

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
KONSENTRASI TEKNIK ENERGI LISTRIK**



**APLIKASI LOGIKA FUZZY EXPERT UNTUK
KONTROLER P.I.D PADA PENGENDALIAN
PENDULUM**

SKRIPSI

Disusun Oleh :

**BAMBANG SUTIAWAN
Nim : 99.12.040**



MARET 2006

LEMBAR PERSETUJUAN

**APLIKASI LOGIKA FUZZY EXPERT UNTUK KONTROLER
P.I.D PADA PENGENDALIAN PENDULUM**

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat Guna
Mencapai Gelar Sarjana Teknik*

Disusun Oleh :

BAMBANG SUTIAWAN
NIM : 99.12.040



Mengetahui
Ketua Jurusan Teknik Elektro S-1

Diperiksa dan Disetujui
Dosen Pembimbing

(Ir. F. Yudi Limpraptono, MT)
Nip. Y. 1039500274

(Ir. Widodo Pudji Muljanto, MT)
Nip. Y. 1028700171

**KONSENTRASI TEKNIK ENERGI LISTRIK
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

Lembar Persembahkan



Kupersembahkan "SKRIPSI" ini untuk:

- ❁ Allah Maha Besar yang telah memberikan kasih sayangNya kepada hamba yang hampir putus asa kuliah di ITN MALANG dan engkau memberikan kekuatan kepada hamba, Jalan hambamu masih panjang untuk melangkah menghadapi masa depan, Semoga pertolonganmu selalu datang untuk hamba yang lemah ini. AMIN YA ALLAH
- ❁ ABAH DAN IBU SERTA KELUARGAKU, Anakmu yang selama ini hanya meminta, suatu saat akan mengembalikan apa yang selama ini engkau berikan pada anakmu dengan tulus ikhlas. AMIN YA ALLAH. Doakan anakmu meraih suatu yang diimpikan dan akan selalu membahagiakan kalian. AMIN YA ALLAH.
- ❁ ADEK DAN KELUARGA, Terima kasih "SEMANGATNYA", Kalian beri dorongan dan semangat "NGEB" untuk tetap menimba ilmu di ITN, kalian semua adalah yang terbaik. Adek isteriku tersayang, tercinta, terbaik, Aik, ngak salah memilih Adek. Doakan aik cepat dapat kerja.
- ❁ Sobat - sobatku di ITN:
 1. Jhon oyi, Teman seperjuangan Selama skripsi
 2. Degan, Kapan nang alas
 3. Ekotel, Ojo rencana tok, ayo nang semeru
 4. Dedy, Mandra, Lutfi, Pi'i kabeh wis angel nyebut siji-siji

🍷 **THANKS TO:**

1. Lopez (Edi Doro), Akhirnya kita sama-sama wisuda, ya opo rek durene dek enggone kakek? Sing rukun karo.....
2. Coowy + Ani (Sekeuarga yang imut- imut), Semoga keluarga kalian mendapat barokah. Amin
3. Joko (@dnie.com), matur kesuwun komputerku sering sampeyan benakno, Semangat kuliahnya di UM! Luluse bareng coowy yo he..he
4. A_ghost (Saleem), Ojo ngegem terus, nasehat Dewi kudu didengerno, ojo mokong, ojo ompila – ompila.
5. Radit kriwul (Iwip),Ojo males nang enggone Pak Nugroho tak kandakno emesmu lo yen nakal - nakal
6. Yushadi jessa (junior), meski hari tak selalu cerah dan malam tak selalu berbintang, tetaplah setia sama Ratna yo!
7. Adit(ABM),ojo gendut-gendut. Keluarga sayang kamu
8. Widodo(tembem),Ghofur (wong alas), Aan Sutape, Kaka (suwun pijetane), Samsul Anis (anti kemapanan), kalian konco-konco sing iso diajak nandi-nandi, makasih kebersamaan yang kita tempuh selama ini.
9. Pak Jamin suwun pak kontrakane.....
10. Rocky yang sempat mengisi kontrakaaan dimana kamu man?

🍷 Abah Taufik,Sekeuarga, Dini Lopez, Nofi Bali, Dewi Saleem.

“Yang tak kan pernah hilang adalah persaudaraan diantara kita, biarkan suatu hari detik – detik waktu mempertemukan kita dalam bahagia

Berjaraknya tempat antara kita semua bukan berarti ada jarak dalam persaudaraan kita”

KATA PENGANTAR

Dengan Rahmat Allah SWT dan mengucapkan syukur kehadirat-Nya atas karunia yang telah dilimpahkan kepada penulis sehingga dapat menyelesaikan skripsi yang berjudul “APLIKASI LOGIKA FUZZY EXPERT UNTUK KONTROLER P.I.D PADA PENGENDALIAN PENDULUM”.

Skripsi ini bertujuan memenuhi kurikulum akademik yang harus ditempuh oleh setiap mahasiswa ITN Malang dalam menempuh jenjang pendidikan Strata Satu pada jurusan Teknik Elektro konsentrasi Teknik Energi Listrik. Diharapkan dengan penulisan skripsi ini, dapat memperdalam dan sekaligus melatih mahasiswa agar dapat menerapkan ilmu yang didapat pada waktu kuliah dengan kondisi yang sesungguhnya.

Penulis menyadari bahwa skripsi ini jauh dari sempurna, karena itu saran dan kritik yang bersifat membangun penulis sangat harapkan.

Atas tersusunnya skripsi ini tidak lupa pula penulis mengucapkan terima kasih kepada:

1. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Mochtar Asroni, MSME, selaku Dekan Fakultas Teknologi Industri, Institut Tehnologi Nasional Malang.
3. Ir. F. Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro, Institut Teknologi Nasional Malang.
4. Ir. Widodo Pudji Muljanto, MT, selaku dosen pembimbing yang memberikan ilmu, dorongan semangat dan bantuannya.

5. Rekan-rekan Elektro atas segala kebersamaannya selama ini.
6. Bapak-bapak dan ibu-ibu dosen jurusan Teknik Elektro Energi Listrik
7. Serta semua pihak yang turut membantu dalam penyelesaian skripsi ini.

Kami menyadari bahwa skripsi ini masih jauh dari sempurna, oleh sebab itu kritik dan saran akan kami terima dengan senang hati.

Akhir kata semoga skripsi ini dapat bermanfaat bagi kita semua dalam memperkaya ilmu pengetahuan.

Malang, September 2005

Penyusun

Abstraksi

APLIKASI LOGIKA FUZZY EXPERT UNTUK CONTROLLER P.I.D PADA PENGENDALIAN PENDULUM

(Bambang Sutiawan, Nim : 99.12.040, Teknik Elektro Energi Listrik)

Pembimbing : Ir. Widodo Pudji Muljanto, M.T.

Institut Teknologi Nasional Malang

Kata Kunci : Kontrol P.I.D berbasis Fuzzy Expert

Pengukuran dan pengontrolan sistem PID (*Proporsional, Integral, Dan Diferensial*), sangat sering digunakan dalam sistem kontrol industri. Salah satu kontrol otomatis yang paling banyak digunakan di industri komersial adalah sistem kontrol PID sebagai komponen pengendaliannya. *Controller* ini mentransformasikan sinyal kesalahan menjadi sinyal kontrol secara proporsional, integral dan diferensial.

Sejalan dengan kemajuan teknologi, instrumen digital mulai menggeser instrumen analog. Instrumen digital yang dimaksudkan di sini adalah keperluan sistem informasi yang menggunakan bantuan perangkat lunak (*software*). Keuntungan menggunakan instrumen digital ini, antara lain; biaya lebih ekonomis, pengguna dapat melakukan perubahan secara fungsional, dengan hanya mengubah programnya saja.

Sistem otomatis yang digunakan di sini adalah pendulum, maksud dan tujuan dari instrumen digital di sini adalah mengontrol kecepatan dari pendulum. Pendulum sendiri yaitu sebuah alat atau sistem aplikasi kinerja dari sebuah motor. Di sini pendulum dikontrol oleh Instrumen digital dengan perangkat lunaknya dan bekerja atas dasar motor. Contoh sederhana dari pendulum adalah Penggerak antena parabola digital, di mana antena tersebut dapat bergerak secara terkontrol tanpa menggerakkan tiang – tiang penyangga dari antena itu sendiri. Dan masih banyak lagi contoh lainnya seperti Penggerak radar di bandara, penggerak peluncur roket, dan lain – lain, inti dari pendulum sendiri adalah suatu alat atau sistem yang gerakannya dinamis, terkontrol dan terangkai pada sebuah motor.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL.....	v
DAFTAR GRAFIK	vi
DAFTAR LAMPIRAN.....	vii
BAB I PENDAHULUAN	1
Latar belakang.....	1
Rumusan masalah	2
Batasan masalah	2
Tujuan	2
Relevansi	3
Sistematika penulisan	3
BAB II DASAR TEORI	4
2.1 Tinjauan umum sistem kontrol	4
2.2 Fungsi kontroler	5
2.3 Macam macam kontroler menurut karakteristiknya dan keunggulannya	5
2.3.1 Aksi kontroler proporsional.	5
2.3.2 Aksi kontroler integral	6
2.3.3 Aksi kontroler proporsional ditambah integral.....	7
2.3.4 Aksi kontrol proporsional ditambah turunan.....	8
2.3.5 Aksi kontrol proporsional ditambah integral ditambah turunan	10
2.4 Kontroler P.I.D Analog	12
2.5 Kontroler P.I.D digital	13
2.6 Pengendalian berbasis logika fuzzy.....	17

2.7 Himpunan fuzzy.....	19
2.8 Struktur pengaturan logika fuzzy.....	25
2.9 Aturan dasar fuzzy logic kontrol	28
2.10 Logika pengambilan keputusan	30
2.11 Perencanaan dari fuzzy logic kontrol	36
2.12 Proses strukturisasi logika fuzzy	36
2.1.3 Blok Diagram Alir Algoritma PID	41
BAB III PERENCANAAN SISTEM	42
3.1 Perencanaan model pendulum dengan fuzzy controler	42
3.2 Perencanaan sumber tegangan Pada Pendulum	44
3.3 Perencanaan penentuan arus beban Pada Pendulum	45
3.4 Perencanaan pengaturan kecepatan pendulum	46
3.5 Penentuan P.I.D	47
3.6 Penentuan sudut teta pd pendulum	48
3.7 Penentuan logika fuzzy.....	49
BAB IV HASIL ANALISA	54
4.1 Gerakan Konvensional.....	50
4.2 Gerakan Pendulum Dengan Pengatur Kecepatan Dan Pengaturan Logika Fuzzy	50
4.3 Gerakan Pendulum Dengan Kontrolir P.I.D dan Pengaturan logika Fuzzy	53
4.4 Gerakan Pendulum Dengan Kontrolir P.I.D dan Pengaturan logika Fuzzy Expert.....	56
BAB V PENUTUP	
5.1 Kesimpulan	59
5.2 Saran	59
DAFTAR PUSTAKA.....	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2-1 Sistem Kontrol Loop Tertutup	4
Gambar 2.2 Aksi Kontroler Integral	6
Gambar 2.3 Diagram blok kontroler Integral.....	7
Gambar 2.4 Diagram Blok proporsional ditambah integral.....	8
Gambar 2.5 Diagram Blok Kontroler Proporsional ditambah turunan.....	10
Gambar 2.6 Diagram Blok Kontroler Proporsional ditambah Integral ditambah turunan	11
Gambar 2.7 Diagram Blok kontroler P.I.D	13
Gambar 2.8 Gambar Metode Euler untuk Integrasi Numerik.....	14
Gambar 2.9 Ilustrasi Diferensial Numerik	15
Gambar 2.10 Filter P.I.D Digital	16
Gambar 2.11 Sistem Kontrol Loop Tertutup	17
Gambar 2.12 Gambar perbandingan bentuk fungsi keanggotaan	20
Gambar 2.13 Penafsiran grafis variabel Linguistik.....	24
Gambar 2.14 Gambar struktur pengaturan Logika Fuzzy.....	26
Gambar 2.15 Gambar Flowchart	41
Gambar 3.1 Parameter Percobaan Rangkaian Pengaturan Kecepatan Pendulum Dengan Logika Fuzzy	50
Gambar 3.2 Parameter Percobaan Rangkaian Pengaturan Kecepatan Pendulum Pada PID Dengan Logika Fuzzy	53
Gambar 3.3 Parameter Percobaan Rangkaian Pengaturan Kecepatan Pendulum Pada PID yang ditambah Purelin Dengan Logika Fuzzy	56

DAFTAR TABEL

Tabel 2.1 Rule Base	3 8
---------------------------	-----

DAFTAR GRAFIK

Grafik 4.1.2 Karakteristik Pengaturan kecepatan Pendulum dengan Switch Sebelum Fuzzy -----	51
Grafik 4.1 3 Karakteristik Pengaturan kecepatan Pendulum dengan switch sesudah Fuzzy -----	52
Grafik 4.2 2 Karakteristik Pengaturan kecepatan Pendulum dengan PID Sebelum Fuzzy -----	54
Grafik 4.2 3 Karakteristik Pengaturan kecepatan Pendulum dengan PID Setelah Fuzzy -----	55
Grafik 4.3 2 Karakteristik Pengaturan kecepatan Pendulum dengan PID Sebelum Fuzzy Expert -----	57
Grafik 4.3 3 Karakteristik Pengaturan kecepatan Pendulum dengan PID Setelah Fuzzy Expert-----	58



BAB I

PENDAHULUAN

1.1 Latar belakang

System pengontrolan PID (proposal, integral dan Diferensial), sangat sering digunakan dalam sistem kontrol industri. Dalam pengontrolan ini dibutuhkan berbagai penerapan, misalnya menyajikan prosedur dalam mendesain program dan mengkompensasikan system kontrol linier, parameternya satu masukan dan satu keluaran. Terdapat beberapa macam metode kontrol yang digunakan untuk mengontrol suatu system. Salah satu kontrol otomatis yang paling banyak digunakan di industri komersial adalah system kontrol PID sebagai komponen pengendaliannya. Kontroller ini mentransformasikan sinyal kesalahan menjadi sinyal kontrol secara proporsional, integral dan diferensial.

Sejalan dengan kemajuan teknologi, instrumen digital mulai menggeser instrumen analog. Instrumen digital yang dimaksud disini adalah keperluan system instrumentasi yang menggunakan perangkat lunak (Pemrograman). Keuntungan menggunakan instrumen digital antara lain; biaya lebih ekonomis, pengguna dapat melakukan perubahan secara fungsional, dengan hanya mengubah programnya saja. Dengan latar belakang kecenderungan bidang instrumentasi ke arah instrumentasi digital dan sering dibutuhkan kontroller PID di dalam dunia industri, maka akan dibuat instrumen digital yang menerapkan kotroller PID sebagai komponen utamanya.

1.2. Rumusan Masalah

Berdasarkan permasalahan yang terdapat pada latar belakang, dapat dibuat rumusan masalah sebagai berikut:

- a. Bagaimana merumuskan parameter PID dengan pengaturan logika fuzzy.
- b. Bagaimana merumuskan parameter PID dengan pengaturan Pendulum
- c. Bagaimana mensimulasikan persamaan kecepatan dengan controller PID dan berbasis logika fuzzy.

1.3. Batasan Masalah

Aspek kajian permasalahan yang telah dimasalahkan dalam fenomena masalah ditekankan pada beberapa hal seperti berikut:

- a. Pada analisa PID dengan Pendulum.
- b. Pembahasan ditekankan pada controller PID yang berbasis logika fuzzy dan tidak membahas system kendali keseluruhan.

1.4. Tujuan

Menganalisa suatu kontroler P.I.D yang dapat digunakan dalam penyelesaian pengaturan kecepatan Pendulum dengan menggunakan Logika Fuzzy Ekpert yang digunakan sebagai pengendali.

1.5. Relevansi

Dari analisa yang dilakukan maka sangat diharapkan menghasilkan data-data yang tepat dan sesuai dengan controller PID pd Pendulum dengan logika fuzzy expert dalam keadaan berbeban dan tanpa beban, sehingga dapat diketahui kebenaran dan ketepatan untuk perhitungan pengaman pada Pendulum dapat dianalisa grafis dan perhitungan matematis.

1.6. Sistematika

Penulisan tugas akhir ini terbagi meliputi 5 bab, dengan garis besar penulisannya sebagai berikut:

- Bab I. Membuat latar belakang, tujuan, rumusan masalah, ruang lingkup, relevansi dan sistematika penulisan.
- Bab II. Membahas tentang Konsep Dasar Controller PID pada Pendulum dengan Logika Fuzzy
- Bab III. Perencanaan analisa Pendulum pada keadaan berbeban dan tanpa beban.
- Bab IV. Membahas Hasil Analisa Simulasi
- Bab V. Membuat kesimpulan dan saran.



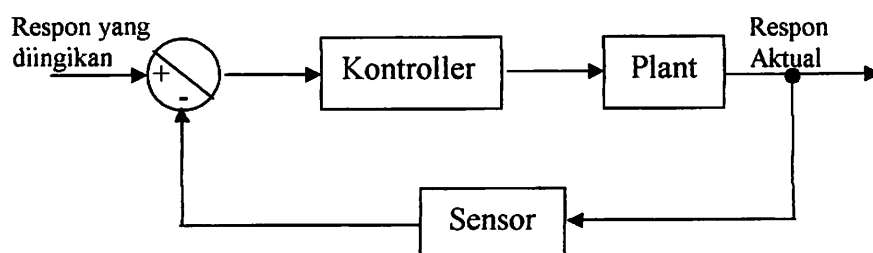
BAB II

DASAR TEORI

2.1 Tinjauan umum Sistem Kontrol

Tinjauan umum yang dimaksud disini merupakan system loop tertutup dengan pemrograman pada system PID. Kontroller berfungsi memodifikasi dinamika system pemograman agar diperoleh respon system yang diinginkan. System kontrol loop tertutup adalah system kontrol umpan balik. System ini cenderung menjaga hubungan yang telah ditentukan antara keluaran dan melakukan acuan dengan membandingkan dengan menggunakan selisihnya sebagai alat pengontrolan.

System loop tertutup sederhana di perlihatkan pada gambar berikut ini:



Gambar 2.1: Sistem kontrol Loop tertutup
Sumber: Philip dan Nogle, 1995; hal:2

System fisik yang dikontrol disebut plant. Sensor yang digunakan untuk mengukur respon plant. Perbedaan antara respon yang diinginkan (Desired Response) dengan actual disebut sinyal beda (Difference Signal) atau sinyal kesalahan (error).

Kontroller adalah komponen yang berfungsi mentransformasikan sinyal kesalahan menjadi sinyal kontrol plant. Pemrograman pada kontroler dapat

membandingkan nilai keluaran plant dengan nilai yang diinginkan, menentukan deviasi, menghasilkan suatu sinyal digital kontrol yang memerkecil deviasi (penyimpanan) sampai nol atau sampai nilai yang paling kecil.

2.2 Fungsi kontroler

Fungsi kontroler adalah untuk mengolah nilai variable kesalahan menjadi nilai variable untuk plant.

Pada prinsip dasarnya suatu system kontroler harus memenuhi persyaratan berikut ini;

1. Respon yang relatif cepat
2. Mempunyai unjuk peredaman yang layak

2.3 Macam-macam Kontroler menurut karakteristik dan keunggulannya

Kontroler otomatis adalah membandingkan nilai sebenarnya dari keluaran system secara keseluruhan (plant) dengan mengacu pada masukan (nilai yang dikehendaki), menentukan penyimpangan, dan menghasilkan sinyal kontrol yang akan mengurangi penyimpangan menjadi nol atau nilai yang kecil. Dengan berbagai cara bagaimana kontroler otomatis tersebut menghasilkan sinyal kontrol dinamakan aksi kontrol.

2.3.1 Aksi Kontrol Proporsional

Kontrol proporsional melakukan tindakan yang sebanding atau proporsional dengan besarnya kesalahan yang terjadi. Untuk kontroler dengan aksi kontrol proporsional, dengan hubungan antara masukan kontroler $u(t)$ dan sinyal pembangkit kesalahan $e(t)$, (Ogata 1996; hal:201).

Persamaan karakteristiknya dapat diberikan;

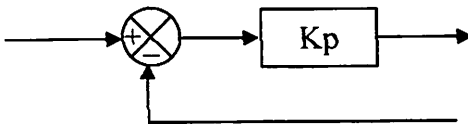
$$u(t) = K_p e(t)$$

Dalam besaran transformasi Laplace;

$$\frac{U(s)}{E(s)} = K_p$$

Dengan K_p adalah suku penguatan proporsional. Pada dasarnya suatu kontroler proporsional adalah penguat dengan penguatan yang dapat diatur.

Dengan blok kontroler ini ditunjukkan sebagai berikut;



Gambar 2.2: Diagram blok kontroler proporsional

Sumber: Ogata, 1996; hal:201

2.3.2 Aksi kontroler Integral

pada kontroler dengan aksi kontrol integral, nilai masukan kontroler $u(t)$ diubah pada laju proporsional dari sinyal pembangkit kesalahan $e(t)$, (Ogata, 1996;hal:2001). Sehingga.

$$\frac{du(t)}{dt} = K_i e(t)$$

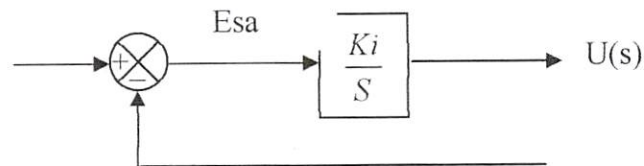
$$\text{atau } u(t) = K_i \int_0^t e(t) dt$$

Dengan K_i adalah konstanta yang dapat diubah. Fungsi alih dari kontroler integral adalah;

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

Jika nilai $e(t)$ ada dua (double), maka nilai $u(t)$ bervariasi dua secara cepat. Untuk pembangkit kesalahan nol, nilai $u(t)$ tetap konstanta. Aksi kontrol integral biasanya disebut kontrol reset.

Gambar dibawah ini menunjukkan diagram blok kontroler integral;



Gambar 2.3: Diagram blok kontroler integral
 Sumber: Ogata, 1996; hal: 202

2.3.3 Aksi kontrol proporsional ditambah integral.

Aksi kontrol kontrolle proporsional ditambah integral didefinisikan dengan persamaan berikut, (Ogata, 1996; hal: 202);

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt$$

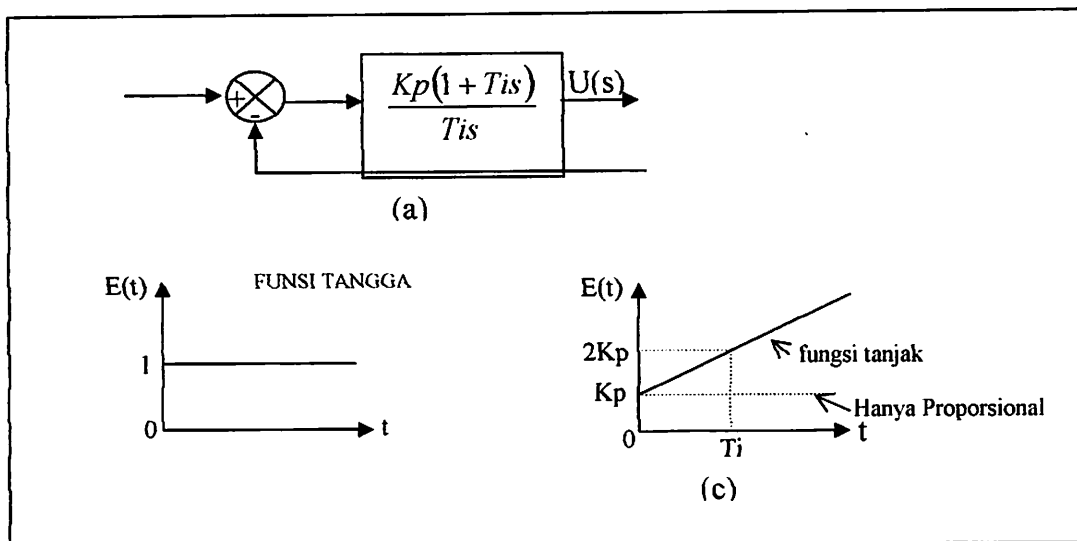
Fungsi alih kontroler ini adalah;

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right)$$

Dengan K_p penguatan proporsional dan t_i disebut waktu integral. Keduanya K_p dan T_i dapat ditentukan. Waktu integral mengatur aksi kontrol integral sedangkan perubahan nilai K_p berakibat pada bagian aksi kontrol

proporsional maupun integral. Kebalikan waktu integral T_i disebut laju reset. Laju reset adalah bilangan yang menunjukkan berapa kali tiap menit bagian proporsional dari aksi kontrol diduplikasi.

Diagram blok proporsional ditambah integral menunjukkan jika sinyal pembangkit kesalahan $e(t)$ adalah fungsi langkah unit seperti gambar (b), maka keluaran controller seperti pada gambar (c);



Gambar 2.4: (a) diagram blok controller proporsional ditambah integral (b) dan (c) diagram masukan langkah unit dan keluaran controller.

Sumber: Ogata, 1996;hal:202

2.3.4 Aksi kontrol proporsional ditambah turunan

aksi kontrol proporsional ditambah turunan didefinisikan dengan persamaan berikut, (Ogata, 1996; hal: 203).

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

Fungsi alihnya adalah;

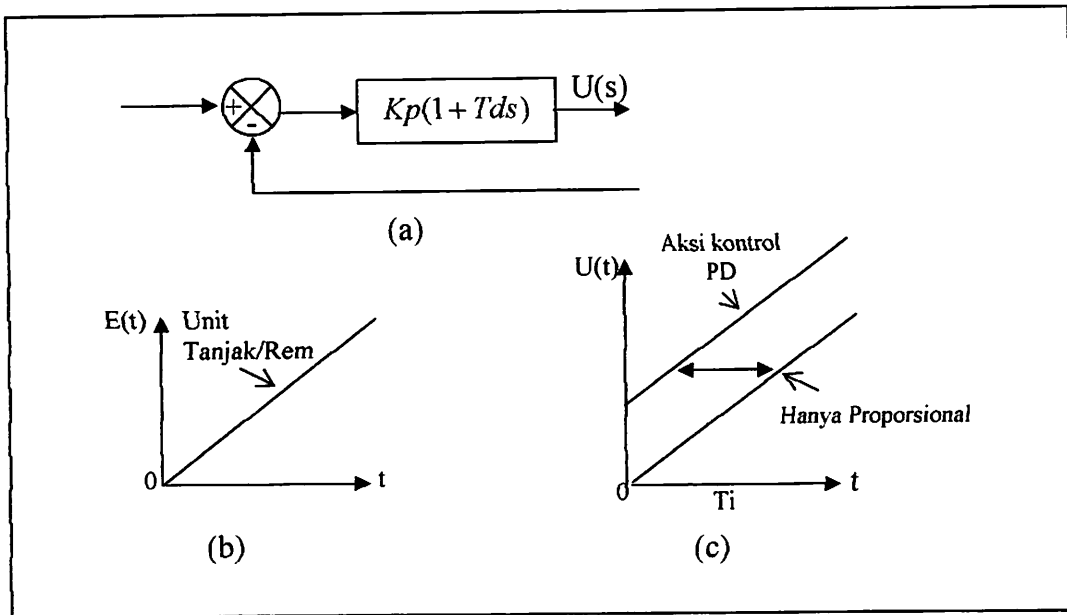
$$\frac{U(s)}{E(s)} = K_p(1 + Tds)$$

Dengan K_p adalah penguatan proporsional dan T_d konstanta yang disebut waktu turunan, K_p dan T_d keduanya dapat ditentukan. Aksi kontrol turunan kadang-kadang disebut laju kontrol dengan besaran keluaran kontroler sebanding dengan laju perubahan sinyal kesalahan. Waktu T_d adalah waktu interval dari laju aksi yang dapat memberikan pengaruh ada aksi kontrol proporsional. Aksi kontrol ini mempunyai keuntungan yang akan melengkapi suatu cara untuk mendapatkan suatu kontroler kepekaan tinggi. Ada juga kelemahannya yaitu adanya gangguan sinyal penguatan yang dapat bercampur pada alat kontrolernya.

Aksi kontrol turunan tidak pernah digunakan sendiri karena aksi kontrol ini hanya efektif selama periode transient. Jika sinyal pembangkit kesalahan $e(t)$ unit fungsi tanjak seperti ditunjukkan pada gambar (b), maka keluaran kontroler menjadi seperti pada gambar (c). seperti dapat dilihat pada gambar (c), aksi kontrol turunan mempunyai karakter antisipasi.

Namun demikian, aksi kontrol turunan tidak mengantisipasi aksi lain yang belum pernah dilakukan

Diagram blok kontroler proporsional ditambah turunan;



Gambar 2.5: (a) Diagram blok kontroler ditambah turunan.
 (b) dan (c) diagram yang menggambarkan unit masukan fungsi landai dan keluaran

Sumber: Ogata, 1996; hal: 203

2.3.5. Aksi Kontrol Proporsional ditambah integral ditambah turunan

Kombinasi aksi kontrol proporsional aksi kontrol integral, dan aksi kontrol turunan disebut aksi kontrol proporsional ditambah integral ditambah turunan. Kombinasi ini mempunyai keuntungan dibanding masing-masing kontroler sebelumnya.

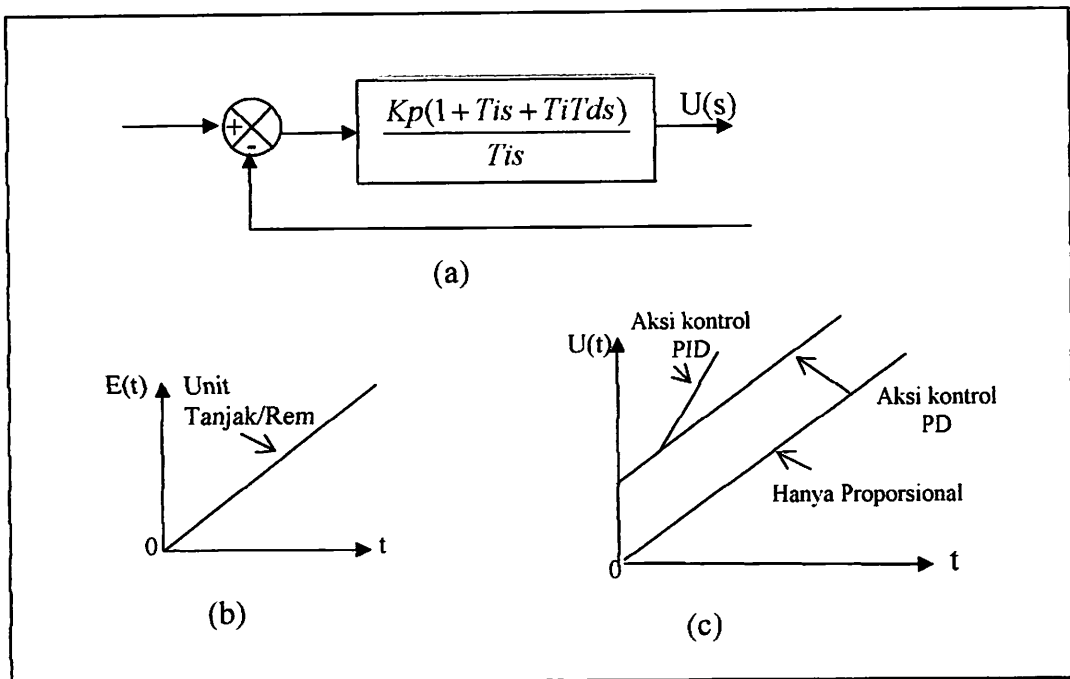
Persamaan ketiga kombinasi ini diberikan oleh : (Ogata, 1996 hal : 203)

$$U(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2-10)$$

Atau fungsi alihnya;

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2-11)$$

dengan K_p penguatan proporsional, T_i waktu integral, dan T_d waktu turunan. Gambar (a), adalah diagram blok kontroler proporsional ditambah integral ditambah turunan. Jika $e(t)$ fungsi landai unit seperti pada gambar (b). keluaran kontroler seperti pada gambar (c) diagram blok dari kontroler diatas adalah sebagai berikut;



Gambar 2.6.

(a) diagram blok kontroler proporsional ditambah integral ditambah turunan.

(b) dan (c) diagram yang menggambarkan unit masukan fungsi landai dan keluaran kontroler

Sumber : Ogata, 1996;hal : 204.

2.4. Kontroler PID Analog

rancangan alat kontrol PID analog (proporsional integral Diferensial) sangat sering digunakan dalam sistem kontrol industri.

PID secara matematis dapat dilukiskan sebagai berikut ; (Ogata, 1996; hal : 166).

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2-12)$$

Dimana ;

K_p = Penguatan proporsional

T_i = Waktu integral

T_d = Waktu turunan

Kontroler PID analog adalah suatu kontroler yang mentransformasikan sinyal kesalahan $e(t)$ menjadi sinyal kontrol $u(t)$ secara proporsional, integral dan diferensial. Kekurangan yang ada pada tipe-tipe sebelumnya.

Bentuk matematis kontroler PID analog dapat dinyatakan dalam persamaan berikut ;

$$u(t) = \text{Klasifikasi}_p \left(e(t) + \frac{1}{T_i} \int_{-\infty}^t e(t) + T_d \frac{de(t)}{dt} \right) \quad (2-13)$$

karakteristik kontroler PID tergantung kepada ketiga sifat yang ada pada parameter P,I dan D. dengan merubah K_p , T_i dan T_d satu atau dua dari ketiga unsur dapat dibuat lebih menonjol dari yang lain. Unsur yang menonjol itulah yang kemudian akan membawa pengaruh pada respon sistem secara keseluruhan.

Untuk konstanta-konstanta K_p , T_i dan T_d adalah parameter-parameter alat kontrol pada rumus yang pertama, dapat juga dituliskan sebagai berikut :

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (2-14)$$

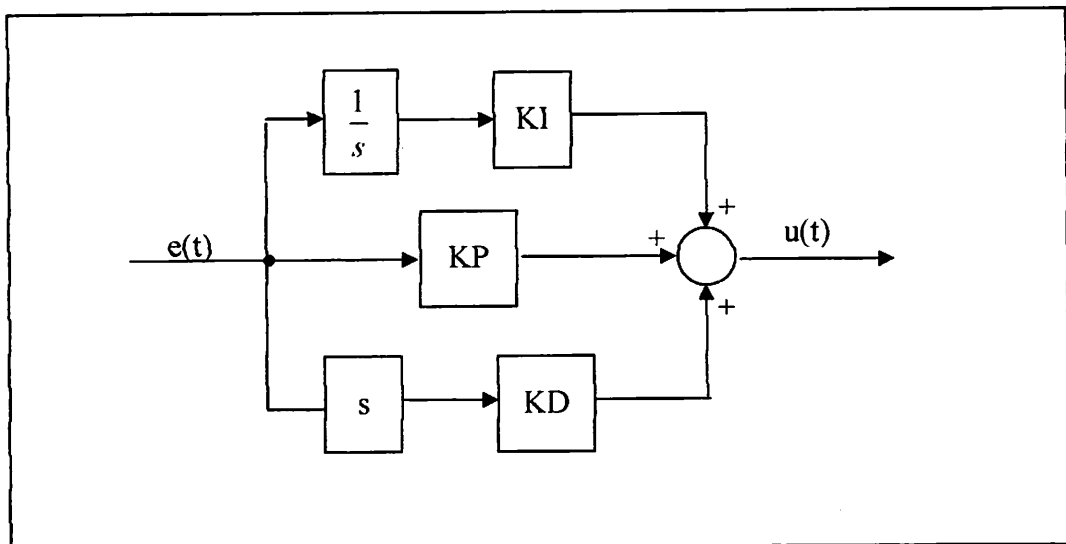
Dengan;

K_p = penguatan proporsional

K_i = penguatan integral

K_d = penguatan turunan (derivatif)

Dalam hal ini, K_p , K_i dan K_d menjadi parameter-parameter alat kontrol PID yang aktual, sebagai pengganti penyetyelan penguatan proporsional.



Gambar 2.7: Diagram Blok Kontroler PID

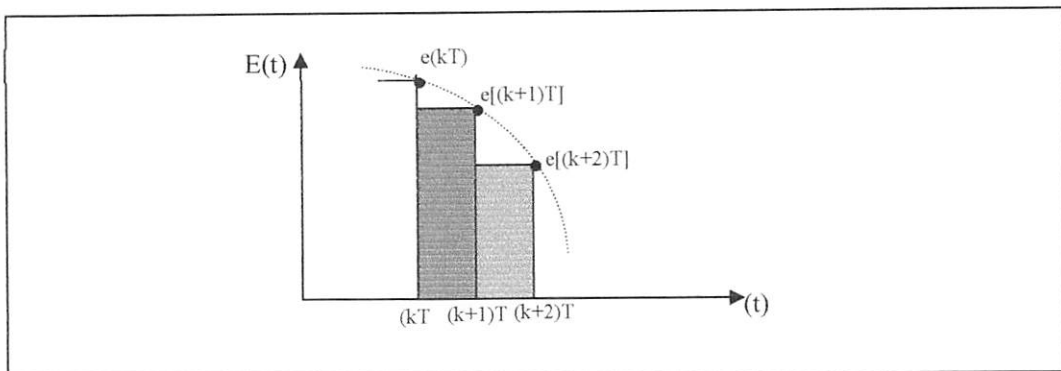
Sumber : Philips & harbor, 1996; hal : 332

2.5. Kontroler PID digital

Kontroler PID digital merupakan teknik lain yang dikembangkan untuk menemukan fungsi keluaran pada sistem waktu diskrit dan waktu linear umum. Pengendali PID digital juga harus memenuhi persamaan untuk sistem waktu

kontinyu sebagaimana persamaan (2-13). Untuk mengubah persamaan 2-13 tersebut menjadi bentuk setara diskritnya, maka dapat digunakan metode Euler (aturan segi empat siku-siku) sebagaimana uraian berikut ini. Untuk mengubah suku integral dari kondisi kontinyu ke diskrit, ditampilkan dalam gambar 2.8.

dalam diagram tersebut, area dibawah $e(t)$ dibagi atas segmen-segmen dengan lebar sama



Gambar 2.8: Metode Euler Untuk Integrasi Numerik

Sumber : Philips & harbor, 1996; hal : 148.

Bila $m(t)$ adalah integral $e(t)$, nilai integral pada $t = [(Klasifikasi + 1) T]$ setara dengan nilai $t = KT$ ditambah area di bawah kurva $e(t)$ antara KT dan $(Klasifikasi + 1)T$

Menurut aturan Euler;

$$M[(K + 1)T] + Te [(Klasifikasi + 1) T] \quad (2 - 15)$$

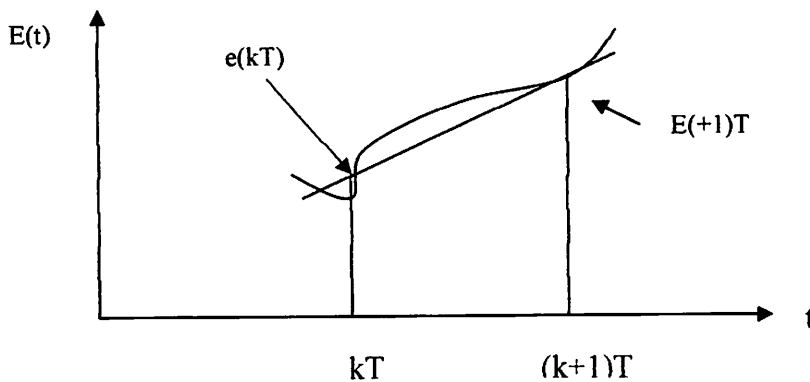
Persamaan beda (2- 15) merupakan aturan Euler untuk; integral numerik, maka transformasi – z dari persamaan ini menghasilkan;

$$z[M(z) - m(0)] = M(z) + Tz [E(z) - e(0)] \quad (2-16)$$

Fungsi alih integrator, dengan mengabaikan kondisi awal adalah;

$$M(z) = \frac{Tz}{z-1} E(z) \quad (2-17)$$

Diferensi numerik dapat dikerjakan dengan menggunakan metode yang digambarkan dalam gambar berikut ini ;



Gambar 2.9.: Ilustrasi diferensial numerik

Sumber : Philips & harbor, 1996; hal : 149.

Metode ini diasumsikan bahwa kemiringan $e(t)$ pada $t = (Klasifikasi + 1)T$ sama dengan kemiringan garis lurus yang menghubungkan $e(kT)$ dengan $e[(Klasifikasi + 1)T]$. bila $m(t)$ adalah turunan $e(t)$, prosedur diferensial numerik ini dinyatakan oleh ;

$$M [(K+1) T] = \frac{e[k + 1]T] - e(kT)}{T} \quad (2-18)$$

Transformasi-z dari persamaan ini menghasilkan fungsi alih;

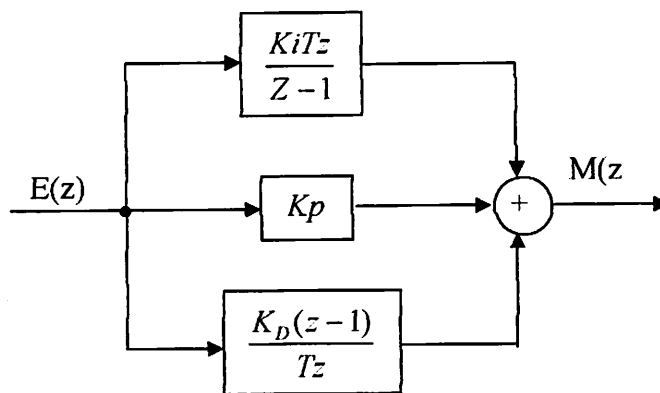
$$M(z) = \frac{z-1}{Tz} E(z)$$

Fungsi alih ini merupakan kebalikan dari integrator aturan Euler tersebut. Pendekatan lain, fungsi alih integrator numerik dan diferensiator

numerik, tidak unik untuk pengendali PID digital namun digunakan untuk banyak pengendali komersial yang ada. Dengan persamaan ini untuk membentuk fungsi alih pengendali $D(z)$;

$$M(z) = D(z)E(z) = \left[K_p + \frac{K_i T z}{z-1} + \frac{K_D (z-1)}{T z} \right] E(z)$$

Pengendalian PID digital dapat diperlihatkan pada gambar di bawah ini;



Gambar 2.10.: Filter PID digital

Sumber: Phillips & Harbor, 1996; hal: 149.

Untuk menentukan beda pengendali PID, kita merealisasikan pengendali dengan menggunakan persamaan beda pengendali di atas;

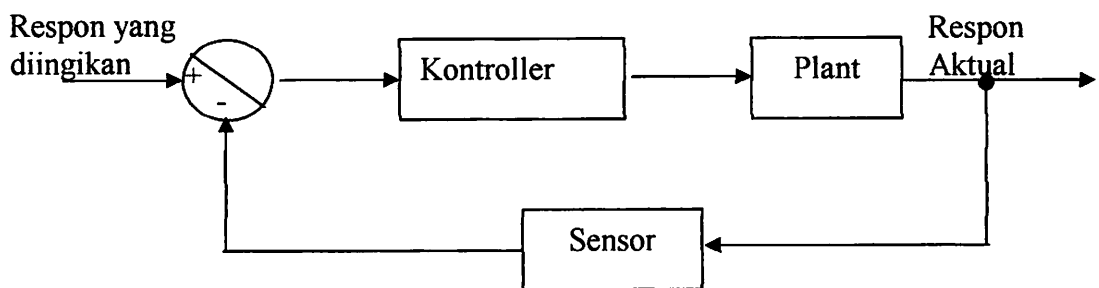
$$m(k+1) = K_p e(k+1) + K_i m_i(k+1) + \frac{K_D [(k+1) - e(k)]}{T}$$

Dengan $m(k+1)$ adalah; $m_i(k+1) = m_i(k) + T e(k+1)$

2.6 Pengendalian Berbasis Logika Fuzzy

Sistem kontrol adalah pengendalian komponen-komponen fisik yang didesain untuk berubah, untuk mengendalikan, atau untuk memerintahkan melalui suatu aksi kontrol (*control action*), sistem fisik yang lain sehingga dia menunjukkan suatu karakteristik atau kinerja yang dikehendaki. Pada *closed-loop control system*, *control action* tergantung pada *output* fisik dari sistem. Untuk mengontrol suatu variabel fisik, kita harus mengukurnya. Sistem untuk pengukuran sinyal kontrol disebut sensor, sistem fisik yang dikontrol disebut plant. Dalam *closed loop control system*, suatu *forcing signal* dari sistem (yang disebut input) ditentukan oleh respon dari sistem (yang disebut output). Untuk mendapatkan respon dan karakteristik yang memuaskan untuk *close loop control system*, perlu menghubungkan suatu sistem tambahan, yang disebut *compensator* atau *controller*, kedalam loop. Bentuk umum dari *close loop control system* terlihat pada gambar 1.

System loop tertutup sederhana di perlihatkan pada gambar berikut ini:



Gambar 1: Sistem kontrol Loop tertutup

Sistem kontrol kadang-kadang dibagi dalam dua kelas. Bila objek dari sistem kontrol adalah untuk mempertahankan variabel fisik pada suatu harga yang konstan bila ada suatu gangguan, maka sistem tersebut dinamakan tipe kontrol regulator. Contoh dari sistem ini adalah pengaturan temperatur dalam ruangan atau pengaturan kecepatan yang konstan dari suatu mesin listrik. Bentuk yang lain adalah *tracking controller*. Dalam sistem invariabel fisik digunakan untuk mengikuti track fungsi waktu yang diinginkan. Contoh dari sistem ini adalah *automatic aircraft landing system*.

Problema dalam sistem kontrol dinyatakan sebagai berikut. Output atau respon dari sistem yang dikontrol (*plant*) diatur seperti yang dikehendaki oleh *signal error*. *Signal error* tersebut merupakan ketimpangan yang terjadi pada *output plant*, sementara *signal error* tersebut terkakulasi pada *plant* yang setiap saat terkoreksi, sehingga terjadilah perubahan error yang dialami oleh motor dan ini dilakukan agar didapatkan perbedaan antara respon yang sebenarnya dari *plant*, yang diukur oleh sensor dari sistem, dan respon yang dikehendaki, seperti yang dinyatakan oleh input referensi.

Untuk suatu sistem dengan *reference* input yang konstan, bentuk dari *controller* adalah *Propositional-Differential and Integral* (PID), namun penggunaan sistem ini memiliki kelemahan yaitu :

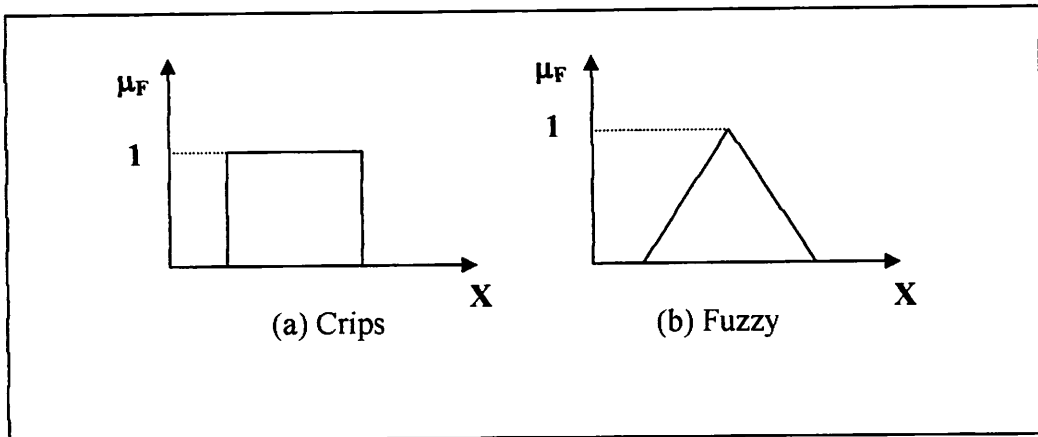
- Adanya overshoot pada respon
- Apabila overshoot ingin dikurangi maka time respon menjadi lambat untuk mencapai settling point yang diinginkan.

Berdasarkan kelemahan controller PID tersebut dewasa ini telah banyak dikembangkan penggunaan *controller fuzzy* sebagai kontrol kecepatan, penggunaan sistem kontrol ini dapat mengatasi kelemahan yang dimiliki oleh PID kontrol.

2.7 Himpunan Fuzzy

Konsep himpunan fuzzy (*fuzzy set*) pertama sekali diperkenalkan oleh Lutfi A. Zadeh pada tahun 1965. Konsep ini dipresentasikan dalam menganalisa fenomena alam yang selalu bersifat tidak tepat (*inexact*), bila ditinjau cara berfikir manusia, cara berfikir manusia pada umumnya tidak hanya berupa bilangan tetapi lebih berupa penggolongan dengan batasan yang samar (*fuzzy*), seperti kecil sedang, besar dan seterusnya. Suatu himpunan fuzzy terdiri dari elemen-elemen yang mempunyai derajat keanggotaan μ , suatu elemen x dalam himpunan bagian fuzzy F mempunyai derajat keanggotaan $\mu_f(x)$ yang terletak antara 0 dan 1. Jika $\mu_f(x) = 1$, maka x adalah keanggotaan himpunan F . jika $\mu_f(x) = \mu$. dengan $0 < \mu < 1$ maka dikatakan fungsi F dengan derajat keanggotaan μ . Dalam merujuk himpunan bagian fuzzy ini sering digunakan besaran yang nonnumerik, misalnya "besar, sedang, kecil, dan sebagainya. Suatu elemen x selain memiliki derajat keanggotaan terhadap himpunan bagian fuzzy F juga bisa memiliki derajat keanggotaan terhadap himpunan bagian fuzzy lain, derajat keanggotaan terhadap suatu elemen x dalam suatu himpunan bagian fuzzy dihitung melalui fungsi distribusi, yang bisa didefinisikan menurut keperluan. Jenis fungsi distribusi yang populer adalah segitiga, trapesium, eksponensial, dari bentuk-bentuk tersebut diperlukan suatu analisa yang berbeda tergantung dari kebutuhan dan keperluan

dari penggunaan sistem tersebut, perbandingan fungsi keanggotaan dari himpunan fuzzy dan crips dapat dilihat pada gambar 2-



Gambar. 2-. Perbandingan Bentuk Fungsi Keanggotaan

Sementara definisi dan operasi dasar himpunan fuzzy, adalah suatu himpunan fuzzy F dalam semesta X, didefinisikan sebagai kumpulan pasangan elemen x, dan fungsi keanggotaan $\mu_f(x)$, dengan nilai interval [0,1] pada tiap x dalam X. nilai fungsi keanggotaan menunjukkan tingkat keanggotaan elemen x dalam F, secara umum himpunan dinotasikan sebagai.

$$F = [x, \mu_f(x) | x \in X] \tag{2-24}$$

Bila diskrit dengan elemen, F dapat dinyatakan sebagai :

$$F = [\mu_1(x)/x_1 + \mu_2(x)/x_2 + \dots + \mu_n(x)/x_n] \tag{2-25}$$

$$F = \sum_{i=1}^n \mu_f(x_i)/x_i$$

Bila x kontinyu, F dapat dinyatakan sebagai :

$$F = \int_x \mu_f(x)/x \tag{2-26}$$

Didalam himpunan fuzzy F , x disebut pendukung (support) F , untuk elemen x dimana $\mu_f(x) = 0,5$ disebut titik silang (*crosss-over*). Himpunan fuzzy yang mempunyai pendukung tunggal dengan $\mu_f(x) = 1$, disebut fuzzy tunggal, misal A dan B adalah himpunan fuzzy dalam semesta X dengan fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$. Beberapa operasi dasar dari himpunan fuzzy yang sering digunakan adalah sebagai berikut :

- Gabungan fuzzy (union), jika C adalah gabungan dari A dan B atau $C = A \cup B$, masing-masing mempunyai fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$. maka fungsi keanggotaan himpunan fuzzy adalah

$$\mu_C(x) = \max [\mu_A(x), \mu_B(x)] \quad (2-27)$$

$$\mu_C(x) = \mu_A(x) \cap \mu_B(x) \quad (2-28)$$

- Irisan fuzzy (intersection) jika C adalah irisan dari A dan B atau $C = A \cap B$, yang masing mempunyai fungsi keanggotaan himpunan fuzzy C adalah :

$$\mu_C(x) = \min [\mu_A(x), \mu_B(x)] \quad (2-29)$$

$$\mu_C(x) = \mu_A(x) \cap \mu_B(x) \quad (2-30)$$

- Komplemen fuzzy adalah fungsi keanggotaan dari komplemen himpunan fuzzy A

$$\mu_A(x) = 1 - \mu_B(x) \quad (2-31)$$

- Perkalian Aljabar adalah himpunan fuzzy A dan B atau A, B adalah :

$$\mu_{AB}(x) = \mu_A(x)\mu_B(x), x \in X \quad (2-32)$$

➤ Penjumlahan Aljabar adalah penjumlahan aljabar himpunan A dan B atau

$$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x), x \in X \quad (2-33)$$

Fungsi keanggotaan fuzzy dalam banyak hal sangat membantu dan perlu untuk menyatakan fungsi keanggotaan fuzzy dalam bentuk standar, untuk menyatakan suatu fungsi keanggotaan dalam pendefinisian himpunan fuzzy pada bentuk yang akan digunakan, secara umum ada dua metode yaitu pendefinisian secara numerik dan bentuk fungsi.

- *Pendefinisian secara numerik*, fungsi keanggotaan yang didefinisikan secara numerik dengan menggunakan pendukung diskrit, selain itu dapat juga menggunakan nilai keanggotaan dari bentuk fungsi yaitu dengan mengambil nilai bentuk fungsi tiap pendukung x yang berhingga jumlahnya, contoh :

$x = \{8,9,10,11,12\}$, $\mu_F(x) = \{5,5; 0,8; 0,5\}$, maka himpunan fuzzy didefinisikan untuk himpunan elemen x diatas adalah :

$$F = \{0,5/8; 0,8/9; 1,0/10; 0,8/11; 0,5/12\}.$$

- *Pendefinisian dalam bentuk fungsi* adalah bentuk fungsi digunakan pada pendukung kontinyu, ada beberapa fungsi yang dapat digunakan dalam himpunan fuzzy, bentuk fungsi yang sering digunakan adalah :

$$\text{- Fungsi Exponensial : } \mu_F(x) = \exp\left\{-\frac{(x-u)}{2\sigma^2}\right\} \quad (2-34)$$

$$\text{- Fungsi Segitiga : } \mu_F(x) = 1 - \sqrt{\frac{(x-a)^2}{b}} \quad (2-35)$$

$$\text{- Fungsi Trapesium : } \mu_F(x) = \left[\begin{array}{l} 1; 0 \leq (x - a) \leq \frac{b}{2} \\ 2 - 2 \frac{\sqrt{(x - a)^2}}{b}; \frac{b}{a} \langle (x - a) \leq b \end{array} \right] \quad (2-36)$$

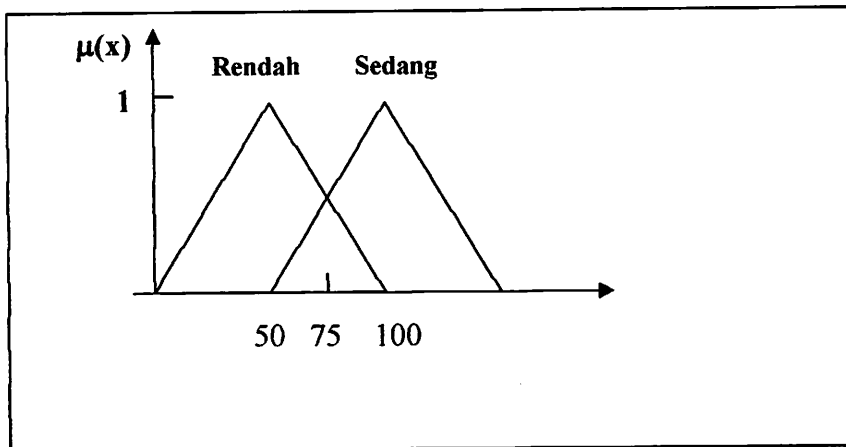
Penggunaan variabel lingustik dilakukan dengan himpunan fuzzy untuk mempresentasikan cara berfikir manusia sebagai pengganti dari variabel numerik, yang biasanya digunakan dalam pendekatan kualitatif. Variabel lingustik berupa ungkapan yang bersifat kualitatif dan berfungsi untuk menyediakan himpunan fuzzy, dalam manusia berfikir menggunakan ungkapan yang bersifat kualitatif terhadap informasi yang diindranya. Ungkapan-ungkapan kualitatif ini terus menyertai proses berfikir manusia sampai diperoleh pengambilan keputusan yang diungkapkan secara kualitatif juga. Dalam bentuk sederhana variabel lingustik dapat dinyatakan dengan pasangan $(u, T(u), X)$, dengan u menunjukkan nama variabel dan $T(u)$ merupakan istilah yang menyatakan himpunan a yaitu seperangkat nama nilai lingustik himpunan fuzzy atas semesta X . sebagai contoh misalkan : variabel level akan dinyatakan dengan variabel lingustik, $T(\text{level})$ dapat ditulis : = [rendah, sedang]

Sementara variabel lingustik didefinisikan untuk semesta $X = [0, 150]$, dalam semesta ini, rendah terletak disekitar 50 mm dan sedang disekitar 100 mm, bila fungsi segi tiga digunakan untuk mendefinisikan secara fungsional kedua himpunan ini maka cara menyatakan adalah :

$$\text{Rendah : } \mu_F(x) = 1 - \frac{\sqrt{(x - 50)^2}}{b}$$

$$\text{Sedang : } \mu_F(x) = 1 - \frac{\sqrt{(x - 100)^2}}{b}$$

Dimana $x \in X$. Harga b dipilih sedemikian rupa sehingga titik silang (cross over) terletak dititik $x = 75$ mm, yaitu pendukung dengan nilai keanggotaan $\mu(75) = 0,5$ baik pada himpunan rendah maupun sedang, penafsiran secara gratis semua pendefinisianya ditunjukkan pada gambar 2.8.



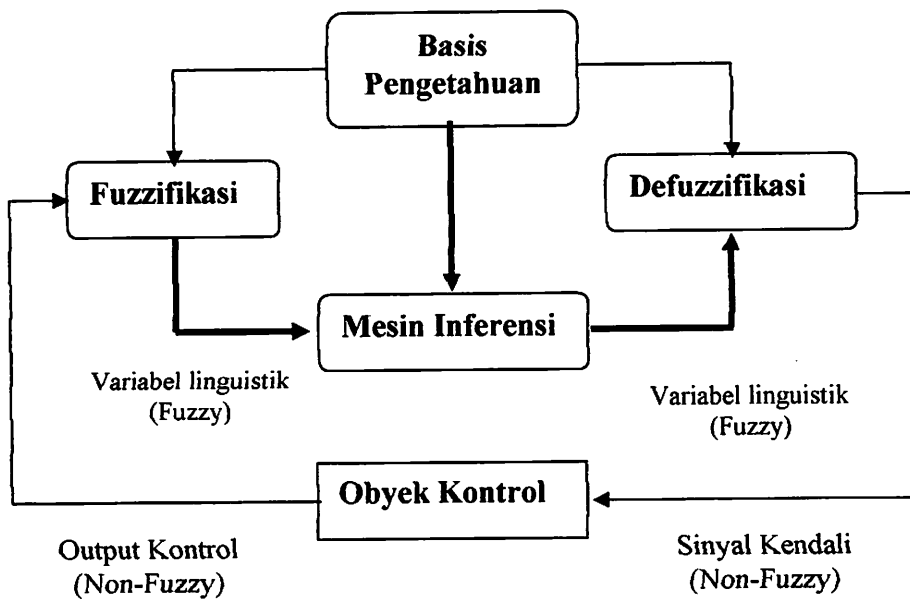
Gambar 2-8. Penafsiran Grafis Variabel Linguistik

Gambar tersebut diatas memberikan penafsiran bahwa nilai keanggotaan untuk suatu titik semakin besar bila titik tersebut berada disekitar titik 50 dan 100, sedangkan contoh pada titik 55 mempunyai nilai keanggotaan yang mendekati 1 pada himpunan rendah, yang berarti titik ini mempunyai nilai kebenaran yang kuta untuk menjadi anggota himpunan rendah, sebaliknya karena mempunyai nilai keanggotaan yang kecil pada himpunan sedang maka titik ini mempunyai nilai kebenaran yang lemah untuk menjadi anggota himpunan yang sedang. Sedangkan untuk titik 75 mempunyai nilai kebenaran yang sama untuk menjadi anggota rendah maupun sedang.

2.8. Struktur Pengaturan Logika Fuzzy

Perancangan pengatur logika fuzzy (*fuzzy logic control*) atau *PLF* menggabungkan aspek pendefinisian himpunan fuzzy dengan aspek logika fuzzy untuk memperoleh suatu controller yang dapat mempresentasikan cara kerja operator manusia. Dengan prosedur perancangan tertentu kedua aspek diterapkan pada masukan dan keluaran untuk membentuk hasil perancangan yang berbentuk suatu algoritma aturan fuzzy, perancangan pengatur logika fuzzy selama ini tidak mempunyai prosedur yang kaku, hal ini disebabkan oleh fleksibilitas pengaturan logika fuzzy terhadap sebagai macam plant. Secara umum pengaturan logika fuzzy mempunyai empat bagian pokok seperti yang ditunjukkan pada gambar 2.9. keempat blok tersebut saling berkaitan dalam menjalankan fungsinya, adapun fungsinya.

1. Fuzzifier yang berfungsi untuk mentransformasikan sinyal masukan yang berbentuk crips (bukan fuzzy) himpunan fuzzy dengan menggunakan operator fuzzifier.
2. Basis pengetahuan berisi basis data dan aturan dasar yang mendefinisikan himpunan fuzzy atas-atas daerah masukan dan keluaran, dan menyusunnya dalam perangkat aturan kontrol.
3. Logika pengambilan keputusan merupakan inti dari pengaturan logika fuzzy yang mempunyai kemampuan seperti manusia dalam mengambil keputusan, aksi atur fuzzy disimpulkan dengan menggunakan implikasi fuzzy mekanisme inference fuzzy.
4. Defuzzifier berfungsi untuk mentransformasikan kesimpulan tentang aksi atur yang bersifat crips dengan menggunakan operator defuzzifier.



Gambar. 2.9. Blok Diagram Pengaturan Berbasis Logika Fuzzy

Dalam penerapan fuzzifikasi pada sistem pengaturan, besaran masukan yang diperoleh dari plant selalu berupa crisp yang bersifat pasti dan kuantitatif. Sedangkan pengolahan data dalam pengaturan logika fuzzy didasarkan pada teori himpunan fuzzy yang menggunakan variabel linguistik yang bersifat fuzzy, oleh karena pada tahap awal pengaturan logika fuzzy, diperlukan adanya fuzzifikasi yang dilakukan oleh fuzzifier. Fuzzifikasi sekaligus dapat dikatakan sebagai pemetaan pemasukan kesemesta himpunan fuzzy, secara simbolis pemetaan ini dinyatakan dalam $X = \text{fuzzifier}(X_0)$, X_0 adalah masukan crisp, x adalah himpunan fuzzy dan fuzzifier merupakan operator fuzzifikasi. Metode yang sering dipakai adalah dengan memperlakukan masukan crisp yang diperoleh sebagai fuzzy tunggal (*singlelato fuzzy*), secara mendasar fuzzy tunggal adalah suatu nilai yang pasti yang bersifat bukan fuzzy, jadi jelas dalam hal ini tidak ada kefuzzian dalam data yang dihasilkan. Sementara basis data yang berfungsi untuk mendefinisikan himpunan-himpunan fuzzy dari sinyal masukan dan sinyal keluaran agar dapat

digunakan oleh variabel linguistik dalam aturan dasar. Perencanaan basis data meliputi *Kuantitas* dan *Normalisasi*, kuantitas berarti mendeskripsikan semesta pembicaraan yang kontinyu, kedalam sejumlah segmen-segmen tertentu yang disebut level kuantitas. Pemberian nomor atau level-level ini membentuk pendukung himpunan fuzzy secara berhingga atau semesta pembicaraan baru yang bersifat diskrit, prosedur ini diperlukan bila mendefinisikan fungsi keanggotaannya dinyatakan secara numerik, yang penerapannya pada suatu tabel pandang (*look-up table*), tabel tersebut mendefinisikan secara numerik misalnya dengan pembagian ruang sebanyak lima (5) variabel linguistik yaitu : Negatif Besar (*NB*), Negatif Kecil (*NK*), Nol (*NL*), positif kecil (*PK*), Positif Besar (*PB*).

Pemilihan jumlah level kuantitas (penyokong) mempengaruhi kepekaan pengatur logika fuzzy terhadap masukan dan kehalusan aksi atur pada output, semakin banyak level kuantitas yang dihasilkan pada semesta pembicaraan masukan dan keluaran maka semakin peka dan semakin halus aksi aturnya. Tetapi karena tabel pandang menggunakan jumlah memori pada komputer maka dalam menentukan jumlah level harus ada kompromi dengan aspek penghematan memori, Normalisasi diperlukan bila diinginkan semesta pembicaraan yang terbatas pada jangkauan tertentu, jadi normalisasi merupakan pemetaan semesta pembicaraan masukan kesemesta pembicaraan baru yang terbatas, skala pemetaan bisa seragam (*uniform*) atau tidak seragam (*non-uniform*), tergantung kebutuhan perancangan. *Pembagian ruang masukan dan keluaran* adalah pendefinisian himpunan fuzzy atas ruang masukan dan keluaran berarti ulang membagi-bagi semesta pembicaraan atas nilai-nilai variabel linguistik seperti *Negatif Besar*, (*NB*), *Negatif Sedang* (*NS*), *Negatif Kecil* (*NK*), *Nol* (*NL*), *Positif Kecil* (*PK*),

Positif Sedang (NS), Positif Besar (PB), dan sebagainya. Pembagian ruang masukan dan keluaran ini menentukan berapa banyak jumlah himpunan, ini juga menentukan kehalusan pengatur logika fuzzy. *Pemilihan Fungsi Keanggotaan* pada pendefinisian secara numerik, tingkat keanggotaan penyokong dalam himpunan fuzzy dinyatakan dalam bentuk tabulasi, pembagian ruang dilakukan dengan tujuh nilai linguistik himpunan fuzzy yaitu negatif besar, negatif sedang, negatif kecil, nol, positif kecil, positif sedang, positif besar. Dengan fungsi keanggotaan adalah segitiga :

Fungsi keanggotaan dapat dipilih secara bebas dengan menentukan dengan cara sembarang nilai keanggotaannya, tetapi pada dasarnya penentuan ini harus tetap dengan menggambarkan karakteristik masing-masing fuzzy. Fungsi yang sering digunakan adalah fungsi eksponen (η), fungsi segitiga. Pembagian ruang biasanya cukup dengan menentukan titik sumbu simetri dan sembarang fungsi yang digunakan, fungsi ini mudah diadaptasi terhadap keadaan semesta pembicaraan dengan cara mengatur parameter retaan yang menentukan titik kerja dan parameter sebarang yang menentukan jangkauan kerja. Parameter retaan (u dan a) dan parameter sebarang (b dan s) dapat dipilih sebarang, tetapi dengan pertimbangan perancangan yang baik, dalam hal parameter, sebarang seragam biasanya sebarang dipilih sedemikian rupa sehingga titik silang (*Cross-over*) terletak ditengah-tengah dua retaan himpunan yang bersebelahan. Contoh penggunaan fungsi eksponen (η), fungsi segitiga dan fungsi tranposium

2.9. Aturan Dasar Fuzzy Logic Control

Sistem fuzzy dikarakteristikan oleh istilah-istilah linguistik yang didasarkan pada pengetahuan pakar dan biasanya berbentuk aturan Jika - Maka (If Then), yang disebut aturan kontrol fuzzy. Hal ini untuk mempermudah implementasi bentuk kondisional fuzzy, seperangkat aturan kontrol fuzzy merupakan penjabaran dari bentuk-bentuk kondisional fuzzy dalam aturan dasar pada pengaturan logika fuzzy. Perencanaan aturan kontrol meliputi :

1. Pemilihan variabel masukan dan keluaran
 2. Penurunan aturan kontrol fuzzy
- *Pemilihan Variabel Masukan Dan Keluaran*

Pemilihan variabel masukan dan keluaran memberikan pengaruh kuat pada karakteristik controller, pengetahuan dan pengalaman rekayasa memainkan peranan yang sangat penting. Pemilihan variabel ini memerlukan pemahaman tentang perilaku plant dan perilaku pengaturannya, secara umum sebagai mana controller yang konvensional, error dan perubahan error dari plant tetap menjadi acuan utama aturan kontrol pengaturan logika fuzzy untuk menghasilkan sinyal atur. Pengaturan logika fuzzy biasanya menggunakan variabel masukan berupa error dan perubahan error, sedangkan keluaran dan aksi atur berupa sinyal atur atau perubahan sinyal atur.

2.10. Logika Pengambilan Keputusan

Inti persoalan dari pengaturan fuzzy terletak pada logika pengambilan keputusan yang meniru mengambil keputusan manusia, untuk memahami proses pengambilan kesimpulan dalam fuzzy, pada bagian ini akan dibahas beberapa konsep logika pengambilan kesimpulan yang meliputi fungsi-fungsi implikasi fuzzy, penafsiran kata hubung, operator konvensional dan mekanisme inference. Adapun aturan kontrol fuzzy mengambil simple pada seorang operator yang sedang bekerja secara sadar atau tidak sadar akan menggunakan aturan kontrol yang berupa hubungan Jika-Maka dalam mengambil keputusan. Aturan kontrol ini dilengkapi dengan variabel-variabel linguistik sehingga terbentuk suatu pernyataan-pernyataan linguistik yang didasarkan pada pengetahuan dan pengalaman tentang objek yang akan diukur.

Perilaku suatu sistem pengaturan fuzzy didasari oleh seperangkat pernyataan linguistik berbasis pengetahuan yang meniru cara kerja operator, pernyataan ini menggunakan variabel linguistik dari himpunan fuzzy, oleh karena itu disebut pernyataan kondisional fuzzy atau aturan kontrol fuzzy, oleh karena itu disebut pernyataan kondisional fuzzy atau aturan kontrol fuzzy. Bentuk umum aturan kontrol fuzzy adalah pernyataan Jika-Maka (*If-Then*) dan pernyataan lengkapnya adalah jika manusia i , maka keluaran j . $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$. masukan dan keluaran merupakan keadaan dari objek atur dan keluaran merupakan tindakan yang harus dilakukan terhadap objek atur sehingga berhubungan dengan keadaan masukan. Dari bentuk aturan kontrol di atas aturan tersebut merupakan bentuk *MIMO* (Multi Input Multi Output). Untuk sistem

MISO (Multi Input Single Output), maka keluaran hanya ada satu. Sistem yang akan dibahas adalah sistem MISO dengan dua masukan dan satu keluaran yang perangkat aturan kontrolnya berbentuk :

R_1 : Jika x adalah A_1 dan y adalah B_1 maka z adalah C_1

R_1 : Jika x adalah A_2 dan y adalah B_2 maka z adalah C_2

:

:

R_n : Jika x adalah A_n dan y adalah B_n maka z adalah C_n

Dimana x dan y adalah variabel masukan dan z variabel keluaran A, B, C adalah nilai linguistik dari variabel $x, y,$ dan z . sementara untuk fungsi implikasi fuzzy aturan dasar kontrol fuzzy adalah relasi fuzzy yang dinyatakan sebagai suatu hubungan sebab akibat yang disebut implikasi fuzzy, aturan kontrol fuzzy bila dinyatakan dengan implikasi fuzzy akan menjadi :

$$\begin{aligned} \mu_{R1} &= \underline{\Delta} \mu(A_1 \text{ dan } B_1 \rightarrow C_1(x, y, z)) \\ &= [\mu A_i \text{ dan } \mu B_{1(y)}] \rightarrow \mu c_{i(z)} \end{aligned} \quad (2.37)$$

A_1 dan B_1 adalah himpunan fuzzy $A_1 \times b_1$ dalam $x \times y$, $R_1 \underline{\Delta} (A_1 \text{ dan } B_1) \rightarrow C_1$

Adalah implikasi fuzzy dalam $X \times Y \times Z$, dalam notasi merupakan fungsi implikasi yang pernah diusulkan oleh peneliti. Fungsi implikasi dapat didefinisikan sebagai sesuatu relasi yang menunjukkan suatu keterkaitan, antara suatu kejadian sebab dan akibat, suatu fungsi implikasi tentang keterkaitan kejadian diatas maka semakin baik implikasi tersebut. Salah satu implikasi yang akan dipakai disini adalah hasil penelitian Mamdani, fungsi ini disebut fungsi implikasi aturan operasi mini (mini - operation rule) yang dinotasikan :

$$R_c = A \times B$$

$$= \int_{X \times Y} \mu_A(x) \mu_B(y) | (x,y) \tag{2-38}$$

$$= \int_{X \times Y \times Z} \mu_A \times B(x, y) \wedge \mu_C(z) | (x,y,z) \tag{2-39}$$

Bentuk $A \times B$ adalah produk kartesian masukan dalam $X \times Y$

Penafsiran kata hubung adalah karena tiap-tiap kontrol fuzzy dinyatakan dengan suatu relasi, maka perilaku dari keseluruhan sistem fuzzy ditentukan oleh relasi fuzzy itu sendiri. Dengan kata lain perilaku sistem fuzzy ditentukan oleh relasi tunggal yang merupakan kombinasi seluruh perangkat relasi fuzzy dalam sekumpulan aturan kontrol, pengkombinasian ini dilakukan oleh kata hubung, untuk kata hubung dan (*end*) biasanya digunakan pada produk kartesian yang dilatarbelakngi oleh nilai yang berbeda. Sebagai contoh "Jika (A dan B) maka C" sedangkan kata hubung (*also*), juga digunakan pada fungsi implikasi fuzzy dan dapat juga sebagai gabungan dan irisan. Mekanisme inferensi adalah seperti yang dicontohkan, pada operator komposisional yang diterapkan pada suatu aturan kontrol, sedangkan sistem pengaturan fuzzy terdiri dari seperangkat aturan kontrol. Untuk mengkombinasi aturan kontrol tersebut digunakan kata hubung, juga, bentuk umum dari relasi kata hubung juga adalah :

Masukan : x adalah A' dan y adalah B'

R_1 : Jika x adalah A_1 dan y adalah B_1 maka z adalah C_1

Juga R_2 : Jika x adalah A_2 dan y adalah B_2 maka z adalah C_2

:

:

Juga R_n : Jika x adalah A_n dan y adalah B_n maka z adalah C_n

Kesimpulan : z adalah C'

Keseluruhan aturan kontrol ini harus dapat menghasilkan suatu keluaran berupa kesimpulan yang dinyatakan dengan C' , kesimpulan C' diperoleh dari mekanisme inferensi dengan komposisi sup-star yang didasarkan dari fungsi implikasi fuzzy dan kata hubung "dan" dan "juga" ada beberapa lemma penting yang berhubungan dengan mekanisme inferensi,

$$\text{Lemma 1} \quad : \left(A', B' \right)_{i=1}^n R_i = (A', B') R_i \quad (2-40)$$

R_i adalah relasi aturan kontrol ke -i

Lemha 2 : Untuk fungsi implikasi operasi mini Mamdani diperoleh :

$$\text{a). } (A', B')(A_i \text{ dan } B_i \rightarrow c_i) = [A' \cdot (A_i \rightarrow C_i)] \cap [B' \cdot (B_i \rightarrow C_i)] \quad (2-41)$$

$$\text{jika } \mu_{A_i \times B_i} = \mu_{A_i} \wedge \mu_{B_i}$$

$$\text{b). } (A', B')(A_i \text{ dan } B_i \rightarrow c_i) = [A' \cdot (A_i \rightarrow C_i)] \cap [B' \cdot (B_i \rightarrow C_i)] \quad (2-42)$$

Lemha 3 : Jika masukan berupa fuzzy tunggal $A' = X_0$ dan $B' = Y_0$ diperoleh :

$$\text{a). } R_c : a_i \wedge \mu_{c_i}(z) \quad a_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0) \quad (2-43)$$

$$\text{b). } R_c : a_i \wedge \mu_{c_i}(z) \quad a_i = \mu_{A_i}(x_0) \mu_{B_i}(y_0) \quad (2-44)$$

Lemha 1 menyatakan bahwa keluaran yang disimpulkan dari semua perangkat aturan kontrol, sama dengan gabungan semua keluaran tiap-tiap aturan kontrol, sedang lemma 2 dan 3 menguraikan penerapan fungsi implikasi operasi mini Mamdani (R_0), dengan menggunakan kedua jenis produk kertesian, dari kedua lemma ini diperoleh hubungan sederhana antara masukan dan keluaran :

$$R_c : \mu_c = \bigcup_{j=1}^n a_j \wedge \mu_{C_j} \quad (2-45)$$

faktor pembobot a_i merupakan besarnya kontribusi aturan ke I untuk aksi kontrol fuzzy, faktor pembobot dapat ditentukan dengan dua pilihan yaitu dengan operasi minimum (a_i) dan produk aljabar (a_i) dalam produk kartesian. Lemma 2 mengisyaratkan masukan bagi aturan kontrol berupa suatu himpunan fuzzy, untuk menjelaskan hal itu, diasumsikan ada dua aturan kontrol fuzzy sebagai berikut :

R_1 : Jika x adalah A_1 dan y adalah B_1 maka z adalah C_1

R_2 : Jika x adalah A_2 dan y adalah B_2 maka z adalah C_2

Yang mendapat masukan himpunan fuzzy A' dan B' , dan penafsiran grafis dari lemma 2 untuk dua aturan kontrol fuzzy diatas yang mekanisme inferensinya menggunakan operasi minimum..

Penafsiran secara grafis lemma Menggambarkan bahwa pada mekanisme inferensi setiap aturan kontrol mempunyai kontribusi terhadap sinyal atur yang diputuskan, kenyataan bahwa kemasukan pengatur logika fuzzy diperlukan sebagai fuzzy tunggal yang menyebabkan permasalahan dalam mekanismwe inferensi menjadi lebih sederhana pada lemma 3 permasalahan tersebut diatasi dengan menyederhanakan permasalahan dengan mengganti himpunan fuzzy tunggal, hal ini karena pada penerapan yang sebenarnya masukan yang bersifat crisp yang diperlukan sebagai fuzzy tunggal. Penafisran grais lemma 3 pada menjelaskan hal ini, dari aturan kontrol R_1 dan R_2 dan dengan menerapkan fungsi implikasi operasi mini Mamdani (R_c), produk kartesian operasi minimum dan penggunaan Lemma 3 diperoleh faktor pembobot

$$a_1 = \mu_{A_1}(X_0) \wedge \mu_{B_1}(y_0) \quad (2-46)$$

$$a_2 = \mu_{A_2}(X_0) \wedge \mu_{B_2}(y_0) \quad (2-47)$$

fungsi keanggotaan sinyal atur adalah fungsi keanggotaan yang didapat dari persamaan diatas merupakan fungsi keanggotaan untuk sinyal atur tersebut, dengan fungsi keanggotaan itu akan didapat nilai dari sinyal atur yang didapat dari proses fuzzifikasi. Fungsi defuzzifikasi merupakan kebalikan dari fuzzifikasi, yaitu mentransformasikan data fuzzy ke data bukan fuzzy, atau pemetaan dari ruang aksi atur fuzzy ke aksi atur crisp, fungsi defuzzifikasi dinyatakan dengan :

$$Z_0 = \text{mem-fuzzikan } (z)$$

Secara garis besar strategi defuzzifikasi bertujuan menghasilkan aksi atur nyata yang dapat mempresentasikan distribusi dari aksi atur masing-masing kaidah atur, ada beberapa metode yang dapat digunakan, salah satu adalah metode yang dikenal dengan metode CAO (Centre of Area) atau yang dikenal dengan metode titik pusat. Sistem metode adalah menghitung titik pusat dari distribusi sinyal atur dari masing-masing kaidah atur, secara umum Z_0 dihitung dengan menggunakan persamaan titik pusat.

$$Z_0 = \frac{\sum_{i=1}^n \mu_z(Z_i) Z_i}{\sum_{i=1}^n \mu_z(Z_i)} \quad (2-48)$$

Pada semesta pembicaraan dengan penyokong berhingga (diskrit) dengan jumlah penyokong (z) n persamaan ini dapat digunakan secara langsung karena setiap penyokong nilai keanggotaannya ditentukan secara implisit (pendefinisian numerik) pada semesta tak terhingga (kontinyu), yang didefinisikan secara fungsional z_i sebagai nilai rata-rata fungsi keanggotaan.

2.11. Perencanaan dari Fuzzy Logic Control

Logika fuzzy digunakan pada pendesainan control untuk plant dinamic yang kompleks dan sering sekali tidak dapat ditentukan dengan tepat pada sebuah sistem kontrol kecepatan motor. Fungsi fuzzy logic control adalah untuk merubah kaidah kontrol linguistik kedalam strategi kontrol berdasarkan informasi heruristik atau *exsper knowledge* (14). Pendekatan fuzzy logic control sangatlah berguna untuk mengontrol kecepatan Pendulum, karena sistem ini tidak menggunakan matematika yang rumit dari Pendulum, dan menggunakan sistem kontrol yang tertutup (*Cloosed Loop Control*).

Diagram fungsional dari fuzzy logic control diilustrasikan dalam diagram tersebut terdapat (4) blok utama, yaitu knowledge Bse, Fuzzification, Inference mekanis, dan Defuzzification. Knowledge bse disusun dari sebuah data base dan rule base, data base berisikan membership fuction input dan output yang menyediakan informasi yang tepat untuk operasi fuzzifikasi. Rule base dibuat dari sehimpunan rule linguistik / kaidah-kaidah linguistik yang menghubungkan variabel input fuzzy kekontrol fuzzy, fuzzifikasi mengubah sinyal input yang kaku, error (e) dan tingkat error (Δe) kedalam sinyal yang difuzzifikasi yang dapat ditunjukkan pada tingkat keanggotaan.

2.12. Proses Strukturisasi Logika Fuzzy

Akhirnya defuzzifikasi mengubah output yang fuzzy kedalam sinyal kontrol yang kaku, dalam hal ini adalah f yang mengatur kecepatan Pendulum, fungsi fuzzifikasi adalah menentukan error, $e(k) = f_R - f_M$ dan perubahan error, $\Delta e(k) = e(k) - e(k-1)$ untuk menandakan himpunan fuzzy. Fuzzifikasi mengisyaratkan

sebuah pengsekalan dan prosedur kwantitasi yang mendefinisikan range - range nilai dalam sebuah membership function, input yang kwantitasi ini dirubah kelabel linguistik yang sesuai, untuk standar linguistik digunakan, positif, negatif, nol, variabel ini ditentukan dalam domain variabel input (e dan Δe) gambar 3.14. positif besar, positif sedang, nol, negatif sedang, negatif sedang, negatif besar ditentukan dalam domain variabel output (Δf), gambar 2- menunjukkan membership function normalisasi variabel input, untuk setiap membership function yang ditentukan. *Knowledge base* mendefinisikan kaidah-kaidah yang dipresentasikan sebagai statement yang membentuk hubungan antara variabel input dan output dalam bentuk fungsi membership function, statement pengemudian dinyatakan dalam himpunan kaidah-kaidah *If-Then* (Jika-Maka) kaidah-kaidah ini dibentuk berdasarkan pengetahuan pakar.

Adapun untuk sistem kontrol fuzzy agar dapat bekerja, dibutuhkan suatu aturan (*Rule Base*) dalam memproses data input untuk pensekalan, pada tabel 2.1. menunjukkan suatu rule base yang digunakan dalam perencanaan sistem kontrol logika fuzzy yang dikombinasikan dengan phase locked loop, penentuan rule base adalah sangat menentukan kinerja dari plant yang digunakan. Adapun rule base yang digunakan, didapat dari hasil coba-coba, ini dilakukan agar didapat suatu hasil yang optimal dalam menentukan aturan yang cocok untuk pengaturan kecepatan, setelah didapatkan rule base yang tepat yang beracuan dari data respon kecepatan motor yang digunakan maka sistem dapat digunakan berdasarkan ketentuan tersebut.

Tabel 2,1, Rule Base Yang Digunakan Dalam pengaturan Kecepatan Pendulum Dengan Fuzzy Logic Control (FLC) dan Phase Locked Loop (PLL)

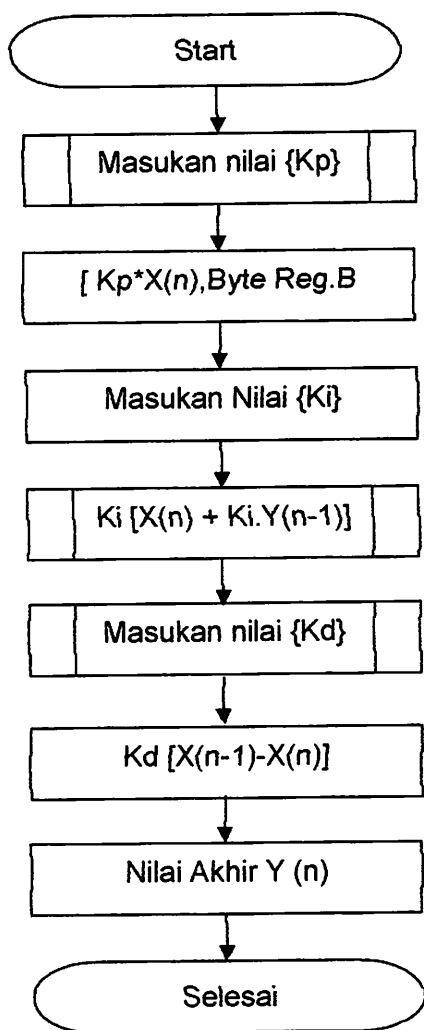
E / ΔE	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NM	NM	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PS
ZE	NM	NS	NS	ZE	PS	PS	PM
PS	NS	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Perencanaan suatu pengaturan kecepatan pendulum menggunakan logika fuzzy, harus dilakukan suatu aturan pemerosean (struktur) yang saling berkaitan antara satu sistem pendefinisian dengan sistem pendefinisian yang lain, yang disebut dengan diagram alir (*flowchart*), ini dilakukan untuk mendapatkan proses kerja yang digunakan dalam mengatur kecepatan motor. Agar lebih jelas sistem kerja dalam sistem pengaturan kecepatan pendulum menggunakan *fuzzy logic control*

Blok Diagram Sistem

dalam perencanaan sistem perangkat keras diperhatikan diagram blok di bawah ini.

. Diagram Alir Alogaritma PID



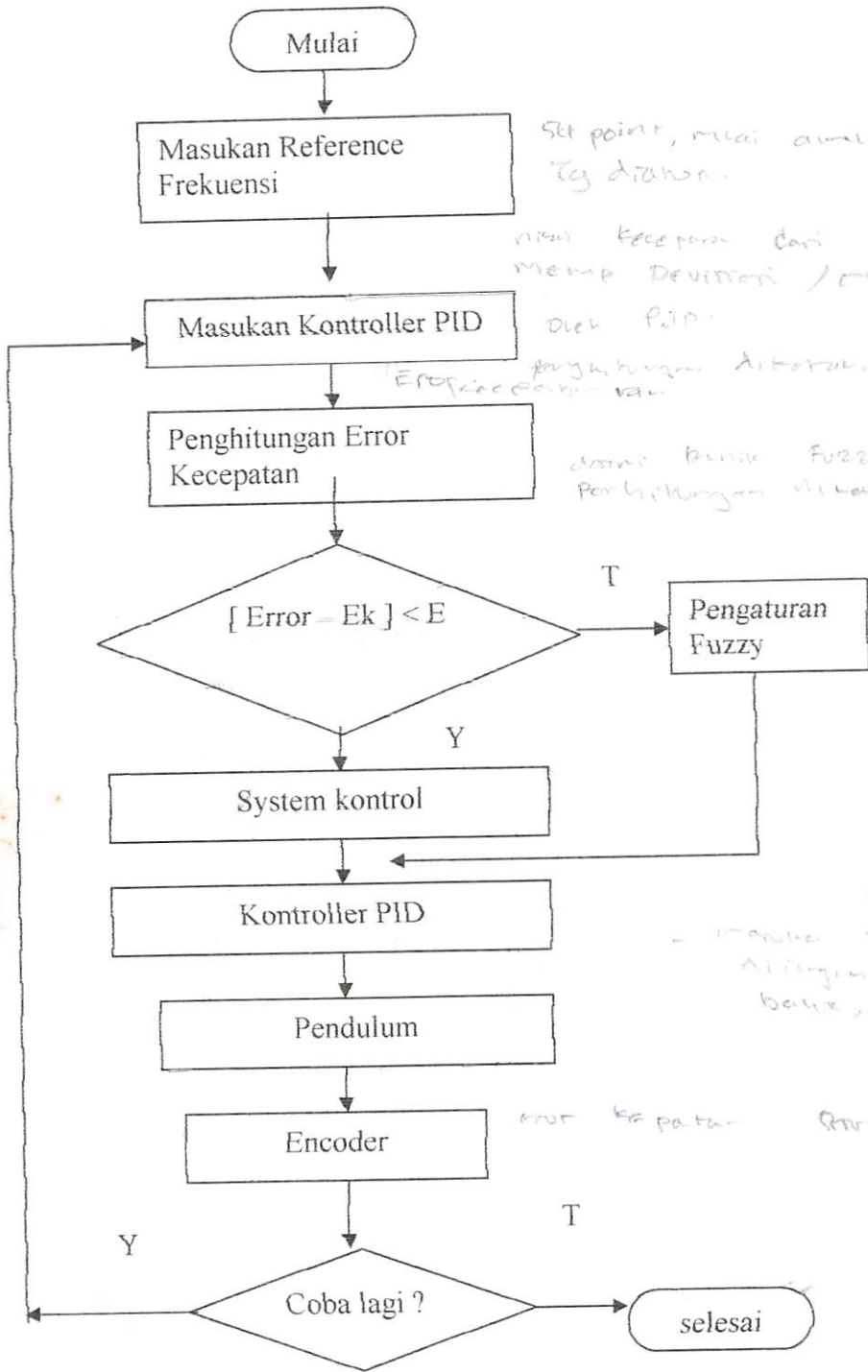
Gambar : Diagram Alir Alogaritma PID

Sumber : Perancangan.

Perancangan diagram alir algoritma PID diatas, merupakan penerapan sistem PID digital. Algoritma diperoleh dari persamaan (2020), bila ditransformasikan dalam bentuk keluaran Y (n) adalah sebagai berikut ;

$$Y(n) = K_p[X(n)] + K_i \{[X(n)] + K_i T [Y(n-1)]\} + \frac{K_d \{[X(n-1)] - [X(n)]\}}{T} \quad (3-6)$$

Pada pembacaan data digital pada penguatan (Kp, Ki, Kd) ditentukan agar mendapatkan suatu nilai register, sehingga dapat menjumlahkan, mengurangi serta mengalihkan bilangannya masing-masing, sesuai dengan persamaan diatas. Model matematis dari kontroler (Kp, Ki, Kd) didapat dengan ; $(V_{in} \times 2^n)$. sedangkan bilangan masukan X (n) didapat dengan; (data X 0,02) serta keluaran bilangan atau konversi balik Y (n) didapat dengan; (Data/0,02).



Set point, nilai awal dari kecepatan yg diukur.

nilai kecepatan dari set point itu nanti mempengaruhi / error

oleh PID:

penghitungan dilakukan melalui error

dan ini bisa fuzzy digunakan untuk perhitungan nilai kecepatan, dll

fuzzy dan ada

di lingkungan

ada di lingkungan

ada di lingkungan

ada di lingkungan

ada di lingkungan

error kecepatan



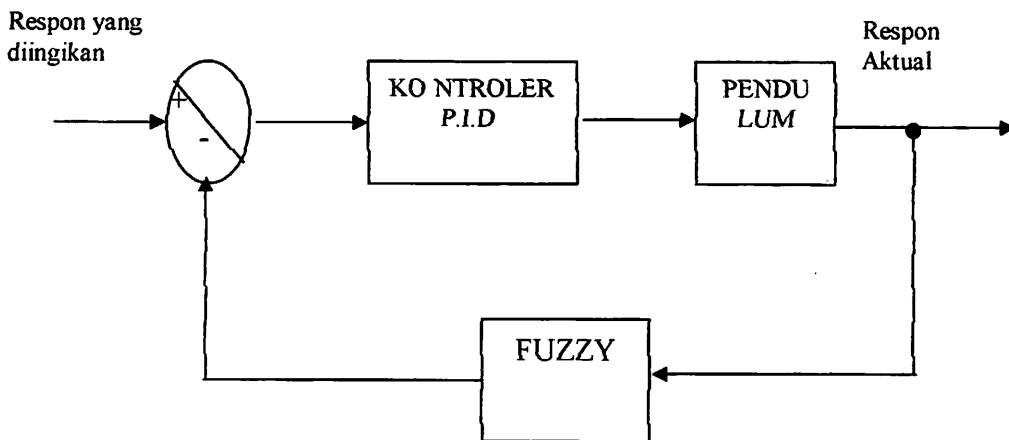
BAB III

PERENCANAAN MODEL PENDULUM DENGAN FUZZY CONTROLLER

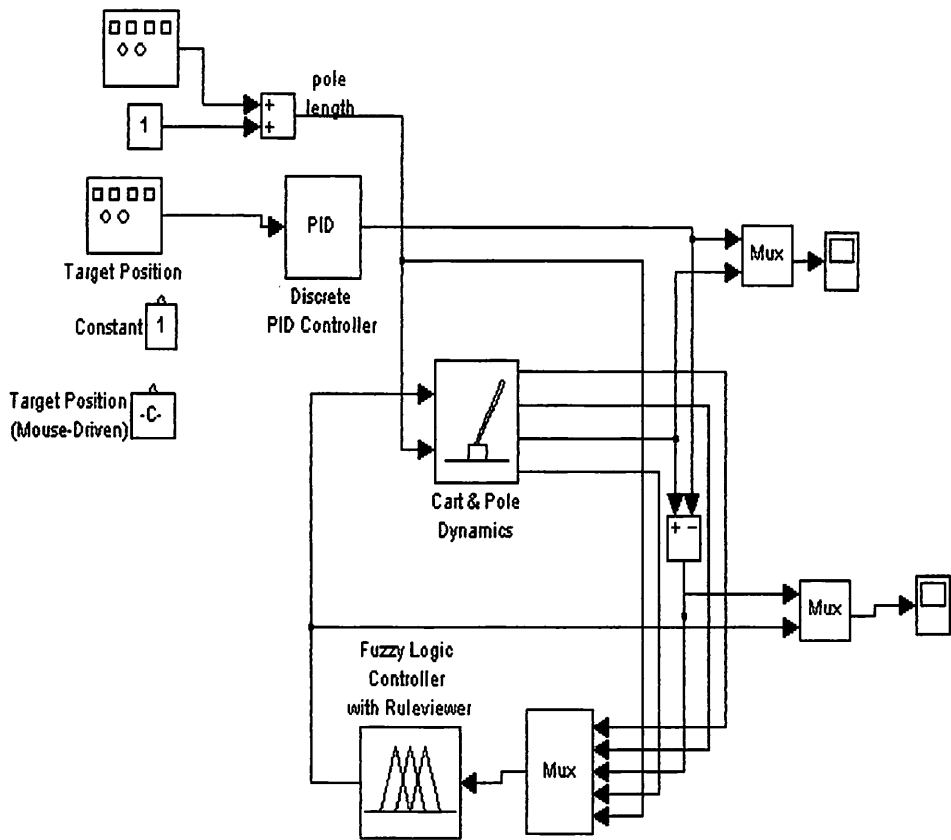
Perencanaan analisa model Pendulum dengan dan pengaturan kecepatan menggunakan fuzzy logic controller sehingga dapat disimulasikan sesuai pada beban. Pendulum dibuat berdasarkan persamaan matematik motor 1 phasa dengan pengaturan fuzzy logic controller pada beban. Dengan menggunakan pemodelan pendulum tersebut dan pengaturan kecepatan menggunakan fuzzy logic controller sangat diharapkan sistem dapat bekerja dengan baik.

Pada gambar 3-1

Blok diagram pemodelan Pendulum. Dan Kontroler P.I.D Berbasis FUZZY EXPERT



Gambar 3-1, Blok Diagram Pendulum dengan Fuzzy Controller



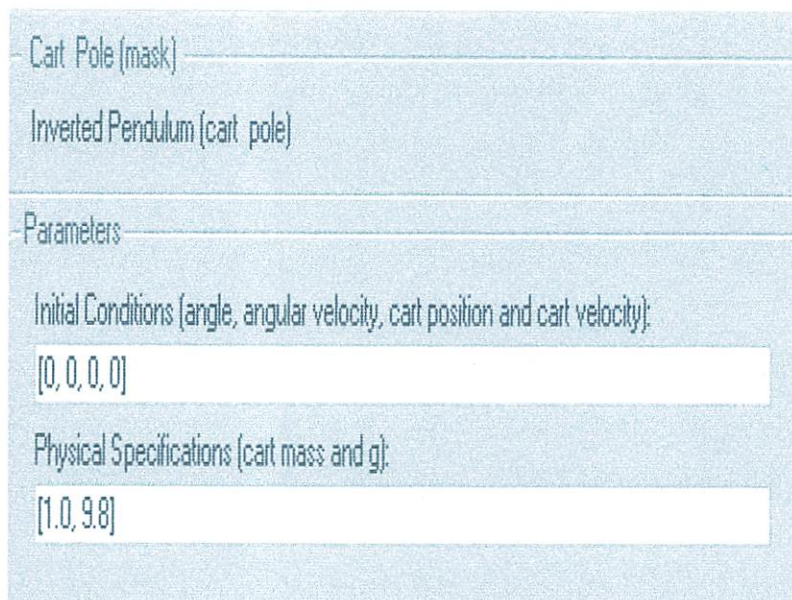
Gambar 3-2, Rangkaian Control Pendulum dengan Fuzzy Controller

3.1. Sumber Tegangan

Sumber tegangan yang diperlukan adalah sumber tegangan ac dengan urutan tegangan R-N, S-N, T-N. Pengaturan tegangan sumber ke Pendulum dilakukan dengan Kontrol PID yang dilakukan mulai dari tegangan 10 volt sampai 220 volt.

3.2. Modelling Pendulum

Bentuk parameter Pendulum yang direncanakan ditunjukkan pada table 3-1, merupakan parameter standart yang tercantum dalam data book Pendulum.



Cart Pole (mask)
Inverted Pendulum (cart pole)
Parameters
Initial Conditions (angle, angular velocity, cart position and cart velocity):
[0, 0, 0, 0]
Physical Specifications (cart mass and g):
[1.0, 9.8]

Tabel 3-1 Parameter Pendulum

3.1. Sumber Tegangan

Sumber tegangan yang diperlukan adalah sumber tegangan ac dengan urutan tegangan R-N, S-N, T-N. Pengaturan tegangan sumber ke Pendulum dilakukan dengan Kontrol PID yang dilakukan mulai dari tegangan 10 volt sampai 220 volt.

3.2. Modelling Pendulum

Bentuk parameter Pendulum yang direncanakan ditunjukkan pada table 3-1, merupakan parameter standart yang tercantum dalam data book Pendulum.

Cart Pole (m):	1
Inverted Pendulum (cart pole)	1
Parameter:	
Initial Conditions (angle, angular velocity, cart position and cart velocity):	
[0, 0, 0, 0]	
Physical Specifications (cart mass and g):	
[1.0, 9.8]	

Tabel 3-1 Parameter Pendulum

3.2. Penentuan Arus Beban

Untuk menentukan besar arus beban pada persamaan Pendulum terlebih dahulu harus mencari penghitungan daya masukan terhadap besar induktansi pada Pendulum tersebut. Besar daya sangat mempengaruhi perubahan pendulum tersebut.

Dalam gambar 3-2 ditunjukkan

perubahan arus beban terhadap besar daya masukan dan perubahan induktansi.

Gambar 3-2 Rangkaian penentuan parameter arus Pendulum

3.3. Penentuan Kecepatan

Pengaturan kecepatan Pendulum disebabkan masukan dari dua buah sinyal referensi dengan konstanta sudut 120 derajat dan sinyal masukan berbentuk persegi. Kemudian masuk dalam pengaturan kecepatan (K_p) yang menghasilkan torsi untuk menggerakkan pendulum .

Pada gambar 3-3

ditunjukkan arus stator yang disebabkan dari masukan torsi dan daya pendulum.



Gambar 3-3, Rangkaian penentuan parameter Sudut

3.4. Penentuan PID

Pemindahan arus pada stator ke arus rotor menimbulkan fluks magnetik pada Pendulum, sehingga menimbulkan daya yang digunakan untuk menghitung besar arus yang menuju ke rotor.

Pada gambar 3-4 ditunjukkan perubahan fluks dari arus stator ke arus rotor.

Parameter:

Proportional gain (Kp):

0.1

Integral gain (Ki):

1

Derivative gain (Kd):

0

Time constant for derivative (s):

0.5e-3

Output limits: [Upper Lower]

[1e6 -1e6]

Output initial value:

0

Sample time:

50e-6

Gambar 3-4 Rangkaian Perubahan Fluks dari arus stator (I_d) ke arus rotor (I_s)

3.5. Penentuan Sudut Teta

Perputaran kecepatan fluks stator pada Pendulum dengan kecepatan fluks pada rotor serta kapasitas daya pendulum dapat mempengaruhi pergeseran sudut teta antara belitan rotor dan belitan stator. Pergeseran sudut teta ini dapat menimbulkan kecepatan pada Pendulum

Pada gambar 3-5 ditunjukkan perubahan sudut teta antara belitan stator dan belitan rotor.

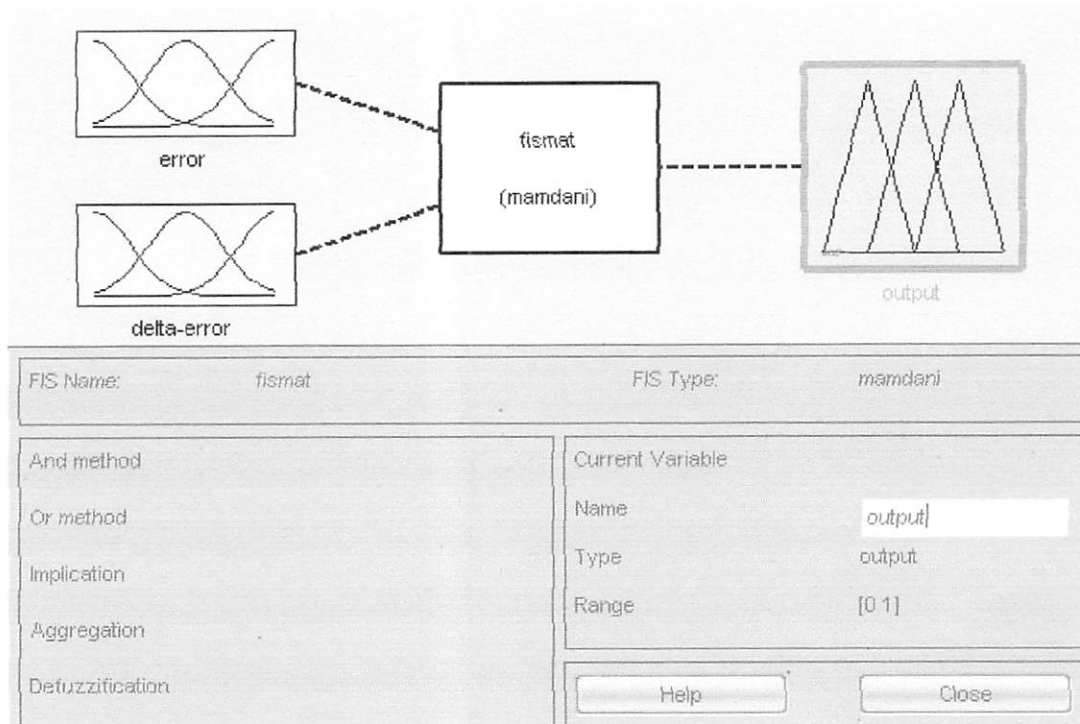


Gambar 3-5 Data Perubahan Sudut Teta antara belitan stator dan belitan rotor

3.6. Penentuan Logika fuzzy

Perubahan logika fuzzy dapat mempengaruhi kinerja Proportional Integral derivatif yang digunakan untuk mengatur kecepatan pendulum melalui perubahan regulator. Penentuan regulator arus ini dapat menimbulkan pengaturan kecepatan pada pendulum.

Pada gambar 3-6 ditunjukkan perubahan putaran kecepatan pendulum.



Gambar 3-6 Rangkaian Perubahan pendulum pada kontrol Logika Fuzzy



BAB IV

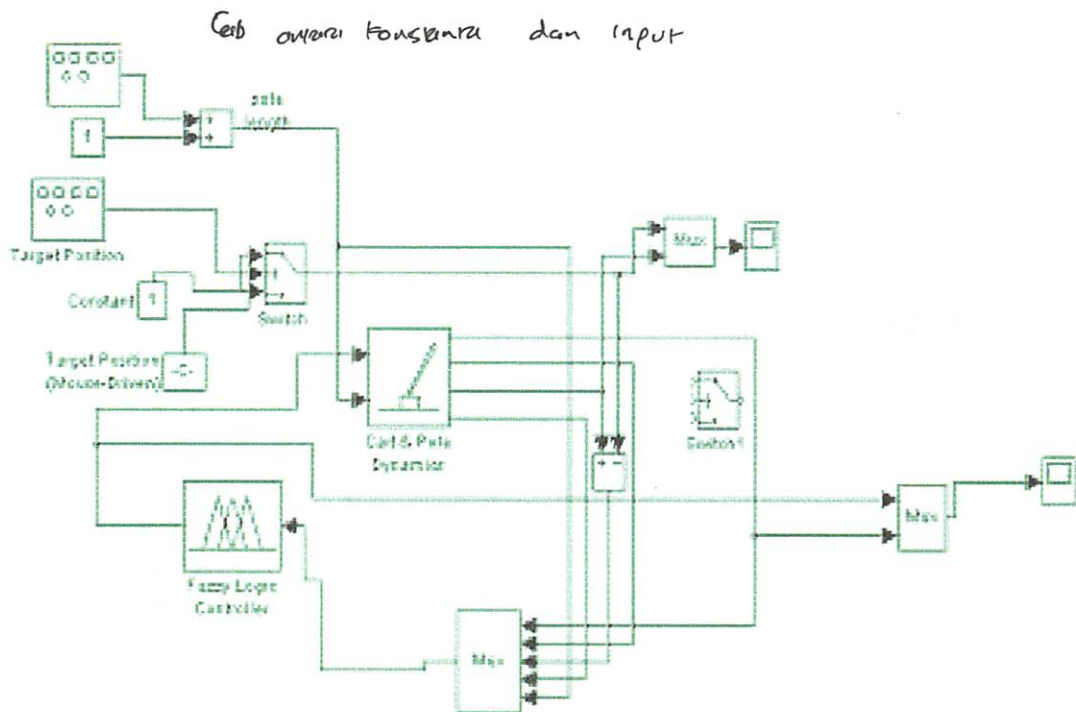
HASIL ANALISA

Dalam analisa ini dibahas simulasi penurunan rumus arus AC dan cara pemasangan Controller untuk mengatur kecepatan beban pendulum. Berikut ini adalah hasil simulasi dengan bahasa pemrograman Matlab.

4.1 Gerakkan Konvensional

4.1.1 Data Percobaan

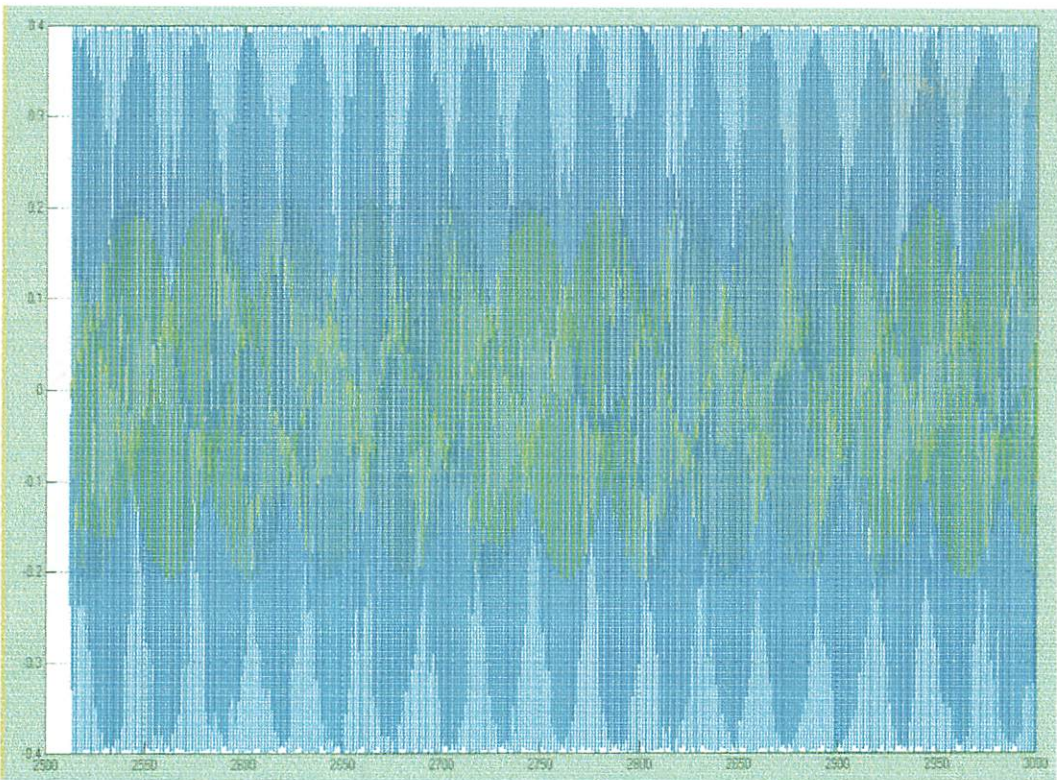
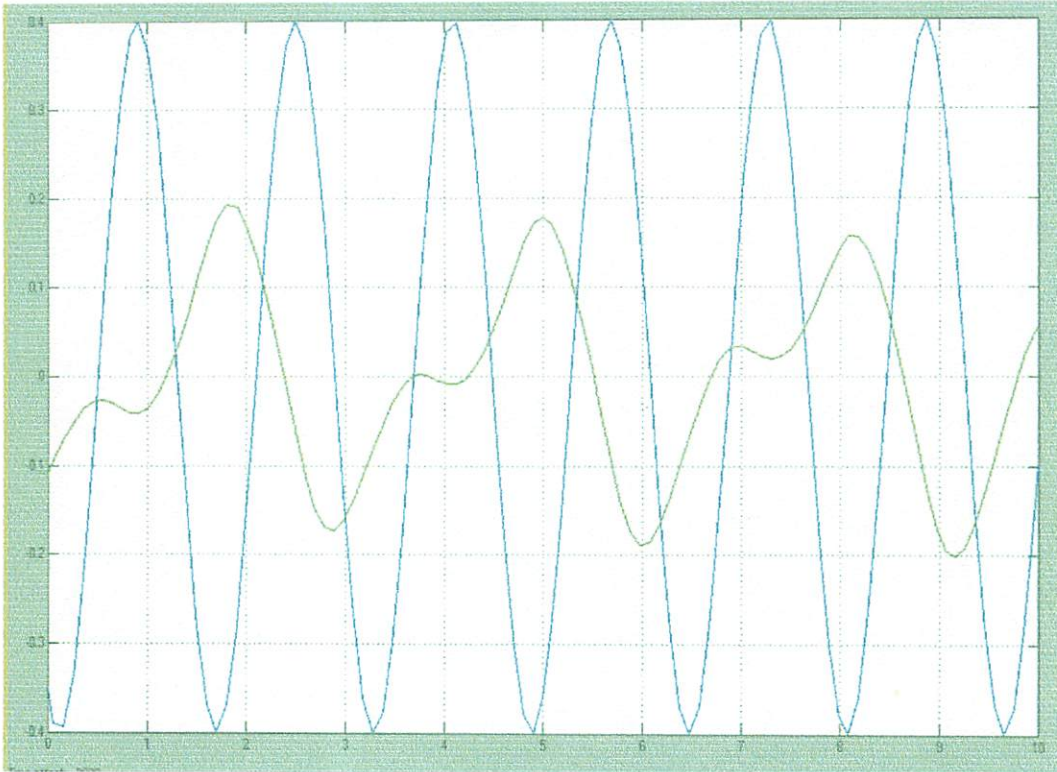
Data pendulum yang dipakai dalam ini adalah sebagai berikut :



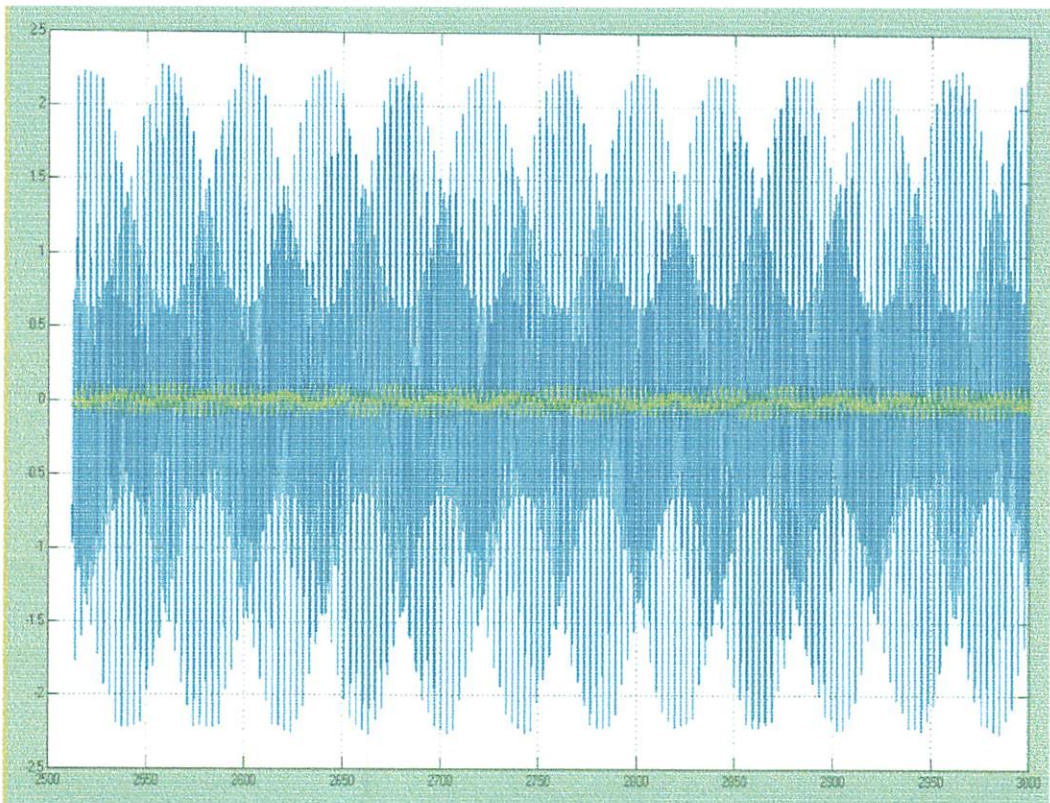
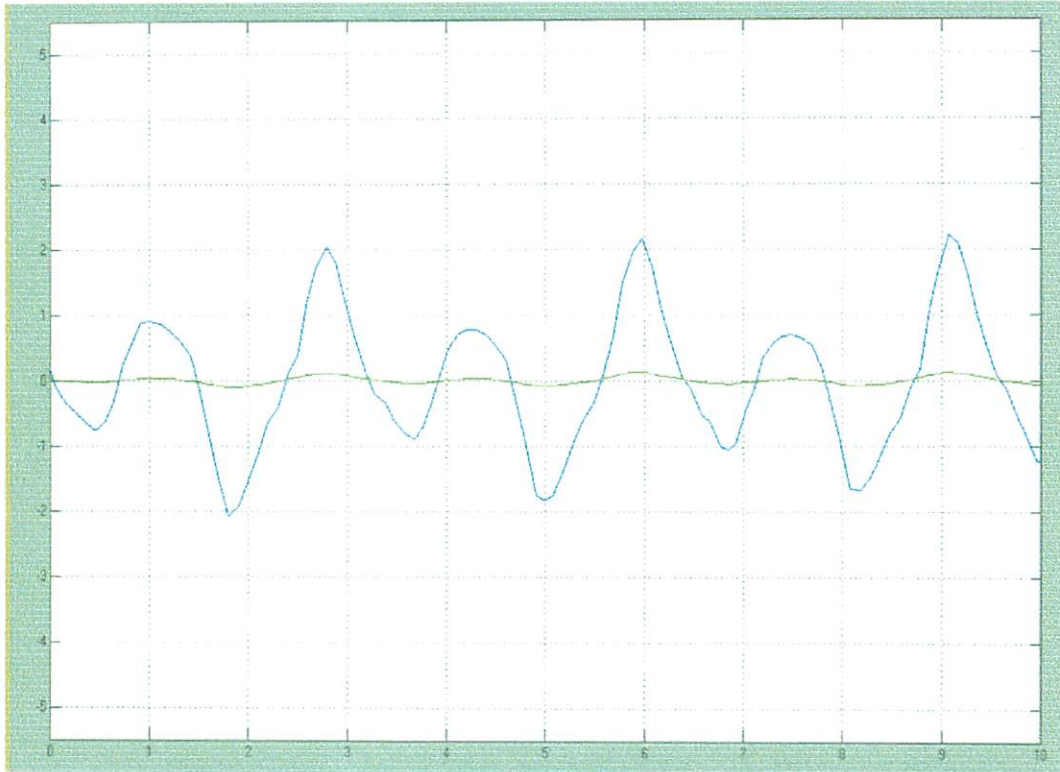
Gambar 4-1-1 Rangkaian Pengatur Kecepatan Pendulum Dengan Logika Fuzzy.

4.1.2 Rangkaian Percobaan

Karakteristik rangkaian sistem pengaturan pendulum tanpa menggunakan pengaturan logika fuzzy yang dipergunakan dalam simulasi ini adalah yang tampak pada gambar 4-1-2 di bawah ini :



Gambar 4-1-2 Karakteristik Pengaturan Kecepatan switch pada Pendulum Sebelum Dengan Logika Fuzzy.

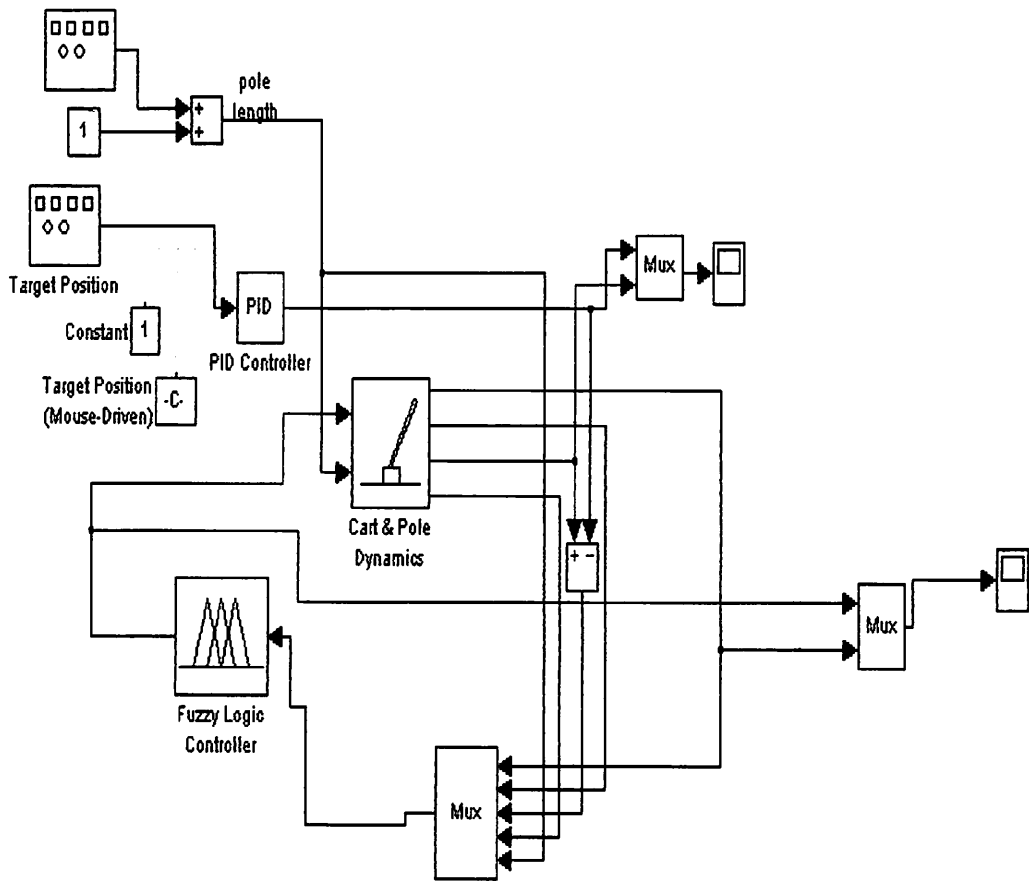


Gambar 4-1-3 Karakteristik Pengaturan Kecepatan switch pada Pendulum Setelah Dengan Logika Fuzzy.

4.2 Gerakan Pendulum dengan Controller PID dan Pengaturan Logika Fuzzy

4.2.1 Data Percobaan

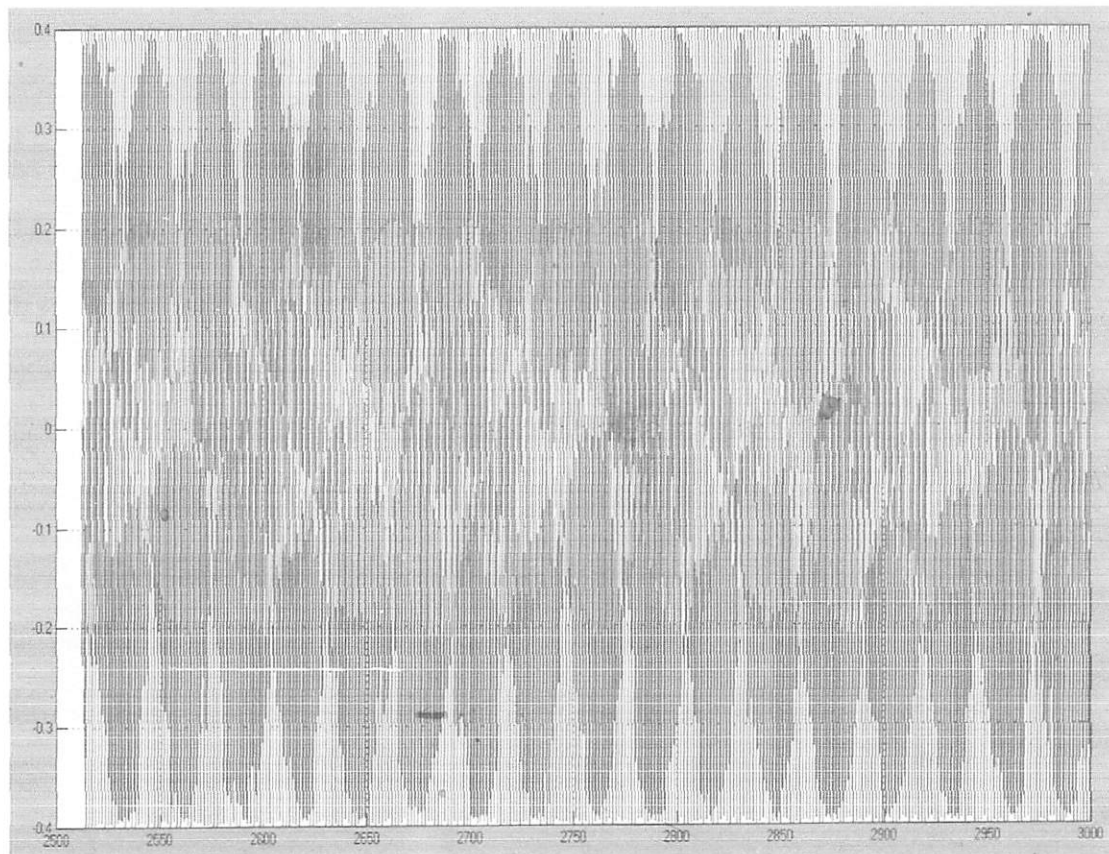
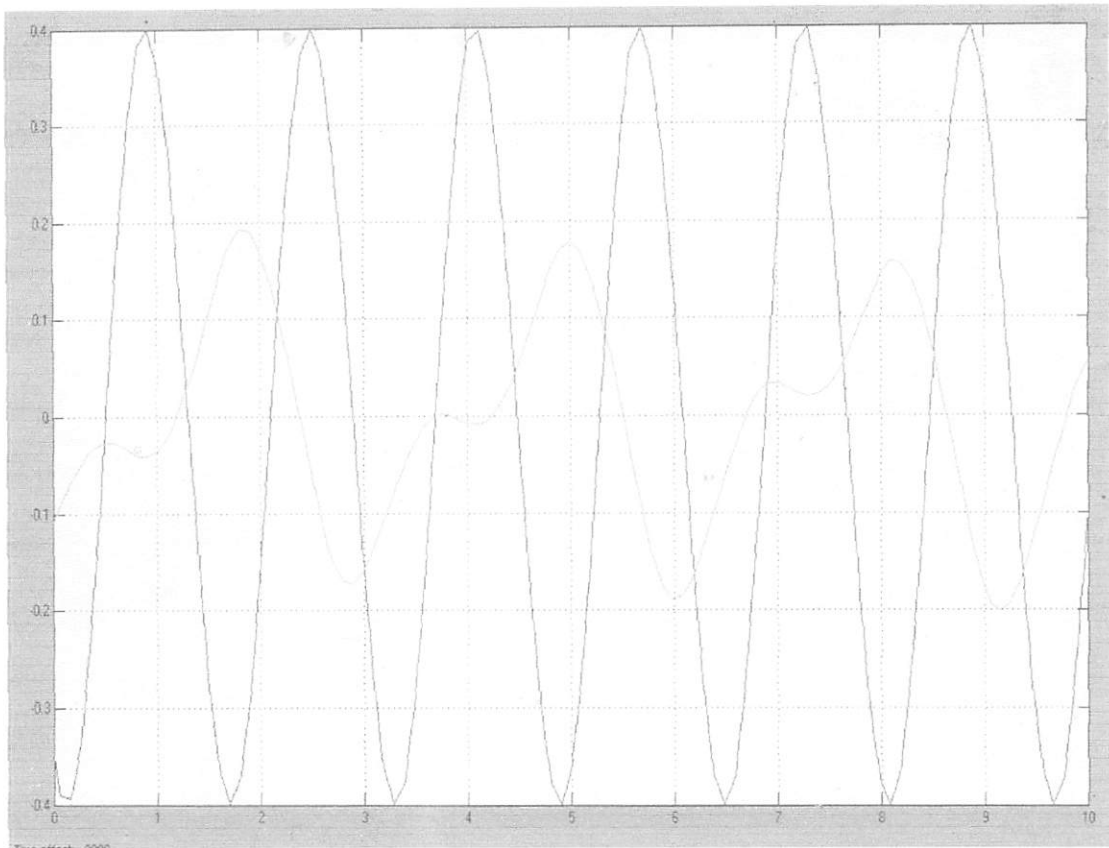
Dengan data-data Pendulum seperti pada Pendulum konvensional dengan controller PID dan ditambah pengaturan logika fuzzy seperti pada gambar 4-2-1 dibawah ini :



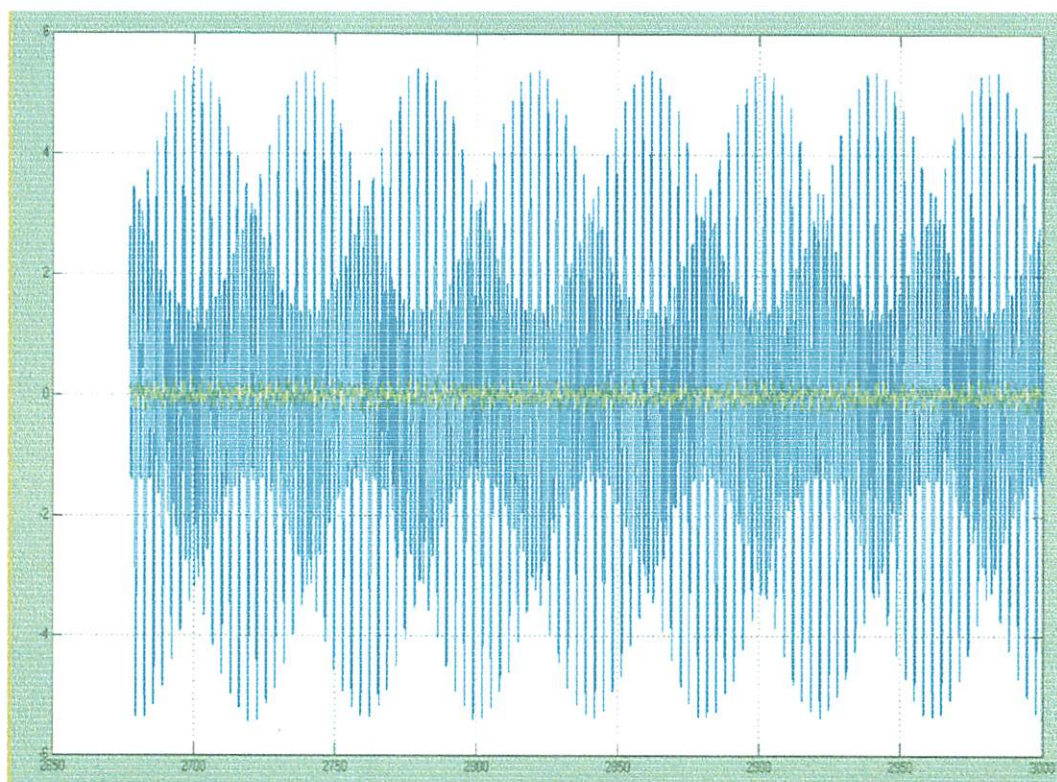
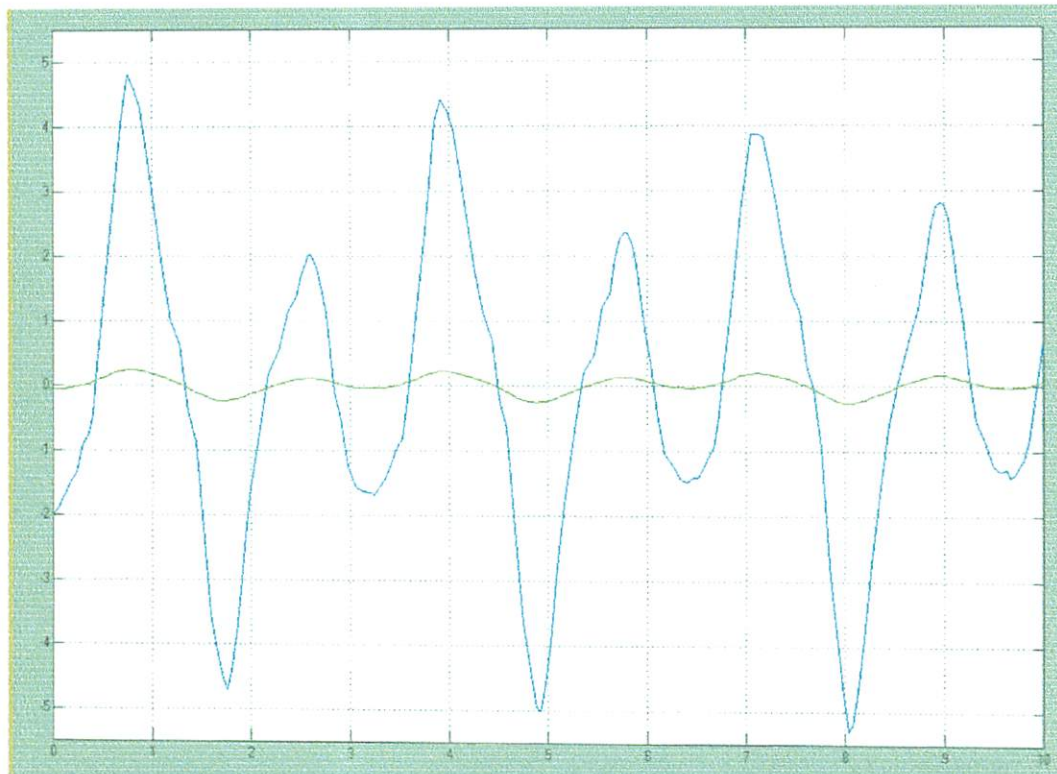
Gambar 4-2-1 Rangkaian Pengaturan Kecepatan Pendulum Dengan Controller PID dan Logika Fuzzy

4.2.2 Rangkaian Percobaan

Karakteristik rangkaian sistem pengaturan pendulum Dengan controller PID dan menggunakan pengaturan logika Fuzzy yang dipergunakan dalam simulasi ini adalah yang tampak pada gambar 4-2-2 dibawah ini :



Gambar 4-1-2 Karakteristik Pengaturan Kecepatan switch pada Pendulum Sebelum Dengan Logika Fuzzy.

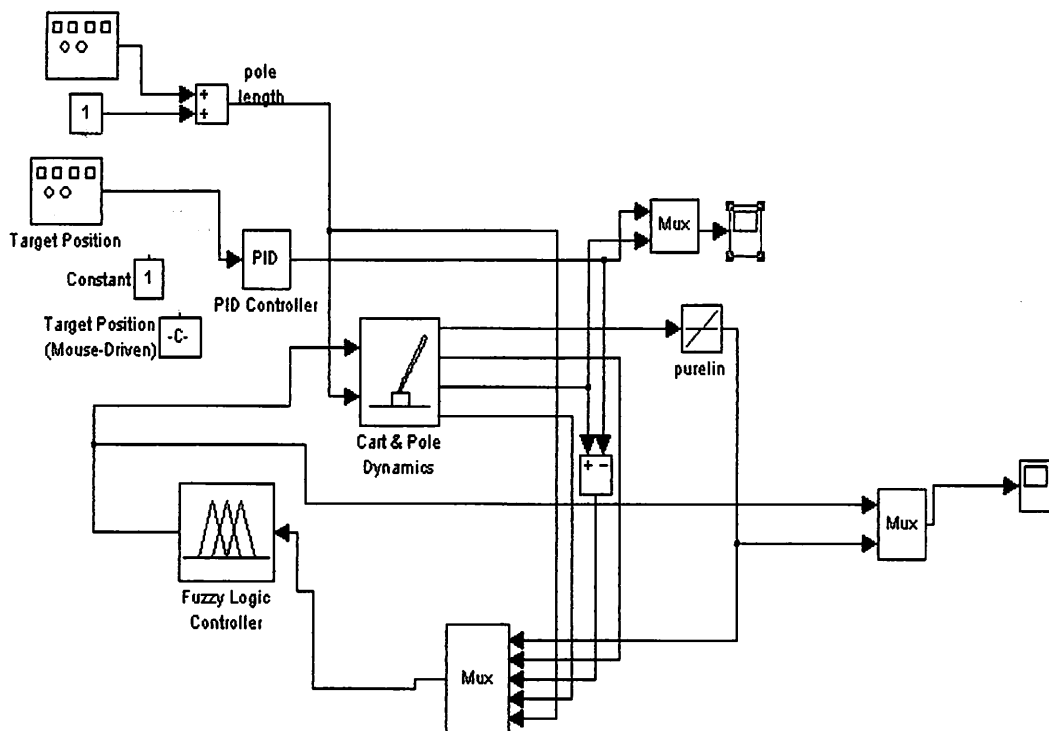


Gambar 4-2-3 Karakteristik Pengaturan Kecepatan Pendulum Pada PID Setelah Dengan Logika Fuzzy.

4.3 Gerakan Pendulum dengan Controller PID ditambah purelin dan Pengaturan Logika Fuzzy

4.3.1 Data Percobaan

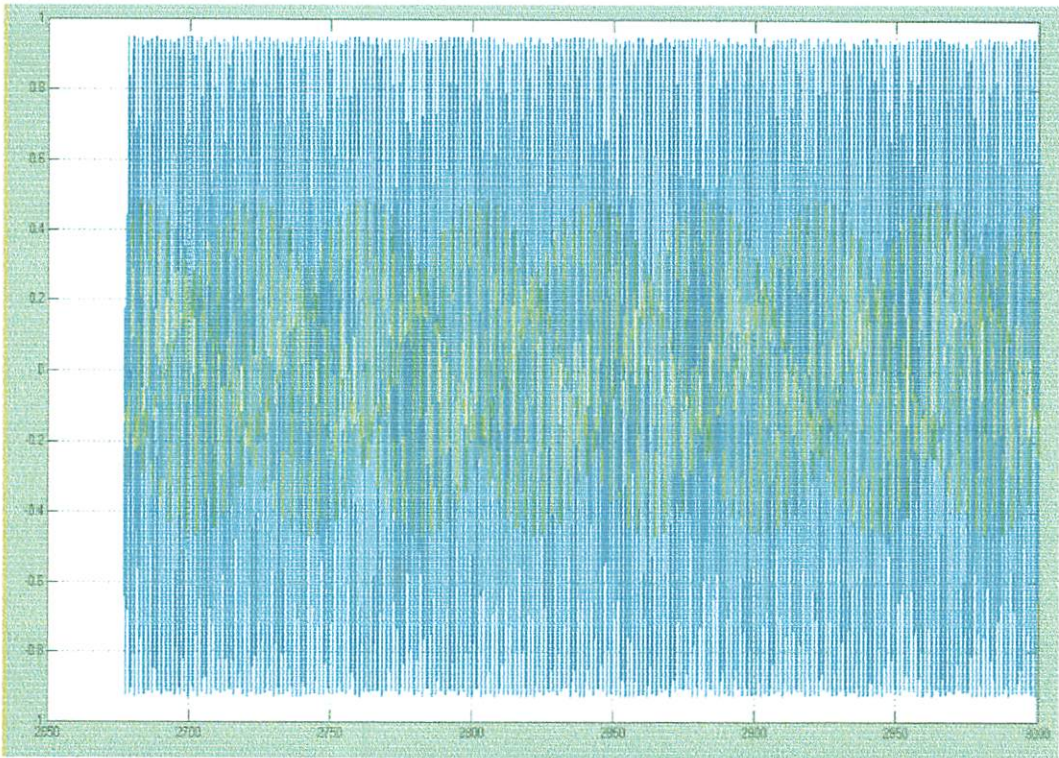
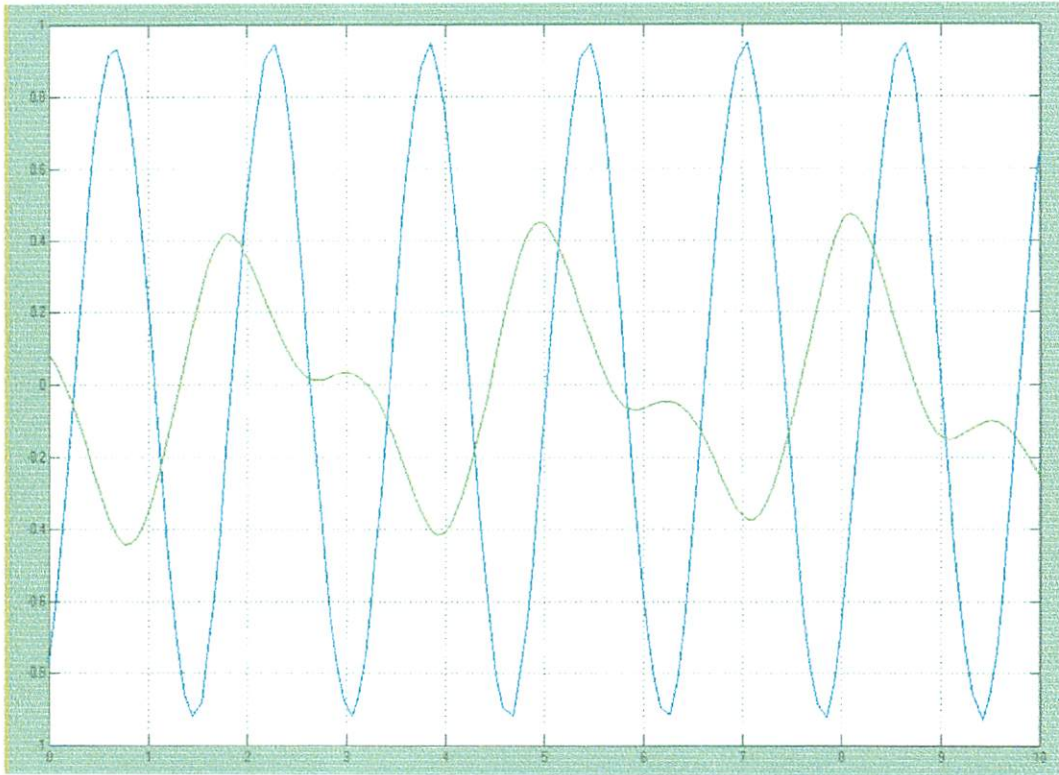
Dengan data-data Pendulum seperti pada Pendulum controller PID dengan pengaturan logika fuzzy dan ditambah purelin seperti pada gambar 4-3-1 dibawah ini :



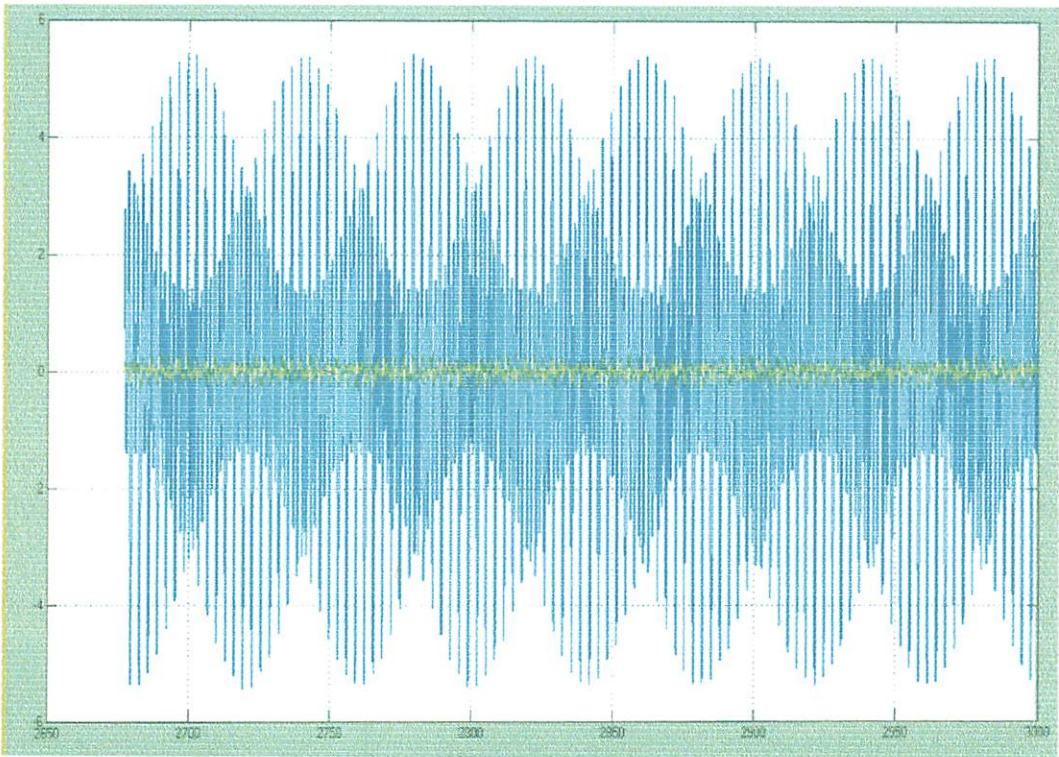
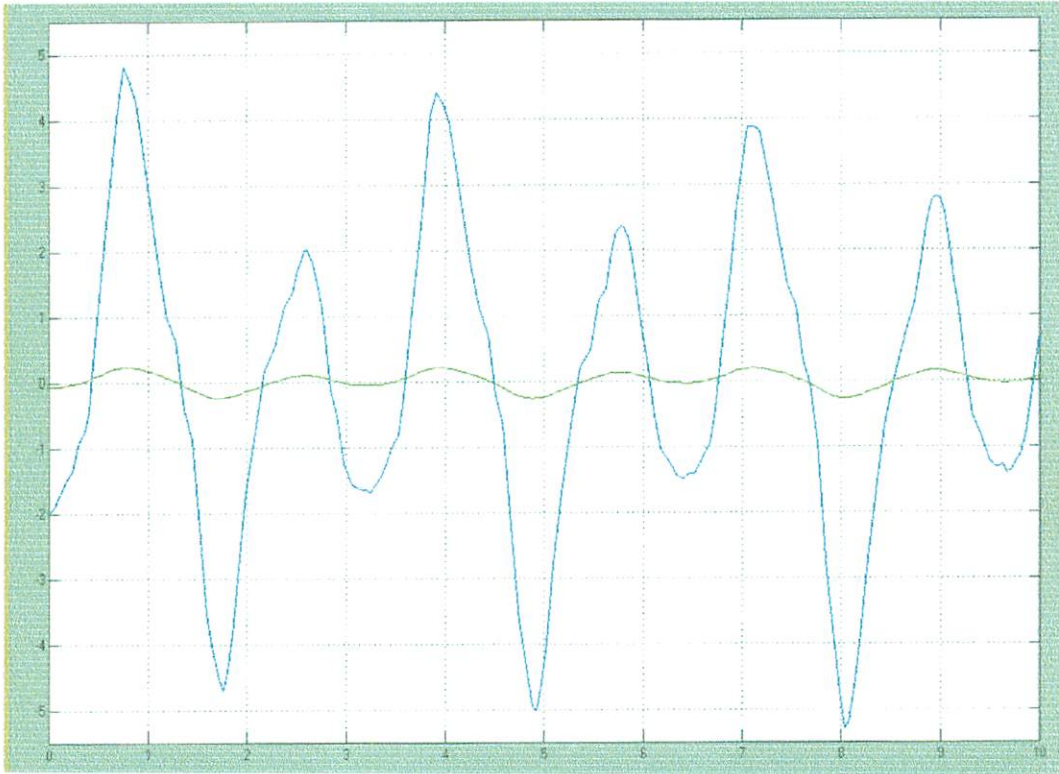
Gambar 4-3-1 Rangkaian Pengaturan Kecepatan Pendulum Dengan Controller PID yang ditambah Purelin dan Logika Fuzzy

4.3.2 Rangkaian Percobaan

Karakteristik rangkaian sistem pengaturan pendulum Dengan controller PID yang ditambah Purelin dan menggunakan pengaturan logika Fuzzy yang dipergunakan dalam simulasi ini adalah yang tampak pada gambar 4-3-2 dibawah ini :



Gambar 4-3-2 Karakteristik Pengaturan Kecepatan Pendulum Pada PID yang ditambah Purelin Sebelum Dengan Logika Fuzzy.



Gambar 4-3-3 Karakteristik Pengaturan Kecepatan Pendulum Pada PID yang ditambah Purelin Setelah Dengan Logika Fuzzy.



BAB V

PENUTUP

5.1. Kesimpulan

Kesimpulan hasil analisa didapatkan dalam Pengaturan Putaran menggunakan PID dengan metode logika fuzzy expert sebagai berikut :

- a. Pengaturan putaran menggunakan PID tanpa metode logika fuzzy expert mengenai nilai kecepatan lebih besar dan berosilasi amplitudo sangat besar sehingga putaran pada peralatan sangat berpengaruh terhadap pengaman.
- b. Pengaturan putaran menggunakan PID dengan metode logika fuzzy expert mengenai nilai kecepatan lebih kecil dan berosilasi amplitudo sangat kecil sehingga putaran pada peralatan tidak berpengaruh terhadap pengaman.
- c. Pengaturan putaran menggunakan logika fuzzy sangat efektif dalam mereduksi nilai kesalahan pada suatu sistem.

5.2. Saran

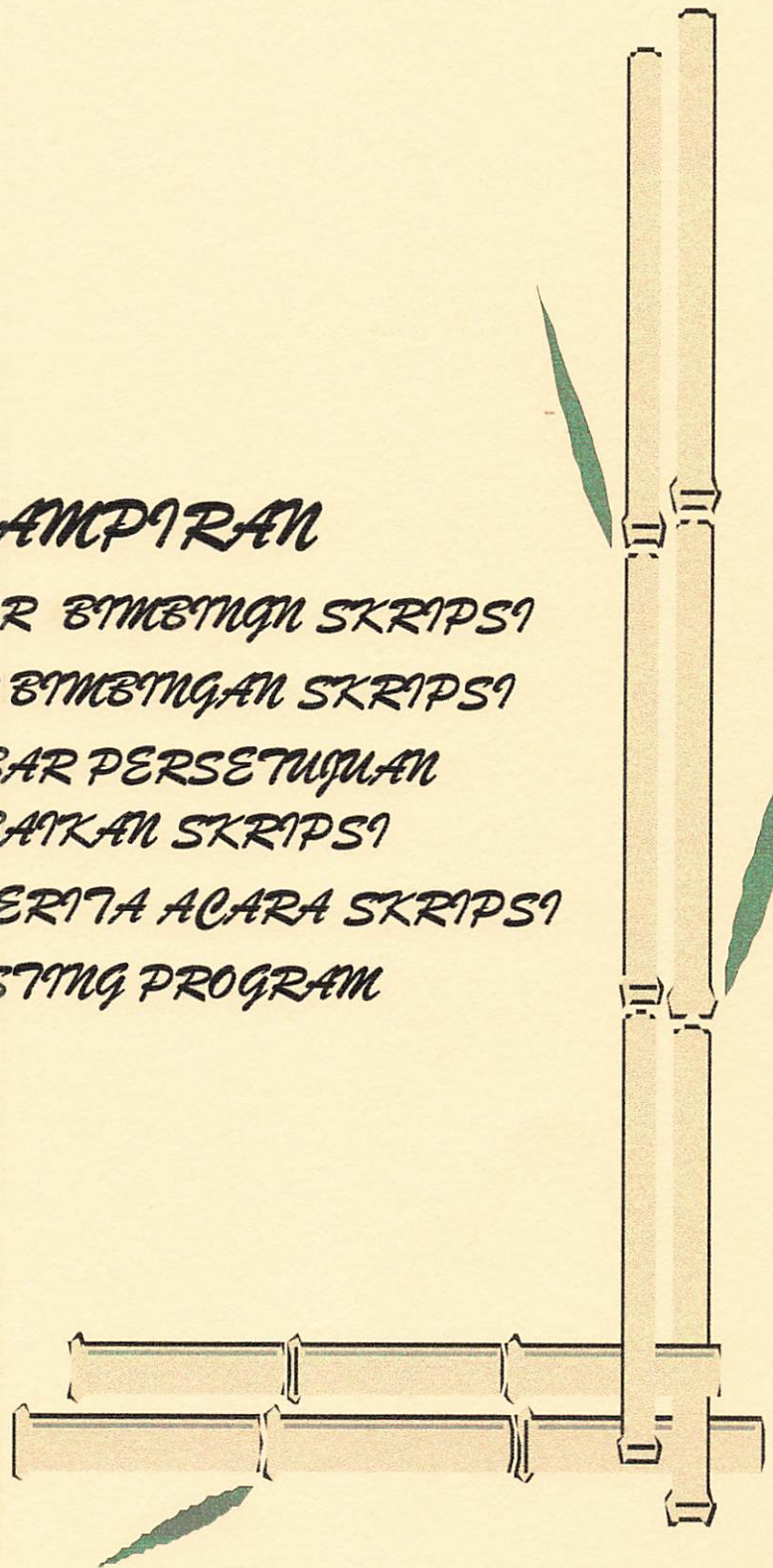
Untuk mendapatkan hasil yang lebih baik dalam pengaturan putaran sebagai bahan utama pada suatu sistem, maka sangat diharapkan pemakain metode menggunakan jaringan saraf tiruan.

DAFTAR PUSTAKA

1. Moti schneider ,abraham kandel,Gideon Langholz and Gerard crew "*Fuzzy Expert System Tools*".
2. Muhammat Jamsidi ,Nader Vadiie ,Timothy J.ros "*Fuzzy Logic And Control Softawre and Hardware Aplication*".
3. Martin Brown ,Criss Harris "*Neuro Fuzzy Adaptive Moelling And Control*".
4. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS ,Vol 37, No.5, OCTOBER 1998
5. S.G. Tzafestas (ED),*Applied Digital Control*. Amterdam:North Holland 1985
- 6 S.Assilian and E.H Mamdani, "An experiment in linguiststic synthesis with fuzzy logic controler, "Int. J .Man-Mach Studies,vol.7,pp 1-13, 1974
- 7 E.H. Mamdani et al," Pratical application of an heuristic fuzzy rule-based controler to the dynamic control of a robot arm,"Proc.Inst.Elec.Eng,vol 66,1985
- 8 L.A.Zadeh,"Fuzzy sets," Inform. Contr,Vol.8.

LAMPIRAN

- ✓ FORMULIR Bimbingan SKRIPSI
- ✓ LEMBAR Bimbingan SKRIPSI
- ✓ LEMBAR PERSETUJUAN
PERBAIKAN SKRIPSI
- ✓ LEMBAR BERITA ACARA SKRIPSI
- ✓ LISTING PROGRAM





INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

LEMBAR BIMBINGAN SKRIPSI

1. Nama : BAMBANG SUTIAWAN
2. Nim : 99.12.040
3. Jurusan : Teknik Elektro
4. Konsentrasi : Energi Listrik
5. Judul : **APLIKASI LOGIKA FUZZY
EXPERT UNTUK KONTROLER
P.LD PADA PENGENDALIAN
PENDULUM**
6. Tanggal mengajukan skripsi : 05 Mei 2005
7. Tanggal menyelesaikan skripsi : 04 Oktober 2005
8. Dosen pembimbing : Ir. Widodo Pudji Muljanto, MT
9. Telah dievaluasi dengan nilai : B+

Menyetujui

Dosen Pembimbing

(Ir. Widodo Pudji Muljanto, MT)
Nip. Y. 1028700171

Mengetahui

Ketua Jurusan Teknik Elektro S-1

(Ir. F. Yudi Ljapraptono, MT)
Nip. Y. 1039500274



FORMULIR BIMBINGAN SKRIPSI

Nama : BAMBANG SUTIAWAN
Nim : 99.12.040
Masa Bimbingan : 27 Juli 2005 - 27 Januari 2006
**Judul Skripsi : APLIKASI LOGIKA FUZZY EXPERT UNTUK
KONTROLER P.I.D PADA PENGENDALIAN
PENDULUM**

NO	Tanggal	Uraian	Paraf Pembimbing
1.	2-08-2005	Pengajuan Bab I	
2.	20-08-2005	Pengajuan Bab II	
3.	27-08-2005	Pengajuan Bab III	
4.	1-09-2005	Pengajuan Bab IV + Program	
5.	3-09-2005	Penyempurnaan Judul	
6.	5-09-2005	Acc Program	
7.	15-09-2005	Acc makalah hasil	
8.	23-09-2005	Acc ujian skripsi	
9.			
10.			

Malang, 200
Dosen Pembimbing

Ir. Widodo Pudi Mulianto, MT
NIP.Y 1028700171

LEMBAR PERSETUJUAN

APLIKASI LOGIKA FUZZY EXPERT UNTUK KONTROLLER PID PADA PENGENDALIAN PENDULUM

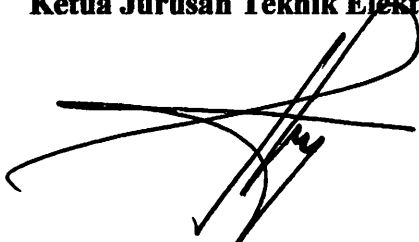
SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat Guna
Mencapai Gelar Sarjana Teknik*

Disusun Oleh :

**BAMBANG SUTIAWAN
99.12.040**

**Mengetahui
Ketua Jurusan Teknik Elektro S-1**



**(Ir. F. Yudi Limpraptono, MT)
Nip. Y. 1039500274**

**Diperiksa dan Disetujui
Dosen Pembimbing**



**(Ir. Widodo Pudji Muljanto, MT)
Nip. Y. 1028700171**

**KONSENTRASI TEKNIK ENERGI LISTRIK
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi jurusan teknik elektro jenjang strata satu

(S-1) Telah dilakukan perbaikan skripsi oleh:

Nama : BAMBANG SUTIAWAN
Nim : 99.12.040
Jurusan : Teknik Elektro
Konsentrasi : Energi Listrik
Judul : APLIKASI LOGIKA FUZZY EXPERT UNTUK
KONTROLER P.I.D PADA PENGENDALIAN
PENDULUM

Perbaikan meliputi :

No.	Materi Perbaikan	Paraf
1.	Penulisan keterangan gambar dan tabel harus ditengah dan menggunakan huruf balok	
2.	Beda fuzzy expert dan fuzzy biasa	

Anggota Penguji

(IRRINE BUDI S,ST,MT)

(Ir. I MADE WARTANA, MT)

Dosen Pembimbing

(Ir. WIDODO PUDJ MULJANTO, MT)

```

Creator          "pascal"
UpdateHistory    "UpdateHistoryNever"
ModifiedByFormat "%<Auto>"
LastModifiedBy  "Nge'b"
ModifiedDateFormat "%<Auto>"
LastModifiedDate "Sat Aug 20 11:33:36 2005"
ModelVersionFormat "1.%<AutoIncrement:44>"
ConfigurationManager "none"
LinearizationMsg  "none"
Profile          off
ParamWorkspaceSource "MATLABWorkspace"
AccelSystemTargetFile "accel.tlc"
AccelTemplateMakefile "accel_default_tmf"
AccelMakeCommand  "make_rtw"
TryForcingSFcnDF  off
ExtModeBatchMode  off
ExtModeEnableFloating on
ExtModeTrigType   "manual"
ExtModeTrigMode   "oneshot"
ExtModeTrigPort   "1"
ExtModeTrigElement "any"
ExtModeTrigDuration 1000
ExtModeTrigDurationFloating "auto"
ExtModeTrigHoldOff 0
ExtModeTrigDelay    0
ExtModeTrigDirection "rising"
ExtModeTrigLevel    0
ExtModeArchiveMode  "off"
ExtModeAutoIncOneShot off
ExtModeIncDirWhenArm off
ExtModeAddSuffixToVar off
ExtModeWriteAllDataToWs off
ExtModeArmWhenConnect on
ExtModeSkipDownloadWhenConnect off
ExtModeLogAll       on
ExtModeAutoUpdateStatusClock on
BufferReuse         on
StrictBusMsg        "None"
ProdHWDeviceType    "32-bit Generic"
ShowModelReferenceBlockVersion off
ShowModelReferenceBlockIO off
Array {
  Type          "Handle"
  Dimension      1
  Simulink.ConfigSet {
    $ObjectID      1
    Version         "1.1.0"
    Array {
      Type          "Handle"
      Dimension      7
      Simulink.SolverCC {
        $ObjectID      2
        Version         "1.1.0"
        StartTime       "0.0"
        StopTime        "1000"
        AbsTol          "1e-6"
      }
    }
  }
}

```

```

FixedStep          "auto"
InitialStep        "auto"
MaxNumMinSteps    "-1"
MaxOrder           5
ExtrapolationOrder 4
NumberNewtonIterations 1
MaxStep            "0.1"
MinStep            "auto"
RelTol             "1e-3"
SolverMode         "SingleTasking"
Solver             "ode45"
SolverName         "ode45"
ZeroCrossControl  "UseLocalSettings"
AlgebraicLoopSolver "TrustRegion"
SolverResetMethod  "Fast"
PositivePriorityOrder off
AutoInsertRateTranBlk off
SampleTimeConstraint "Unconstrained"
RateTranMode       "Deterministic"
}
Simulink.DataIOCC {
  $ObjectID        3
  Version           "1.1.0"
  Decimation        "1"
  ExternalInput     "[t, u]"
  FinalStateName    "xFinal"
  InitialState      "xInitial"
  LimitDataPoints   off
  MaxDataPoints     "1000"
  LoadExternalInput off
  LoadInitialState  off
  SaveFinalState    off
  SaveFormat        "Array"
  SaveOutput        off
  SaveState         off
  SignalLogging     on
  SaveTime          off
  StateSaveName     "xout"
  TimeSaveName      "tout"
  OutputSaveName    "yout"
  SignalLoggingName "logout"
  OutputOption      "RefineOutputTimes"
  OutputTimes       "[]"
  Refine            "1"
}
Simulink.OptimizationCC {
  $ObjectID        4
  Array {
    Type            "Cell"
    Dimension       5
    Cell            "ZeroExternalMemoryAtStartup"
    Cell            "ZeroInternalMemoryAtStartup"
    Cell            "InitFltsAndDblsToZero"
    Cell            "OptimizeModelRefInitCode"
    Cell            "NoFixptDivByZeroProtection"
    PropName        "DisabledProps"
  }
}

```

```

}
Version          "1.1.0"
BlockReduction   off
BooleanDataType  off
ConditionallyExecuteInputs on
InlineParams     off
InlineInvariantSignals off
OptimizeBlockIOStorage on
BufferReuse      on
EnforceIntegerDowncast on
ExpressionFolding on
FoldNonRolledExpr on
LocalBlockOutputs on
ParameterPooling on
RollThreshold    5
SystemCodeInlineAuto off
StateBitsets     off
DataBitsets      off
UseTempVars      off
ZeroExternalMemoryAtStartup on
ZeroInternalMemoryAtStartup on
InitFltsAndDblsToZero on
NoFixptDivByZeroProtection off
EfficientFloat2IntCast off
OptimizeModelRefInitCode off
LifeSpan         "inf"
BufferReusableBoundary on
}
Simulink.DebuggingCC {
  $ObjectID      5
  Version        "1.1.0"
  RTPrefix       "error"
  ConsistencyChecking "none"
  ArrayBoundsChecking "none"
  SignalInfNanChecking "none"
  AlgebraicLoopMsg "warning"
  ArtificialAlgebraicLoopMsg "warning"
  CheckSSInitialOutputMsg on
  CheckExecutionContextPreStartOutputMsg off
  CheckExecutionContextRuntimeOutputMsg off
  SignalResolutionControl "TryResolveAllWithWarning"
  BlockPriorityViolationMsg "warning"
  MinStepSizeMsg "warning"
  SolverPrmCheckMsg "none"
  InheritedTsInSrcMsg "warning"
  DiscreteInheritContinuousMsg "warning"
  MultiTaskDSMMsg "warning"
  MultiTaskRateTransMsg "error"
  SingleTaskRateTransMsg "none"
  TasksWithSamePriorityMsg "warning"
  CheckMatrixSingularityMsg "none"
  IntegerOverflowMsg "warning"
  Int32ToFloatConvMsg "warning"
  ParameterDowncastMsg "error"
  ParameterOverflowMsg "error"
  ParameterPrecisionLossMsg "warning"
}

```

```

UnderSpecifiedDataTypeMsg "none"
UnnecessaryDatatypeConvMsg "none"
VectorMatrixConversionMsg "none"
InvalidFcnCallConnMsg "error"
FcnCallInpInsideContextMsg "Use local settings"
SignalLabelMismatchMsg "none"
UnconnectedInputMsg "warning"
UnconnectedOutputMsg "warning"
UnconnectedLineMsg "warning"
SFcnCompatibilityMsg "none"
UniqueDataStoreMsg "none"
BusObjectLabelMismatch "warning"
RootOutportRequireBusObject "warning"
AssertControl "UseLocalSettings"
EnableOverflowDetection off
ModelReferenceIOMsg "none"
ModelReferenceVersionMismatchMessage "none"
ModelReferenceIOMismatchMessage "none"
ModelReferenceCSMismatchMessage "none"
ModelReferenceSimTargetVerbose off
UnknownTsInhSupMsg "warning"
ModelReferenceDataLoggingMessage "warning"
ModelReferenceSymbolNameMessage "warning"
ModelReferenceExtraNoncontSigs "error"
}
Simulink.HardwareCC {
  $ObjectID 6
  Version "1.1.0"
  ProdBitPerChar 8
  ProdBitPerShort 16
  ProdBitPerInt 32
  ProdBitPerLong 32
  ProdIntDivRoundTo "Undefined"
  ProdEndianess "Unspecified"
  ProdWordSize 32
  ProdShiftRightIntArith on
  ProdHWDeviceType "32-bit Generic"
  TargetBitPerChar 8
  TargetBitPerShort 16
  TargetBitPerInt 32
  TargetBitPerLong 32
  TargetShiftRightIntArith on
  TargetIntDivRoundTo "Undefined"
  TargetEndianess "Unspecified"
  TargetWordSize 32
  TargetTypeEmulationWarnSuppressLevel 0
  TargetPreprocMaxBitsSint 32
  TargetPreprocMaxBitsUint 32
  TargetHWDeviceType "Specified"
  TargetUnknown on
  ProdEqTarget on
}
Simulink.ModelReferenceCC {
  $ObjectID 7
  Version "1.1.0"

```

```

    UpdateModelReferenceTargets
    "IfOutOfDateOrStructuralChange"
    CheckModelReferenceTargetMessage "error"
    ModelReferenceNumInstancesAllowed "Multi"
    ModelReferencePassRootInputsByReference
    ModelReferenceMinAlgLoopOccurrences off
}
Simulink.RTWCC {
    $BackupClass          "Simulink.RTWCC"
    $ObjectID             8
    Array {
        Type              "Cell"
        Dimension         1
        Cell              "IncludeHyperlinkInReport"
        PropName          "DisabledProps"
    }
    Version               "1.1.0"
    SystemTargetFile     "grt.tlc"
    GenCodeOnly          off
    MakeCommand           "make_rtw"
    TemplateMakefile     "grt_unix.tmf"
    GenerateReport       off
    SaveLog              off
    RTWVerbose           on
    RetainRTWFile        off
    ProfileTLC           off
    TLCDebug             off
    TLCCoverage          off
    TLCAssert           off
    ProcessScriptMode    "Default"
    ConfigurationMode    "Optimized"
    ConfigAtBuild        off
    IncludeHyperlinkInReport off
    LaunchReport         off
    TargetLang           "C"
    Array {
        Type              "Handle"
        Dimension         2
        Simulink.CodeAppCC {
            $ObjectID     9
            Array {
                Type      "Cell"
                Dimension 9
                Cell      "IgnoreCustomStorageClasses"
                Cell      "InsertBlockDesc"
                Cell      "SFDataObjDesc"
                Cell      "SimulinkDataObjDesc"
                Cell      "DefineNamingRule"
                Cell      "SignalNamingRule"
                Cell      "ParamNamingRule"
                Cell      "InlinedPrmAccess"
                Cell      "CustomSymbolStr"
                PropName  "DisabledProps"
            }
            Version      "1.1.0"
            ForceParamTrailComments off
        }
    }
}

```



```

GenerateComments          on
IgnoreCustomStorageClasses on
IncHierarchyInIds        off
MaxIdLength              31
PreserveName              off
PreserveNameWithParent   off
ShowEliminatedStatement  off
IncAutoGenComments       off
SimulinkDataObjDesc      off
SFDataObjDesc            off
IncDataTypeInIds         off
PrefixModelToSubsysFcnNames on
CustomSymbolStr          "$R$N$M"
MangleLength             1
DefineNamingRule         "None"
ParamNamingRule          "None"
SignalNamingRule         "None"
InsertBlockDesc          off
SimulinkBlockComments   on
EnableCustomComments     off
InlinedPrmAccess        "Literals"
ReqsInCode               off
}
Simulink.GRTTargetCC {
  $BackupClass            "Simulink.TargetCC"
  $ObjectID                10
  Array {
    Type                  "Cell"
    Dimension              12
    Cell                   "IncludeMdlTerminateFcn"
    Cell                   "CombineOutputUpdateFcns"
    Cell                   "SuppressErrorStatus"
    Cell                   "ERTCustomFileBanners"
    Cell                   "GenerateSampleERTMain"
    Cell                   "MultiInstanceERTCode"
    Cell                   "PurelyIntegerCode"
    Cell                   "SupportNonFinite"
    Cell                   "SupportComplex"
    Cell                   "SupportAbsoluteTime"
    Cell                   "SupportContinuousTime"
    Cell                   "SupportNonInlinedSFcns"
    PropName               "DisabledProps"
  }
  Version                  "1.1.0"
  TargetFcnLib             "ansi_tfl_tmw.mat"
  TargetLibSuffix         ""
  TargetPreCompLibLocation ""
  GenFloatMathFcnCalls    "ANSI_C"
  UtilityFuncGeneration   "Auto"
  GenerateFullHeader      on
  GenerateSampleERTMain   off
  IsPILTarget             off
  ModelReferenceCompliant on
  IncludeMdlTerminateFcn  on
  CombineOutputUpdateFcns off
  SuppressErrorStatus     off
}

```

```

        IncludeFileDelimiter      "Auto"
        ERTCustomFileBanners      off
        SupportAbsoluteTime        on
        LogVarNameModifier         "rt_"
        MatFileLogging             on
        MultiInstanceERTCode       off
        SupportNonFinite           on
        SupportComplex              on
        PurelyIntegerCode          off
        SupportContinuousTime      on
        SupportNonInlinedSFCns    on
        ExtMode                     off
        ExtModeStaticAlloc         off
        ExtModeTesting             off
        ExtModeStaticAllocSize     1000000
        ExtModeTransport           0
        ExtModeMexFile             "ext_comm"
        RTWCAPISignals             off
        RTWCAPIParams              off
        RTWCAPIStates              off
        GenerateASAP2              off
    }
    PropName                       "Components"
}
}
PropName                          "Components"
}
Name                              "Configuration"
SimulationMode                     "normal"
CurrentDlgPage                     "Solver"
}
PropName                          "ConfigurationSets"
}
Simulink.ConfigSet {
    $PropName                       "ActiveConfigurationSet"
    $ObjectID                       1
}
BlockDefaults {
    Orientation                      "right"
    ForegroundColor                 "black"
    BackgroundColor                 "white"
    DropShadow                      off
    NamePlacement                   "normal"
    FontName                        "Helvetica"
    FontSize                        10
    FontWeight                      "normal"
    FontAngle                       "normal"
    ShowName                        on
}
BlockParameterDefaults {
    Block {
        BlockType                   ActionPort
        InitializeStates             "held"
        ActionType                   "unset"
    }
    Block {

```

```

BlockType          Constant
Value              "1"
VectorParams1D    on
OutDataTypeMode   "Inherit from 'Constant value'"
OutDataType        "sfixed(16)"
ConRadixGroup     "Use specified scaling"
OutScaling         "2^0"
SampleTime        "inf"
}
Block {
BlockType          DataTypeConversion
OutDataTypeMode   "Inherit via back propagation"
OutDataType        "sfixed(16)"
OutScaling         "2^0"
LockScale         off
ConvertRealWorld  "Real World Value (RWV)"
RndMeth           "Zero"
SaturateOnIntegerOverflow on
SampleTime        "-1"
}
Block {
BlockType          Demux
Outputs            "4"
DisplayOption     "none"
BusSelectionMode  off
}
Block {
BlockType          Derivative
LinearizePole     "inf"
}
Block {
BlockType          Fcn
Expr              "sin(u[1])"
SampleTime        "-1"
}
Block {
BlockType          Gain
Gain              "1"
Multiplication    "Element-wise (K.*u)"
ParameterDataTypeMode "Same as input"
ParameterDataType "sfixed(16)"
ParameterScalingMode "Best Precision: Matrix-wise"
ParameterScaling  "2^0"
OutDataTypeMode   "Same as input"
OutDataType        "sfixed(16)"
OutScaling         "2^0"
LockScale         off
RndMeth           "Floor"
SaturateOnIntegerOverflow on
SampleTime        "-1"
}
Block {
BlockType          If
NumInputs          "1"
IfExpression       "u1 > 0"
ShowElse           on
}

```

```

    ZeroCross          on
    SampleTime         "-1"
}
Block {
    BlockType          Inport
    UseBusObject       off
    BusObject          "BusObject"
    BusOutputAsStruct  off
    PortDimensions     "-1"
    SampleTime         "-1"
    DataType           "auto"
    OutDataType        "sfix(16)"
    OutScaling         "2^0"
    SignalType         "auto"
    SamplingMode       "auto"
    Interpolate        on
}
Block {
    BlockType          Integrator
    ExternalReset      "none"
    InitialConditionSource "internal"
    InitialCondition   "0"
    LimitOutput        off
    UpperSaturationLimit "inf"
    LowerSaturationLimit "-inf"
    ShowSaturationPort off
    ShowStatePort      off
    AbsoluteTolerance  "auto"
    IgnoreLimit        off
    ZeroCross          on
}
Block {
    BlockType          Math
    Operator           "exp"
    OutputSignalType   "auto"
    SampleTime         "-1"
    OutDataTypeMode    "Same as first input"
    OutDataType        "sfix(16)"
    OutScaling         "2^0"
    LockScale          off
    RndMeth            "Floor"
    SaturateOnIntegerOverflow on
}
Block {
    BlockType          Merge
    Inputs             "2"
    InitialOutput      "[]"
    AllowUnequalInputPortWidths off
    InputPortOffsets  "[]"
}
Block {
    BlockType          Mux
    Inputs             "4"
    DisplayOption      "none"
    UseBusObject       off
    BusObject          "BusObject"
}

```

```

    NonVirtualBus          off
}
Block {
    BlockType              Outputport
    Port                   "1"
    UseBusObject           off
    BusObject              "BusObject"
    BusOutputAsStruct      off
    PortDimensions         "-1"
    SampleTime             "-1"
    DataType               "auto"
    OutDataType            "sfix(16)"
    OutScaling             "2^0"
    SignalType             "auto"
    SamplingMode           "auto"
    OutputWhenDisabled     "held"
    InitialOutput          "[]"
}
Block {
    BlockType              Product
    Inputs                 "2"
    Multiplication          "Element-wise(.*)"
    InputSameDT            on
    OutDataTypeMode        "Same as first input"
    OutDataType            "sfix(16)"
    OutScaling             "2^0"
    LockScale              off
    RndMeth                "Zero"
    SaturateOnIntegerOverflow on
    SampleTime             "-1"
}
Block {
    BlockType              RelationalOperator
    Operator               ">="
    InputSameDT            on
    LogicOutDataTypeMode  "Logical (see Configuration
Parameters: Optimiza"
"tion)"
    LogicDataType          "uint(8)"
    ZeroCross              on
    SampleTime             "-1"
}
Block {
    BlockType              Scope
    Floating               off
    ModelBased             off
    TickLabels             "OneTimeTick"
    ZoomMode               "on"
    Grid                   "on"
    TimeRange              "auto"
    YMin                   "-5"
    YMax                   "5"
    SaveToWorkspace        off
    SaveName               "ScopeData"
    LimitDataPoints        on
    MaxDataPoints          "5000"
}

```

```

    Decimation          "1"
    SampleInput         off
    SampleTime          "0"
}
Block {
    BlockType           SignalGenerator
    WaveForm            "sine"
    TimeSource          "Use simulation time"
    Amplitude           "1"
    Frequency           "1"
    Units               "Hertz"
    VectorParams1D      on
}
Block {
    BlockType           SubSystem
    ShowPortLabels      on
    Permissions         "ReadWrite"
    PermitHierarchicalResolution "All"
    SystemSampleTime    "-1"
    RTWFcnNameOpts     "Auto"
    RTWFileNameOpts    "Auto"
    SimViewingDevice    off
    DataTypeOverride    "UseLocalSettings"
    MinMaxOverflowLogging "UseLocalSettings"
}
Block {
    BlockType           Sum
    IconShape           "rectangular"
    Inputs              "++"
    InputSameDT         on
    OutDataTypeMode     "Same as first input"
    OutDataType         "sfix(16)"
    OutScaling          "2^0"
    LockScale           off
    RndMeth             "Floor"
    SaturateOnIntegerOverflow on
    SampleTime          "-1"
}
Block {
    BlockType           Switch
    Criteria             "u2 >= Threshold"
    Threshold           "0"
    InputSameDT         on
    OutDataTypeMode     "Inherit via internal rule"
    RndMeth             "Floor"
    SaturateOnIntegerOverflow on
    ZeroCross           on
    SampleTime          "-1"
}
Block {
    BlockType           Terminator
}
Block {
    BlockType           TransferFcn
    Numerator           "[1]"
    Denominator         "[1 2 1]"
}

```

```

        AbsoluteTolerance      "auto"
        Realization            "auto"
    }
}
AnnotationDefaults {
    HorizontalAlignment        "center"
    VerticalAlignment          "middle"
    ForegroundColor            "black"
    BackgroundColor            "white"
    DropShadow                  off
    FontName                    "Helvetica"
    FontSize                    10
    FontWeight                  "normal"
    FontAngle                   "normal"
}
LineDefaults {
    FontName                    "Helvetica"
    FontSize                    9
    FontWeight                  "normal"
    FontAngle                   "normal"
}
System {
    Name                        "slcpnn"
    Location                     [2, 82, 1014, 739]
    Open                          on
    ModelBrowserVisibility       off
    ModelBrowserWidth            200
    ScreenColor                   "white"
    PaperOrientation              "landscape"
    PaperPositionMode            "auto"
    PaperType                     "usletter"
    PaperUnits                    "inches"
    ZoomFactor                    "100"
    ReportName                    "simulink-default.rpt"
    Block {
        BlockType                SubSystem
        Name                      "Cart & Pole\nDynamics"
        Ports                     [2, 4]
        Position                   [265, 185, 315, 240]
        ShowPortLabels             off
        TreatAsAtomicUnit          off
        MinAlgLoopOccurrences      off
        RTWSystemCode              "Auto"
        MaskType                   "Cart & Pole"
        MaskDescription             "Inverted Pendulum (cart & pole)"
        MaskHelp                   "Inverted Pendulum"
        MaskPromptString           "Initial Conditions (angle, angular
velocity, ca"
"rt position and cart velocity):|Physical Specifications (cart
mass and g):"
        MaskStyleString            "edit,edit"
        MaskTunableValueString     "on,on"
        MaskCallbackString         "|"
        MaskEnableString           "on,on"
        MaskVisibilityString       "on,on"
        MaskToolTipString          "on,on"
    }
}

```

```

MaskVarAliasString      ","
MaskInitialization      "init_cond = @1; spec =
@2;mc=spec(1);g=spec(2);"
MaskDisplay             "plot(-4,-1,4,6,[0 3 2.8 -0.2 0],[1 5 5.15
1.15 "
"1],[-1 -1 1 1 -1],[1 0 0 1 1],[-3.5 3.5],[0 0])"
MaskIconFrame           on
MaskIconOpaque          on
MaskIconRotate          "none"
MaskIconUnits           "autoscale"
MaskValueString         "[0, 0, 0, 0]||[1.0, 9.8]"
MaskTabNameString       ","
System {
Name                    "Cart & Pole\Dynamics"
Location                [261, 554, 1061, 859]
Open                    off
ModelBrowserVisibility off
ModelBrowserWidth      200
ScreenColor             "white"
PaperOrientation        "landscape"
PaperPositionMode      "auto"
PaperType               "usletter"
PaperUnits              "inches"
ZoomFactor              "100"
Block {
BlockType              Inport
Name                   "in_1"
Position               [60, 50, 80, 70]
Port                   "1"
IconDisplay            "Port number"
LatchInput             off
}
Block {
BlockType              Inport
Name                   "in_2"
Position               [55, 210, 75, 230]
Port                   "2"
IconDisplay            "Port number"
LatchInput             off
}
Block {
BlockType              Fcn
Name                   "Fcn2"
Position               [515, 220, 555, 240]
ShowName               off
Expr                   "(u(1)+u(7)*u(6)*(u(4)*u(4)*u(2)-
u(5)*u(3)))"
"/(mc+u(7))"
}
Block {
BlockType              Fcn
Name                   "Fcn3"
Position               [425, 85, 465, 105]
ShowName               off
Expr                   "(g*u(2)+u(3)*((-u(1)-
u(6)*u(5)*u(4)*u(4)*u("

```



```

"2))/(mc+u(6)))/(u(5)*(4/3-(u(6)*u(3)*u(3))/(mc+u(6))))"
}
Block {
  BlockType      Fcn
  Name            "Fcn4"
  Position        [145, 69, 200, 91]
  ShowName        off
}
Block {
  BlockType      Fcn
  Name            "Fcn5"
  Position        [145, 98, 200, 122]
  ShowName        off
  Expr            "cos(u[1])"
}
Block {
  BlockType      Gain
  Name            "Gain"
  Position        [110, 230, 145, 260]
  Gain            "0.1"
}
Block {
  BlockType      Gain
  Name            "Gain1"
  Position        [155, 205, 190, 235]
  Gain            "0.5"
}
Block {
  BlockType      Mux
  Name            "Mux1"
  Ports          [4, 1]
  Position        [280, 54, 315, 111]
  ShowName        off
}
Block {
  BlockType      Mux
  Name            "Mux2"
  Ports          [3, 1]
  Position        [465, 208, 495, 252]
  ShowName        off
  Inputs          "[4,1,2]"
}
Block {
  BlockType      Mux
  Name            "Mux3"
  Ports          [2, 1]
  Position        [375, 73, 405, 117]
  ShowName        off
  Inputs          "[4,2]"
}
Block {
  BlockType      Mux
  Name            "Mux5"
  Ports          [2, 1]
  Position        [270, 207, 300, 258]
  ShowName        off
}

```

```

    Inputs          "2"
  }
  Block {
    BlockType      Integrator
    Name           "theta"
    Ports          [1, 1]
    Position       [570, 85, 590, 105]
    InitialCondition "init_cond(1)"
  }
  Block {
    BlockType      Integrator
    Name           "theta dot"
    Ports          [1, 1]
    Position       [500, 85, 520, 105]
    InitialCondition "init_cond(2)"
  }
  Block {
    BlockType      Integrator
    Name           "x"
    Ports          [1, 1]
    Position       [635, 220, 655, 240]
    InitialCondition "init_cond(3)"
  }
  Block {
    BlockType      Integrator
    Name           "x dot"
    Ports          [1, 1]
    Position       [575, 220, 595, 240]
    InitialCondition "init_cond(4)"
  }
  Block {
    BlockType      Outputport
    Name           "out_1"
    Position       [735, 85, 755, 105]
    IconDisplay    "Port number"
    InitialOutput  "0"
  }
  Block {
    BlockType      Outputport
    Name           "out_2"
    Position       [740, 125, 760, 145]
    Port          "2"
    IconDisplay    "Port number"
    InitialOutput  "0"
  }
  Block {
    BlockType      Outputport
    Name           "out_3"
    Position       [740, 165, 760, 185]
    Port          "3"
    IconDisplay    "Port number"
    InitialOutput  "0"
  }
  Block {
    BlockType      Outputport
    Name           "out_4"

```

```

    Position      [740, 205, 760, 225]
    Port          "4"
    IconDisplay   "Port number"
    InitialOutput "0"
}
Line {
  SrcBlock      "theta"
  SrcPort      1
  Points        [0, 0]
  Branch {
    DstBlock    "out_1"
    DstPort    1
  }
  Branch {
    Points      [-25, 0; 0, -65; -445, 0; 0, 70]
    Branch {
      Points    [0, 10]
      DstBlock  "Fcn5"
      DstPort   1
    }
    Branch {
      DstBlock  "Fcn4"
      DstPort   1
    }
  }
}
Line {
  SrcBlock      "Fcn5"
  SrcPort      1
  Points        [20, 0; 0, -20]
  DstBlock      "Mux1"
  DstPort      3
}
Line {
  SrcBlock      "Fcn4"
  SrcPort      1
  Points        [10, 0; 0, -5]
  DstBlock      "Mux1"
  DstPort      2
}
Line {
  SrcBlock      "theta dot"
  SrcPort      1
  Points        [15, 0]
  Branch {
    DstBlock    "theta"
    DstPort    1
  }
  Branch {
    Points      [0, 40]
    Branch {
      DstBlock  "out_2"
      DstPort   1
    }
  }
  Branch {
    Points      [-305, 0; 0, -30]
  }
}

```

```

        DstBlock      "Mux1"
        DstPort      4
    }
}
Line {
    SrcBlock      "Fcn3"
    SrcPort      1
    Points      [10, 0]
    Branch {
        DstBlock      "theta dot"
        DstPort      1
    }
    Branch {
        Points      [0, 30; -45, 0; 0, 105]
        DstBlock      "Mux2"
        DstPort      2
    }
}
Line {
    SrcBlock      "Fcn2"
    SrcPort      1
    DstBlock      "x dot"
    DstPort      1
}
Line {
    SrcBlock      "Mux2"
    SrcPort      1
    DstBlock      "Fcn2"
    DstPort      1
}
Line {
    SrcBlock      "x dot"
    SrcPort      1
    Points      [0, 0]
    Branch {
        DstBlock      "x"
        DstPort      1
    }
    Branch {
        Points      [15, 0; 0, -15]
        DstBlock      "out_4"
        DstPort      1
    }
}
Line {
    SrcBlock      "x"
    SrcPort      1
    Points      [25, 0; 0, -55]
    DstBlock      "out_3"
    DstPort      1
}
Line {
    SrcBlock      "in_1"
    SrcPort      1
    DstBlock      "Mux1"

```

```

    DstPort          1
  }
  Line {
    SrcBlock         "Mux3"
    SrcPort          1
    DstBlock         "Fcn3"
    DstPort          1
  }
  Line {
    SrcBlock         "Mux1"
    SrcPort          1
    Points           [10, 0]
    Branch {
      Points         [0, 130]
      DstBlock       "Mux2"
      DstPort        1
    }
    Branch {
      DstBlock       "Mux3"
      DstPort        1
    }
  }
  Line {
    SrcBlock         "Gain"
    SrcPort          1
    DstBlock         "Mux5"
    DstPort          2
  }
  Line {
    SrcBlock         "Mux5"
    SrcPort          1
    Points           [15, 0]
    Branch {
      Points         [0, -130]
      DstBlock       "Mux3"
      DstPort        2
    }
    Branch {
      Points         [130, 0]
      DstBlock       "Mux2"
      DstPort        3
    }
  }
  Line {
    SrcBlock         "in_2"
    SrcPort          1
    Points           [15, 0]
    Branch {
      DstBlock       "Gain"
      DstPort        1
    }
    Branch {
      DstBlock       "Gain1"
      DstPort        1
    }
  }
}

```

```

Line {
  SrcBlock      "Gain1"
  SrcPort       1
  DstBlock      "Mux5"
  DstPort       1
}
Annotation {
  Name          "half pole length"
  Position      [224, 199]
  VerticalAlignment "top"
}
Annotation {
  Name          "pole mass"
  Position      [222, 257]
  VerticalAlignment "top"
}
}
}
Block {
  BlockType     Constant
  Name          "Constant"
  Position      [121, 155, 139, 175]
  Orientation   "up"
  NamePlacement "alternate"
  MaskIconFrame on
  MaskIconOpaque on
  MaskIconRotate "none"
  MaskIconUnits "autoscale"
}
Block {
  BlockType     Constant
  Name          "Constant1"
  Position      [80, 70, 100, 90]
  NamePlacement "alternate"
  ShowName      off
}
Block {
  BlockType     Reference
  Name          "Fuzzy Logic\nController"
  Ports        [1, 1]
  Position      [150, 270, 210, 320]
  Orientation   "left"
  FontName     "Arial"
  SourceBlock   "fuzblock/Fuzzy Logic \nController"
  SourceType    "FIS"
  ShowPortLabels on
  fis          "fismat"
}
Block {
  BlockType     Mux
  Name          "Mux1"
  Ports        [5, 1]
  Position      [320, 342, 360, 398]
  Orientation   "left"
  ShowName      off
  Inputs        "5"
}

```

```

MaskIconFrame          on
MaskIconOpaque         on
MaskIconRotate         "none"
MaskIconUnits          "autoscale"
}
Block {
  BlockType            Mux
  Name                  "Mux2"
  Ports                 [2, 1]
  Position              [450, 124, 480, 151]
  ShowName              off
  Inputs                "2"
}
Block {
  BlockType            Mux
  Name                  "Mux3"
  Ports                 [2, 1]
  Position              [595, 272, 625, 308]
  ShowName              off
  Inputs                "2"
}
Block {
  BlockType            Reference
  Name                  "PID Controller"
  Ports                 [1, 1]
  Position              [190, 140, 220, 170]
  SourceBlock           "simulink_extras/Additional\nLinear/PID
Controll"
"er"
  SourceType            "PID Controller"
  ShowPortLabels        on
  P                      "1"
  I                      "0.1"
  D                      "0.5"
}
Block {
  BlockType            Scope
  Name                  "Scope"
  Ports                 [1]
  Position              [500, 127, 520, 153]
  ShowName              off
  Location              [5, 56, 1029, 737]
  Open                  off
  NumInputPorts         "1"
  TickLabels            "on"
  List {
    ListType            AxesTitles
    axes1                "%<SignalLabel>"
  }
  TimeRange              "10.000000"
  YMin                   "-1.4e+008"
  YMax                   "1e+007"
  DataFormat             "Array"
}
Block {
  BlockType            Scope

```

```

Name "Scope1"
Ports [1]
Position [685, 257, 705, 283]
ShowName off
Location [1, 97, 1025, 767]
Open off
NumInputPorts "1"
TickLabels "on"
ZoomMode "yonly"
List {
ListType AxesTitles
axes1 "%<SignalLabel>"
}
TimeRange "10.000000"
YMin "-5.5"
YMax "5.5"
SaveName "ScopeData1"
DataFormat "Array"
}
Block {
BlockType Sum
Name "Sum"
Ports [2, 1]
Position [405, 255, 425, 275]
Orientation "down"
ShowName off
Inputs "+-"
MaskIconFrame on
MaskIconOpaque on
MaskIconRotate "none"
MaskIconUnits "autoscale"
}
Block {
BlockType Sum
Name "Sum1"
Ports [2, 1]
Position [160, 65, 180, 85]
ShowName off
}
Block {
BlockType SignalGenerator
Name "Target Position"
Ports [0, 1]
Position [55, 103, 100, 137]
Amplitude "0.400000"
Frequency "0.628320"
}
Block {
BlockType Constant
Name "Target Position\n(Mouse-Driven)"
Position [137, 185, 163, 205]
Orientation "up"
NamePlacement "alternate"
Value "-1.3651"
MaskIconFrame on
MaskIconOpaque on
}

```



```

MaskIconRotate      "none"
MaskIconUnits      "autoscale"
}
Block {
  BlockType          SignalGenerator
  Name               "Target Position1"
  Ports              [0, 1]
  Position           [65, 28, 110, 62]
  ShowName           off
  Amplitude          "0.500000"
  Frequency          "6.000000"
  Units              "rad/sec"
}
Block {
  BlockType          Reference
  Name               "purelin"
  Ports              [1, 1]
  Position           [470, 178, 495, 202]
  SourceBlock        "neural/Transfer Functions/purelin"
  SourceType         ""
  ShowPortLabels     on
}
Line {
  SrcBlock           "Sum"
  SrcPort            1
  Points             [0, 90]
  DstBlock           "Mux1"
  DstPort            3
}
Line {
  SrcBlock           "Cart & Pole\nDynamics"
  SrcPort            2
  Points             [110, 0; 0, 155]
  DstBlock           "Mux1"
  DstPort            2
}
Line {
  SrcBlock           "Cart & Pole\nDynamics"
  SrcPort            3
  Points             [90, 0]
  Branch {
    DstBlock          "Sum"
    DstPort           1
  }
  Branch {
    Points            [0, -75]
    DstBlock          "Mux2"
    DstPort           2
  }
}
Line {
  SrcBlock           "Cart & Pole\nDynamics"
  SrcPort            4
  Points             [80, 0; 0, 145]
  DstBlock           "Mux1"
  DstPort            4
}

```

```

}
Line {
  SrcBlock      "Mux2"
  SrcPort      1
  DstBlock     "Scope"
  DstPort      1
}
Line {
  SrcBlock      "Target Position\n(Mouse-Driven)"
  SrcPort      1
  Points       [0, -40; 20, 0]
}
Line {
  SrcBlock      "Target Position"
  SrcPort      1
  Points       [70, 0]
  DstBlock     "PID Controller"
  DstPort      1
}
Line {
  SrcBlock      "Sum1"
  SrcPort      1
  Points       [60, 0; 0, 65]
  Branch {
    Points     [0, 85]
  }
  DstBlock     "Cart & Pole\nDynamics"
  DstPort      2
}
Branch {
  Points     [145, 0; 0, 250]
  DstBlock  "Mux1"
  DstPort   5
}
}
Line {
  SrcBlock      "Target Position1"
  SrcPort      1
  Points       [10, 0; 0, 25]
  DstBlock     "Sum1"
  DstPort      1
}
Line {
  SrcBlock      "Constant1"
  SrcPort      1
  DstBlock     "Sum1"
  DstPort      2
}
Line {
  SrcBlock      "Mux3"
  SrcPort      1
  Points       [0, -20]
  DstBlock     "Scope1"
  DstPort      1
}
Line {
  SrcBlock      "Constant"

```

```

    SrcPort          1
    Points           [0, -20; 40, 0]
  }
  Line {
    SrcBlock         "Mux1"
    SrcPort          1
    Points           [-15, 0; 0, 5; -65, 0; 0, -50; -10,
0]
    DstBlock         "Fuzzy Logic\nController"
    DstPort          1
  }
  Line {
    SrcBlock         "Fuzzy Logic\nController"
    SrcPort          1
    Points           [-50, 0; 0, -45]
    Branch {
      Points         [0, -20; 125, 0; 0, -30]
      DstBlock       "Cart & Pole\nDynamics"
      DstPort        1
    }
    Branch {
      Points         [135, 0; 0, 30]
      DstBlock       "Mux3"
      DstPort        1
    }
  }
  Line {
    SrcBlock         "PID Controller"
    SrcPort          1
    Points           [195, 0]
    Branch {
      Points         [10, 0; 0, -25]
      DstBlock       "Mux2"
      DstPort        1
    }
    Branch {
      DstBlock       "Sum"
      DstPort        2
    }
  }
  Line {
    SrcBlock         "Cart & Pole\nDynamics"
    SrcPort          1
    DstBlock         "purelin"
    DstPort          1
  }
  Line {
    SrcBlock         "purelin"
    SrcPort          1
    Points           [5, 0; 0, 110]
    Branch {
      Points         [0, 50]
      DstBlock       "Mux1"
      DstPort        1
    }
    Branch {

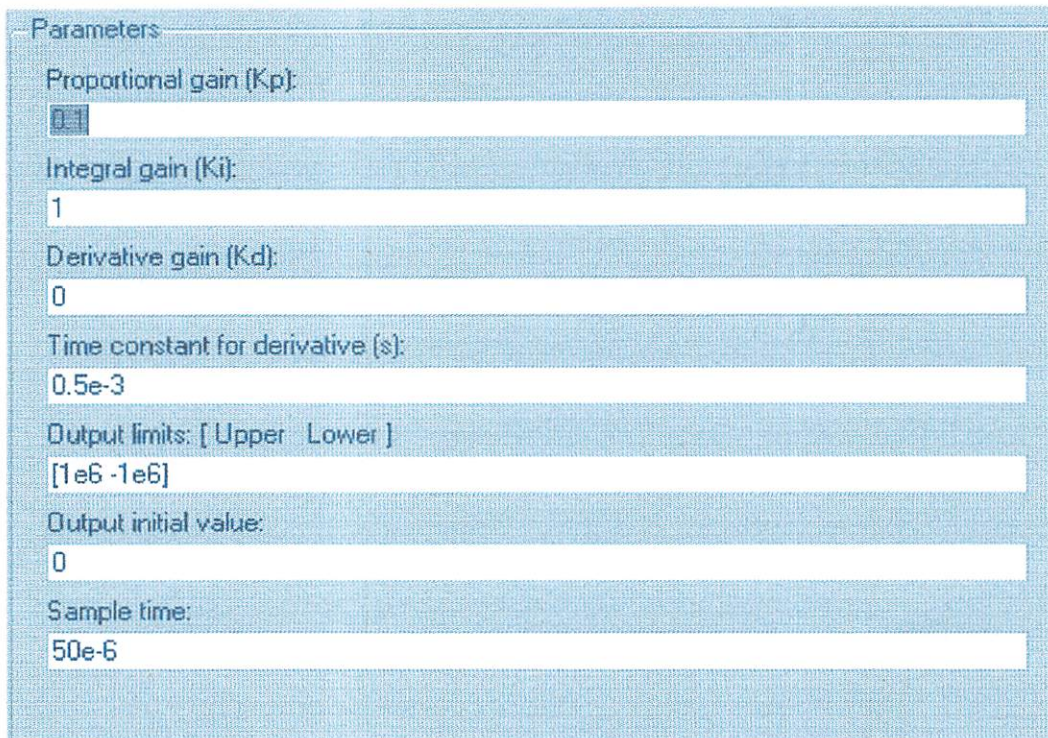
```

```
    DstBlock      "Mux3"
    DstPort       2
  }
}
Annotation {
  Name           "pole\nlength"
  Position       [202, 52]
  VerticalAlignment "top"
}
}
```

3.4. Penentuan PID

Pemindahan arus pada stator ke arus rotor menimbulkan fluks magnetik pada Pendulum, sehingga menimbulkan daya yang digunakan untuk menghitung besar arus yang menuju ke rotor.

Pada gambar 3-4 ditunjukkan perubahan fluks dari arus stator ke arus rotor.



The image shows a screenshot of a PID controller parameter configuration window. The window has a light blue background and a title bar that says "Parameters". Below the title bar, there are several input fields for different parameters:

- Proportional gain (Kp): 0.1
- Integral gain (Ki): 1
- Derivative gain (Kd): 0
- Time constant for derivative (s): 0.5e-3
- Output limits: [Upper Lower] [1e6 -1e6]
- Output initial value: 0
- Sample time: 50e-6

GAMBAR 3-4

**RANGKAIAN PERUBAHAN FLUKS DARI ARUS STATOR (ID) KE
ARUS ROTOR (IS)**