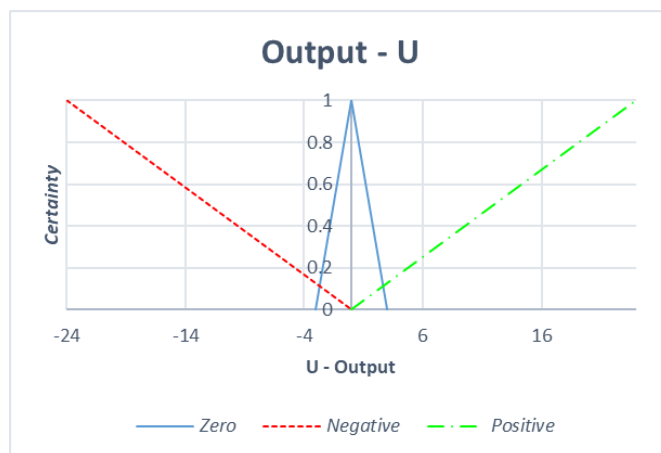
Fuzzy Compensator's Parameters		Input						Output		
		$u_p$			$u_i$			$u$		
Linguistic Values		N	Z	P	N	Z	P	N	Z	P
Min	Universe of Discourse	$-13.23209$			$-60.69559683$			$-24$		
Max		$13.23209$			$60.69559683$			$24$		
Min	Range of Value	$-13.232$	$-0.411$	$-0.411$	$-60.695$		$-60.695$	$-24$	$-3$	$0$
Mid		--	0	--	--	0	--	-	0	-
Max		$0.411$	$0.411$	$13.232$	$60.695$		$60.695$	$0$	$3$	$24$
Min	Certainty's Range of Value	$1$	$0$	$0$	$1$	$0$	$0$	$1$	$0$	$0$
Mid		-	1	-	-	1	-	-	1	-
Max		$0$	$0$	$1$	$0$	$0$	$1$	$0$	$0$	$1$



Gambar 3-30 Input  $u_p$

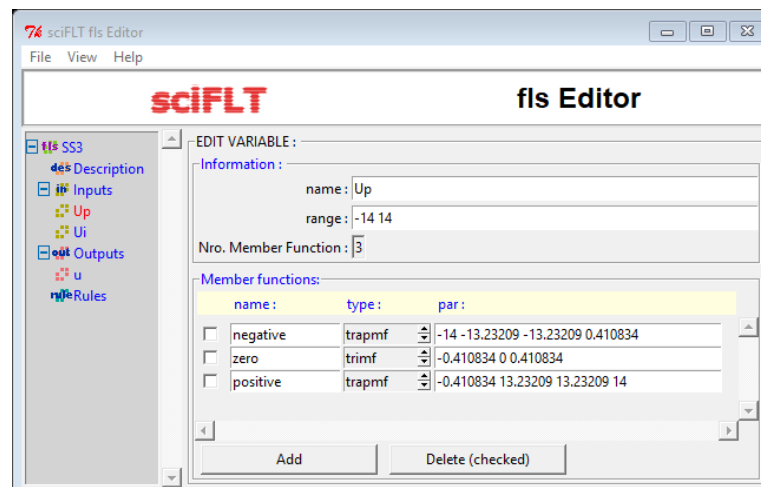


Gambar 3-31 Input  $u_i$

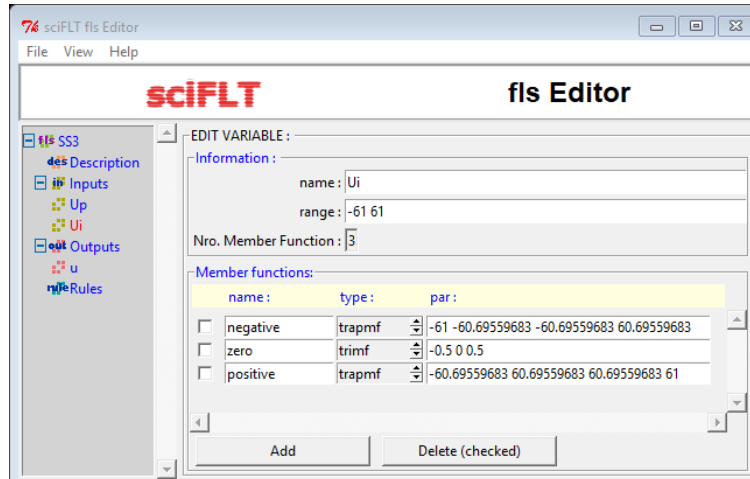


Gambar 3-32 Output  $u$

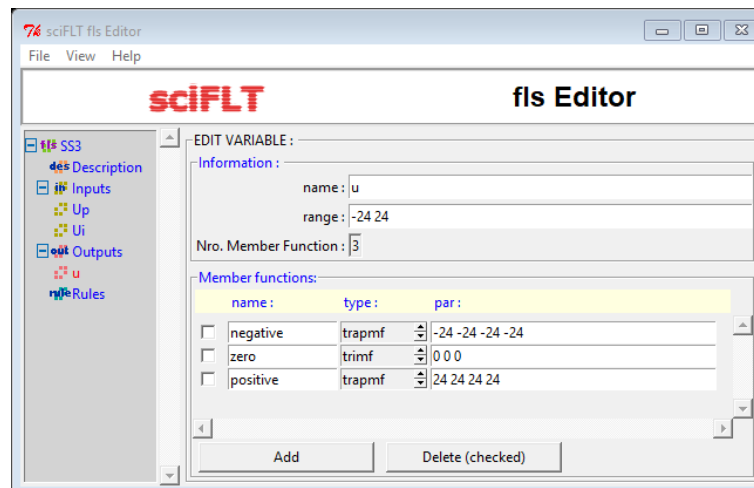
Parameter dari hasil konstruksi *membership function* ini dimasukkan ke “fls Editor” seperti ditunjukkan pada Gambar 3-33, Gambar 3-34, dan Gambar 3-35.



Gambar 3-33 Memasukkan parameter  $u_p$  ke “fls Editor”



Gambar 3-34 Memasukkan parameter  $u_i$  ke “fls Editor”



Gambar 3-35 Memasukkan parameter  $u$  ke “fls Editor”

### 3.3.8.2 Rule Base

Rule base ditentukan berdasarkan *logic reason* terhadap kondisi *real*, lihat Tabel 3-6. Rule base ini kemudian dijadikan sebagai dasar pada langkah selanjutnya, yaitu *matching*.

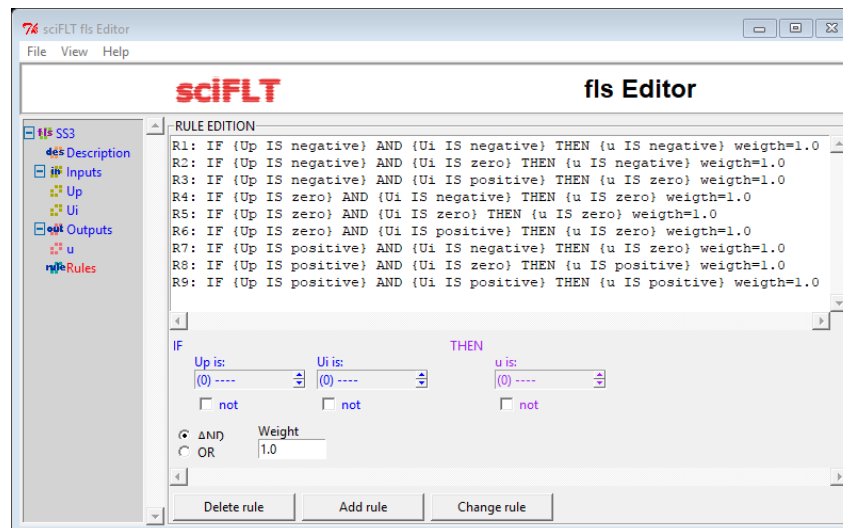
Tabel 3-6 Fuzzy Rule Base

Object		Fuzzy Rule		
		$u_p$		
		Positive	Zero	Negative
$u_i$	Positive	P	Z	Z
	Zero	P	Z	N
	Negative	Z	Z	N

### 3.3.8.3 Mekanisme Inferensi

#### 1. *Matching*

Pada langkah ini, setiap kemungkinan kondisi didefinisikan berdasarkan *rule base* yang telah didefinisikan pada langkah sebelumnya. Memasukkan *rules* pada “fls Editor” dapat dilihat pada Gambar 3-36.



Gambar 3-36 *Matching* berdasarkan *Rule Base*

#### 2. *Inference Step*

Pada langkah ini terdapat dua pengambilan keputusan, yakni *Aggregation* dan *Implication*. Pada *Aggregation* dipilih metode *Maximum* dan pada *Implication* dipilih metode *Minimum*. Untuk memilih kedua metode ini pada “fls Editor” adalah dengan meng-klik radio *button Maximum* pada “*Aggregation method*” dan radio *button Minimum* pada “*Implication Method*” yang ditunjukkan pada Gambar 3-29 bagian bawah.

### 3.3.8.4 *Defuzzification*

Metode *defuzzification* yang digunakan pada penelitian ini adalah *Center of Area*. Untuk memilih metode ini pada “fls Editor” adalah dengan meng-klik radio *button Centroide* pada “*Defuzzification Method*” yang ditunjukkan pada Gambar 3-29 bagian paling bawah.

### 3.3.8.5 Konstanta *Compensator fuzzy*

Bagian terakhir yang harus ditentukan adalah konstanta fuzzy. Konstanta ini dapat dicari dengan menggunakan Pers. (14) yaitu

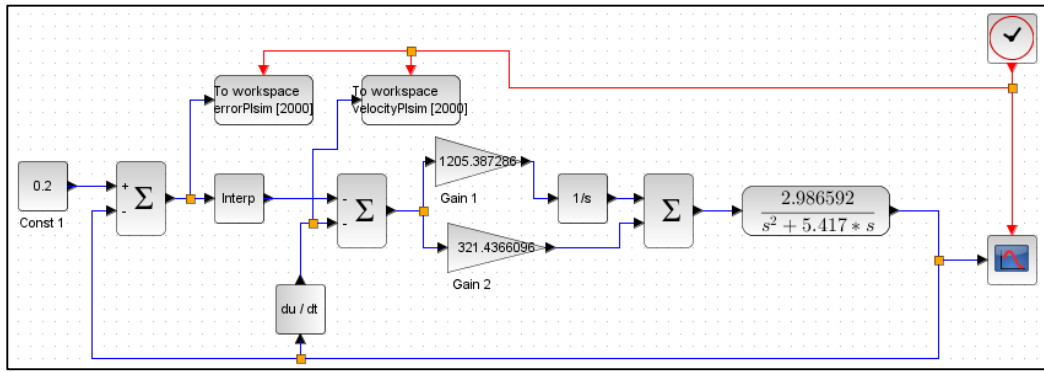
$$K_f = \frac{h}{u_r} = \frac{13.23209}{24} = 0.551337$$

## 3.4 Pengujian dengan Simulasi Komputer dan Eksperimen objek

Pada penelitian ini, setiap hasil desain perlu diuji untuk mengetahui kualitas *controller* yang didesain. Pengujian dilakukan dengan simulasi di komputer dan eksperimen objek. Setiap jenis simulasi menggunakan kedua *compensator* yang dirancang, yakni *compensator PI* dan *compensator fuzzy*. Setiap pengujian dengan masing-masing *compensator* akan dilakukan sebanyak dua kali dengan input yang berbeda, yakni 0.2 dan 2, dalam satuan mm. Penentuan nilai input ini didasarkan pada nilai *error* yang berada di dalam *range error* NCT. Dapat diperhatikan pada Gambar 3-25, *range error* NCT adalah -15 sampai 15,

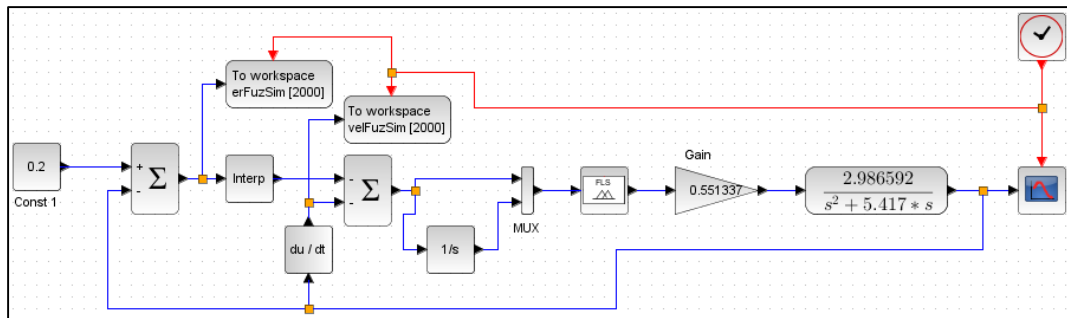
Penggantian nilai input dilakukan dengan mengganti nilai pada blok Const 1. Pada eksperimen objek, terdapat *sampling period*, di mana *sampling period* adalah resolusi waktu penerimaan dan pengiriman data dari dan ke Arduino. Blok Const 2 pada eksperimen objek berfungsi mengirimkan sinyal sehingga dapat mengaktifkan *driver* Escon melalui pin digital 8. Pada bagian ini hanya menampilkan blok diagram.

Gambar 3-37 menunjukkan blok diagram controller NCTF menggunakan *compensator PI*. Blok *interp* adalah blok dengan input error  $e = \theta_r - \theta$  dan output error rate  $\dot{e}$ . Secara berurut, blok Gain 1 dan blok Gain 2 adalah blok yang berisi nilai  $K_i$  dan  $K_p$  dengan hasil terbaik dibandingkan dengan nilai  $K_i$  dan  $K_p$  pada Tabel 3-3. Sebelum mendapatkan hasil terbaik ini telah dilakukan proses trial-error dengan memvariasikan nilai  $K_i$  dan  $K_p$ .



Gambar 3-37 Simulasi Komputer – *Compensator PI*

Pada Gambar 3-38 terdapat blok baru yaitu blok FLS. Blok ini merupakan blok yang menyimpan data hasil desain *compensator fuzzy* yang telah dibahas pada bagian 3.3.8.1 sampai 3.3.8.4. Kemudian, blok Gain pada Gambar 3-38 berisi konstanta *compensator fuzzy* yang telah diketahui pada bagian 3.3.8.5.

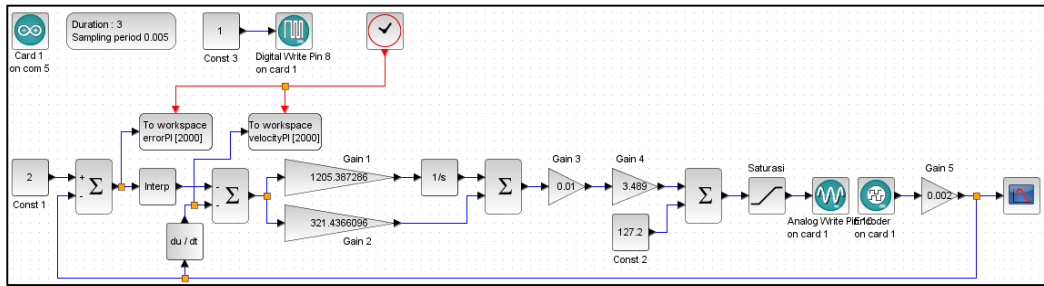


Gambar 3-38 Simulasi Komputer – *Compensator fuzzy*

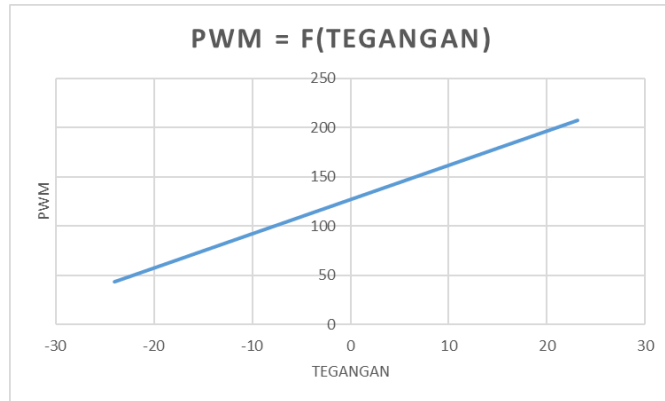
Pada Gambar 3-39, blok Gain 3 berfungsi untuk mereduksi nilai sehingga dapat berubah ke dalam *range* tegangan kerja objek kendali. Kemudian, nilai yang telah terduksi tersebut diubah menjadi nilai PWM dengan menggunakan blok Gain 4 dan blok Const 2. Nilai pada blok Gain 4 dan blok Const 2 didapat dari persamaan garis lurus menggunakan keterangan yang ditunjukkan pada Gambar 3-40. Grafik pada Gambar 3-40 didapat dengan cara memanipulasi data pada Tabel 3-1, sehingga didapat persamaan berikut

$$P = 3.489T + 127.2 \quad \text{Pers. (16)}$$

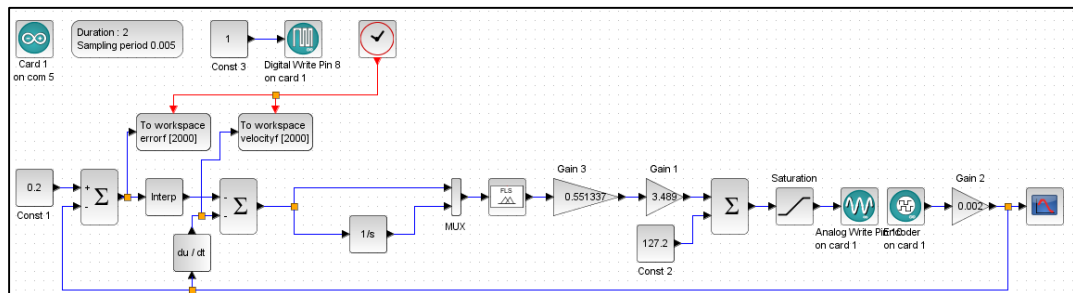
di mana  $P$  adalah nilai PWM,  $T$  adalah nilai tegangan, koefisien  $T$  3.489 digunakan untuk blok Gain 4 pada Gambar 3-39, dan konstanta pada term kedua Pers. (16) 127.2 digunakan untuk blok Const 2 pada Gambar 3-39. Blok Saturasi berguna untuk membatasi range nilai PWM yang masuk ke objek kendali. Batas bawah nilai PWM adalah 26 dan batas atas nilai PWM adalah 229.



Gambar 3-39 Eksperimen objek – *Compensator PI*



Gambar 3-40 Grafik konversi tegangan menjadi PWM



Gambar 3-41 Eksperimen objek – *Compensator fuzzy*

Pada Gambar 3-41, sama dengan blok diagram pada Gambar 3-39, hanya saja di sini terdapat blok FLS, yang fungsinya telah dijelaskan sebelumnya, yaitu untuk menyimpan hasil rancangan *compensator fuzzy*.

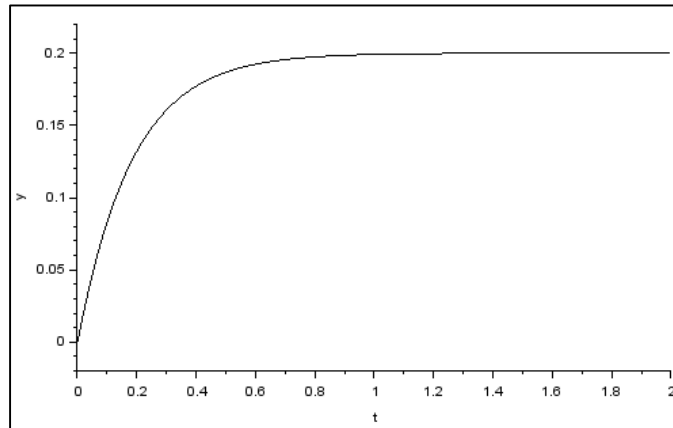


## BAB 4

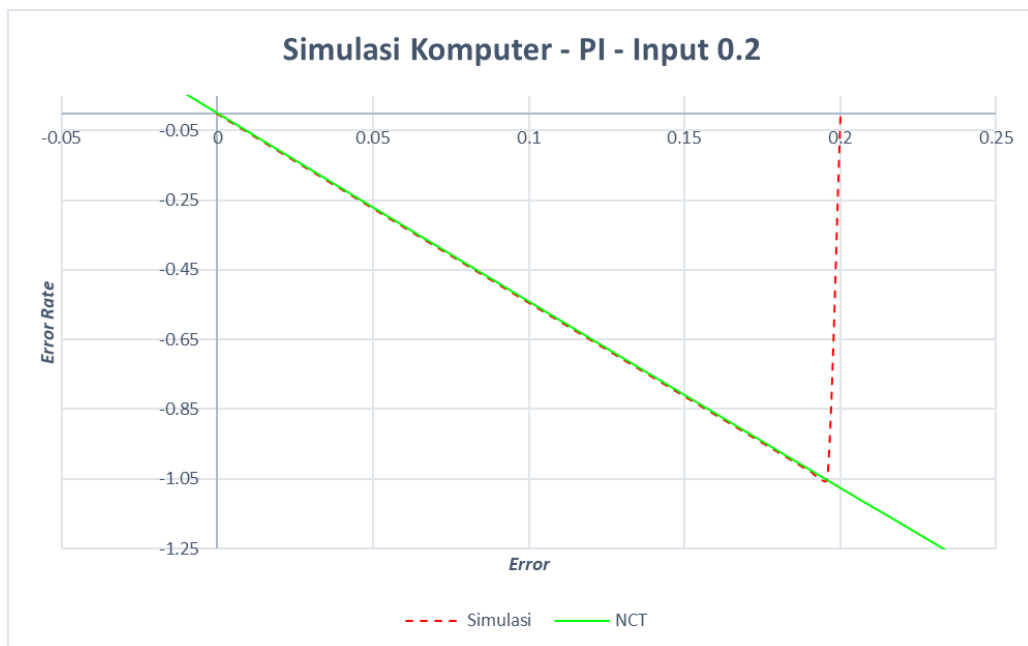
### HASIL DAN PEMBAHASAN

#### 4.1 Analisis terhadap *Compensator PI*

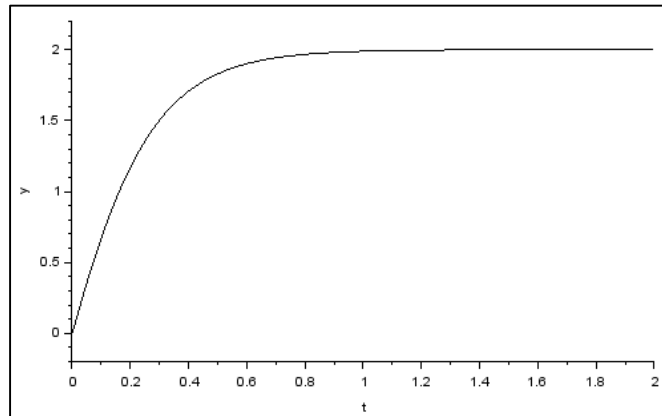
Hasil ideal simulasi komputer menggunakan *compensator PI* dapat dilihat pada Gambar 4-1 sampai Gambar 4-4. Hasil eksperimen objek menggunakan *compensator PI* dapat dilihat pada Gambar 4-5 sampai Gambar 4-8.



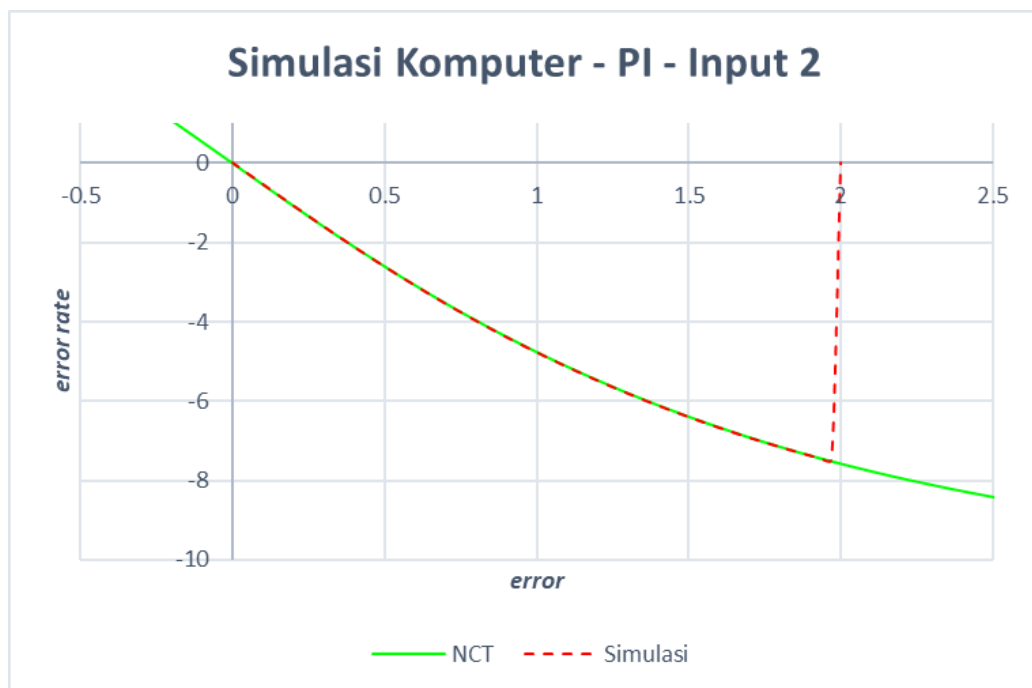
Gambar 4-1 Grafik jarak-waktu simulasi komputer – *compensator PI* – input 0.2



Gambar 4-2 Grafik NCTF simulasi komputer – *compensator PI* – input 0.2

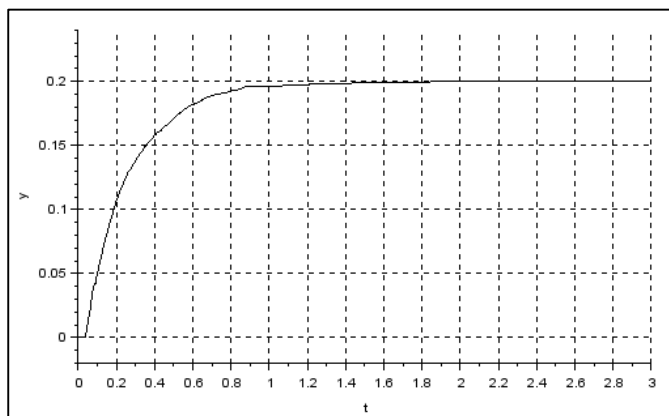


Gambar 4-3 Grafik jarak-waktu simulasi komputer – *compensator PI* – input 2

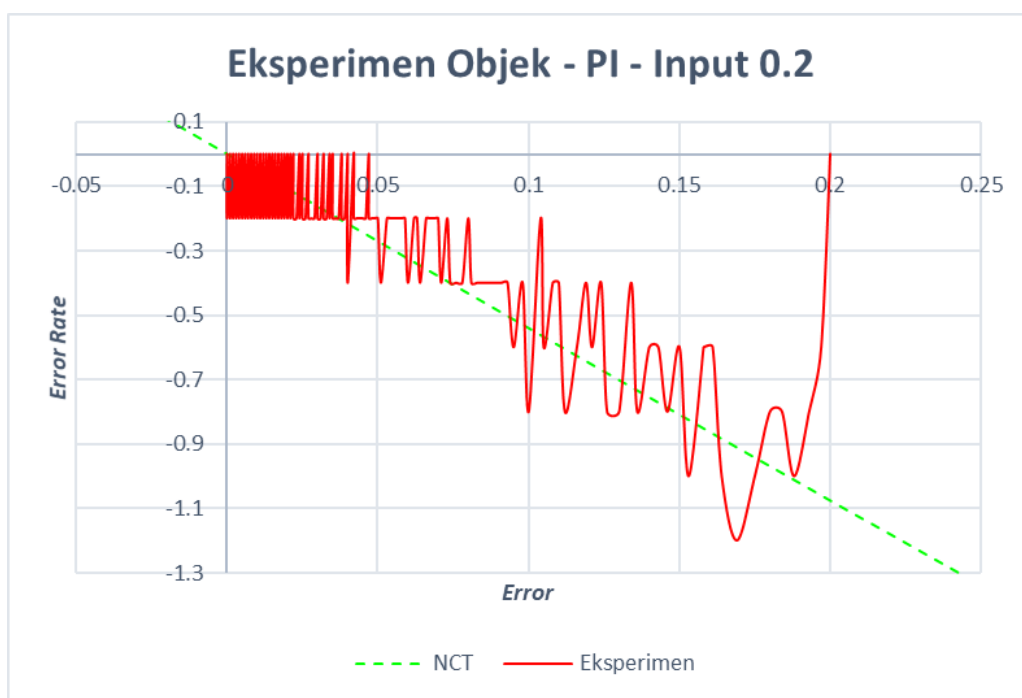


Gambar 4-4 Grafik NCTF simulasi komputer – *compensator PI* – input 2

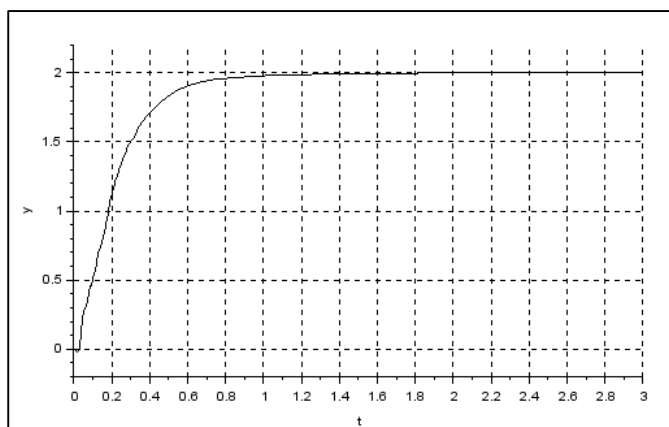
Pada Gambar 4-6 dan Gambar 4-8 menunjukkan respon objek (kurva simulasi) yang berusaha mengikuti NCT. Dapat diperhatikan bahwa kurva simulasi bergerak dari *error rate* nol menuju kurva NCT. Ketika menyentuh NCT, kurva simulasi menstabilkan berusaha mengikuti NCT, sehingga membentuk kurva naik-turun. Naik-turunnya kurva tersebut disebabkan oleh dua faktor, yaitu *sampling period* dan resolusi *encoder*.



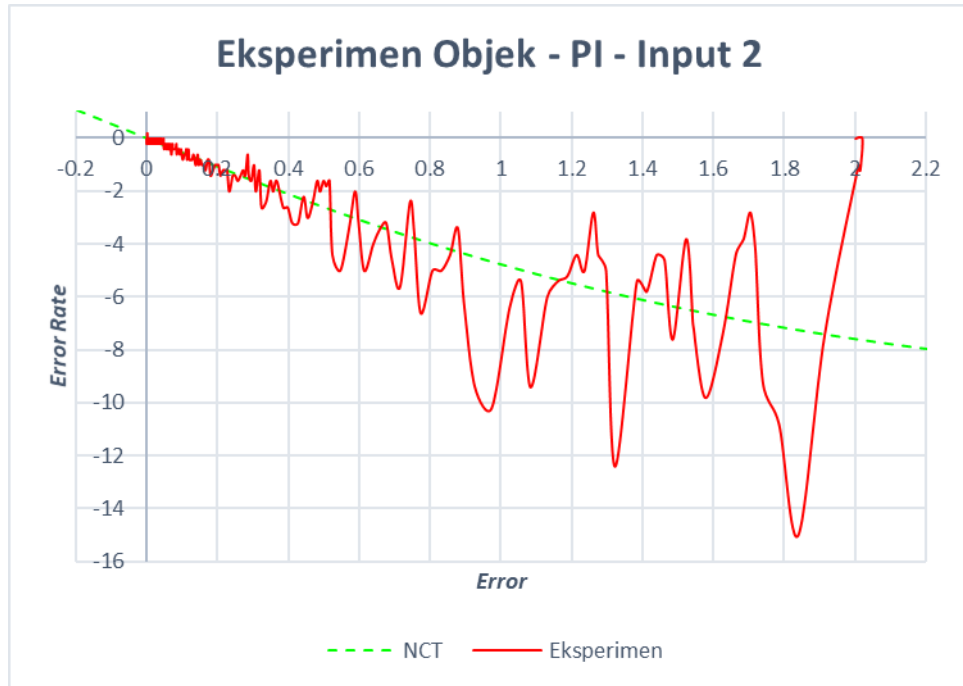
Gambar 4-5 Grafik jarak-waktu eksperimen objek – *compensator PI* – input 0.2



Gambar 4-6 Grafik NCTF eksperimen objek – *compensator PI* – input 0.2



Gambar 4-7 Grafik jarak-waktu eksperimen objek – *compensator PI* – input 2



**Gambar 4-8 Grafik NCTF eksperimen objek – compensator PI – input 2**

*Sampling period* yang digunakan pada eksperimen objek adalah 0.005 sekon. Ini merupakan *sampling period* terkecil pada Scilab. Pada Gambar 4-6 dapat dilihat bahwa kurva simulasi tidak terlalu jauh melewati NCT dibandingkan dengan kurva simulasi pada Gambar 4-8. Hal ini terjadi karena pada Gambar 4-6 input yang digunakan adalah 0.2. Jika melihat NCT pada Gambar 4-6, input 0.2 (*error*) bersesuaian dengan nilai -1.1 (*error-rate*). Sementara batas terluar *error-rate* adalah  $\pm 13.232$ , yang mana mengindikasikan bahwa kecepatan objek masih sangat rendah. Karena kecepatan rendah dan *sampling period* 0.005s, maka ketika *controller* menerima data dari *encoder* bahwa kondisi objek belum menyentuh NCT, dan 0.005 detik setelahnya *controller* menerima data dari *encoder* bahwa objek telah menyentuh atau melewati NCT, maka *controller* langsung memaksa objek agar mengikuti NCT. Hal inilah (kecepatan dan *sampling period*) yang menyebabkan naik-turun kurva simulasi pada Gambar 4-6 tidak sebesar naik-turunnya kurva simulasi pada Gambar 4-8.

Analisis karakteristik terhadap respon objek yang dapat dilakukan adalah karakteristik *settling time* dan *rise time*. *Settling time* adalah waktu respon untuk mencapai, dan bertahan pada 2% dari nilai akhir. *Rise time* adalah waktu respon dari 10% ke 90% terhadap nilai akhir (Nise, 2015, hlm. 163). Tabel karakteristik

hasil pengujian simulasi komputer dan eksperimen objek menggunakan *compensator PI* dapat dilihat pada Tabel 4-1.

**Tabel 4-1 Karakteristik respon objek menggunakan *compensator PI***

Karakteristik	<i>Settling time</i> (s)		<i>Rise time</i> (s)	
	<b>0.2</b>	<b>2</b>	<b>0.2</b>	<b>2</b>
<b>Simulasi Komputer</b>	0.72	0.767	0.404	0.443
<b>Eksperimen Objek</b>	1.42	1.235	0.644	0.42

*Steady state error*<sup>13</sup> pada simulasi komputer dan eksperimen objek menggunakan *compensator PI* dapat mencapai 0%, tetapi pada waktu yang berbeda. Nilai *steady state* ini adalah hasil analisis terhadap grafik jarak-waktu. Pada Tabel 4-2 dapat dilihat bahwa pada simulasi komputer *steady state error* tercapai pada waktu ke 2.0684 dan 2.2534 untuk input 0.2 dan 2. Pada eksperimen objek, 1.822 dan 1.8151 untuk input 0.2 dan 2 secara berurut.

**Tabel 4-2 *Steady state error* 0% PI**

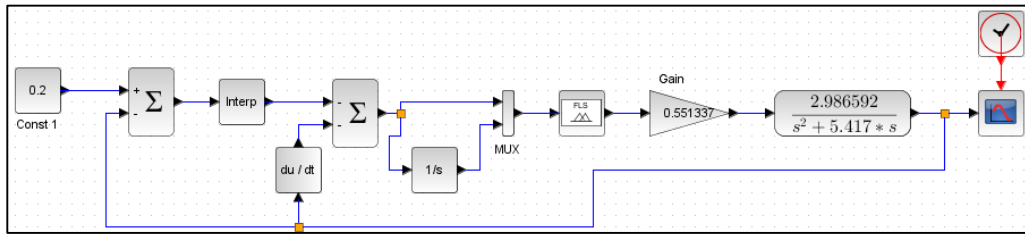
<b>Input</b>	<b>0.2</b>	<b>2</b>
<b>Simulasi Komputer</b>	2.068	2.253
<b>Eksperimen Objek</b>	1.822	1.815

## 4.2 Analisis terhadap *Compensator Fuzzy*

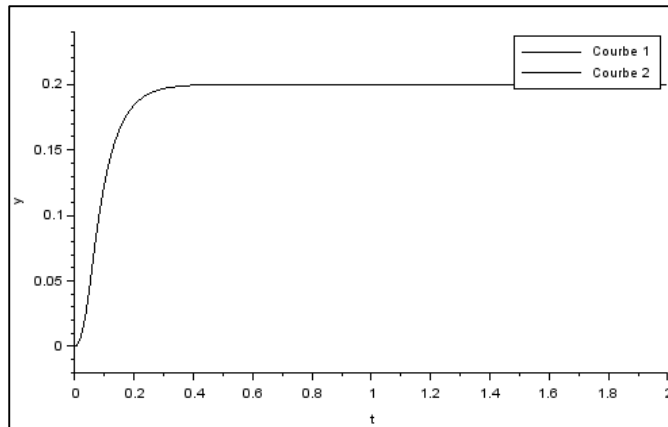
Gambar 4-9 adalah hasil perbaikan blok diagram pada Gambar 3-38. Blok diagram pada Gambar 3-38 tidak bisa disimulasi dikarenakan terdapat masalah pada *software* yang bukan merupakan kajian dari penelitian ini. Permasalahan yang terjadi terletak pada pengambilan data *error* dan *error rate* menggunakan blok “To Workspace”. Karena data *error* dan *error rate* pada simulasi komputer menggunakan *compensator fuzzy* tidak bisa diperoleh, maka grafik NCTF tidak bisa dikonstruksi.

---

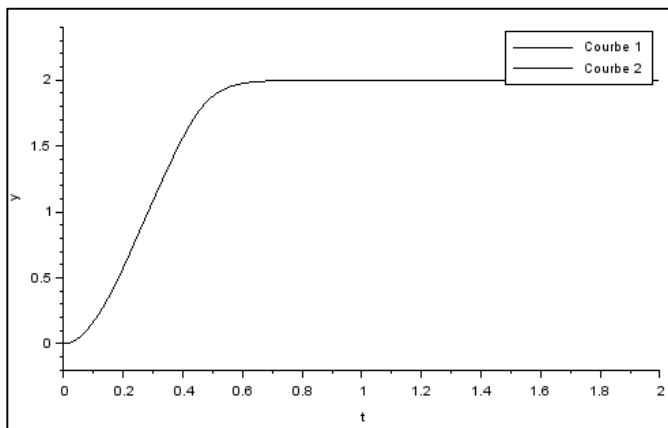
<sup>13</sup> *Steady state error* dalam konteks *point-to-point* adalah ketepatan objek berhenti pada titik akhir.



Gambar 4-9 Blok diagram hasil perbaikan



Gambar 4-10 Grafik jarak-waktu simulasi komputer – *compensator fuzzy* – input 0.2



Gambar 4-11 Grafik jarak-waktu simulasi komputer – *compensator fuzzy* – input 2

Permasalahan juga terjadi pada saat eksperimen objek menggunakan *compensator fuzzy*. Ketika melakukan eksekusi blok diagram, Scilab mengalami *error* sehingga tidak merespon eksekusi, dan secara mendadak *software* Scilab berhenti bekerja yang mengakibatkan *software* Scilab harus di-*restart*. Setelah di-*restart*, kemudian mengeksekusi ulang blok diagram, hal yang sama terulang kembali. Permasalahan ini juga terletak pada internal *software* yang mana bukan merupakan kajian dari penelitian ini.

Data yang didapat dari pengujian menggunakan *compensator fuzzy* hanyalah data dari pengujian simulasi komputer saja, dikarenakan terdapat kendala pada *software* Scilab. Hasil pengujian simulasi komputer hanya dalam bentuk grafik jarak-waktu saja. Oleh karena itu, analisis yang dapat dilakukan adalah analisis karakteristik terhadap grafik jarak-waktu hasil simulasi komputer, yang mana merupakan hasil ideal. Tabel karakteristik dapat dilihat pada Tabel 4-3. *Steady state error* pada simulasi komputer menggunakan *compensator fuzzy* adalah 0%.

**Tabel 4-3 Karakteristik respon objek menggunakan *compensator fuzzy***

Karakteristik	<i>Settling time</i> (s)		<i>Rise time</i> (s)	
<b>Input</b>	<b>0.2</b>	<b>2</b>	<b>0.2</b>	<b>2</b>
Simulasi Komputer	0.293	0.576	0.152	0.356

*Steady state error* pada simulasi komputer menggunakan *compensator fuzzy* juga dapat mencapai 0%. Pada Tabel 4-4 dapat dilihat bahwa *steady state error* mencapai 0% pada simulasi komputer menggunakan *compensator fuzzy* dapat diraih pada detik ke 0.5662 dan 0.9883 untuk input 0.2 dan 2 secara berturut-turut.

**Tabel 4-4 *Steady state error* 0% fuzzy**

<b>Input</b>	<b>0.2</b>	<b>2</b>
<b>Simulasi Komputer</b>	0.566	0.988

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Kesimpulan penelitian ini adalah :

1. *Controller* NCTF dengan menggunakan *compensator PI* dan *compensator fuzzy* telah selesai dirancang dan diimplementasikan untuk operasi PTP pada sistem linier menggunakan *ball-screw*.
2. Kinerja *controller* NCTF (*settling time*, *rise time*, dan *steady state error*) telah diketahui melalui simulasi komputer dan eksperimen objek .
3. Pada simulasi komputer, *compensator fuzzy* lebih baik daripada *compensator PI* dalam hal waktu untuk mencapai titik akhir. Dengan kata lain, pada sistem *point-to-point*, *compensator fuzzy* membutuhkan waktu lebih sedikit untuk menggerakkan objek kendali dari titik awal ke titik akhir daripada *compensator PI*.

#### 5.2 Saran

*Software* Scilab tidak bisa mengeksekusi eksperimen objek menggunakan *compensator fuzzy*. Maksudnya adalah Scilab tidak bisa mengirim data ke Arduino ketika menggunakan *toolbox* fuzzy. Scilab juga tidak bisa mengambil data menggunakan blok “To Workspace” pada saat simulasi komputer menggunakan *compensator fuzzy*. Kemungkinan letak permasalahannya ada pada *toolbox* fuzzy. Penelitian yang sama dapat dilakukan dengan hasil yang lebih baik apabila *toolbox* fuzzy telah mengalami perbaikan. Karena sifatnya yang *open-source*, bagi para peneliti yang *expert* di bidang pemrograman juga dapat melakukan penelitian terhadap *bug* yang terdapat di *toolbox* fuzzy. *Source-code toolbox* fuzzy untuk Scilab dapat dilihat pada <http://atoms.scilab.org/toolboxes/sciFLT/0.4.7>.



## DAFTAR PUSTAKA

- C. J. Kempf. (1996). Step and settle positioning algorithm for electro-mechanical system with damping. *Proceedings of the 4th International Workshop on Advanced Motion Control, Tsukuba, Japan.*
- C. J. Kempf, & S. Kobayashi. (1999). Disturbance observer and feedforward design for a high-speed direct-drive positioning table. *IEEE Trans Control Syst Technol.*
- M. L. Workman, R. L. Kosut, & G. F. Franklin. (1987). Adaptive proximate time-optimal servomechanisms: continuous time case. *Proceedings of the American Control Conference, Minneapolis, USA.*
- Masayoshi Tomizuka. (1996). Robust digital motion controllers for mechanical systems. *Robotics Autonomous Syst.*
- Min-Ho Park, & Chung-Yeun Won. (1991). Time optimal control for induction motor servo system. *IEEE Trans Power Electron.*
- Nise, N. S. (2015). *Control systems engineering*. Hoboken (Nueva Jersey, Estados Unidos: Wiley.
- Passino, K. M., & Yurkovich, S. (1998). *Fuzzy control*. Menlo Park, Calif: Addison-Wesley.
- Purtojo, Wahyudi, Akmeliawati, R., & Shafie, A. A. (2009). Fuzzy-based NCTF control of PTP positioning system. Dalam *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on* (hlm. 1741–1747). IEEE.

- S. Sankaranarayanan, & F. Khorrami. (1997). Adaptive variable structure control and applications to friction compensations. *Proceedings of the 36th IEEE Conference on Decision & Control, San Diego USA.*
- Sato Kaiji, Wahyudi, & Shimokohbe Akira. (2001). Design and characteristics of practical control system for PTP positioning. *Trans Jpn Soc Mech Eng (Ser C).*
- T. Umeno, T. Kaneko, & Y. Hori. (1993). Robust servo-system design with two degree of freedom and its application to novel motion control of robot manipulators. *IEEE Trans Ind Electron.*
- Wahyudi, Sato, K., & Shimokohbe, A. (2003). Characteristics of practical control for point-to-point (PTP) positioning systems. *Precision Engineering*, 27(2), 157–169. [https://doi.org/10.1016/S0141-6359\(02\)00226-X](https://doi.org/10.1016/S0141-6359(02)00226-X)
- WU Shang-Teh, & FU Jeng-Young. (1998). Time-optimal control of servo systems using PD algorithms. *JSME Int J Ser C.*
- Y. Fujimoto, & A. Kawamura. (1995). Robust servo-system based on two-degree-of-freedom control with sliding mode. *IEEE Trans Ind Electron.*

# LAMPIRAN 1

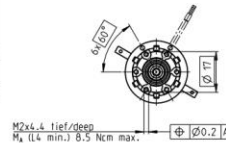
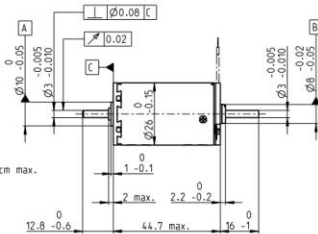
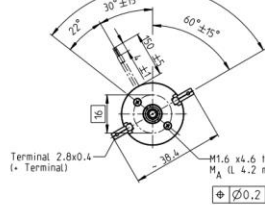
## DATASHEET MAXON DC MOTOR 26 MM

maxon A-max

### A-max 26 Ø26 mm, Precious Metal Brushes CLL, 4.5 Watt

High Power

Kabel AWG 24/7  
cable UL Style 1061  
• Kabel rot  
cable red



M 1:2

Stock program  
Standard program  
Special program (on request)

#### Part Numbers

with terminals	110204	110205	110206	110207	110208	110209	110210	110211	110212	110213	110214
with cables	353109	353110	353111	353112	353113	353114	353115	353116	353117	353118	353119

#### Motor Data

		2.4	3.6	6	7.2	9	12	15	18	24	30	36
<b>Values at nominal voltage</b>												
1 Nominal voltage	V	2.4	3.6	6	7.2	9	12	15	18	24	30	36
2 No load speed	rpm	3890	5190	4090	4060	4020	4440	3530	3640	4510	4680	4520
3 No load current	mA	67.7	69.9	29.2	24	19	16.5	9.41	8.2	8.45	7.16	5.67
4 Nominal speed	rpm	3460	4640	2940	2650	2620	3030	2070	2180	3060	3210	3050
5 Nominal torque (max. continuous torque)	mNm	4.53	5.08	11.3	13.3	13.4	13.2	12.9	12.9	12.8	12.6	12.5
6 Nominal current (max. continuous current)	A	0.84	0.84	0.84	0.814	0.647	0.529	0.33	0.284	0.262	0.214	0.171
7 Stall torque	mNm	35.9	44.1	39.2	38.1	38.2	41.4	31.4	32.5	40.1	40.3	38.5
8 Stall current	A	6.15	6.71	2.83	2.27	1.8	1.62	0.763	0.697	0.797	0.865	0.513
9 Max. efficiency	%	81	81	81	81	81	81	80	80	81	81	81
<b>Characteristics</b>												
10 Terminal resistance	Ω	0.39	0.536	2.12	3.17	4.99	7.41	19.2	25.8	30.1	45.1	70.2
11 Terminal inductance	mH	0.0402	0.0509	0.227	0.332	0.528	0.77	1.9	2.57	2.99	4.34	6.68
12 Torque constant	mNm/A	5.84	6.57	13.9	16.8	21.2	25.5	40.1	46.7	50.3	60.6	75.2
13 Speed constant	rpm/V	1640	1450	689	569	451	374	238	205	190	158	127
14 Speed / torque gradient	rpm/mNm	109	119	105	108	106	108	114	113	114	117	119
15 Mechanical time constant	ms	16.6	16.1	15	14.9	14.9	14.9	14.9	14.9	14.9	15	15
16 Rotor inertia	gcm <sup>2</sup>	14.4	12.9	13.6	13.2	13.3	13.1	12.6	12.6	12.5	12.2	12.1

#### Specifications

17 Thermal resistance housing-ambient	13.2 K/W
18 Thermal resistance winding-housing	3.2 K/W
19 Thermal time constant winding	12.5 s
20 Thermal time constant motor	473 s
21 Ambient temperature	-30...+85°C
22 Max. winding temperature	+85°C

#### Mechanical data (sleeve bearings)

23 Max. speed	6700 rpm
24 Axial play	0.1 - 0.2 mm
25 Radial play	0.012 mm
26 Max. axial load (dynamic)	1.7 N
27 Max. force for press fits (static) (static, shaft supported)	80 N
28 Max. radial load, 5 mm from flange	1200 N

#### Mechanical data (ball bearings)

23 Max. speed	6700 rpm
24 Axial play	0.1 - 0.2 mm
25 Radial play	0.025 mm
26 Max. axial load (dynamic)	5.0 N
27 Max. force for press fits (static) (static, shaft supported)	75 N
28 Max. radial load, 5 mm from flange	1200 N

#### Other specifications

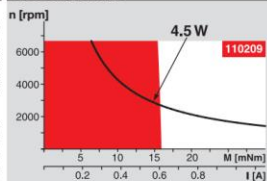
29 Number of pole pairs	1
30 Number of commutator segments	13
31 Weight of motor	119 g

CLL = Capacitor Long Life

Values listed in the table are nominal. Explanation of the figures on page 151.

**Option**  
Ball bearings in place of sleeve bearings  
Without CLL

#### Operating Range



#### Comments

**Continuous operation**  
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.  
= Thermal limit.

**Short term operation**  
The motor may be briefly overloaded (recurring).

Assigned power rating

#### maxon Modular System

<b>Planetary Gearhead</b> Ø26 mm 0.75 - 4.5 Nm Page 336		<b>Spur Gearhead</b> Ø30 mm 0.07 - 0.2 Nm Page 337		<b>Planetary Gearhead</b> Ø32 mm 0.75 - 6.0 Nm Page 338/339/342		<b>Spur Gearhead</b> Ø38 mm 0.1 - 0.6 Nm Page 348		<b>Spindle Drive</b> Ø32 mm Page 370-372		<b>Encoder MR</b> 128 - 1000 CPT, 3 channels Page 392		<b>Encoder Enc</b> 22 mm 100 CPT, 2 channels Page 398		<b>Encoder HED_5540</b> 500 CPT, 3 channels Page 400/402		<b>Encoder MEnc</b> Ø13 mm 16 CPT, 2 channels Page 410	
--	--	---	--	--	--	--	--	--	--	--	--	--	--	---	--	---	--

**Recommended Electronics:**  
 Notes Page 24  
 ESCON Module 24/2 416  
 ESCON 36/2 DC 416  
 EPOS2 24/2 424  
 EPOS2 Module 36/2 424  
 EPOS2 24/5, EPOS2 50/5 425  
 EPOS2 P 24/5 428  
 MAXPOS 50/5 435

Overview on page 20-27

## LAMPIRAN 2

### PENGATURAN PARAMETER ESCON DRIVER

*Driver* Escon perlu diatur sesuai tujuan penggunaan. Pengaturan parameter harus disesuaikan dengan *datasheet* pada motor DC yang digunakan. Maxon motor DC tipe 126691 adalah kombinasi antara *gear drive train* 110395 dan motor DC 110212. Data yang digunakan adalah data dengan tipe motor DC 110212 yang ditunjukkan pada Gambar Lampiran 2-1.

		with terminals	110204	110205	110206	110207	110208	110209	110210	110211	110212
		with cables	353109	353110	353111	353112	353113	353114	353115	353116	353117
<b>Motor Data</b>											
<b>Values at nominal voltage</b>											
1	Nominal voltage	V	2.4	3.6	6	7.2	9	12	15	18	24
2	No load speed	rpm	3890	5190	4090	4060	4020	4440	3530	3640	4510
3	No load current	mA	67.7	69.9	29.2	24	19	16.5	9.41	8.2	8.45
4	Nominal speed	rpm	3460	4640	2940	2650	2620	3030	2070	2180	3060
5	Nominal torque (max. continuous torque)	mNm	4.53	5.08	11.3	13.3	13.4	13.2	12.9	12.9	12.8
6	Nominal current (max. continuous current)	A	0.84	0.84	0.84	0.814	0.647	0.529	0.33	0.284	0.262
7	Stall torque	mNm	35.9	44.1	39.2	38.1	38.2	41.4	31.4	32.5	40.1
8	Stall current	A	6.15	6.71	2.83	2.27	1.8	1.62	0.783	0.697	0.797
9	Max. efficiency	%	81	81	81	81	81	81	80	80	81
<b>Characteristics</b>											
10	Terminal resistance	$\Omega$	0.39	0.536	2.12	3.17	4.99	7.41	19.2	25.8	30.1
11	Terminal inductance	mH	0.0402	0.0509	0.227	0.332	0.528	0.77	1.9	2.57	2.99
12	Torque constant	mNm/A	5.84	6.57	13.9	16.8	21.2	25.5	40.1	46.7	50.3
13	Speed constant	rpm/V	1640	1450	689	569	451	374	238	205	190
14	Speed / torque gradient	rpm/mNm	109	119	105	108	106	108	114	113	114
15	Mechanical time constant	ms	16.6	16.1	15	14.9	14.9	14.9	14.9	14.9	14.9
16	Rotor inertia	gcm <sup>2</sup>	14.4	12.9	13.6	13.2	13.3	13.1	12.6	12.6	12.5

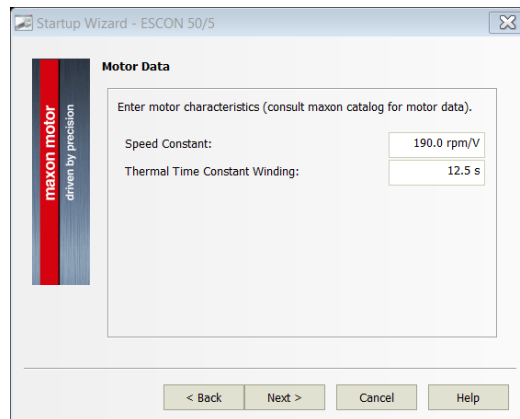
**Gambar Lampiran 2-1 Zoom motor data dari *datasheet* maxon DC motor 26 mm**

<b>Specifications</b>	
<b>Thermal data</b>	
17 Thermal resistance housing-ambient	13.2 K/W
18 Thermal resistance winding-housing	3.2 K/W
19 Thermal time constant winding	12.5 s
20 Thermal time constant motor	473 s
21 Ambient temperature	-30...+65°C
22 Max. winding temperature	+85°C
<b>Mechanical data (sleeve bearings)</b>	
23 Max. speed	6700 rpm
24 Axial play	0.1 - 0.2 mm
25 Radial play	0.012 mm
26 Max. axial load (dynamic)	1.7 N
27 Max. force for press fits (static)	80 N
(static, shaft supported)	1200 N
28 Max. radial load, 5 mm from flange	5.5 N

**Gambar Lampiran 2-2 Zoom *thermal* dan *mechanical* data dari *datasheet* maxon DC motor 26 mm**

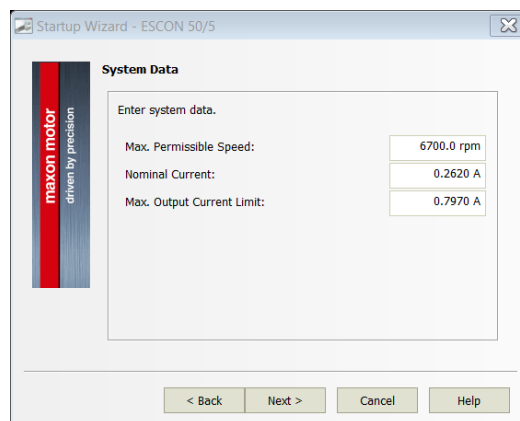
Langkah-langkah pengaturan *driver* motor Escon menggunakan *software* Escon Studio adalah sebagai berikut :

1. *Speed constant* didapat dari *datasheet* motor data nomor 13. *Thermal time constant winding* didapat dari *datasheet* bagian *thermal* data nomor 19.



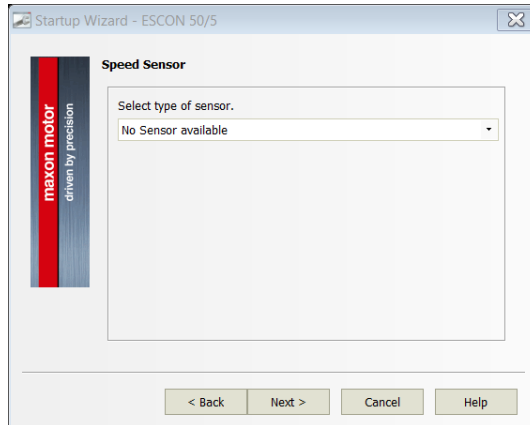
**Gambar Lampiran 2-3 Langkah 1 pengaturan *driver* motor Escon**

2. Nilai *Max. Permissible Speed* didapat dari *datasheet* bagian *mechanical* data nomor 23. Secara berturut-turut, nominal *current* dan *Max. output current limit* didapat dari bagian motor data nomor 6 dan 8.



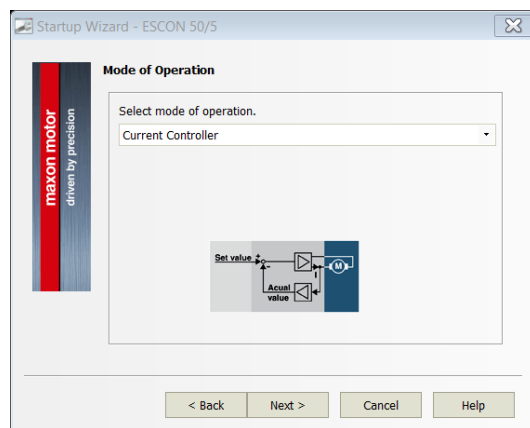
**Gambar Lampiran 2-4 Langkah 2 pengaturan *driver* motor Escon**

3. Pilih no sensor available untuk *speed* sensor, dikarenakan *driver* motor akan digunakan sebagai *converter* nilai PWM menjadi tegangan bukan sebagai *controller* yang membaca nilai kecepatan dari motor.



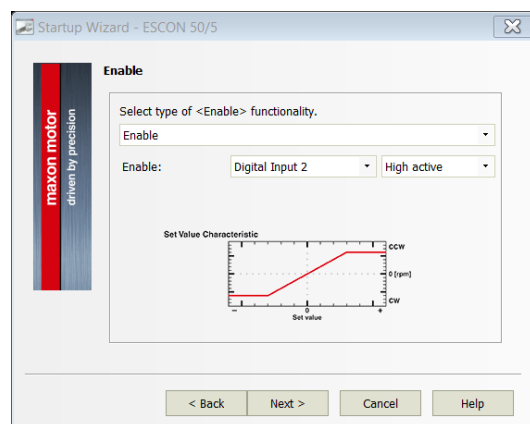
**Gambar Lampiran 2-5 Langkah 3 pengaturan *driver* motor Escon**

4. Pilih *current controller* sebagai *mode of operation*, dikarenakan *driver* Escon digunakan untuk mengatur arus yang masuk ke motor.



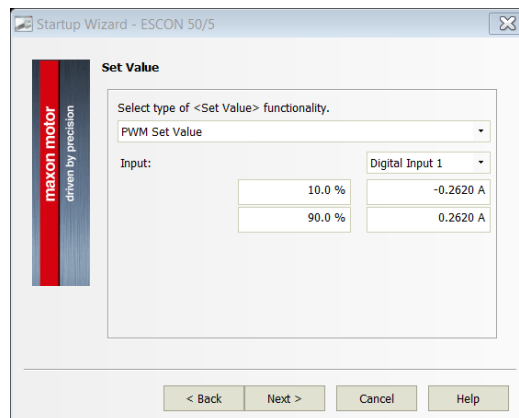
**Gambar Lampiran 2-6 Langkah 4 pengaturan *driver* motor Escon**

5. Pilih *enable* untuk tipe fungsionalitas *<enable>* dan pin *digital input 2* untuk pin *enable* dengan *mode active high*. Pin *digital input 2* digunakan untuk mengaktifkan dan menonaktifkan *driver* motor ketika menerima nilai 1 dan 0 secara berturut-turut.



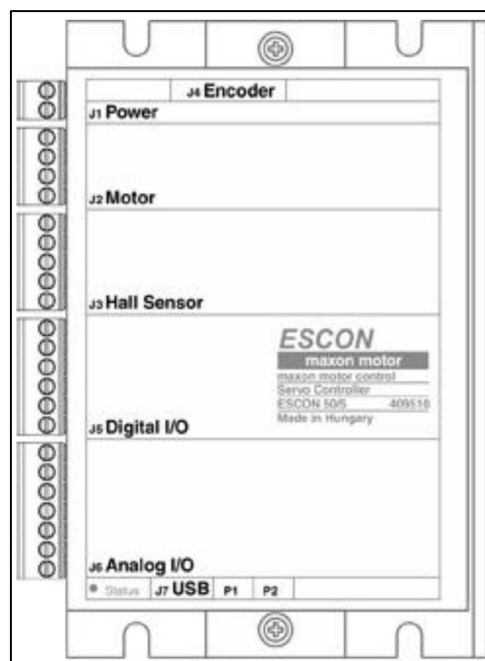
**Gambar Lampiran 2-7 Langkah 5 pengaturan *driver* motor Escon**

6. Arduino UNO sebagai *controller card* akan menggunakan sinyal PWM untuk mengontrol motor DC melalui *driver* Escon. Oleh karena itu maka *PWM Set Value* dipilih sebagai tipe fungsionalitas *<set value>*. Pin yang digunakan untuk menerima sinyal PWM dari Arduino adalah pin *digital* input 1. Lebar pulsa 10.0% diatur dengan nilai arus sebesar -0.2620 A dan 90.0% diatur dengan nilai arus sebesar 0.2620 A, agar dapat mengendalikan motor di kedua arah (searah jarum jam *Clock-Wise* dan berlawanan arah jarum jam *Counter-Clock-Wise*).



Gambar Lampiran 2-8 Langkah 6 pengaturan *driver* motor Escon

Hasil dari pengaturan parameter *driver* motor Escon dapat dilihat pada Gambar Lampiran 2-10 dan Gambar Lampiran 2- 11.



Gambar Lampiran 2-9 Gambar tampak atas *driver* motor Escon 50/5

J1 Power		
Pin	Signal	Configured Purpose
-	Power_GND	Ground of supply voltage
+	+V <sub>cc</sub>	Power supply voltage (+10...+50 VDC)
J2 Motor		
Pin	Signal	Configured Purpose
1	DC: Motor (+M) EC: Motor winding 1	DC motor: Motor+
2	DC: Motor (-M) EC: Motor winding 2	DC motor: Motor-
3	DC: not connected EC: Motor winding 3	-
4	Motor shield	Cabel shield

**Gambar Lampiran 2-10 Pengaturan pin *Power* dan *Motor* pada *driver* motor Escon 50/5**

J5 Digital I/Os		
Pin	Signal	Configured Purpose
1	DigIN1	PWM - Set Value
2	DigIN2	Enable
3	DigIN/DigOUT3	
4	DigIN/DigOUT4	
5	GND	Ground
6	+5 VDC	Auxiliary output voltage (+5 VDC; <=10 mA)

**Gambar Lampiran 2- 11 Pengaturan pin *Digital* pada *driver* motor Escon 50/5**