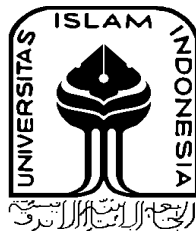


**PERANCANGAN, IMPLEMENTASI DAN EVALUASI
KONTROLER PID UNTUK SISTEM KENDALI ANTI-
SWAY PADA *ROTARY PENDULUM* DENGAN
KONFIGURASI *NORMAL ANTI-SWAY, POSITION
CONTROL ONLY, DAN DELAYED FEEDBACK
CONTROL***

TUGAS AKHIR

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



Disusun Oleh :

Nama : M. Ali Akbar Makkituju

No. Mahasiswa : 19525146

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

2023

LEMBAR PENGESAHAN DOSEN PEMBIMBING

**PERANCANGAN, IMPLEMENTASI DAN EVALUASI
KONTROLER PID UNTUK SISTEM KENDALI ANTI-
SWAY PADA *ROTARY PENDULUM* DENGAN
KONFIGURASI *NORMAL ANTI-SWAY, POSITION
CONTROL ONLY, DAN DELAYED FEEDBACK
CONTROL***

TUGAS AKHIR

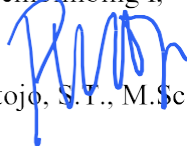
Disusun Oleh :

Nama : M. Ali Akbar Makkituju

No. Mahasiswa : 19525146

Yogyakarta, 13 November 2023

Pembimbing I,



Purtojo, S.T., M.Sc.

LEMBAR PENGESAHAN DOSEN PENGUJI

PERANCANGAN, IMPLEMENTASI DAN EVALUASI
KONTROLER PID UNTUK SISTEM KENDALI ANTI-
SWAY PADA *ROTARY PENDULUM* DENGAN
KONFIGURASI *NORMAL ANTI-SWAY, POSITION
CONTROL ONLY, DAN DELAYED FEEDBACK
CONTROL*

TUGAS AKHIR

Disusun Oleh :

Nama : M. Ali Akbar Makkituju

No. Mahasiswa : 19525146

Tim Penguji

Purtojo, S.T., M.Sc

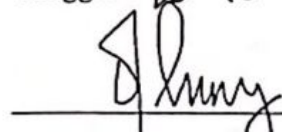
Ketua



Tanggal : 20-12-2023

Finny Pratama Putera, S.T., M.Eng.

Anggota I

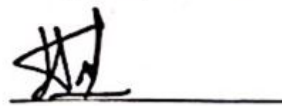


Tanggal : 5 Desember 2023

Donny Suryawan, Ir., S.T., M.Eng.,

IPP

Anggota II



Tanggal : 11-12-2023

Mengetahui

Dekan Fakultas Teknik Mesin



Donny Suryawan, Ir., S.T., M.T., IPP

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : M. Ali Akbar Makkituju
NIM : 19525146
Program Studi : Teknik Mesin
Fakultas : Fakultas Teknologi Industri
Judul : Perancangan, implementasi dan evaluasi kontroler
PID untuk sistem kendali *anti-sway* pada *rotary pendulum* dengan konfigurasi *normal anti-sway, position control only*, dan *delayed feedback control*

Dengan ini saya menyatakan bahwa karya tulis ilmiah yang saya buat merupakan karya sendiri dan bukan hasil plagiasi dari karya tulis orang lain. Semua referensi dan kutipan yang terdapat pada karya tulis ini telah saya cantumkan sitasi dan sumber Pustaka nya menyikuti tata cara pengutipan karya ilmiah yang benar. Apabila kemudian hari saya di anggap melakukan pelanggaran hak kekayaan intelektual dan terbukti melanggar hak tersebut, maka saya akan bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Yogyakarta, 22 Desember 2023



M. Ali Akbar Makkituju

HALAMAN PERSEMBAHAN

Tugas akhir ini kami persembahkan kepada keluarga, saudara dan orang-orang terdekat saya yang senantiasa dan mendukung saya disetiap langkah yang saya ambil, terkhusus kepada kedua orang tua saya :

Ayah dan Ibu tercinta:

Bapak Aksar dan Ibu Azizah

Sebagai tanda bukti atas Amanah yang diberikan kepada saya untuk melanjutkan Pendidikan ke jenjang Strata 1. Terimakasih telah berusaha mendidik dan mengantarkan saya mencapai gelar serjana. Semoga gelar ini dapat menjadi jembatan saya untuk membanggakan kedua orang tua saya, dan segala proses perjuangan beliau-beliau dapat menjadi pahala yang bermanfaat di akhirat kelak.

Aamin ya rabbal alamin

HALAMAN MOTTO

وَلِرَبِّكَ فَاصْبِرْ

“Dan untuk (memenuhi perintah) Tuhanmu, bersabarlah.”. (QS. Al-Muddassir :
7).

يَا أَيُّهَا الَّذِينَ آمَنُوا اسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ ۚ إِنَّ اللَّهَ مَعَ
الصَّابِرِينَ

“Hai orang-orang yang beriman, jadikanlah sabar dan shalat sebagai penolongmu,
sesungguhnya Allah beserta orang-orang yang sabar.”(QS AlBaqarah : 153)

KATA PENGANTAR

Assalamualaikum warahmatullah wabarakatuh

Puji syukur penulis panjatkan kepada Allah SWT atas nikmatNya yang telah diberikan sehingga penulis dapat menulis laporan yang berjudul : Perancangan, implementasi dan evaluasi kontroler PID untuk sistem kendali *anti-sway* pada *rotary pendulum* dengan konfigurasi *normal anti-sway, position control only*, dan *delayed feedback control* hingga selesai. Shalawat serta salam penulis haturkan kepada baginda Nabi Muhammad SAW, dan juga kepada keluarganya, sahabatnya, dan pengikut setianya hingga akhir zaman.

Laporan Tugas akhir ini disusun sebagai salah satu syarat bagi mahasiswa untuk memperoleh gelar Sarjana Teknik Mesin (S.T.) pada Program Studi Teknik Mesin Fakultas Teknologi Industri di Universitas Islam Indonesia. Penulis haturkan terimakasih banyak kepada pihak-pihak yang telah bersedia membrikan bantuan dan saran sehingga penulis dapat menyelesaikan skripsi ini dengan baik, *wa bil khusus* untuk :

1. Cinta pertama penulis, ibunda Azizah dan figure teladan penulis ayahanda Aksar yang tidak berhenti-hentinya selalu mendoakan, mendukung, memberikan kasih sayang kepada penulis sampai penulis dapat menyelesaikan laporan ini. Terimakasih atas doa dan motivasi yang selalu diberikan, terimakasih sudah menjadi *support system* terbaik bagi penulis.
2. Bapak dosen pembimbing, bapak Purtojo, S.T., M.Sc., terimakasih atas bimbingan dan ilmu yang telah bapak berikan dalam menyelesaikan Tugas Akhir ini, terimakasih sudah meluangkan waktu untuk membimbing penulis di sela kesibukan bapak. Menjadi salah satu mahasiswa bimbingan bapak merupakan anugrah yang sampai saat ini selalu penulis syukuri. Terimakasih bapak, semoga bapak dan keluarga selalu diberikan kesehatan dan dilimpahkan banyak kebahagiaan.
3. Teman-teman penulis di Teknik Mesin, Hafiz halomoan Sormin, Reyhan Hilmi Arsyad, dan Deza Alfarend terimakasih sudah memberikan *support* dan motivasi agar penulisan ini dapat terseleikan

4. Seluruh jajaran Prodi Teknik Mesin FTI UII yang telah membantu kelancaran penulisan berupa petunjuk dan arahan, baik dalam hal yang bersifat administrasi maupun hal akademik.
5. Naura Maysahaq yang selalu memberikan dukungan serta motivasi, menjadi tempat penulis untuk bercerita dan berkeluh kesah, sehingga penulis dapat terus bersemangat dalam penulisan ini hingga selesai. Semoga kita selalu dalam lindungan Allah SWT, dan semoga apa yang kita cita-citakan dapat tercapai dan dimudahkan dalam ridhoNya. *Aamin aamin ya rabbal aalamiin*

Akhir kata, penulis memohon maaf sekiranya apabila terdapat kesalahan-kesalahan, baik khilaf maupun disengaja, selama proses penulisan tugas akhir ini. Penulis sadar atas kekurangan penulis, dan sebagai manusia penulis sadar atas ketidaksempurnaan pada diri penulis. Penulis juga membuka kritik dan saran jikalau terdapat hal-hal yang perlu diperbaiki di masa mendatang. Semoga penulisan tugas akhir ini dapat menjadi manfaat dan ikut andil dalam pembuatan karya lain selanjutnya.

Sekian dan terimakasih.

Wassalamualaikum warahmatullahi wabarakatuh.

ABSTRAK

Crane merupakan sebuah alat pengangkut barang-barang berat menggunakan derek jangkung yang berfungsi untuk memindahkan beban dari satu titik ke titik yang lain dengan jarak tertentu. Dalam skala eksperimen, *crane* dapat dimodelkan dalam bentuk sederhana yang bertujuan untuk simulasi kendali yang cocok digunakan pada *crane*. Salah satu model sederhana dari *crane* adalah *rotary pendulum*. Di dalam penelitian kendali *rotary pendulum*, posisi sudut pada lengan pendulum dan pendulum ayun menjadi objek utama yang akan dikendalikan. Hal ini bertujuan agar *rotary pendulum* dapat berputar dan berhenti pada sudut yang diinginkan. *Rotary pendulum* digerakkan menggunakan motor DC yang perputaran sudut pada lengan pendulum dan pendulum ayunnya dilacak menggunakan sensor encoder. Penelitian ini akan berfokus pada perancangan, implementasi, dan evaluasi pada tiga konfigurasi yang berbeda yaitu *normal anti-sway control*, *position control only*, dan *delayed feedback control* dengan menggunakan kendali PID +PD. Alat *rotary pendulum* akan dimodelkan dalam bentuk model matematis, hal ini dilakukan untuk mencari nilai dari konstanta PID+PD yang akan digunakan.

Kata kunci: *rotary pendulum*, posisi sudut, *normal anti-sway control*, *position control only*, *delayed feedback control*, PID+PD

Daftar Isi

Halaman Judul	i
Lembar Pengesahan Dosen Pembimbing	ii
Lembar Pengesahan Dosen Penguji	Error! Bookmark not defined.
Pernyataan Keaslian	v
Halaman Persembahan	vi
Halaman Motto	vii
Kata Pengantar.....	ix
Abstrak	ixx
Daftar Isi	x
Daftar Tabel.....	xiii
Daftar Gambar	xiii
Daftar Notasi	xvii
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	4
Bab 2 Tinjauan Pustaka	6
2.1 Kajian Pustaka	6
2.2 Dasar Teori	9
2.2.1 <i>Eksperimen setup</i>	9
2.2.2 Sistem kendali	23
2.2.3 Model matematis	25
2.2.4 PID.....	28
Bab 3 Metode Penelitian.....	31
3.1 Alur Penelitian	31
3.2 Metode Pengujian	32
3.2.1 Simulasi model matematis.....	32
3.2.2 Pengujian kendali PID objek kendali <i>rotary pendulum</i>	32

Bab 4 Hasil dan Pembahasan	34
4.1 Hasil Perhitungan Model Matematis	34
4.1.1 Hasil fungsi <i>transfer</i> motor DC	35
4.1.2 Hasil fungsi <i>transfer</i> pendulum ayun	36
4.2 Pengujian Fungsi <i>Transfer</i> dan Alat Rotary Pendulum	38
4.2.1 Pengujian menggunakan <i>normal anti-sway control</i> dengan PID+PD <i>Ziegler Nichols</i>	38
4.2.2 Pengujian menggunakan <i>normal anti-sway control</i> dengan PID Tyreus-Luyben +PD <i>Ziegler Nichols</i>	43
4.2.3 Pengujian menggunakan <i>anti-swing control using only position control</i> dengan pengendali PID <i>Ziegler Nichols</i>	48
4.2.4 Pengujian menggunakan <i>anti-swing control using only position control</i> dengan pengendali PID <i>Tyreus-Luyben</i>	50
4.2.5 Pengujian menggunakan <i>delayed feedback control</i> dengan PID+PD <i>Ziegler-Nichols</i>	53
4.2.6 Pengujian menggunakan <i>delayed feedback control</i> dengan PID <i>Tyreus-Luyben</i> + PD <i>Ziegler Nichols</i>	57
4.3 Perbandingan	62
4.3.1 Perbandingan fungsi <i>transfer</i> sistem kendali	62
4.3.2 Perbandingan sistem <i>anti-sway</i> pada objek kendali	63
4.4 Pembahasan	64
Bab 5 Penutup.....	66
5.1 Kesimpulan	66
5.2 Saran	66
Daftar Pustaka	68

DAFTAR TABEL

Tabel 2-1 Blok-blok yang digunakan	15
Tabel 2-2 Pengaruh masing-masing konstanta PID (Toochinda, 2016).....	29
Tabel 2-3 Perhitungan konstanta PID metode <i>Ziegler-Nichols</i>	30
Tabel 2-4 Perhitungan konstanta PID metode <i>Tyreus-Luyben</i>	30
Tabel 4-1 Identifikasi data alat <i>rotary pendulum</i>	34

DAFTAR GAMBAR

Gambar 2-1 Skema kontroler <i>normal anti-sway control</i> (Solihin & Wahyudi, 2006)	7
Gambar 2-2 Skema kontroler <i>anti-swing control using only position control</i> (Solihin & Wahyudi, 2006)	7
Gambar 2-3 Skema kontroler <i>delayed feedback</i> (Omar, 2003).....	8
Gambar 2-4 Mekanisme <i>rotary pendulum</i>	9
Gambar 2-5 Arduino Mega 2560	10
Gambar 2-6 <i>Driver</i> motor L298N	11
Gambar 2-7 <i>Optical rotary encoder</i>	12
Gambar 2-8 Contoh penggunaan aplikasi Xcos	12
Gambar 2-9 Model perancangan perangkat keras alat <i>rotary pendulum</i>	13
Gambar 2-10 Koneksi elektrik alat <i>rotary pendulum</i>	14
Gambar 2-11 Objek kendali <i>rotary pendulum</i>	15
Gambar 2-12 Xcos pengecekan pulsa sensor encoder	18
Gambar 2-13 Konfigurasi sensor encoder motor DC	19
Gambar 2-14 Konfigurasi sensor encoder pendulum ayun	19
Gambar 2-15 Grafik pembacaan pulsa encoder motor DC	19
Gambar 2-16 Grafik pembacaan pulsa encoder pendulum	20
Gambar 2-17 Skema Xcos untuk respon motor DC	20
Gambar 2-18 Konfigurasi pada blok Xcos motor DC	21
Gambar 2-19 Respon posisi motor DC	21
Gambar 2-20 Skema Xcos untuk respon <i>osilasi</i> pendulum.....	22
Gambar 2-21 Respon pendulum ayun	23
Gambar 2-22 Sistem kendali sederhana	23
Gambar 2-23 Sistem <i>open loop</i>	24
Gambar 2-24 Sistem <i>closed loop</i>	24
Gambar 2-25 Grafik <i>closed loop</i>	25
Gambar 2-26 Ilustrasi pergerakan pendulum sederhana	26
Gambar 2-27 Respon sistem menggunakan <i>damping</i>	27
Gambar 3-1 Alur penelitian.....	31

Gambar 3-2 Skema kontroler <i>normal anti-sway control</i>	33
Gambar 3-3 Skema kontroler <i>position only control</i>	33
Gambar 3-4 Skema kontroler <i>delayed feedback</i>	33
Gambar 4-1 Konfigurasi pengecekan data kecepatan pada rumus fungsi transfer motor DC	35
Gambar 4-2 Hasil pemeriksaan kecepatan pada rumus fungsi transfer motor DC	36
Gambar 4-3 Rancangan pendulum ayun	37
Gambar 4-4 Konfigurasi mencari nilai PID menggunakan <i>Ziegler-Nichols</i>	39
Gambar 4-5 Grafik respon fungsi <i>transfer</i> motor DC ketika diberinilai $K_p = 10$.	40
Gambar 4-6 Grafik respon model matematis setelah diberi nilai PID menggunakan metode <i>Ziegler-Nichols</i> ,	40
Gambar 4-7 Konfigurasi mencari nilai PD metode <i>Ziegler-Nichols</i>	41
Gambar 4-8 Grafik respon fungsi <i>transfer</i> pendulum ketika diberi nilai $K_p = 0,5$	41
Gambar 4-9 Grafik respon model matematis setelah diberi kendali PID+PD <i>Ziegler-Nichols</i>	42
Gambar 4-10 Skema kendali <i>normal anti-sway pada alat rotary pendulum</i>	42
Gambar 4-11 Grafik respon kendali alat <i>rotary pendulum</i> dengan kendali PID+PD <i>Ziegler-Nichols</i> menggunakan skema <i>normal anti-sway</i>	43
Gambar 4-12 Konfigurasi mencari nilai PID menggunakan metode <i>Tyresus -Luyben</i>	44
Gambar 4-13 Grafik respon fungsi <i>transfer</i> motor DC ketika diberiniali $K_p = 10$	44
Gambar 4-14 Respon model matematis ketika diberi kendali PID <i>Tyresus -Luyben</i>	45
Gambar 4-15 Konfigurasi mencari nilai PD metode <i>Ziegler-Nichols</i>	45
Gambar 4-16 Grafik respon fungsi <i>transfer</i> pendulum ketika diberinilai $K_p = 0,1$	46
Gambar 4-17 Respon model matematis kendali setelah diberi kendali PID <i>Tyresus -Luyben</i> +PD <i>Ziegler-Nichols</i>	46
Gambar 4-18 Grafik respon kendali alat <i>rotary pendulum</i> dengan kendali PID <i>Tyresus -Luyben</i> + PD <i>Ziegler-Nichols</i> menggunakan skema <i>normal anti-sway</i> .	47

Gambar 4-19 Konfigurasi mencari nilai PID menggunakan <i>Ziegler-Nichols</i>	48
Gambar 4-20 Grafik respon fungsi <i>transfer</i> motor DC ketika diberiniali $K_p = 0,001$	49
Gambar 4-21 Grafik respon model matematis ketika diberi nilai kendali PID <i>Ziegler-Nichols</i>	49
Gambar 4-22 Skema sistem kendali <i>control position only</i> menggunakan kendali PID <i>Ziegler-Nichols</i>	50
Gambar 4-23 Grafik respon kendali alat <i>rotary pendulum</i> dengan kendali PID <i>Ziegler-Nichols</i> menggunakan skema <i>anti-swing control using only position control</i>	50
Gambar 4-24 Konfigurasi mencari nilai PID menggunakan <i>Tryeus-Luyben</i>	51
Gambar 4-25 Grafik respon fungsi <i>transfer</i> motor DC ketika diberiniali $K_p = 0,001$	51
Gambar 4-26 Respon model matematis ketika diberi nilai kendali PID <i>Tryeus- Luyben</i>	52
Gambar 4-27 Grafik respon kendali alat <i>rotary pendulum</i> dengan kendali PID <i>Tryeus-Luyben</i> menggunakan skema <i>anti-wing control using only position control</i>	52
Gambar 4-28 Konfigurasi blok PID pada pendulum ayun.....	53
Gambar 4-29 Konfigurasi dan skema mencari nila PID menggunakan <i>Ziegler- Nichols</i>	54
Gambar 4-30 Grafik respon fungsi <i>transfer</i> motor DC setelah diberi nilai $K_p = 9$	54
Gambar 4-31 Respon fungsi <i>transfer</i> motor DC setelah diberi nilai PID menggunakan metode <i>Ziegler-Nichols</i>	55
Gambar 4-32 Konfigurasi mencari nilai PD metode <i>Ziegler-Nichols</i>	55
Gambar 4-33 Grafik respon fungsi <i>transfer</i> pendulum ketika diberinilai $K_p = 0,2$	56
Gambar 4-34 <i>Error</i> pada simulasi X_{cos}	56
Gambar 4-35 Skema kendali <i>delayed feedback</i> pada alat <i>rotary pendulum</i>	57
Gambar 4-36 Grafik respon kendali alat <i>rotary pendulum</i> dengan kendali PID+PD <i>Ziegler-Nichols</i> menggunakan skema <i>delayed feedback</i>	57

Gambar 4-37 Konfigurasi blok PID pada pendulum ayun.....	58
Gambar 4-38 Konfigurasi dan skema mencari nilai PID menggunakan <i>Tyresus-Luyben</i>	58
Gambar 4-39 Grafik respon fungsi <i>transfer</i> motor DC (kiri) setelah diberi nilai $K_p = 9$	59
Gambar 4-40 Grafik respon fungsi <i>transfer</i> motor DC setelah diberi nilai PID menggunakan metode <i>Tyresus-Luyben</i>	59
Gambar 4-41 Konfigurasi mencari nilai PD metode <i>Tyresus-Luyben</i>	60
Gambar 4-42 Grafik respon fungsi <i>transfer</i> pendulum ketika diberinilai $K_p = 0,3$	60
Gambar 4-43 <i>Error</i> pada simulasi X_{cos}	61
Gambar 4-44 Skema kendali <i>delayed feedback</i> pada alat <i>rotary pendulum</i>	61
Gambar 4-45 Grafik respon kendali alat <i>rotary pendulum</i> dengan kendali PID <i>Tyresus-Luyben</i> + PD <i>Ziegler-Nichols</i> menggunakan skema <i>delayed feedback</i> ...	62
Gambar 4-46 Grafik respon perbandingan fungsi <i>transfer</i> sistem kendali	62
Gambar 4-47 Objek kendali tanpa kontrol pengendali.....	63
Gambar 4-48 Perbandingan sistem <i>anti-sway</i> pada objek kendali	64

DAFTAR NOTASI

PID	: <i>Proportional, integral, dan derivative</i>	
PD	: <i>Proportional dan derivative</i>	
K_P	: <i>Konstanta proportional</i>	
K_i	: <i>Konstanta integral</i>	
K_d	: <i>Konstanta derivative</i>	
$\theta_m(s)$: <i>Posisi sudut</i>	(Rad)
$E_a(s)$: <i>Tegangan Input</i>	(V)
K_t	: <i>Torque constant motor DC</i>	(Nm/A)
R_a	: <i>Terminal resistance motor DC</i>	(Ω)
K_b	: <i>Speed constant motor DC</i>	(rpm/V)
J_m	: <i>Momen inersia motor DC</i>	($kg \cdot m^2$)
D_m	: <i>Redaman</i>	(rpm / mNm)
ω_n	: <i>Frekuensi natural</i>	
ζ	: <i>Konstanta damping</i>	
$H(s)$: <i>Fungsi transfer pendulum ayun</i>	

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sistem kendali atau sistem kendali merupakan suatu sistem yang berfungsi untuk mengendalikan, memerintah dan mengatur keadaan dari suatu sistem. Penggunaan sistem kendali sudah banyak diterapkan atau digunakan pada segala bidang baik di bidang industri manufaktur ataupun bidang-bidang lainnya. Sistem kendali pada saat ini terus mengalami perkembangan seiring dengan perkembangan teknologi dimana perkembangan ini membawa kemajuan dan kemudahan kepada manusia dalam hal pemecahan masalah.

Salah satu dari penggunaan sistem kendali dapat ditemukan pada alat *crane*. *Crane* merupakan suatu derek jangkung yang dioperasikan oleh operator untuk mengangkat material konstruksi maupun material lainnya dari satu tempat ke tempat lainnya. Penggunaan alat *crane* sangat bermanfaat karena mampu mengurangi tenaga manusia untuk mengangkut barang-barang yang berat.

Pemindahan barang-barang yang berat menimbulkan ayunan yang dapat beresiko menimbulkan bahaya pada orang-orang yang berada di sekitar *crane*. Ayunan tersebut menjadi semakin berbahaya ketika ayunan *crane* mengakibatkan terjatuhnya material yang sedang diangkut.

Untuk mengurangi hal tersebut, perlu dilakukan pengendalian sistem *anti-sway* pada alat *crane*. Sistem *anti-sway* ini akan mengurangi terjadinya ayunan ketika alat sedang beroperasi. Terdapat banyak penelitian yang membahas tentang penerapan *anti-sway* seperti penelitian oleh (Solihin, Legowo, & Wahyudi, 2010) yang dimana pada penelitian ini menggunakan pengendali fuzzy dan PID untuk mengendalikan ayunan pada *gantry crane*. Penelitian oleh (Omar, 2003) tentang *control of gantry crane and tower crane* menggunakan pengendali PD dan juga fuzzy untuk mengendalikan motor penggerak dan ayunan pendulum. Pada penelitian ini terdapat dua skema sistem kendali yaitu *normal anti-sway control* dan *delayed feedback control*. Terdapat juga penelitian sistem kendali *anti-sway overhead crane* oleh (Akbar, Rahmdani, & Wibowo, 2016) yang dimana penelitian

ini menggunakan metode sistem kendali PD untuk mengendalikan *anti-sway* pada *prototipe overhead crane*. Penelitian mengenai pengendalian ayunan pendulum akibat pergerakan *linier* yang disebabkan oleh motor DC yang dilakukan oleh (Triyogi, 2017) dimana pada penelitian ini metode pengendali yang digunakan ialah *Ziegler-Nichols* dan metode *trial and error*. Penelitian-penelitian diatas merupakan penelitian yang membahas sistem kendali bagi objek kendali yang memiliki pergerakan *linier*. Lalu bagaimana dengan pengendalian ayunan yang terjadi akibat objek kendali bergerak *rotasi*?

Penelitian mengenai *anti-sway* akibat gerak *rotasi* sudah dilakukan pada penelitian sebelumnya oleh (Hadiyan, 2021). Pada penelitian ini menggunakan pengendali PID+PD untuk pengendali dari motor penggerak dan pengendali ayunan pendulum. Pada penelitian ini objek yang dikendalikan adalah prototipe *rotary pendulum* yang dimana objek kendali yang digunakan sudah terintegrasi dengan perangkat lunak Scilab Xcos. Tetapi penelitian (Hadiyan, 2021) ini hanya menggunakan satu skema kontroler yaitu skema kontroler *normal anti-sway*.

Penelitian kali ini akan menggunakan perangkat lunak Scilab Xcos dan juga Arduino IDE. Scilab Xcos berfungsi untuk melakukan simulasi objek kendali *rotary pendulum* dan merancang model sistematis penggunaan sistem kendali. Model sistem kendali yang akan digunakan merupakan metode kendali PID (*Propotional, Integral, Derivativ*). Penggunaan metode kendali PID ini bertujuan untuk mengurangi ayunan yang terjadi pada pendulum akibat dari pergerakan rotasi yang dihasilkan dari motor DC. Pada penelitian ini dilakukan perancangan, implementasi dan evaluasi sistem sistem *anti-sway* menggunakan kendali PID dengan 3 skema kontroler yang berbeda.

1.2 Rumusan Masalah

Berikut ini adalah rumusan masalah penelitian terkait perancangan, implementasi dan evaluasi kontroler PID untuk sistem kendali *anti-sway* pada *rotary pendulum* dengan konfigurasi *normal anti-sway control, position control only dan delayed feedback control* :

1. Bagaimana cara mensimulasikan model matematis objek kendali *rotary pendulum*?

2. Bagaimana cara merancang sistem kendali *anti-sway* menggunakan metode PID dengan konfigurasi *normal anti-sway control*, *position control only* dan *delayed feedback control* kemudian mengimplementasikan sistem tersebut pada objek kendali *rotary pendulum*?
3. Bagaimana hasil dari penerapan sistem *anti-sway* pada objek kendali *rotary pendulum*?

1.3 Batasan Masalah

Berikut ini adalah batasan masalah untuk penelitian perancangan, implementasi dan evaluasi kontroler PID untuk sistem kendali *anti-sway* pada *rotary pendulum* dengan konfigurasi *normal anti-sway control*, *position control only* dan *delayed feedback control*:

1. Alat uji yang digunakan adalah alat uji *rotary pendulum*.
2. Metode kendali yang digunakan adalah PID.
3. Penelitian yang dilakukan sampai dengan implementasi sistem kendali PID pada alat *rotary pendulum*.
4. Perangkat lunak yang digunakan adalah Scilab versi 5.5.2 dan Arduino IDE.
5. Arduino Mega 2560 digunakan sebagai mikrokontroler.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Mensimulasikan model matematis objek kendali *rotary pendulum*.
2. Merancang sistem kendali *anti-sway* menggunakan kendali PID dengan konfigurasi *normal anti-sway control*, *position control only* dan *delayed feedback control* yang dapat diimplementasikan pada objek kendali *rotary pendulum*.
3. Mengevaluasi penerapan sistem *anti-sway* dengan kendali PID pada kendali *rotary pendulum*.

1.5 Manfaat Penelitian

Manfaat pada penelitian ini adalah dapat mengevaluasi kinerja dari objek kendali menggunakan sistem *anti-sway* dengan kendali PID pada konfigurasi yang berbeda. Penelitian ini juga mampu menerapkan pembelajaran yang telah dipelajari pada masa kuliah. Penelitian ini juga dapat digunakan sebagai referensi pembelajaran penggunaan perangkat lunak Scilab Xcos untuk simulasi *anti-sway* yang menggunakan PID sebagai sistem kendali. Bagi dunia industri, penelitian ini dapat menunjukkan efektifitas skema kontroler yang berbeda seperti pengendali *normal anti-sway control*, *anti-swing control using only position sensor*, dan *delayed feedback* dengan menggunakan kendali PID untuk sebuah sistem kendali, seperti pada *crane*, pengendalian *balancing system*, dan sistem-sistem kendali lainnya.

1.6 Sistematika Penulisan

Sistematika penulisan pada ini dibagi menjadi lima bab, diantaranya:

BAB 1 PENDAHULUAN

Bab ini terdiri dari latar belakang, rumusan masalah yang dihadapi, batasan masalah, tujuan serta manfaat dari pelaksanaan penelitian, dan sistematika penulisan laporan.

BAB II TINJAUAN PUSTAKA

Bab ini terdiri dari landasan teori dari penelitian, hal ini terkait dengan rumus-rumus yang akan digunakan dan juga referensi-referensi dari penelitian sebelumnya dan perangkat keras yang digunakan, serta pengaturan perangkat lunak seperti pengaturan pada aplikasi Scilab Xcos.

BAB III METODE PENELITIAN

Bab ini berisi tentang alur penelitian yang dikerjakan dan metode penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan juga pembahasan dari penelitian yang telah dilakukan.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian serta saran untuk penelitian selanjutnya.

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

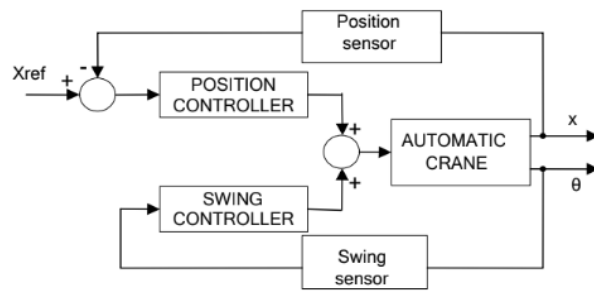
Penelitian mengenai sistem *anti-sway* sudah banyak dilakukan dan juga dikembangkan oleh para peneliti-peneliti sebelumnya. Tetapi kebanyakan referensi yang membahas mengenai *anti-sway* pada sistem derek yang terjadi akibat pergerakan linier. Untuk pembahasan mengenai sistem *anti-sway* akibat pergerakan rotasi seperti yang terdapat pada alat *rotary pendulum* masih jarang dilakukan.

Sebelum melakukan penelitian, diperlukan kajian pustaka yang dapat menjadi referensi penelitian yang akan dilakukan. Kajian pustaka pada penelitian ini terfokus pada beberapa topik bahasan, diantaranya adalah:

1. Penggunaan sistem kendali seperti PID + PD untuk mendapatkan sistem peredam ayunan yang diinginkan dan pengendalian motor DC.
2. Perbandingan konfigurasi pengendali yang berbeda untuk sistem kendali *anti-sway*.

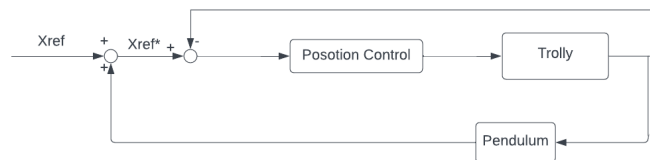
Berdasarkan topik pembahasan diatas, didapatkan beberapa referensi yang membahas mengenai sistem *anti-sway*. Seperti pada jurnal yang membahas tentang penerapan kontrol *anti-sway* pada alat *gantry crane* otomatis dengan sensor minimum (Solihin & Wahyudi, 2006). Pada penelitian ini menggunakan metode PID+PD untuk mengendalikan *gantry crane* otomatis. Pada penelitian ini kendali PID digunakan untuk mengendalikan motor penggerak sedangkan pengendali PD digunakan untuk mengendalikan ayunan pendulum. Pada penelitian ini terdapat dua skema kontroler diantaranya: *Normal anti-sway control* dan *anti-swing control using position control only*.

Normal anti-sway control, skema kontroler ini memiliki *normal feedback control* dan memiliki dua pengendali yaitu kendali untuk mengatur posisi motor gerak dan kendali untuk pendulum ayun. Gambar 2-1 menunjukkan bentuk dari skema kontroler *normal anti-sway control*.



Gambar 2-1 Skema kontroler *normal anti-sway control* (Solihin & Wahyudi, 2006)

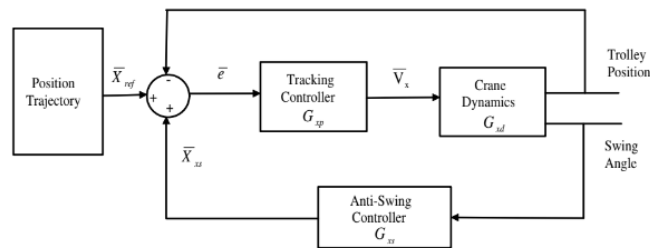
Kemudian ada skema *anti-swing control using position control only*, Skema kontroler ini memiliki perbedaan dengan skema kontroler sebelumnya. Pada skema kontroler ini kendali yang digunakan hanyalah pengendali untuk motor penggerak saja, sedangkan untuk ayunan pendulum tidak memiliki pengendali. *Feedback* pendulum ayun pada skema ini juga berbeda pada skema sebelumnya yang dimana *feedback* pada pendulum ayun masuk menuju input daripada sistem kendali. Gambar 2-2 menunjukkan skema kontroler *anti-swing control using position control only*.



Gambar 2-2 Skema kontroler *anti-swing control using position control only* (Solihin & Wahyudi, 2006)

Terdapat jurnal lain yang membahas tentang skema kontroler *anti-swing control using position control only* yang dilakukan oleh (Yong-Seok Kim, Han-Suk Seo, and Sul 2001). Pada penelitian tersebut penulis hanya menggunakan pengendali *speed control* untuk mengendalikan posisi dari *crane*. Selain dari jurnal tersebut terdapat juga penelitian yang membahas tentang *control of gantry crane and tower crane* (Omar, 2003). Pada penelitian ini pengendalian *trolley* dan pendulum ayun menggunakan pengendali PD dan juga menggunakan *fuzzy controller*. Pada penelitian ini juga memiliki dua skema kontroler yang mana pada skema kontroler pertama sama dengan skema kontroler *normal anti-sway control*,

sedangkan skema kedua pada penelitian ini menggunakan skema seperti pada Gambar 2-3.



Gambar 2-3 Skema kontroler *delayed feedback* (Omar, 2003)

Gambar 2-3 menunjukkan skema kontroler *delayed feedback*. *Feedback* dari pendulum ayun pada skema kontroler ini masuk dan diteruskan ke input sistem yang dimana hasil perbandingan antara *feedback* dari trolley, input sistem, dan pendulum menghasilkan *error* total yang kemudian *error* total ini akan masuk kedalam kendali trolley. Skema kontroler *delayed feedback* juga digunakan pada penelitian yang dilakukan oleh (Masoud, 2002) yang dimana penelitian ini membahas tentang reduksi pendulum ayun pada crane pemindahan *cargo* pada kapal. Penelitian ini menggunakan pengendali *delayed feedback* yang digunakan sebagai pengganti dari pengendali PD.

Selain dari dua jurnal diatas terdapat juga jurnal yang membahas tentang *anti-sway control* dengan sensorless menggunakan *artificial neural network-based* (Solihin & Wahyudi, 2009). Pada penelitian ini pengendali yang digunakan adalah kendali PID dan juga penelitian menunjukkan kinerja dari sistem *anti-sway* ketika menggunakan *swing angle sensor* dan *soft sensor*. Pada penelitian ini skema kontroler yang digunakan ialah skema kontroler *normal anti-sway control*. Pengendali PID pada penelitian ini dioptimasi menggunakan *simulink respon optimization liberty block*.

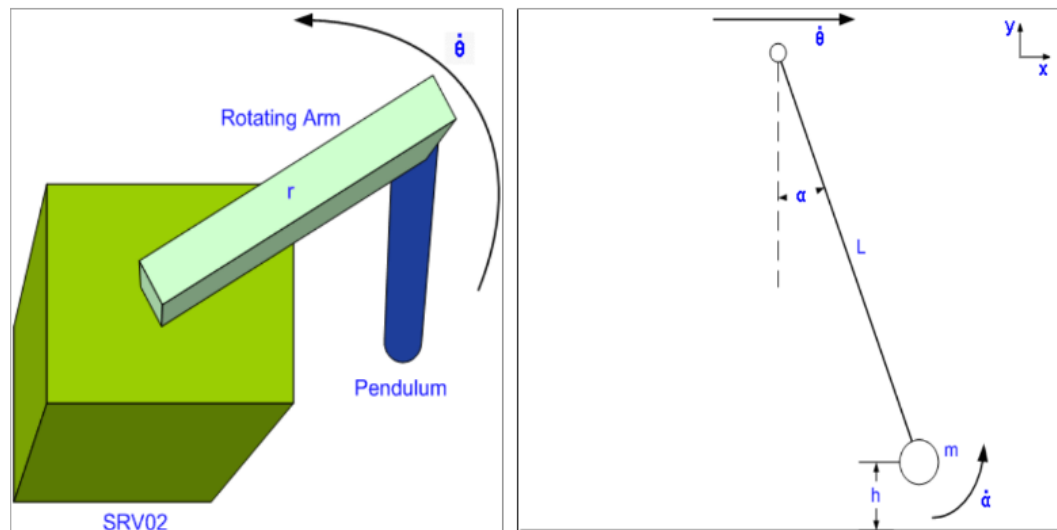
Dua referensi tersebut memiliki pembahasan yang sama yaitu membahas tentang penerapan kendali peredam ayun, meskipun dengan metode kendali yang berbeda dan juga memiliki skema kontroler yang berbeda. Perbedaan referensi-referensi tersebut dengan penelitian yang akan dilakukan adalah ayunan pendulum yang terjadi pada jurnal-jurnal diatas diakibatkan karena pergerakan *linier* sedangkan pada penelitian yang akan dilakukan ayunan pendulum terjadi akibat

pergerakan rotasi dari motor DC, dan pada penelitian yang akan dilakukan metode pengendali yang akan digunakan adalah PID+PD.

2.2 Dasar Teori

2.2.1 Eksperimen setup

Pada penelitian kali ini objek yang akan dikendalikan adalah *rotary pendulum*. *Rotary pendulum* merupakan suatu alat yang memiliki komponen lengan pendulum berputar yang dihubungkan dengan motor DC dan pendulum lainnya yang diletakkan pada ujung dari lengan pendulum (Muntari & Nurhdadi, 2013). Terdapat dua komponen yang bergerak pada *rotary pendulum* yaitu, batang lengan pendulum yang terhubung dengan motor DC dan lengan pendulum yang tersambung dengan batang lengan pendulum.

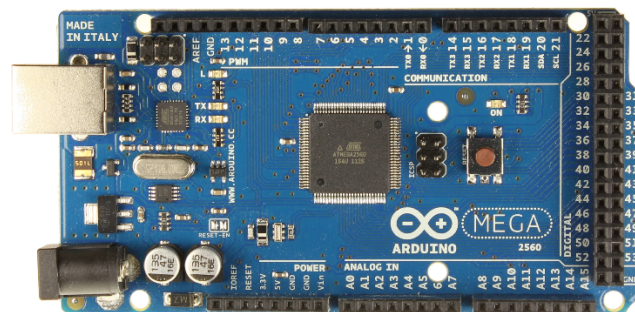


Gambar 2-4 Mekanisme *rotary pendulum*

Gambar 2-4 menunjukkan gambar dari mekanisme *rotary pendulum*. *Rotary pendulum* memiliki mekanisme ketika batang lengan pendulum digerakkan ke posisi tertentu oleh motor DC, pendulum mengikuti pergerakan tersebut dan berayun sampai berhenti ketika motor berhenti berputar. Untuk mengatur ketepatan perpindahan posisi pada batang lengan pendulum dan untuk mengurangi ayunan pada lengan pendulum maka diberikan *controller* PID yang akan dibahas pada sub bab selanjutnya.

Sistem kendali *rotary pendulum* terdiri dari beberapa komponen penting yang masing-masing komponen tersebut mempunyai kegunaan tertentu. Komponen yang pertama adalah arduino, arduino merupakan suatu papan *mikrokontroller* yang bersifat *open-source* yang banyak digunakan oleh berbagai kalangan dari pelajar hingga professional. Terdapat beberapa jenis arduino yang beredar pada saat ini, jenis-jenis ini dapat disesuaikan dengan kebutuhan para penggunanya. Selain memiliki *hardware*, arduino juga memiliki perangkat lunak (*software*) tersendiri bernama arduino IDE yang berfungsi sebagai aplikasi untuk pembuatan programnya. Bahasa program yang digunakan pada arduino adalah bahasa program C++.

Pada penelitian kali ini arduino yang digunakan adalah jenis Arduino Mega 2560. Arduino Mega 2560 memiliki pin *digital input/output* (I/O) yang mencapai 54 pin. Ketersediaan pin ini lebih banyak daripada perangkat keras Arduino jenis lainnya. Gambar 2-5 menunjukkan gambar dari Arduino Mega 2560.



Gambar 2-5 Arduino Mega 2560

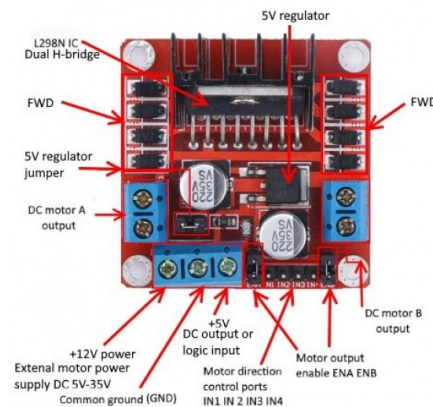
Komponen selanjutnya adalah motor DC, motor DC merupakan motor listrik yang terdiri dari kerangka magnet (*yoke*), kutub motor (*field pole*), kumparan magnet (*field winding*), kumparan jangkar (*armature*), komutator, dan *brushes* (Ahmad & Alnaib, 2019). Pada penelitian ini motor DC digunakan sebagai alat penggerak pada *rotary pendulum* dimana motor DC terhubung dengan batang lengan pendulum untuk menggerakkan batang tersebut ke sudut tertentu dengan kecepatan yang telah ditentukan.

Pada sistem kendali, motor DC memiliki fungsi *transfer* tersendiri untuk memformulasikan karakteristiknya. Berikut adalah fungsi *transfer* dari motor DC (Nise, 2015).

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t/(R_a J_m)}{s \left[s + \frac{1}{J_m} \left(D_m + \frac{K_t K_b}{R_a} \right) \right]} \quad (2.1)$$

Dimana, $E_a(s)$ merupakan *input* tegangan dan $\theta_m(s)$ adalah *output* berupa posisi sudut motor DC dari fungsi *transfer*, sedangkan J_m merupakan nilai dari inersia motor DC, D_m merupakan nilai dari redaman / *viscous damping*. Satuan dari R_a , K_t , dan K_b masing-masing adalah *terminal resistance* (Ω), (Nm/A) dan *speed constant* (rpm/V). Nilai-nilai tersebut bisa didapatkan pada *datasheet* motor DC yang tersedia.

Komponen selanjutnya adalah *torque constant driver* motor DC, pada penelitian ini digunakan *driver* motor L298N sebagai pengatur kerja dari motor DC. *Driver* L298N merupakan sebuah *driver* motor dengan *integrated circuit* (IC) L298 *dual H-Bridge*. Penggunaan IC yang terdapat pada *driver* motor L298N ini mampu menggerakkan dan mengarahkan 2 motor DC sekaligus dengan arus 2A. Pada *driver* memiliki pin-pin dengan fungsinya masing-masing. Motor DC terhubung dengan pin *output* A ataupun B, pin +5 dan +12 merupakan pin untuk *input* tegangan, dan pin GND berfungsi sebagai koneksi *ground*. Gambar 2-6 menampilkan bentuk dari *driver* motor L298N.



Gambar 2-6 *Driver* motor L298N

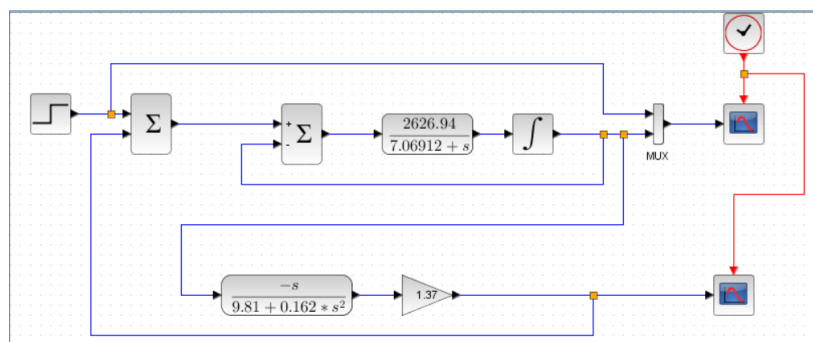
Pada penelitian ini menggunakan sensor *rotary encoder*. *Rotary encoder* merupakan sensor yang dapat mengukur posisi dan kecepatan sudut. Salah satu jenis encoder yang tersedia adalah *optical encoder*. Terdapat dua sensor rotary encoder pada alat rotary pendulum untuk penelitian kali ini yaitu *optical encoder* yang diperlukan untuk mengukur perubahan sudut dari pendulum ayun dan *high*

precision encoders yang digunakan untuk mengukur perubahan posisi dari motor DC. Gambar 2-7 menunjukkan encoder *optical rotary encoder*.



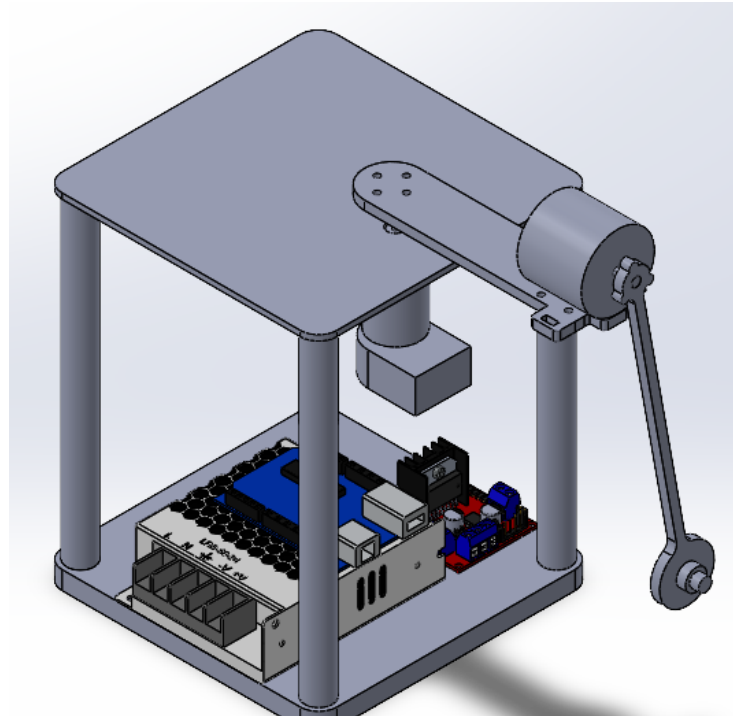
Gambar 2-7 *Optical rotary encoder*

Perangkat lunak yang digunakan pada penelitian ini adalah Scilab. Scilab merupakan suatu perangkat lunak (*software*) yang dapat digunakan untuk melakukan pemodelan matematis seperti kalkulasi integral, diferensial, aljabar, statistic dan lain sebagainya (Handayani, Pramudya, Suparwoto, & Muchlas, 2018). Scilab merupakan perangkat lunak yang bersifat *open source* dimana Scilab dapat diakses secara gratis oleh berbagai kalangan. Terdapat berbagai fitur yang tersedia didalam Scilab, salah satunya adalah fitur Xcos. Pada penelitian ini fitur tersebut digunakan untuk melakukan simulasi sistem kendali. Xcos merupakan bagian dari Scilab GUI (*Graphic User Interface*) yang digunakan sebagai pemodelan simulasi suatu sistem (Kurniawan, Simbolon, & MN, 2020). Sistem *control* dapat disimulasikan menggunakan Xcos, hal ini dikarekan Xcos memiliki model-model blok matematis yang memiliki fungsinya masing-masing. Blok matematis ini juga dapat saling dihubungkan antara satu dengan yang lain untuk membuat diagram sistem kendali. Gambar 2-8 menunjukkan contoh penggunaan aplikasi Xcos pada perangkat lunak Scilab.



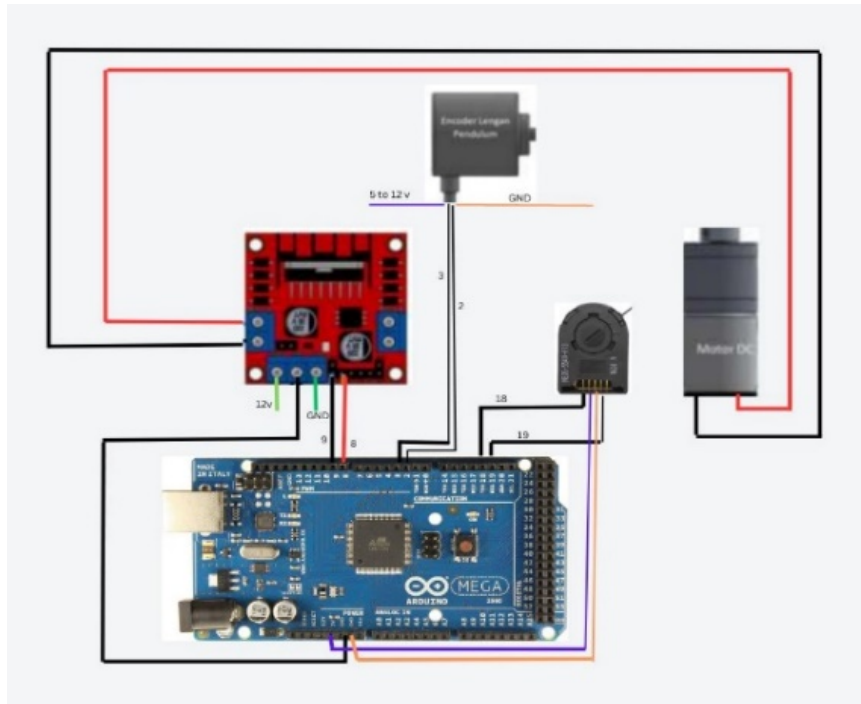
Gambar 2-8 Contoh penggunaan aplikasi Xcos

Pada penelitian ini terdapat dua perancangan yang dilakukan, yang pertama adalah perancangan perangkat keras kemudian perancangan perangkat lunak. Untuk perancangan perangkat keras *rotary pendulum* menggunakan referensi Quanser sebagai acuan bentuk objek kendali.



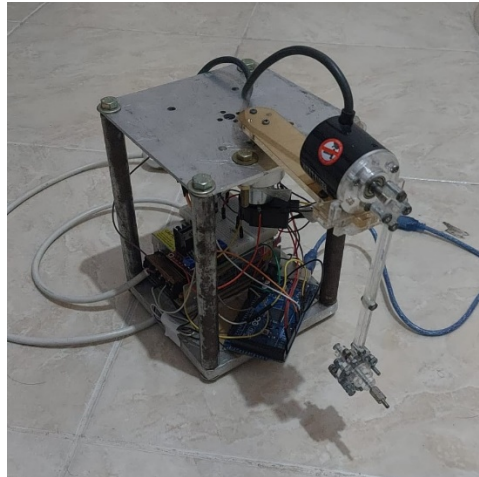
Gambar 2-9 Model perancangan perangkat keras alat *rotary pendulum*

Gambar 2-9 menunjukkan bentuk dari model alat kendali *rotary pendulum*. Berdasarkan gambar diatas, dapat dilihat perancangan model objek kendali *rotary pendulum* diatur dengan 2 alas yang disusun bertingkat. Alas pertama merupakan tempat perangkat keras elektrik seperti *power supply* dan arduino. Alas kedua tersambung dengan komponen-komponen mekanikal seperti motor DC, sensor encoder, lengan pendulum, *coupling*, dan pendulum ayun.



Gambar 2-10 Koneksi elektrik alat *rotary pendulum*

Gambar 2-10 menunjukkan skema koneksi elektrik alat *rotary pendulum*. Setelah merancang dan menyusun komponen perangkat keras, langkah selanjutnya adalah melakukan koneksi elektrik untuk alat *rotary pendulum*. Dari gambar diatas, dilihat bahwa motor DC tidak langsung tersambung ke Arduino Mega 2560, melainkan dihubungkan terlebih dahulu dengan motor *driver* L298N. Hal ini bertujuan agar motor *driver* L298N dapat mengatur arah dan kecepatan motor DC. Motor *driver* L298N dihubungkan dengan pin digital 8 dan 9 pada Arduino Mega dan juga terkoneksi dengan *power supply* dan *ground* (GND). Kemudian mengkoneksikan dua sensor encoder ke Arduino Mega yang dimana masing-masing encoder terkoneksi ke pin digital 2,3,18, dan 19. Koneksi sensor encoder sesuai dengan pengaturan pada Xcos yang akan ditunjukkan pada pembahasan selanjutnya. Gambar 2-11 menunjukkan bentuk dari alat *rotary pendulum* setelah dirancang.




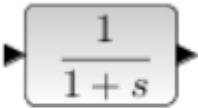
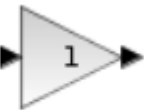





Gambar 2-11 Objek kendali *rotary pendulum*


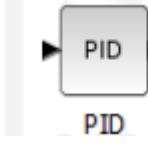
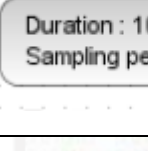


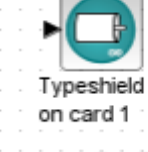
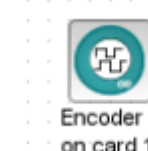
Setelah selesai melakukan perancangan pada perangkat keras, selanjutnya adalah melakukan perancangan pada perangkat lunak. Terdapat dua perangkat lunak yang digunakan untuk penelitian ini, yaitu Arduino IDE dan Scilab Xcos.

Arduino IDE yang digunakan untuk mengupload program *toolbox V3*. Program *toolbox V3* merupakan program yang digunakan untuk menghubungkan mikrokontroler arduino ke perangkat lunak Scilab. Sedangkan Scilab Xcos digunakan untuk memodelkan sistem kendali dari objek kendali *rotary pendulum*. Perangkat lunak Scilab memiliki beberapa fitur yang dapat digunakan pada berbagai macam keperluan, salah satu fitur yang terdapat pada perangkat lunak scilab ini adalah Xcos. Pada fitur Xcos ini dapat digunakan untuk membuat perancangan model sistem kendali, pada Xcos ini juga terdapat blok-blok diagram yang mempresentasikan fungsi-fungsi berbeda, yang dimana fungsi-fungsi tersebut dapat diatur sesuai dengan model sistem kendali yang telah dirancang.

Untuk menghubungkan antara objek kendali *rotary pendulum* ke perangkat lunak Scilab ada beberapa langkah yang harus dilakukan, pertama adalah dengan mengupload program *toolbox V3* ke dalam perangkat keras arduino. Selanjutnya, mengupload *loader toolbox V3* ke dalam perangkat lunak Scilab. Setelah *loader* di *upload*, selanjutnya dapat membuka fitur Xcos pada scilab untuk merancang blok diagram sistem kendali. Berikut ini adalah beberapa blok-blok diagram yang digunakan pada penelitian kali ini.

Tabel 2-1 Blok-blok yang digunakan

Blok Diagram	Nama	Fungsi
 STEP_FUNCTION	STEP_FUNCTION	Berfungsi untuk memberikan input berupa step. Bisa dalam nilai voltase dan juga radian
 CLR	CLR	Berfungsi untuk menyatakan nilai fungsi <i>transfer</i> pada Xcos
 GAINBLK	GAINBLK	Sebagai blok pengkali nilai pada Xcos
 SUMMATION	Summation	Berfungsi untuk melakukan perbandingan antara nilai <i>input</i> dan <i>output</i> pada sistem kendali
 MUX	MUX	Berfungsi untuk menyatukan beberapa input menjadi satu output
 CSCOPE	CSCOPE	Berfungsi untuk menampilkan respon sistem simulasi
 Deriv	Deriv	Berfungsi untuk melakukan proses turunan pada <i>input</i>
 INTEGRAL_m	INTEGRAL_m	Berfungsi untuk melakukan proses integral pada input

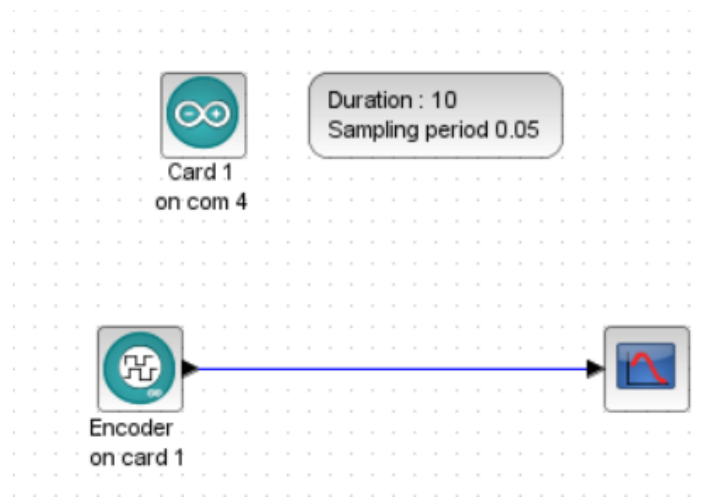
	CLOCK_c	Berfungsi untuk mengatur periode waktu sistem kendali
	PID	Berguna untuk memasukkan nilai kendali PID pada sistem
	TIME_SAMPLE	Berfungsi untuk mengatur waktu dan periode pengambilan sampel pada arduino
	ARDUINO_SETUP	Melakukan pengaturan pembacaan port Arduino yang akan dibaca oleh sistem Xcos
	ARDUINO_SCOPE	Menunjukkan grafik keluaran dari sistem yang menggunakan arduino
	DCMOTOR_SB	Berfungsi untuk mengatur dan menjalankan motor DC yang terhubung dengan arduino
	ENCODER_SB	Berfungsi mengatur dan membaca data dari sensor encoder yang terhubung dengan arduino

Dari blok-blok diagram yang digunakan, objek kendali *rotary pendulum* dapat dilakukan uji coba respon untuk mengetahui performa dari motor DC dan pembacaan kedua sensor encoder yang digunakan.

Setelah perangkat keras dan perangkat lunak sudah selesai dirancang, tahap selanjutnya untuk menyiapkan alat pengujian ini adalah dengan memeriksa sensor encoder pada motor DC dan pendulum ayun, pemeriksaan respon motor DC, dan pemeriksaan pendulum ayun.

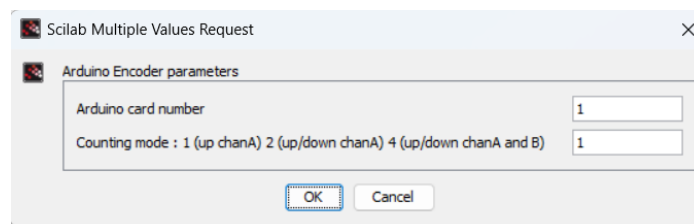
Pengecekan sensor encoder motor DC dan pendulum dilakukan dengan cara mencocokkan nilai pulsa bacaan sensor yang terdapat pada spesifikasi produk

dan pulsa bacaan sensor secara langsung. Berikut ini adalah konfigurasi Xcos untuk pengecekan sensor Encoder pada motor DC.

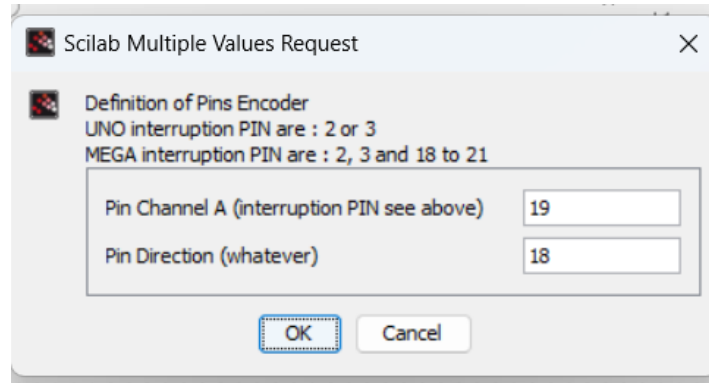


Gambar 2-12 Xcos pengecekan pulsa sensor encoder

Gambar 2-12 menunjukkan skema Xcos untuk pengecekan pulsa sensor encoder. Skema Xcos diatas hanya menghubungkan blok diagram encoder dan blok diagram *arduino scope*, skema Xcos diatas berfungsi untuk menunjukkan grafik pembacaan sensor. Pada blok diagram encoder diatur konfigurasi sensor yang terkoneksi pada arduino. Sensor encoder motor DC terhubung dengan pin 19 dan pin 18. Berikut ini adalah konfigurasi sensor encoder motor DC. Gambar 2-13 menunjukkan konfigurasi dari sensor encoder motor DC.

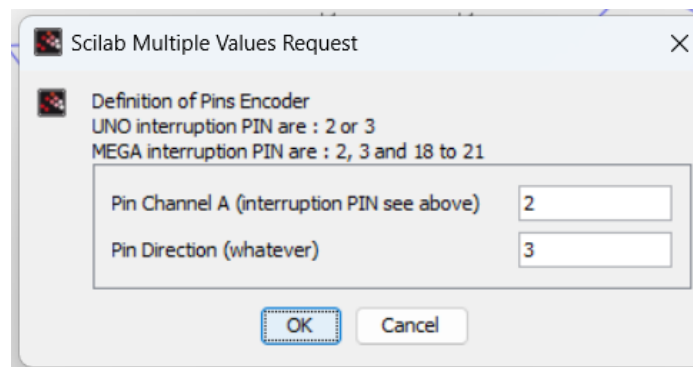


(a) Konfigurasi parameter encoder motor DC



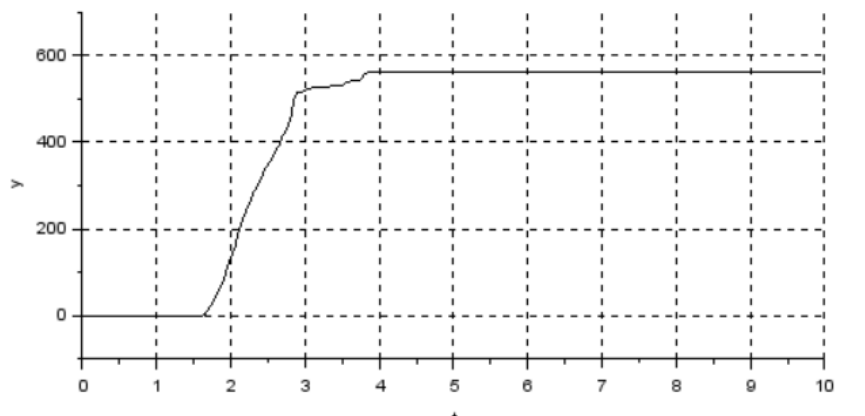
(b) Konfigurasi pin Arduino pada sensor encoder motor DC

Gambar 2-13 Konfigurasi sensor encoder motor DC



Gambar 2-14 Konfigurasi sensor encoder pendulum ayun

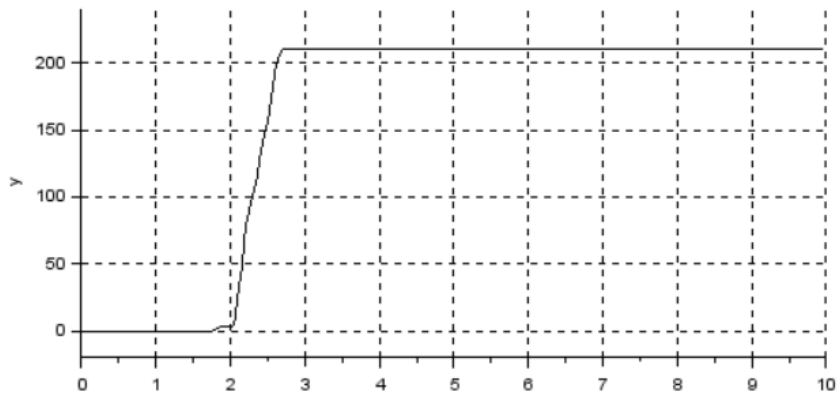
Gambar 2-14 menunjukkan konfigurasi dari sensor encoder pendulum ayun. Untuk sensor pendulum ayun, sensor terhubung ke pin 2 dan pin 3. Pada sensor encoder pendulum memiliki spesifikasi pembacaan 200 pulsa setiap 1 kali putaran.



Gambar 2-15 Grafik pembacaan pulsa encoder motor DC

Gambar 2-15 menunjukkan grafik pembacaan dari pulsa encoder motor DC. Dari grafik diatas, bisa dilihat bahwa untuk 1 kali putaran pulsa yang terbaca

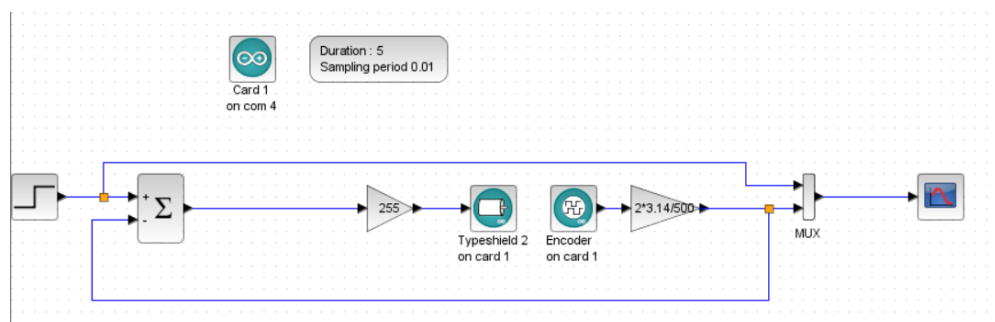
berada disekitaran angka 500 pulsa. Dari hasil pembacaan pulsa encoder motor DC, bisa dibilang bahwa pembacaan sensor encoder motor DC secara langsung sudah sesuai dengan pulsa bacaan sensor encoder spesifikasi. Dimana pulsa dari spesifikasi sensor encoder motor DC adalah 500 pulsa untuk 1 kali putaran.



Gambar 2-16 Grafik pembacaan pulsa encoder pendulum

Gambar 2-16 menunjukkan grafik hasil pembacaan pulsa sensor encoder pendulum. Dari grafik diatas, bisa dilihat bahwa untuk 1 kali putaran pulsa yang terbaca berada disekitaran angka 200 pulsa. Dari hasil pembacaan pulsa encoder pendulum, bisa dibilang bahwa pembacaan sensor encoder pendulum secara langsung sudah sesuai dengan pulsa bacaan sensor encoder spesifikasi. Dimana pulsa dari spesifikasi sensor encoder pendulum adalah 200 pulsa untuk 1 kali putaran.

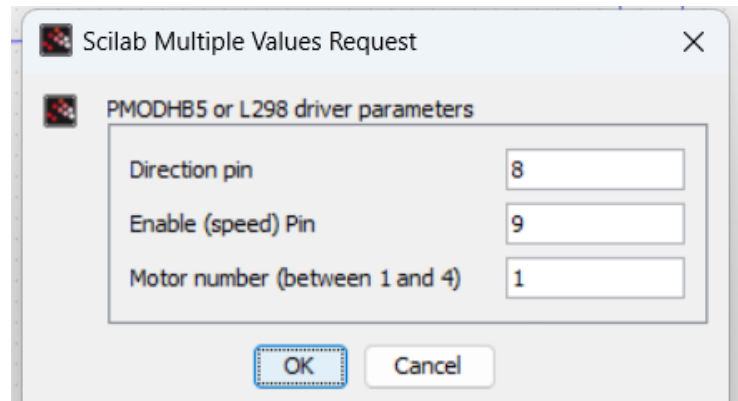
Setelah didapatkan nilai pulsa dari sensor encoder motor DC, selanjutnya dapat memeriksa respon karakteristik motor DC dengan skema Xcos seperti dibawah ini.



Gambar 2-17 Skema Xcos untuk respon motor DC

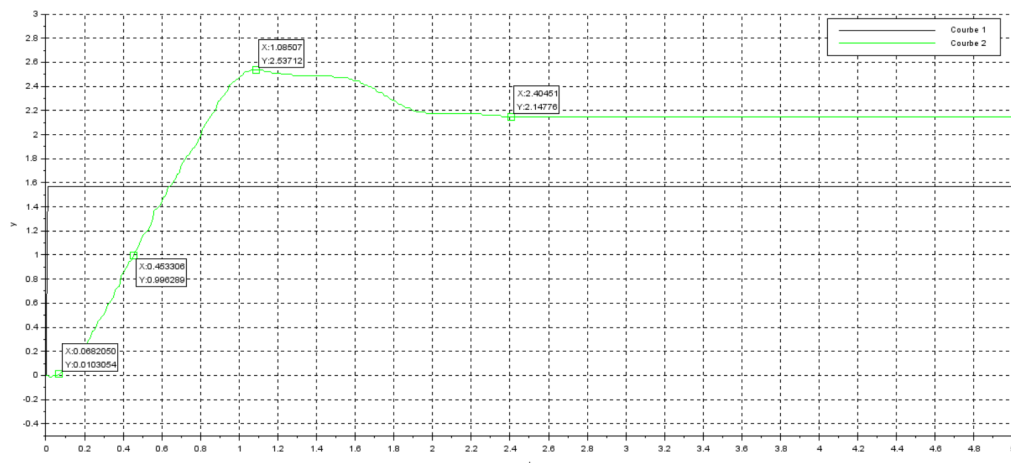
Gambar 2-17 menunjukkan skema Xcos untuk pemeriksaan respon motor DC. Pada pemeriksaan motor DC, posisi yang ingin dicapai adalah 90° atau 1,57 rad.

Karena posisi yang ingin dicapai adalah 1,57 rad maka blok step *input* pada bernilai 1,57 rad. Blok gain yang memiliki nilai 255 adalah nilai dari PWM motor DC yang dimana nilai ini diatur agar motor DC memiliki *akselerasi* yang maksimal. Untuk konfigurasi sensor encoder motor DC sama dengan konfigurasi sebelumnya, sedangkan untuk motor DC terhubung pada pin 8 dan pin 9. Gambar 2-18 menunjukkan konfigurasi pada blok Xcos motor DC.



Gambar 2-18 Konfigurasi pada blok Xcos motor DC

Hasil dari pembacaan sensor encoder motor DC selanjutnya dikalikan dengan nilai $2\pi/500$, yang dimana nilai ini berfungsi untuk merubah nilai pulsa hasil pembacaan sensor encoder motor DC kedalam bentuk satuan radian.



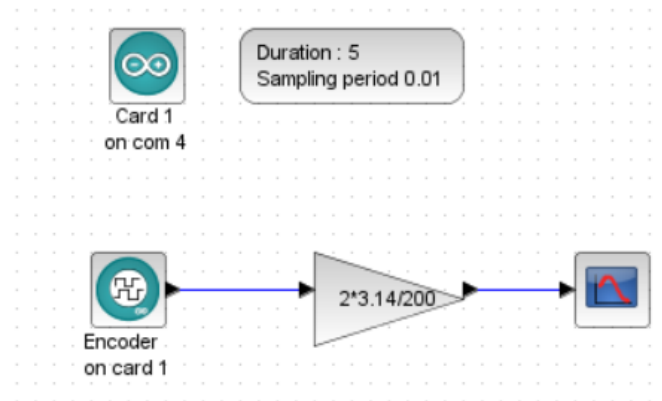
Gambar 2-19 Respon posisi motor DC

Gambar 2-19 menunjukkan respon dari motor DC ketika diberikan nilai *input*. Dari respon posisi motor DC diatas, diketahui untuk melakukan perpindahan posisi motor DC membutuhkan waktu 0,068 detik. Posisi puncak dari perpindahan posisi motor DC berada di nilai 2,53 radian atau pada posisi 144,9 ° di waktu 1,08 detik. Sementara respon motor DC yang diinginkan barada di nilai 1,57 radian atau pada

posisi 90° . Dari hal tersebut diketahui *error steady state* pada respon posisi motor DC bernilai 0,57 radian dan *overshoot* yang muncul sebesar 61,15% dari hasil *steady state* yang diinginkan. Respon dari motor DC membutuhkan waktu 2,4 detik untuk mencapai keadaan *steady state*.

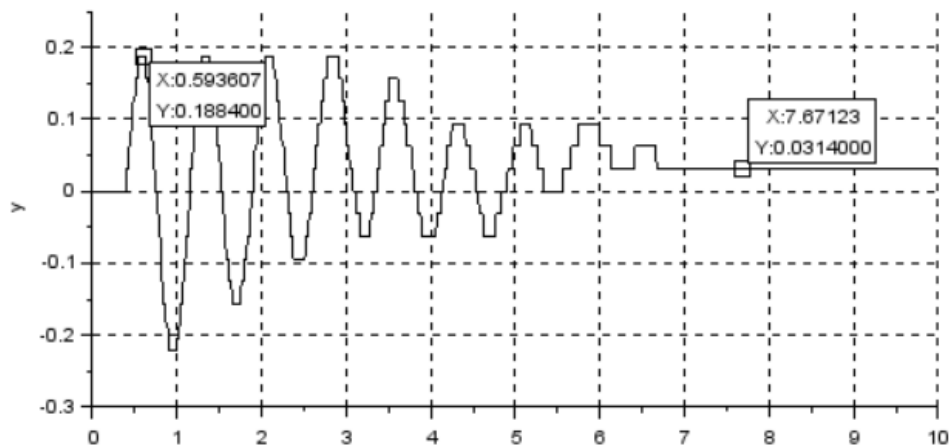
Dari informasi yang didapat, dapat diketahui bahwa respon posisi motor DC masih belum sesuai dengan respon yang diinginkan. Sehingga perlu dilakukan kendali sistem agar respon yang diinginkan dapat tercapai.

Setelah mendapatkan nilai pulsa dari sensor encoder pendulum ayun. Selanjutnya yang dilakukan adalah memeriksa respon karakteristik *osilasi* dari pendulum ayun.



Gambar 2-20 Skema Xcos untuk respon *osilasi* pendulum

Gambar 2-20 menunjukkan skema Xcos untuk pengujian respon *osilasi* dari pendulum ayun. Konfigurasi blok diagram encoder pendulum ayun masih sama dengan konfigurasi saat melakukan pemeriksaan nilai pulsa encoder pendulum ayun. Saat melakukan pemeriksaan karakteristik pendulum ayun, pendulum diayunkan secara manual sehingga pendulum ayun berayun secara bebas. Hasil pembacaan sensor encoder pendulum ayun dikalikan dengan nilai $2\pi/200$, hal ini dilakukan untuk merubah nilai pulsa hasil pembacaan sensor encoder pendulum ayun kedalam bentuk satuan radian.



Gambar 2-21 Respon pendulum ayun

Gambar 2-21 menunjukkan hasil dari respon osilasi pendulum. Dari grafik diatas dapat didapatkan informasi bahwa pendulum berayun bebas selama 6,6 detik hingga akhirnya mencapai keadaan *steady* atau berhenti berayun. Ayunan dari pendulum perlu dikendalikan agar waktu pendulum berayun bebas menjadi berkurang dan besar dari ayunan pendulum berkurang.

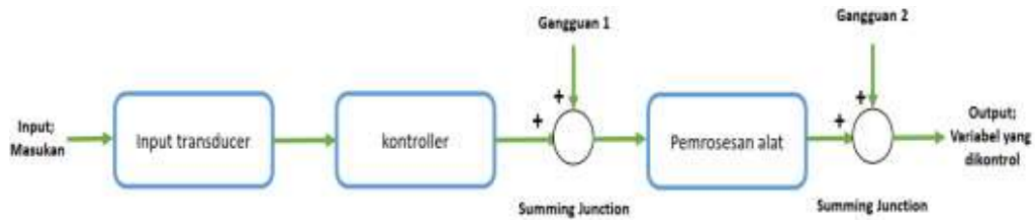
Setelah perancangan perangkat keras dan perangkat lunak selesai kemudian komponen dari objek kendali sudah dilakukan pemeriksaan dan semua kondisi telah sesuai dengan yang diinginkan maka penelitian ini bisa dilakukan.

2.2.2 Sistem kendali

Sistem control memiliki sebuah sub-sistem atau proses yang bertujuan untuk mendapatkan keluaran (*output*) yang diinginkan dengan masukan (*input*) yang tersedia atau ditentukan (Nise, 2015). Terdapat 2 bentuk utama dari sistem control, yaitu sistem *open loop* dan *closed loop*.. Gambar 2-22 menunjukkan bentuk dari sistem kendali sederhana.

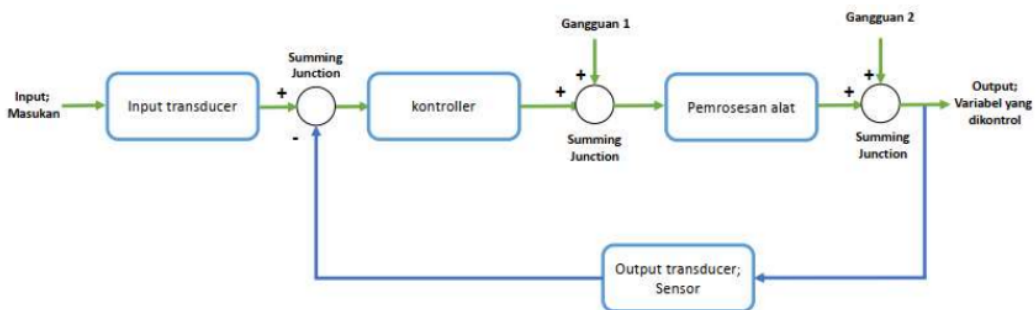


Gambar 2-22 Sistem kendali sederhana



Gambar 2-23 Sistem *open loop*

Gambar 2-23 menampilkan bentuk sistem *open loop*, dari gambar tersebut bisa dilihat alur kerja dari sistem *open loop* dimulai dari *input transducer* menerima sebuah masukan (*input*), masukan tersebut akan diteruskan ke sebuah kontroler. Masukan yang telah diterima oleh kontroler akan diproses sehingga menghasilkan sebuah keluaran (*output*) yang diinginkan. Dalam proses sistem kerja terdapat gangguan-gangguan yang bisa mempengaruhi hasil dari proses sebuah sistem menjadi kurang efektif. Kekurangan dari sistem *open loop* adalah sistem ini tidak dapat mendeteksi gangguan-gangguan yang terjadi ketika proses kerja sistem sedang berlangsung (Nise, 2015).

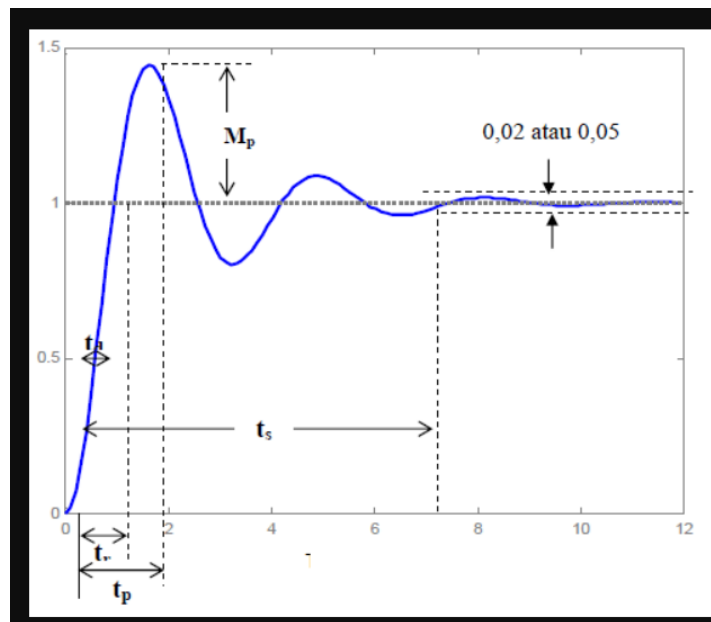


Gambar 2-24 Sistem *closed loop*

Gambar 2-24 menunjukkan skema dari sistem *closed loop*. Berbeda dengan sistem *open loop* dimana tidak bisa mendeteksi gangguan ketika sistem beroperasi, sistem *closed loop* ini mampu mendeteksi gangguan yang terjadi ketika sistem beroperasi. Dari gambar 2-24 terlihat perbedaan dengan skema *open loop*. Sistem *closed loop* memiliki sensor yang mengembalikan data hasil keluaran (*output*). Sensor ini mengembalikan data keluaran ke bagian *summing junction* untuk diteruskan ke kontroler. Lalu kontroler akan menyesuaikan *error* yang terdeteksi dari data pembacaan sensor untuk menghasilkan keluaran yang lebih efektif (Nise, 2015).

Pengambilan data hasil keluaran ke bagian *summing junction* pada sistem *closed loop* ini dikenal dengan istilah *feedback*. *Feedback* pada sistem *closed loop*

ini berfungsi untuk melacak performa sistem dan juga mendeteksi *error* sistem (Toochinda, 2016).

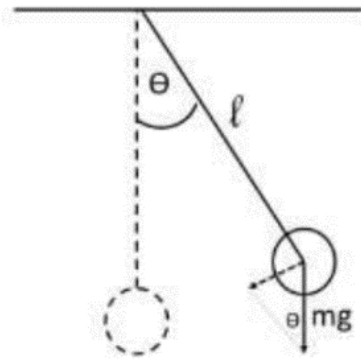


Gambar 2-25 Grafik *closed loop*

Gambar 2-25 menampilkan grafik karakteristik dari sistem *closed loop*, dapat dilihat grafik karakteristik dari kendali *closed loop*. Karakteristik dari sistem *closed loop* adalah waktu sistem mengalami kenaikan respon yang disebut dengan *rise time* (t_r), waktu sistem kendali berada dipuncak atau disebut dengan *time peak* (t_p), waktu sistem sudah berada di kondisi *steady state* atau *settling time* (t_s), dan persen *overshoot* sistem kendali (M_p). Dari beberapa karakteristik tersebut, dapat ditentukan respon sistem yang diinginkan. Untuk mencapai respon sistem yang diinginkan dapat dilakukan dengan mengatur kendali sistem menggunakan kontroler yang dibahas pada sub-bab selanjutnya.

2.2.3 Model matematis

Model matematis digunakan untuk memodelkan pergerakan pendulum sederhana.



Gambar 2-26 Ilustrasi pergerakan pendulum sederhana

Gambar 2-26 menunjukkan pergerakan dari pendulum sederhana. Pemodelan pada gerak pendulum menggunakan pemodelan *osilasi* sederhana yang menggunakan hukum Newton II yang berbentuk :

$$\sum F = m a \quad (2.2)$$

Dimana F merupakan resultan gaya (N), m merupakan massa benda (kg), dan a merupakan percepatan benda (m/s^2). Pendulum memiliki gerak rotasi yang mengakibatkan hukum Newton II berubah menjadi :

$$\tau = J a \quad (2.3)$$

Dimana τ merupakan torsi yang ada pada pendulum, J merupakan momen inersia dari pendulum ($kg.m^2$), dan $a = \ddot{\theta}$ adalah percepatan sudut dari pendulum (m/s^2). Dari formula diatas, maka didapatkan :

$$-(m g \sin \theta)l = J\ddot{\theta} = m l^2 \ddot{\theta} \quad (2.4)$$

$$-g \sin \theta = l\ddot{\theta} \quad (2.5)$$

$$\ddot{\theta} + \frac{g}{l} \sin \theta = 0 \quad (2.6)$$

Dimana θ merupakan posisi sudut, g merupakan nilai dari gravitasi (m/s^2), dan l merupakan panjang dari pendulum (m). Persamaan (2.6) memiliki kesamaan dengan persamaan *osilasi harmonic* sederhana yang berbentuk :

$$\ddot{x} + \frac{k}{m} = 0 \quad (2.7)$$

Dengan asumsi nilai θ kecil, maka nilai $\sin \theta \approx \theta$. Sehingga persamaan (2.6) menjadi:

$$-\theta \frac{g}{l} = \ddot{\theta} \quad (2.8)$$

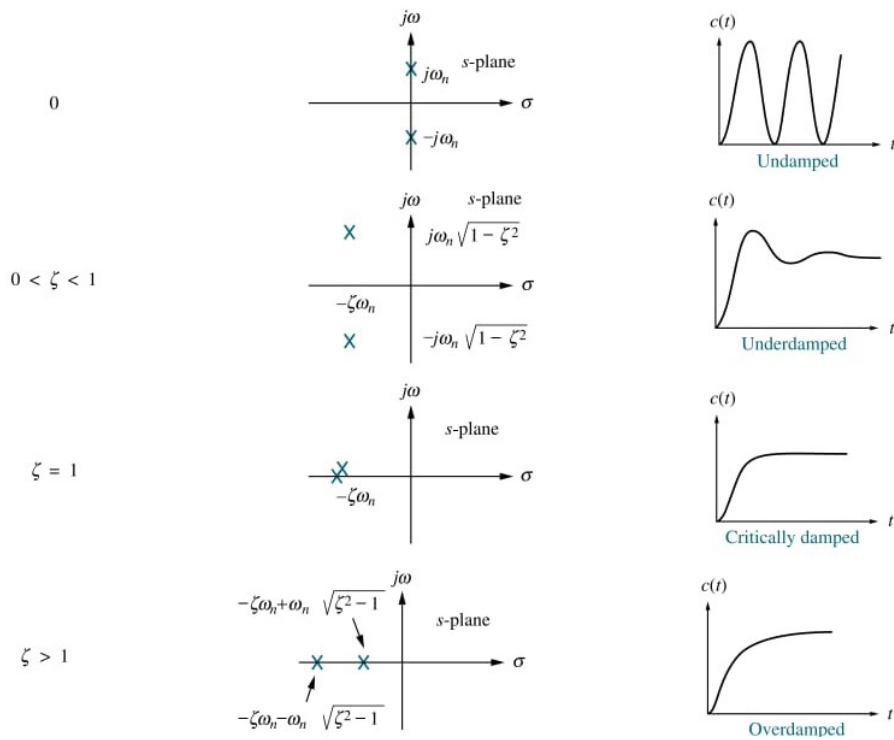
Berdasarkan persamaan (2.8) maka frekuensi natural ω_n dapat diformulakan menjadi:

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{g}{l}} \quad (2.9)$$

Nilai frekuensi natural digunakan untuk mendapatkan fungsi *transfer* bagi pendulum ayun pada penelitian kali ini. Fungsi *transfer* yang digunakan merupakan model matematis untuk sistem control orde 2:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.10)$$

Nilai dari ζ merupakan konstanta *damping*. Apabila nilai ζ sama dengan 0 maka fungsi *transfer* sistem orde 2 akan mengalami osilasi secara terus menerus (*undamped*). Sedangkan apabila nilai dari ζ berada diantara 0 dan 1 maka fungsi *transfer* orde 2 akan berosilasi tetapi memiliki redaman hingga mencapai keadaan *steady state* (*under damped*). Dan apabila nilai dari ζ sama dengan 1 maka fungsi *transfer* orde 2 akan langsung mencapai keadaan *steady state* tanpa mengalami osilasi (*critically damped*). Gambar 2-27 menunjukkan respon sistem ketika menggunakan *damping*.



Gambar 2-27 Respon sistem menggunakan *damping*

Nilai ζ dapat dicari dengan menggunakan teori *viscous damping*, dimana nilai ζ bisa didapatkan menggunakan rumus :

$$\zeta = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}} \quad (2.11)$$

Dimana :

$$n\delta = \ln \frac{x_i}{x_{i+n}} \quad (2.12)$$

Dimana δ merupakan nilai redaman dari ayunan pendulum, x_i merupakan nilai amplitudo dari nilai ayunan pendulum, dan n merupakan nilai dari fase gelombang ayunan pendulum.

2.2.4 PID

Sebuah kontroler *proportional-integral-derivatif* (PID controller) adalah mekanisme kontroler yang memiliki prinsip loop umpan balik (*closed loop*), yang secara luas digunakan dalam sistem kendali didunia industri. Sistem kendali PID menghitung besarnya sebuah “kesalahan” pada sistem sebagai perbedaan antara nilai *output* dari proses dan *set point* yang diinginkan. Kontroler PID berfungsi untun mengupayakan dan meminimalkan kesalahan dengan menyesuaikan *output* proses kontrol.

Algoritma pada PID kontroler melibatkan tiga parameter kontrol konstan yang terpisah yaitu P, I, dan D.

Perumusan kendali PID berdasarkan *error* terhadap waktu dapat dirumuskan sebagai berikut :

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dx} e(t)$$

Variabel u dan e masing-masing merupakan variabel keluaran yang diinginkan, dan masukan yang diberikan ke sistem objek kendali. Selain dari kedua variabel tersebut, pada persamaan PID juga terdapat 3 variabel kontrol, ketiga variabel tersebut adalah K_p sebagai nilai dari konstanta *error proportional*, variabel K_i sebagai nilai dari konstanta *error integral*, dan variabel K_d sebagai nilai konstanta *error derivative*. Masing-masing konstanta K_p , K_i , K_d dapat didefinisikan sebagai berikut :

$$K_p = K ; K_i = \frac{K}{T_i} ; K_d = K T_d$$

Dimana T_i merupakan nilai dari *time of integral* dan T_d merupakan nilai dari *time of derivative*.

Masing-masing dari konstanta PID tersebut memiliki pengaruh untuk mengendalikan respon sistem. Kendali PID dapat meningkatkan atau mengurangi karakteristik *overshoot*, *settling time*, *rise time*, dan *steady state error*. Tabel 2-2 menunjukkan pengaruh penggunaan masing-masing konstanta PID terhadap sistem *closed loop*.

Tabel 2-2 Pengaruh masing-masing konstanta PID (Toochinda, 2016)

Respon closed loop	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Steady State Error</i>
K_p	Berkurang	Bertambah	Berubah sedikit	Berkurang
K_i	Berkurang	Bertambah	Bertambah	Lenyap
K_d	Berubah sedikit	Berkurang	Berkurang	Berubah sedikit

2.2.4.1 Metode *Ziegler-Nichols*

Terdapat banyak metode untuk menentukan nilai konstanta PID, salah satu metode yang umum digunakan adalah metode *Ziegler-Nichols*. Dalam menggunakan metode *Ziegler-Nichols* memiliki 2 cara, yaitu menentukan nilai secara manual dan otomatis. Cara menentukan konstanta PID menggunakan metode manual dapat dilakukan dengan cara sebagai berikut (Toochinda, 2016).

1. Konstanta *integral* dan *derivative* terlebih dahulu dianggap bernilai 0 ($K_i = K_d = 0$).
2. Nilai konstanta K_p diatur hingga sistem kendali closed-loop mulai berosilasi. Catat nilai konstanta sebagai nilai K_u .
3. Ukur periode *osilasi* sebagai T_u .
4. Gunakan tabel dibawah ini sebagai perhitungan konstanta PID.

Tabel 2-3 Perhitungan konstanta PID metode *Ziegler-Nichols*

Bentuk kontoller	K_p	K_i	K_d
P	$0.5 K_u$	-	-
PI	$0.4 K_u$	$0.5 K_u / T_u$	-
PD	$0.8 K_u$	-	$0.1 K_u T_u$
PID	$0.6 K_u$	$1.2 K_u / T_u$	$0.075 K_u T_u$

Dari tabel diatas, akan didapat nilai konstanta masing-masing PID. Nilai yang didapat menggunakan metode *Ziegler-Nichols* manual bukan menjadi nilai mutlak. Perhitungan menggunakan metode *Ziegler-Nichols* dapat digunakan juga sebagai dasar nilai konstanta, kemudian nilai konstanta PID dapat disesuaikan untuk mendapatkan hasil respon sistem yang sesuai dengan karakteristik yang diinginkan.

2.2.4.2 Metode *Tyreus-Luyben*

Metode *Tyreus-Luyben* hampir sama dengan metode *Ziegler-Nichols* tetapi pada perhitungan untuk menentukan nilai konstanta PID kedua metode ini memiliki parameter yang berbeda. Pada metode *Tyreus-Luyben* ini hanya bisa digunakan untuk menentukan nilai konstanta dari PI dan PID *controller* saja. Sama seperti metode *Ziegler-Nichols*, metode *Tyreus-Luyben* ini juga menggunakan *ultimate gain* (K_{cu}) dan periode *osilasi* (P_u). Tabel dibawah ini dapat digunakan sebagai perhitungan untuk mencari nilai PID (Shahrokhi & Zomorodi, 2012).

Tabel 2-4 Perhitungan konstanta PID metode *Tyreus-Luyben*

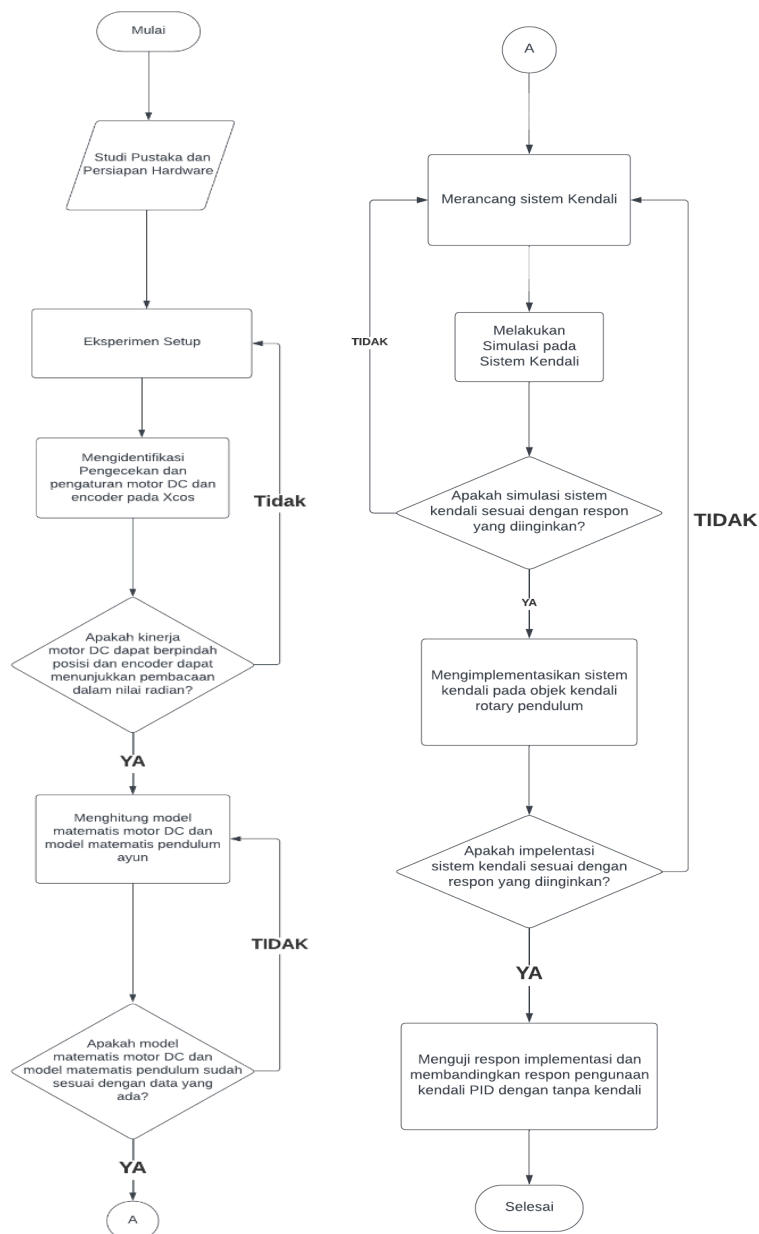
Kontroller	K_c	τ_I	τ_D
PI	$K_{cu}/3.2$	$2.2P_u$	
PID	$K_{cu}/3.2$	$2.2 P_u$	$P_u/6.3$

BAB 3

METODE PENELITIAN

3.1 Alur Penelitian

Pada penelitian ini terdapat alur penelitian yang dimana alur penelitian ini menunjukkan proses tahapan yang dilakukan selama penelitian berlangsung. Gambar 3-1 menunjukkan alur dari penelitian yang akan dilaksanakan.



Gambar 3-1 Alur penelitian

3.2 Metode Pengujian

3.2.1 Simulasi model matematis

Terdapat 2 model matematis yang perlu dihitung, yang pertama adalah model matematis dari motor DC dan model matematis pendulum ayun. Perhitungan model matematis dari motor DC dapat dilakukan dengan cara mengumpulkan data motor DC yang digunakan pada *datasheet*. Setelah data motor DC yang diperlukan telah didapatkan perhitungan model matematis untuk motor DC bisa menggunakan rumus (2.1). Selanjutnya untuk model matematis pendulum ayun dapat dilakukan dengan natural frekuensi pendulum dan data yang ditunjukkan pada pemeriksaan respon *osilasi* pendulum ayun.

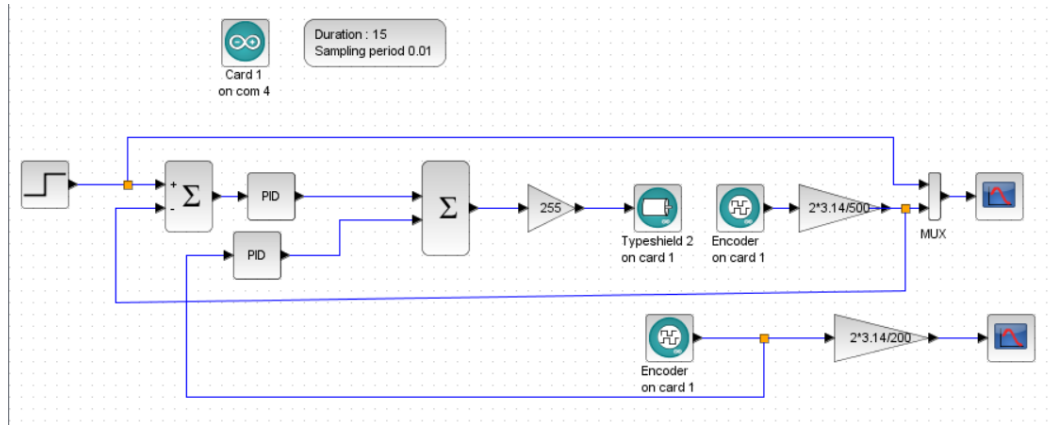
Untuk simulasi model matematis juga dilakukan menggunakan Xcos. Skema dari model matematis yang digunakan seperti yang ditunjukkan pada Gambar 2-1, Gambar 2-2, dan Gambar 2-3.

Input step yang diberikan adalah *input* posisi dalam bentuk radian. Posisi yang diinginkan adalah motor DC berpindah sejauh 90° yang dimana dalam bentuk satuan radian adalah 1,57 rad.

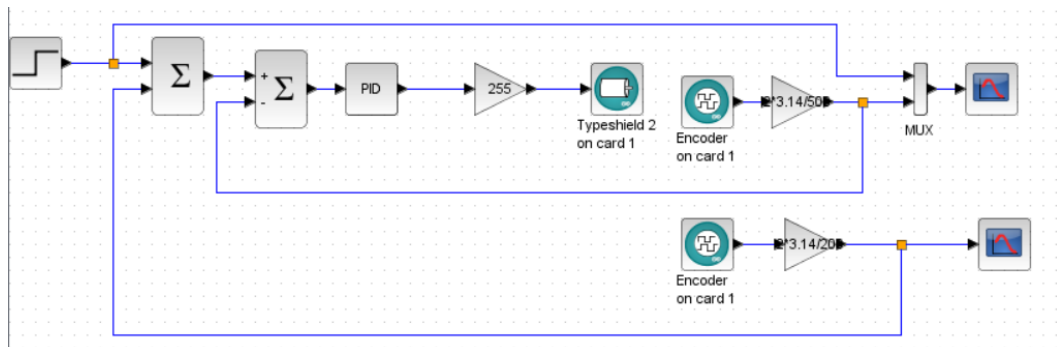
3.2.2 Pengujian kendali PID objek kendali *rotary pendulum*

Dalam pengujian objek kendali *rotary pendulum*, terdapat dua tahapan dalam melakukan pengujian. Yang pertama adalah uji coba alat sebelum diberikan kendali PID, yang kedua menguji alat dengan kendali PID+PD yang telah didapatkan dengan metode yang ada.

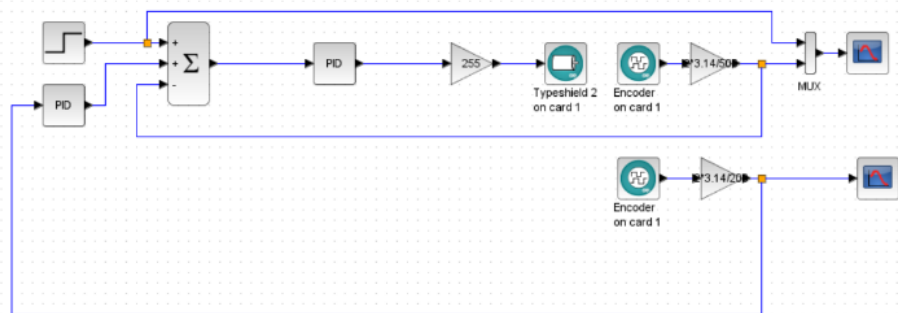
Uji coba dilakukan menggunakan skema Xcos seperti yang ditunjukkan pada Gambar 3-2, Gambar 3-3, dan Gambar 3-4.



Gambar 3-2 Skema kontroler *normal anti-sway control*



Gambar 3-3 Skema kontroler *position only control*



Gambar 3-4 Skema kontroler *delayed feedback*

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Perhitungan Model Matematis

Berdasarkan perumusan model matematis pada bab 2, diketahui terdapat dua fungsi *transfer* pada objek kendali *rotary pendulum*. Fungsi *transfer* yang pertama menunjukkan respon posisi dari motor DC dan fungsi *transfer* kedua menunjukkan respon ayunan dari pendulum ayun. Data yang telah teridentifikasi didapatkan dari *datasheet* motor DC (terlampir) serta pengukuran langsung pada objek kendali *rotary pendulum*. Data yang telah didapatkan selanjutnya digunakan untuk menghitung fungsi *transfer* objek kendali *rotary pendulum*.

Tabel 4-1 Identifikasi data alat *rotary pendulum*

No	Identifikasi Data	Nilai	Sumber
1	<i>Terminal resistance</i> (R_a)	7,41 Ω	<i>Datasheet</i> motor DC
2	<i>Torque constant</i> (K_t)	25,5 mNm/A 225,5 \times 10 ⁻³ Nm/A	<i>Datasheet</i> motor DC
3	<i>Speed constant</i> (K_b)	374 rpm/V	<i>Datasheet</i> motor DC
4	D_m	$\frac{1}{108 \times 10^{-3}}$	<i>Datasheet</i> motor DC
5	Rotor inertia (J_m)	13,1 gcm ² 13,1 \times 10 ⁻³ kgm ²	<i>Datasheet</i> motor DC
6	$I_{\text{pendulum ayun}}$	16,2 cm 0,162 m	Pengukuran langsung
7	$M_{\text{pendulum ayun}}$	46,1 g	Pengukuran langsung

4.1.1 Hasil fungsi *transfer* motor DC

Berdasarkan data diatas dapat dihitung fungsi *transfer* untuk motor DC dengan menggunakan persamaan (2.1). Dengan asumsi bahwa beban yang bertumpuan pada motor DC diabaikan, sehingga nilai momen inersia menggunakan dari datasheet motor.

$$\frac{\theta_m(s)}{E_a(s)} = \frac{25,5 \times 10^{-3} / (7,41 \times 13,110^{-3})}{s \left[s + \frac{1}{13,1 \times 10^{-3}} \left(9,26 + \frac{25,5 \times 10^{-3} \times 374 \times 10^{-3}}{374 \times 10^{-3}} \right) \right]} \quad (4.1)$$

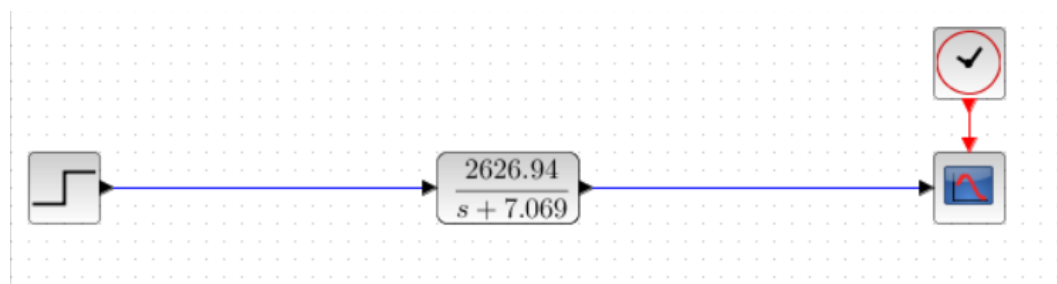
Dari persamaan (4.1) bisa didapatkan hasil dari fungsi *transfer* motor DC.

$$\frac{\theta_m(s)}{E_a(s)} = \frac{2626,94}{s[s + 7,06912]} \quad (4.2)$$

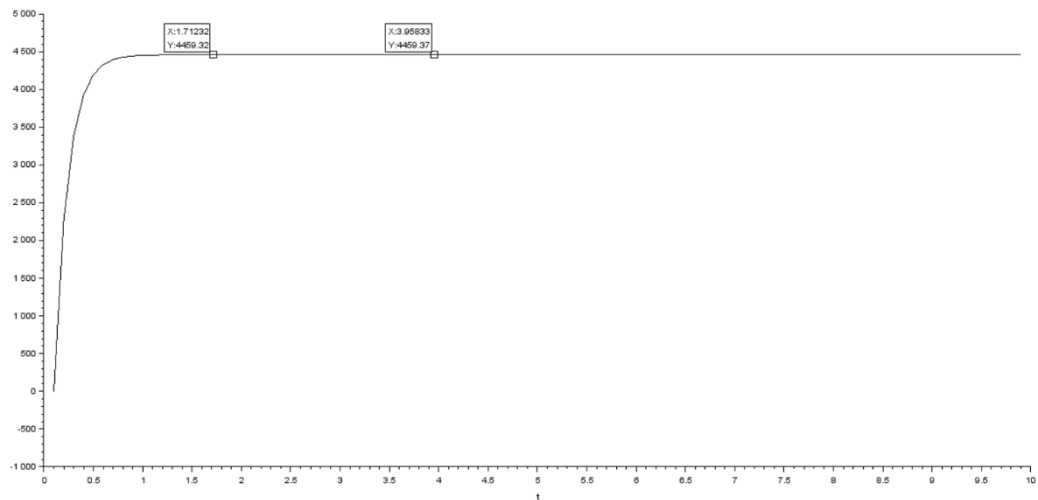
Persamaan (4.2) dapat diubah untuk mendapatkan nilai dari kecepatan sudut dengan memindahkan s ke bagian kiri sehingga persamaan (4.2) menjadi.

$$\frac{\omega_m(s)}{E_a(s)} = \frac{2626,94}{s + 7,07} \quad (4.3)$$

Sebelum melakukan simulasi model matematis, fungsi transfer motor DC yang sudah didapat harus dicek terlebih dahulu. Apakah fungsi transfer yang didapat sudah sesuai dengan data motor DC yang digunakan. Gambar 4-1 menunjukkan konfigurasi untuk pengecekan data kecepatan pada rumus fungsi transfer motor DC.



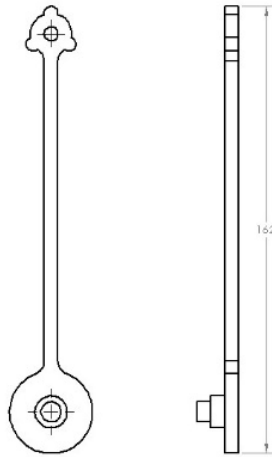
Gambar 4-1 Konfigurasi pengecekan data kecepatan pada rumus fungsi transfer motor DC



Gambar 4-2 Hasil pemeriksaan kecepatan pada rumus fungsi transfer motor DC
 Gambar 4-2 menunjukkan hasil pemeriksaan data kecepatan pada rumus fungsi transfer motor DC. Dari Gambar 4-2 bisa dilihat bahwa rumus fungsi transfer yang didapat memiliki kecepatan sebesar 4459,32 rpm, nilai tersebut sudah mendekati dengan nilai kecepatan motor DC yang terdapat pada datasheet motor DC (terlampir) yaitu sebesar 4440 rpm.

4.1.2 Hasil fungsi *transfer* pendulum ayun

Berdasarkan model matematis yang ditunjukkan pada Gambar 2-26, dapat diketahui bahwa model ilustrasi pendulum merupakan pendulum yang menggunakan tali dengan beban pada ujung talinya. Hal ini berbeda dengan model pendulum dari alat *rotary pendulum* seperti yang ditunjukkan pada gambar dibawah ini.



Gambar 4-3 Rancangan pendulum ayun

Gambar 4-3 menunjukkan rancangan pada pendulum ayun. Berdasarkan gambar tersebut, diasumsikan bahwa model pendulum ayun memiliki sistem yang sama dengan model pendulum menggunakan tali. Asumsi ini disebabkan karena frekuensi natural dari pendulum tidak bergantung dengan massa beban pendulum seperti yang ditunjukkan pada persamaan (2.9). Dengan menggunakan persamaan ini dapat dihitung frekuensi natural untuk pendulum ayun.

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{g}{l}} \quad (2.9)$$

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{9,81 \text{ m/s}^2}{0,162 \text{ m}}} \quad (4.4)$$

$$\omega_n = 1,24 \text{ s}^{-1} \quad (4.5)$$

Nilai frekuensi natural dari alat *rotary pendulum* adalah $1,24 \text{ s}^{-1}$. Setelah dapat nilai dari frekuensi dari *alat rotary pendulum*, selanjutnya dapat mencari nilai fungsi *transfer* dari pendulum ayun. Pertama yang harus dilakukan adalah mencari nilai konstanta *damping* terlebih dahulu, dimana konstanta *damping* dapat dicari menggunakan persamaan (2.11) dan persamaan (2.12).

$$\zeta = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}} \quad (2.11)$$

Dimana :

$$n\delta = \ln \frac{x_i}{x_{i+n}} \quad (2.12)$$

Dari data percobaan ayunan pendulum yang di tunjukkan pada gambar 4-4. Diketahui bahwa amplitude pada fase ke 8 mengalami penurunan, data ini dapat digunakan untuk menghitung nilai dari δ . Berikut ini ditunjukkan perhitungan nilai δ .

$$8\delta = \ln \frac{0,188}{0,031} \quad (4.6)$$

$$8\delta = 1,8 \quad (4.7)$$

$$\delta = \frac{1,8}{8} \quad (4.8)$$

$$\delta = 0,22 \quad (4.9)$$

Sehingga nilai konstanta damping dapat dihitung sebagai berikut:

$$\zeta = \frac{0,22}{\sqrt{(2\pi)^2 + 0,22^2}} \quad (4.10)$$

$$\zeta = 0,008 \quad (4.11)$$

Berdasarkan persamaan (2.10) nilai fungsi *transfer* pendulum ayun didefinisikan sebagai berikut:

$$H(s) = \frac{1,24^2}{s^2 + 2(0,008)(1,24) + 1,24^2} \quad (4.12)$$

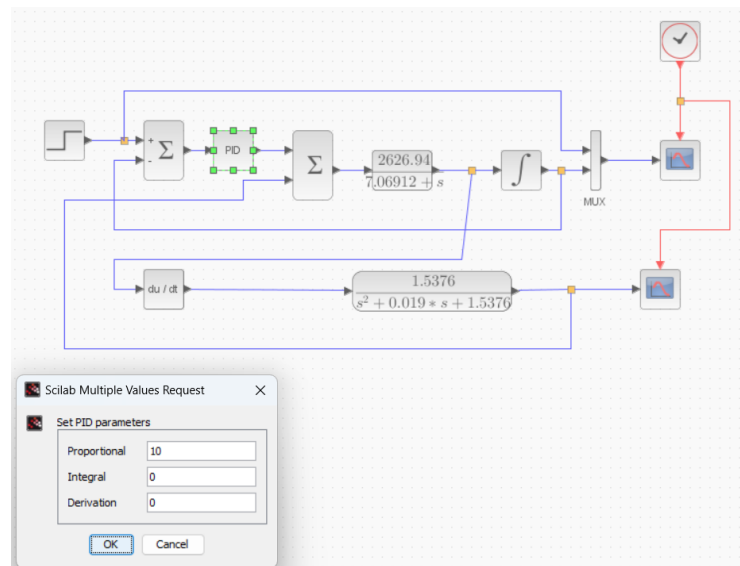
$$H(s) = \frac{1,5376}{s^2 + 0,019 + 1,5376} \quad (4.13)$$

4.2 Pengujian Fungsi *Transfer* dan Alat Rotary Pendulum

4.2.1 Pengujian menggunakan *normal anti-sway control* dengan PID+PD *Ziegler Nichols*

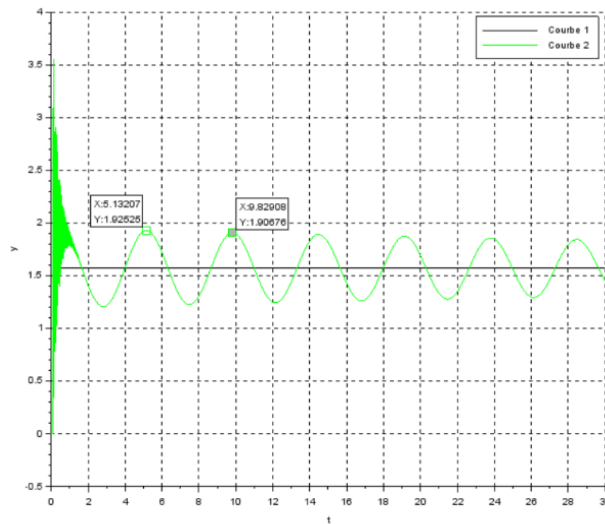
Setelah didapatkan fungsi *transfer* dari *rotary pendulum*, selanjutnya dilakukan pengujian menggunakan kendali PID+PD pada fungsi *transfer dan alat*

rotary pendulum. Pengujian pertama dilakukan pada fungsi *transfer* untuk mencari nilai dari kendali PID+PD. Pada pengujian ini kendali PID untuk kendali motor DC dicari terlebih dahulu, metode yang digunakan adalah metode *Ziegler-Nichols*. Caranya adalah dengan tidak memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberinilai hingga grafik respon menunjukkan pergerakan *osilasi*.

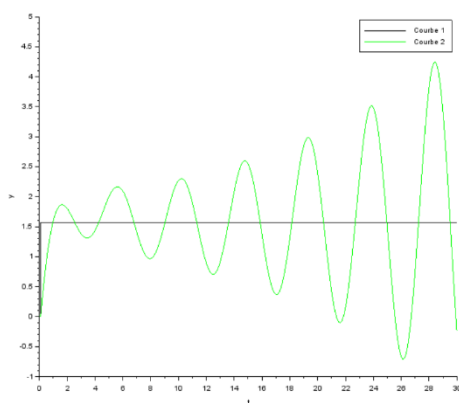


Gambar 4-4 Konfigurasi mencari nilai PID menggunakan *Ziegler-Nichols*

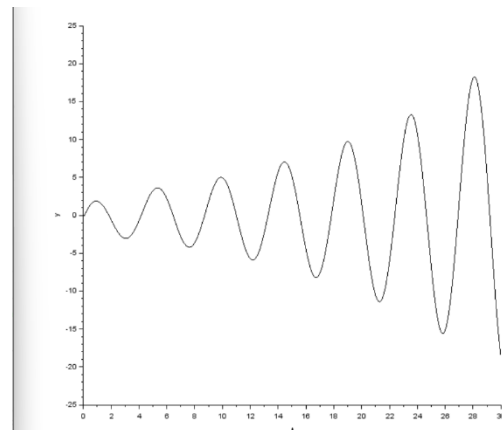
Gambar 4-4 menunjukkan konfigurasi untuk mencari nilai PID menggunakan metode *Ziegler-Nichols*. Dari hasil iterasi didapatkan nilai K_u untuk mencari PID ialah 10.



Gambar 4-5 Grafik respon fungsi *transfer* motor DC ketika diberikan nilai $K_p = 10$
 Gambar 4-5 menunjukkan hasil respon fungsi *transfer* kendali *rotary pendulum* ketika diberikan nilai $K_p = 10$. Dari hasil grafik respon diatas didapatkan nilai dari T_u , nilai T_u yang didapat adalah 4,7. Tabel 2-3 dapat digunakan untuk mencari nilai dari PID menggunakan metode *Ziegler-Nichols* dengan K_u adalah 10 dan T_u adalah 4,7. Hasil yang didapat dari Tabel 2-3 ialah $K_p = 6$, $K_i = 2,57$, dan $K_d = 3,48$.



(a) Grafik respon motor DC

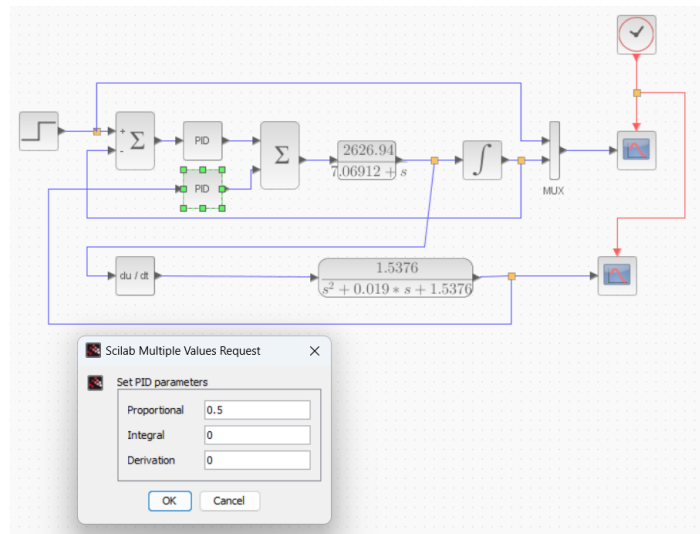


(b) Grafik respon pendulum ayun

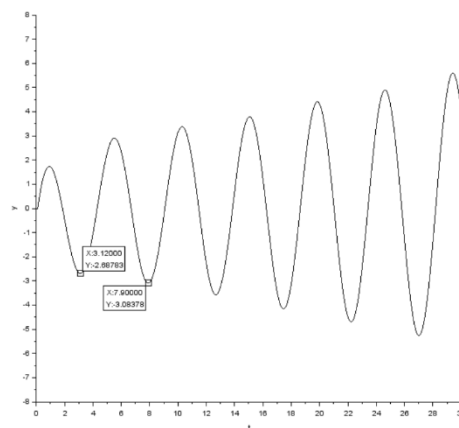
Gambar 4-6 Grafik respon model matematis setelah diberi nilai PID menggunakan metode *Ziegler-Nichols*

Gambar 4-6 menunjukkan respon dari fungsi *transfer* motor DC setelah diberi nilai PID. Dari gambar diatas, dilihat bahwa setelah diberikan nilai pengendali PID pada motor DC, sistem menjadi tidak stabil, hal ini dikarenakan belum adanya pengendali dari ayunan pendulum. Untuk mengendalikan pendulum ayun perlu ditambahkan pengendali PD. Caranya adalah dengan tidak

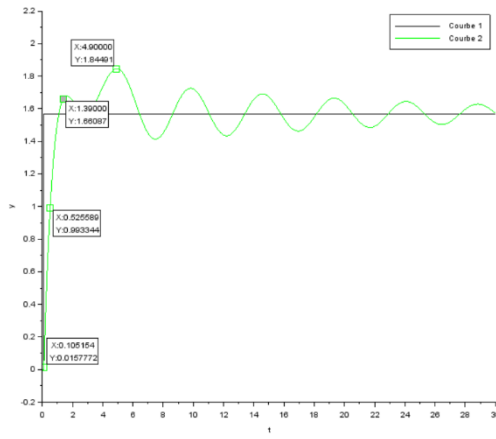
memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberi nilai hingga grafik respon menunjukkan pergerakan *osilasi*. Gambar 4-7 menunjukkan nilai dari K_u untuk mencari nilai dari PD. Dari hasil iterasi didapatkan nilai K_u untuk mencari PD ialah 0,5.



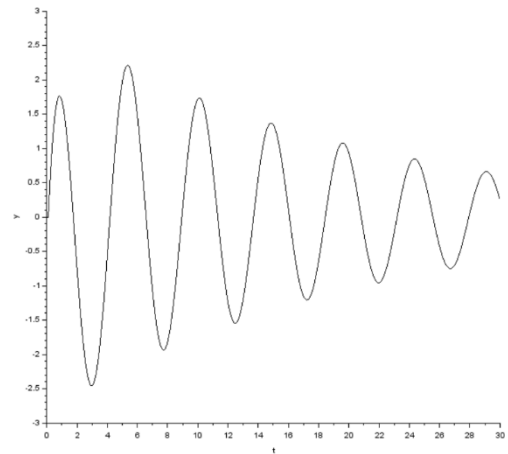
Gambar 4-7 Konfigurasi mencari nilai PD metode *Ziegler-Nichols*



Gambar 4-8 Grafik respon fungsi *transfer* pendulum ketika diberi nilai $K_p=0,5$ Gambar 4-8 menunjukkan hasil respon pendulum ayun ketika diberikan nilai $K_p=0,5$. Berdasarkan grafik respon diatas, nilai T_u untuk mencari nilai PD menggunakan metode *ziegler-nichols* adalah 4,8. Setelah mendapatkan nilai dari K_u dan T_u , selanjutnya dapat mencari nilai dari PD menggunakan Tabel 2-3. Hasil yang didapatkan untuk nilai PD adalah $K_p = 0,4$ dan $K_d = 0,3$.



(a) Grafik respon motor DC



(b) Grafik respon pendulum ayun

Gambar 4-9 Grafik respon model matematis setelah diberi kendali PID+PD

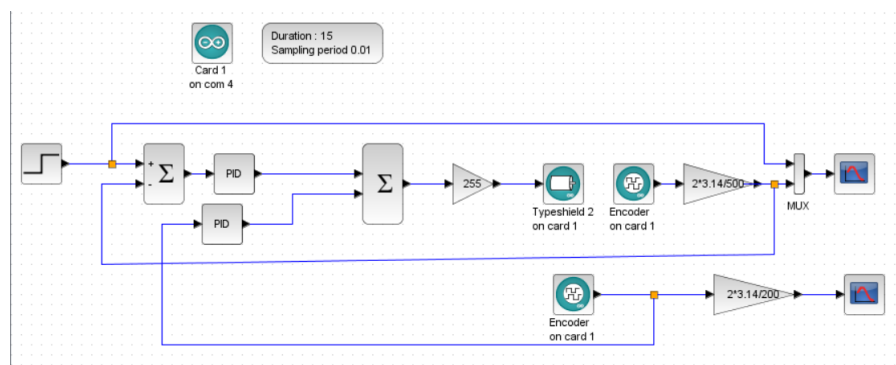
Ziegler-Nichols

Gambar 4-9 menunjukkan grafik respon fungsi *transfer* kendali *rotary pendulum* setelah diberi kendali PID+PD *Ziegler-Nichols*. Dari gambar diatas didapatkan informasi berupa:

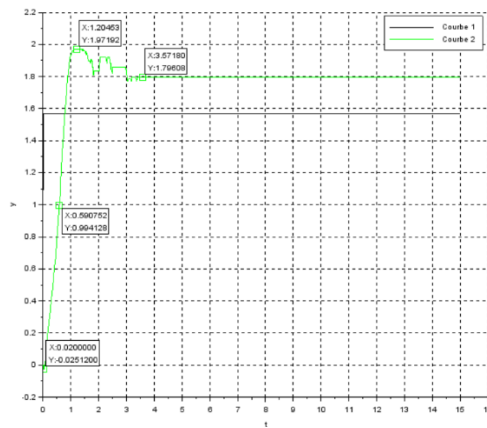
- *Overshoot* = 17,20%
- *Time Constant* = 0,52 detik

Grafik respon fungsi *transfer* dari sistem kendali menunjukkan bahwa motor DC masih belum menunjukkan kondisi *steady state* tetapi *osilasi* pada motor DC mengalami penurunan setiap fasenya. Pada pendulum ayun, grafik respon belum menunjukkan keadaan yang diinginkan tetapi sudah menunjukkan gerakan osilasi yang teredam dimana nilai dari *osilasi* pendulum terus mengalami penurunan.

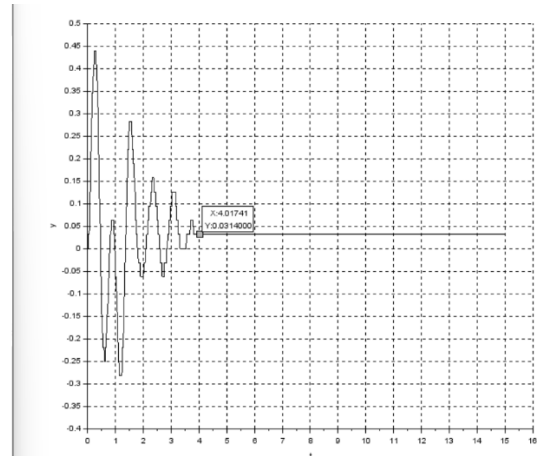
Setelah mendapatkan nilai dari kendali PID+PD, selanjutnya nilai kendali tersebut bisa diujikan pada alat *rotary pendulum*. Gambar 4-10 menunjukkan skema kendali *normal anti-sway*.



Gambar 4-10 Skema kendali *normal anti-sway* pada alat *rotary pendulum*



(a) Grafik respon motor DC



(b) Grafik respon pendulum ayun

Gambar 4-11 Grafik respon kendali alat *rotary pendulum* dengan kendali

PID+PD *Ziegler-Nichols* menggunakan skema *normal anti-sway*

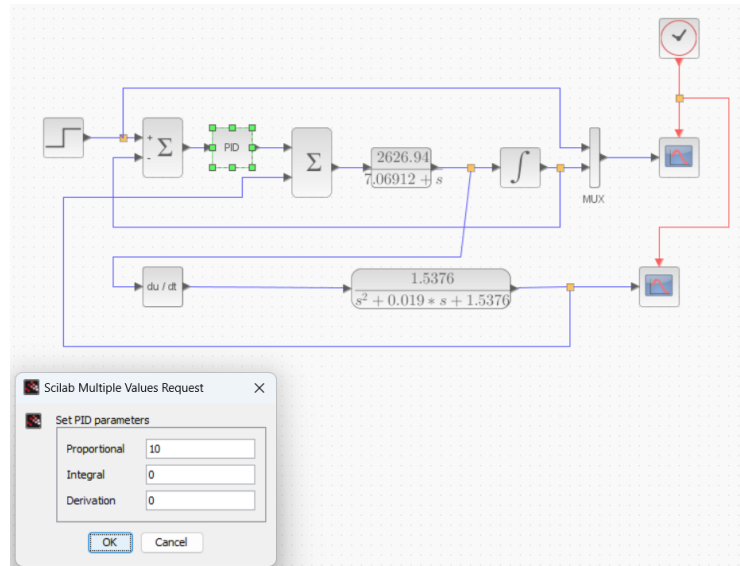
Gambar 4-11 grafik respon kendali alat *rotary pendulum*. Dari Gambar 4-11 didapatkan respon objek kendali *rotary pendulum* memiliki parameter spesifikasi respon sistem sebagai berikut:

- *Error steady state* = 0,22 rad
- *Input* = 1,57 rad
- *Settling time* = 3,57 s
- *Overshoot* = 25,48%
- *Time Constant* = 0,59 s

Pada pendulum ayun objek kendali *rotary pendulum* berayun selama 4 detik dan setelah itu pendulum ayun berhenti berayun.

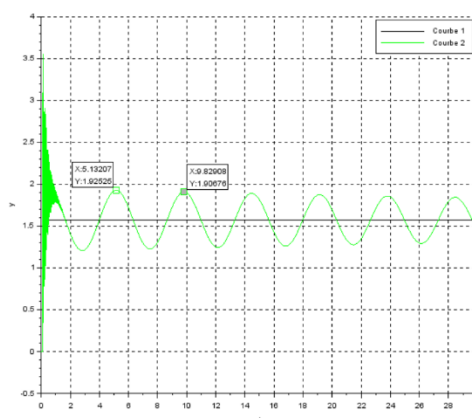
4.2.2 Pengujian menggunakan *normal anti-sway control* dengan PID Tyreus-Luyben +PD *Ziegler Nichols*

Pada pengujian menggunakan kendali PID *Tyreus -Luyben* skema yang digunakan masih sama seperti Gambar 4-4. Pada pengujian ini kendali PID dicari terlebih dahulu. Caranya adalah dengan tidak memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberi nilai hingga grafik respon menunjukkan pergerakan *osilasi*.

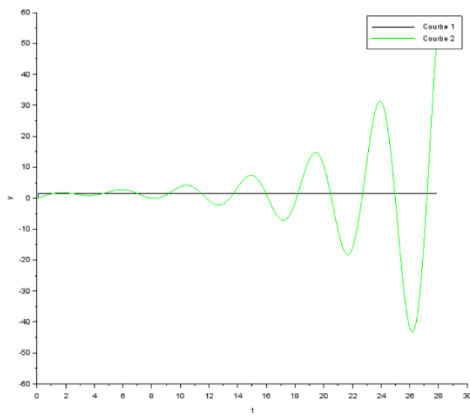


Gambar 4-12 Konfigurasi mencari nilai PID menggunakan metode *Tyreus - Luyben*

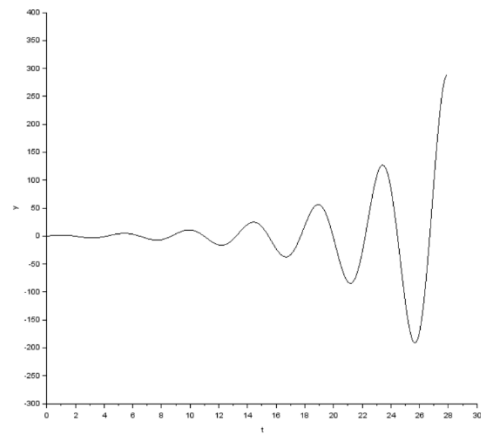
Gambar 4-12 menunjukkan konfigurasi untuk mencari nilai PID menggunakan metode *Tyreus -Luyben*. Dari hasil iterasi didapatkan nilai K_{cu} untuk mencari PID ialah 10.



Gambar 4-13 Grafik respon fungsi *transfer* motor DC ketika diberiniali $K_p = 10$
 Gambar 4-13 menunjukkan hasil respon fungsi *transfer* motor DC. Dari hasil grafik respon diatas bisa dapatkan nilai dari P_u , nilai P_u yang didapat adalah 4,7. Tabel 2-4 dapat digunakan untuk mencari nilai dari PID menggunakan metode *Tyreus -Luyben* dengan K_{cu} adalah 10 dan P_u adalah 4,7. Hasil yang didapat dari Tabel 2-4 ialah $K_p = 3,125$, $K_i = 0,3$, dan $K_d = 2,31$.



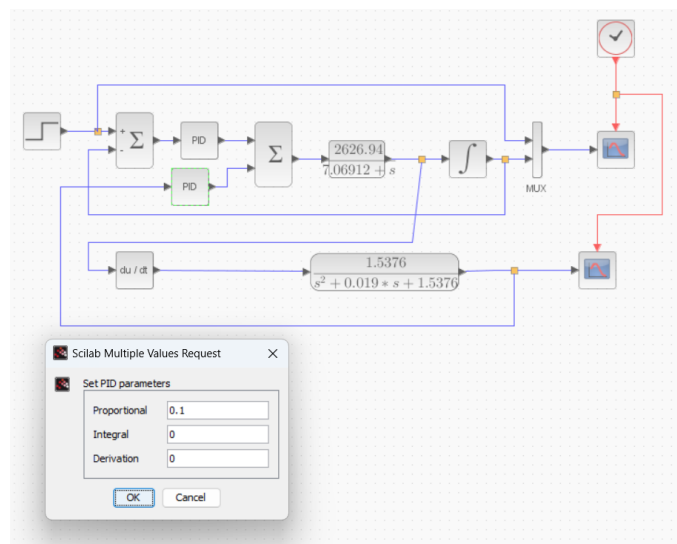
(a) Grafik respon motor DC



(b) Grafik respon pendulum ayun

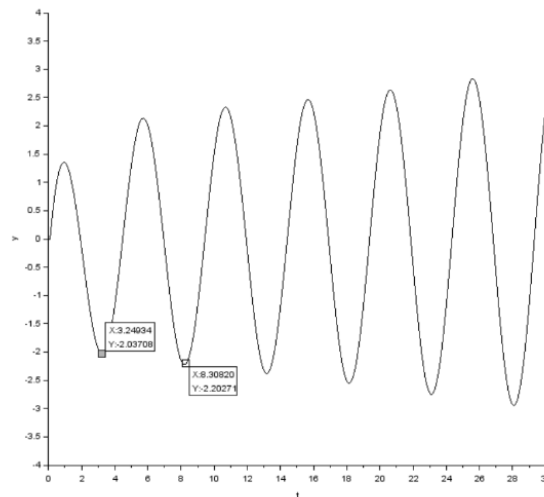
Gambar 4-14 Respon model matematis ketika diberi kendali PID *Tyreus -Luyben*

Gambar 4-14 menunjukkan grafik respon fungsi *transfer* motor DC setelah diberi nilai dari PID. Dari gambar diatas, dilihat bahwa setelah diberikan nilai pengendali PID pada motor DC, sistem menjadi tidak stabil hal ini dikarenakan belum adanya pengendali dari ayunan pendulum. Untuk mengendalikan pendulum ayun perlu ditambahkan pengendali PD. Kendali PD yang digunakan adalah kendali PD menggunakan metode *Ziegler-Nichols*. Caranya adalah dengan tidak memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberinilai hingga grafik menunjukkan pergerakan *osilasi*.

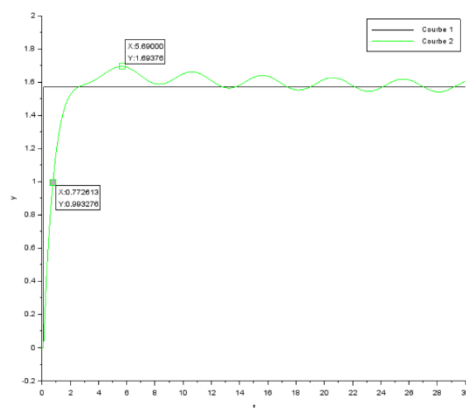


Gambar 4-15 Konfigurasi mencari nilai PD metode *Ziegler-Nichols*

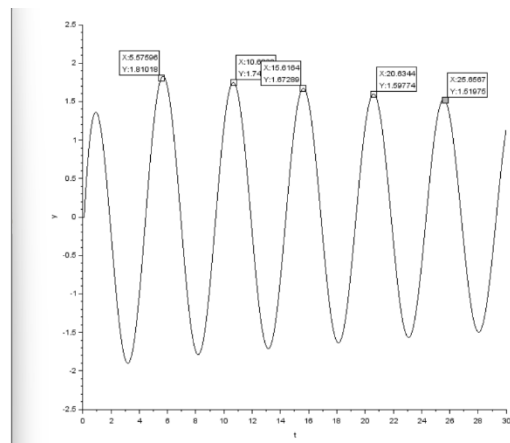
Gambar 4-15 menunjukkan konfigurasi pada blok Xcos PID untuk mencari nilai PD. Dari hasil iterasi didapatkan nilai K_u untuk mencari PD ialah 0,1.



Gambar 4-16 Grafik respon fungsi *transfer* pendulum ketika diberikan nilai $K_p = 0,1$. Gambar 4-16 menunjukkan grafik respon dari fungsi *transfer* pendulum. Berdasarkan grafik diatas, nilai T_u untuk mencari nilai PD menggunakan metode *Ziegler-Nichols* adalah 5,1. Setelah mendapatkan nilai dari K_u dan T_u , selanjutnya dapat mencari nilai dari PD menggunakan Tabel 2-3. Hasil yang didapatkan untuk nilai PD adalah $K_p = 0,08$ dan $K_d = 0,05$.



(a) Grafik respon motor DC



(b) Grafik respon pendulum ayun

Gambar 4-17 Respon model matematis kendali setelah diberi kendali PID *Tyres-Luyben* +PD *Ziegler-Nichols*

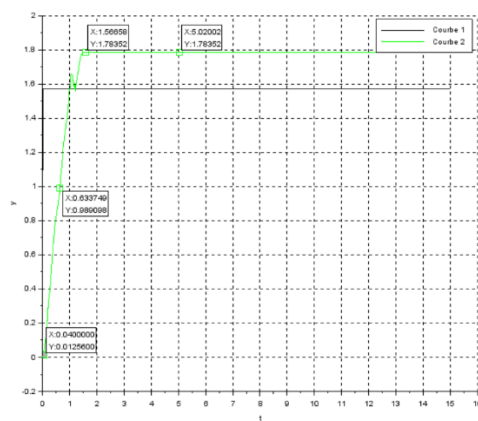
Gambar 4-17 menunjukkan respon model matematis kendali setelah diberi kendali PID *Tyres-Luyben* +PD *Ziegler-Nichols*, dari gambar diatas didapatkan informasi berupa:

- $Overshoot = 7,64\%$

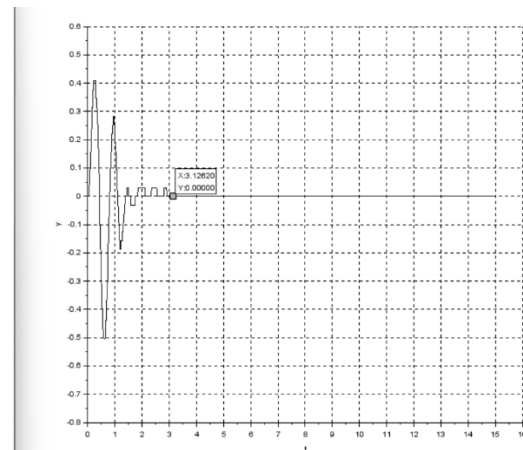
- *Time Constant* = 0,77 detik

Grafik respon fungsi *transfer* dari sistem kendali menunjukkan bahwa motor DC masih belum menunjukkan kondisi *steady state*. Pada pendulum ayun, grafik respon belum menunjukkan keadaan yang diinginkan tetapi sudah menunjukkan gerakan osilasi yang teredam dimana nilai dari *osilasi* pendulum terus mengalami penurunan.

Setelah mendapatkan nilai dari kendali PID+PD, selanjutnya nilai kendali tersebut bisa diujikan pada alat *rotary pendulum*. Untuk skema kendali yang digunakan pada objek kendali *rotary pendulum* sama seperti Gambar 4-10.



(a) Grafik respon motor DC



(b) Grafik respon pendulum ayun

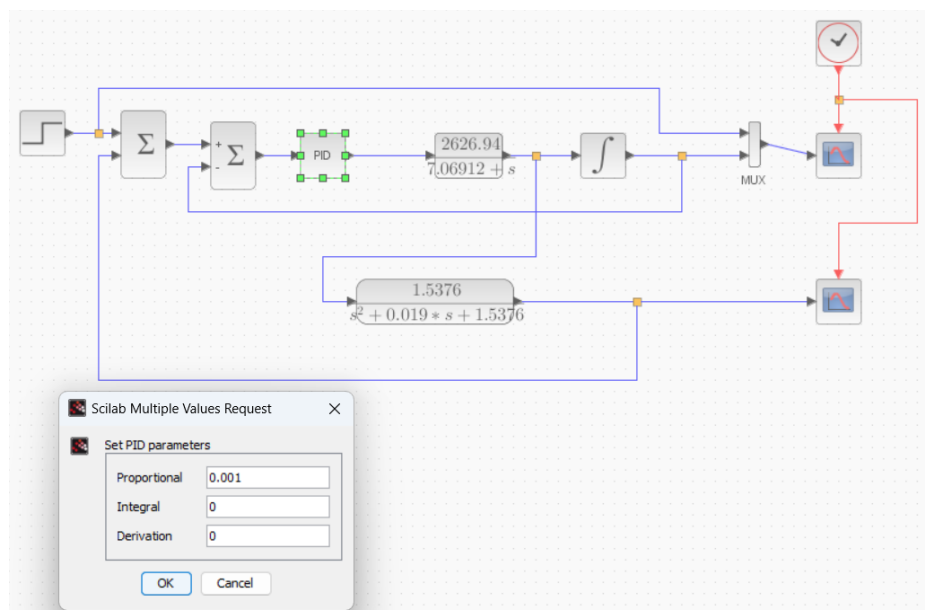
Gambar 4-18 Grafik respon kendali alat *rotary pendulum* dengan kendali PID *Tyres -Luyben* + PD *Ziegler-Nichols* menggunakan skema *normal anti-sway* Gambar 4-18 menunjukkan grafik respon kendali alat *rotary pendulum*. Dari gambar diatas didapatkan respon objek kendali *rotary pendulum* memiliki parameter spesifikasi respon sistem sebagai berikut:

- *Error steady state* = 0,21 rad
- *Input* = 1,57 rad
- *Settling time* = 1,5 s
- *Overshoot* = 13,38%
- *Time Constant* = 0,63 s

Pada pendulum ayun objek kendali *rotary pendulum* berayun selama 3,1 detik dan setelah itu pendulum ayun berhenti berayun.

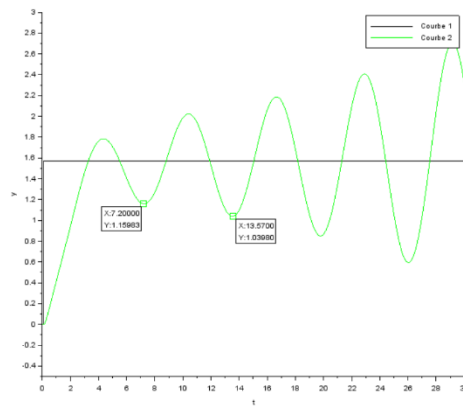
4.2.3 Pengujian menggunakan *anti-swing control using only position control* dengan pengendali PID *Ziegler Nichols*

Pada pengujian kali ini kendali yang digunakan hanyalah pengendali PID untuk motor DC, tidak terdapat pengendali untuk *feedback* dari pendulum ayun . Pengujian pertama dilakukan pada fungsi *transfer* untuk mencari nilai dari kendali PID. Metode yang digunakan adalah metode *Ziegler-Nichols*. Caranya adalah dengan tidak memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberinilai hingga grafik menunjukkan pergerakan *osilasi*.



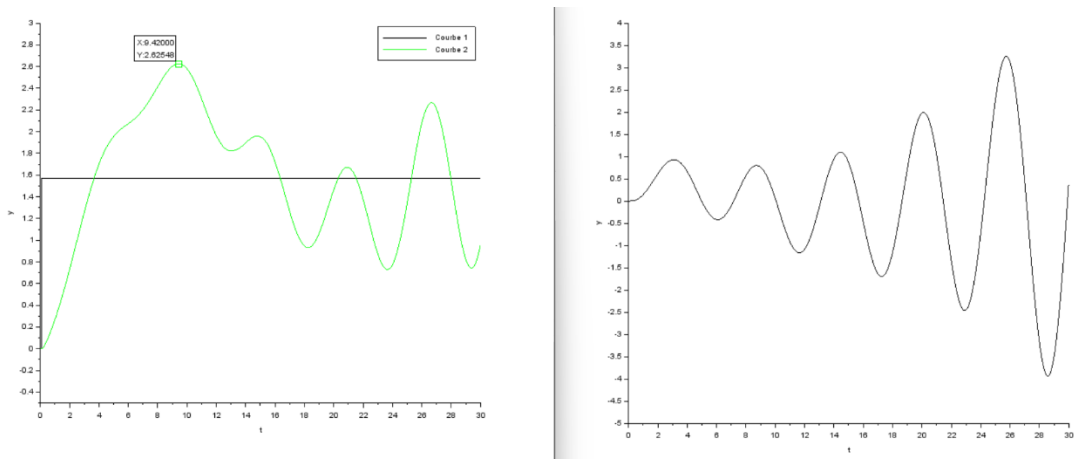
Gambar 4-19 Konfigurasi mencari nilai PID menggunakan *Ziegler-Nichols*

Gambar 4-19 menunjukkan konfigurasi untuk mencari nilai PID menggunakan metode *Ziegler-Nichols*. Dari hasil iterasi didapatkan nilai K_u untuk mencari PID ialah 0,001.



Gambar 4-20 Grafik respon fungsi *transfer* motor DC ketika diberiniali $K_p = 0,001$

Gambar 4-20 menunjukkan grafik respon dari fungsi *transfer* motor DC. Dari hasil grafik respon diatas bisa dapatkan nilai dari T_u , nilai T_u yang didapat adalah 6,7. Tabel 2-3 dapat digunakan untuk mencari nilai dari PID menggunakan metode *Ziegler-Nichols* dengan K_u adalah 0,001 dan T_u adalah 6,7. Hasil yang didapat dari Tabel 2-3 ialah $K_p = 0,0006$, $K_i = 0,0002$, dan $K_d = 0,0005$.

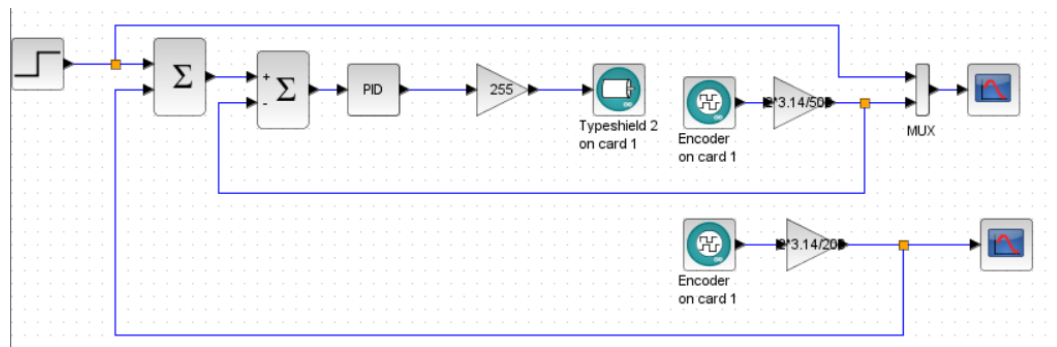


(a) Grafik respon posisi motor DC (b) Grafik respon pendulum ayun

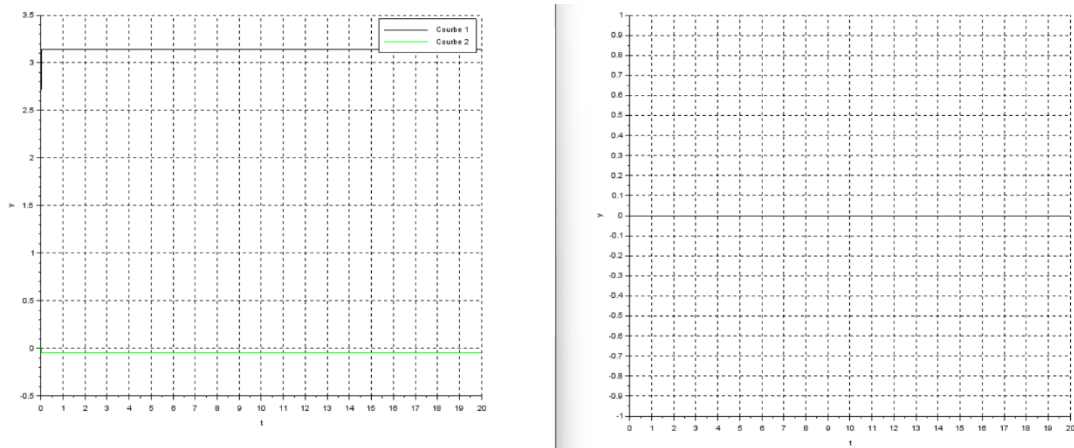
Gambar 4-21 Grafik respon model matematis ketika diberi nilai kendali PID *Ziegler-Nichols*

Gambar 4-21 menunjukkan grafik respon model matematis ketika diberi nilai kendali PID *Ziegler-Nichols*. Dari grafik respon diatas dilihat bawah sistem kendali masih menunjukkan respon tidak stabil baik pada grafik posisi motor DC (kiri) ataupun grafik respon pendulum ayun (kanan) dan motor DC memiliki *overshoot* sebesar 65,61%. Setelah mendapatkan nilai PID, selanjutnya nilai PID

tersebut dapat diujikan pada objek kendali *rotary pendulum*. Gambar 4-22 menunjukkan skema kendali *anti-swing control using only position control*



Gambar 4-22 Skema sistem kendali *control position only* menggunakan kendali PID Ziegler-Nichols



(a) Grafik respon posisi motor DC

(b) Grafik respon pendulum ayun

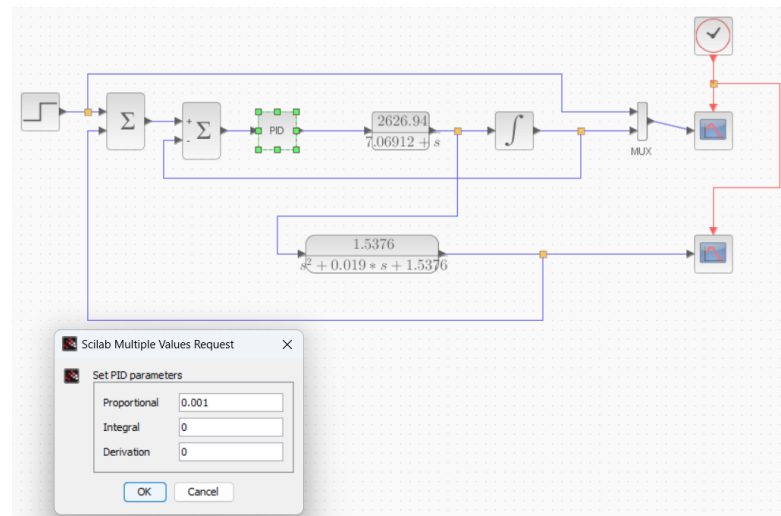
Gambar 4-23 Grafik respon kendali alat *rotary pendulum* dengan kendali PID Ziegler-Nichols menggunakan skema *anti-swing control using only position control*

Gambar 4-23 grafik respon alat *rotary pendulum*. Pada Gambar 4-23 menunjukkan bahwa alat *rotary pendulum* tidak bergerak, hal ini dikarenakan nilai dari pengendali PID yang diberikan sangatlah kecil, nilai dari K_p , K_i , dan K_d pada pengendali ini bisa dianggap 0.

4.2.4 Pengujian menggunakan *anti-swing control using only position control* dengan pengendali PID Tyreus-Luyben

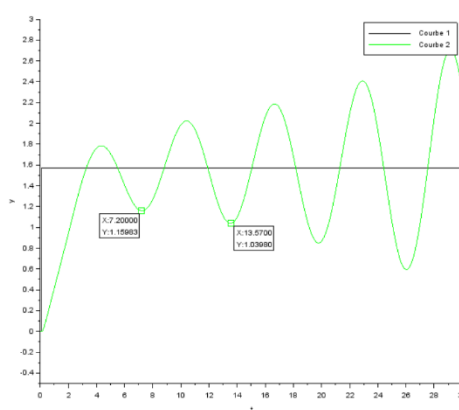
Pada pengujian ini skema yang digunakan masih sama seperti Gambar 4-19. Pada pengujian ini menggunakan metode pengendali Tyreus-Luyben.

Caranya adalah dengan tidak memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberikan nilai hingga grafik menunjukkan pergerakan *osilasi*.



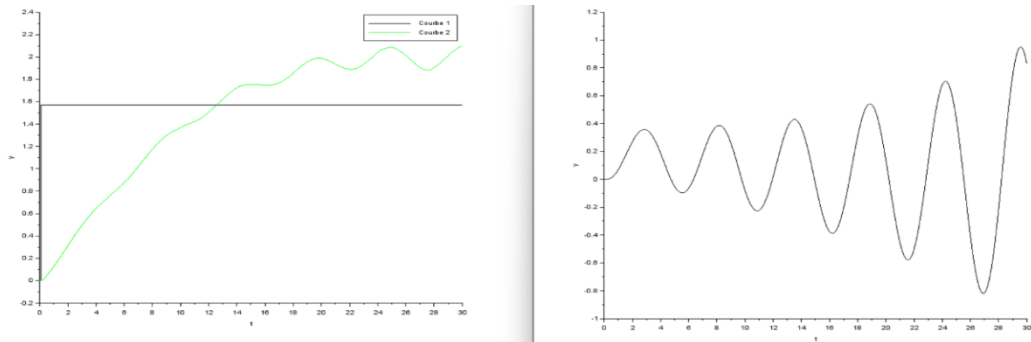
Gambar 4-24 Konfigurasi mencari nilai PID menggunakan *Tryeus-Luyben*

Gambar 4-24 menunjukkan konfigurasi untuk mencari nilai PID menggunakan metode *Tyresus -Luyben*. Dari hasil iterasi didapatkan nilai K_{cu} untuk mencari PID ialah 0,001.



Gambar 4-25 Grafik respon fungsi *transfer* motor DC ketika diberikan nilai $K_p = 0,001$

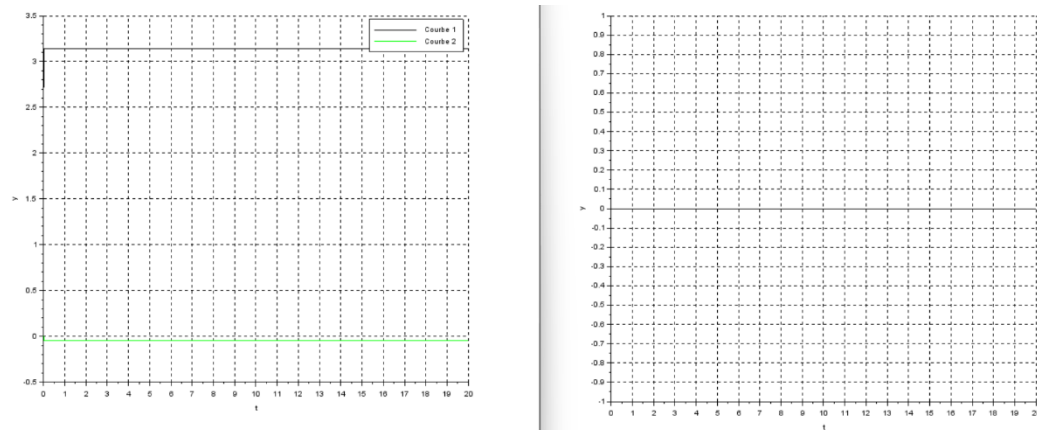
Gambar 4-25 menunjukkan grafik respon dari fungsi *transfer* motor DC. Dari hasil grafik respon diatas bisa dapatkan nilai dari P_u , nilai P_u yang didapat adalah 6,7. Tabel 2-4 dapat digunakan untuk mencari nilai dari PID menggunakan metode Ziegler-Nichols dengan K_{cu} adalah 0,001 dan P_u adalah 6,7. Hasil yang didapat dari Tabel 2-4 ialah $K_p = 0,0003$, $K_i = 0,00002$, dan $K_d = 0,0003$.



(a) Grafik respon posisi motor DC (b) Grafik respon pendulum ayun

Gambar 4-26 Respon model matematis ketika diberi nilai kendali PID *Tryeus-Luyben*

Gambar 4-26 respon model matematis ketika diberi nilai kendali PID *Tryeus-Luyben*. Dari grafik respon diatas dilihat bawah sistem kendali masih menunjukkan respon tidak stabil baik pada grafik posisi motor DC (kiri) ataupun grafik respon pendulum ayun (kanan) dan nilai grafik respon posisi motor DC akan terus mengalami kenaikan hal ini menunjukkan respon yang tidak stabil. Setelah mendapatkan nilai PID, selanjutnya nilai PID tersebut dapat diujikan pada objek kendali *rotary pendulum*. Untuk skema kendali yang digunakan pada objek kendali *rotary pendulum* sama seperti Gambar 4-22.



(a) Grafik respon posisi motor DC (b) Grafik respon pendulum ayun

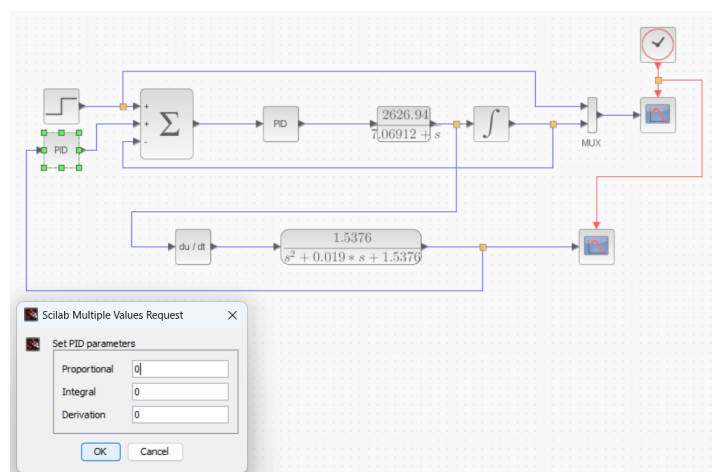
Gambar 4-27 Grafik respon kendali alat *rotary pendulum* dengan kendali PID *Tryeus-Luyben* menggunakan skema *anti-wing control using only position control*

Gambar 4-27 menunjukkan grafik respon alat *rotary pendulum*. Pada gambar diatas dilihat bahwa alat *rotary pendulum* tidak bergerak, hal ini

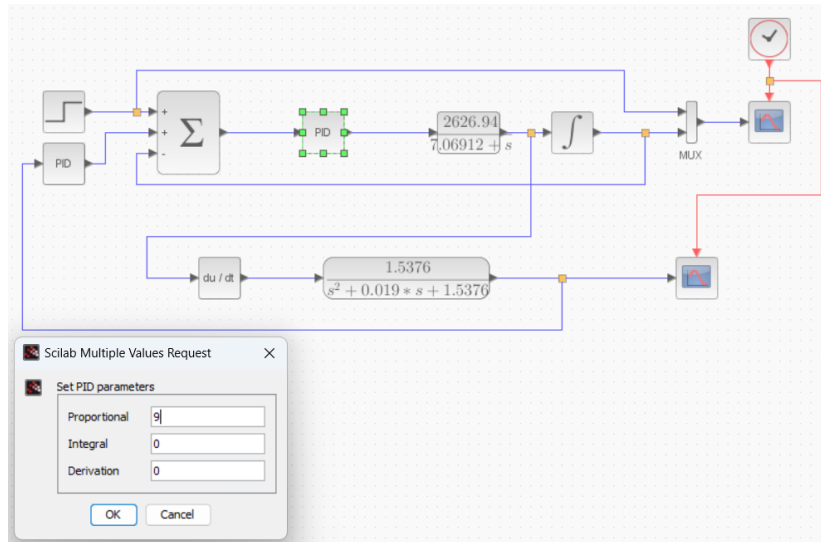
dikarenakan nilai dari pengendali PID yang diberikan sangatlah kecil, nilai dari K_p , K_i , dan K_d pada pengendali ini bisa dianggap 0.

4.2.5 Pengujian menggunakan *delayed feedback control* dengan PID+PD *Ziegler-Nichols*

Pada pengujian kali ini cara yang dilakukan tidaklah jauh berbeda dengan pengujian-pengujian sebelumnya. Pada pengujian ini menggunakan skema *delayed feedback control* dengan kendali PID+PD *Ziegler-Nichols*. Pada pengujian ini kendali yang pertama dicari ialah nilai dari PID untuk mengendalikan motor DC. Pada pengujian kali ini ada sedikit perbedaan ketika mencari nilai dari PID, yang dimana ketika mencari nilai PID untuk motor DC *feedback* dari pendulum ayun berpengaruh terhadap motor DC tetapi pada pengujian kali ini *feedback* dari pendulum tidak berpengaruh kepada motor DC. Hal ini dikarenakan ketika *feedback* dari pendulum ayun dimasukkan ke dalam sistem kendali/berpengaruh terhadap motor DC maka sistem kendali akan menjadi *error* dan grafik respon pada Scilab Xcos tidaklah muncul. Oleh karena itu untuk mencari nilai dari PID motor DC kendali konfigurasi dari blok PID pendulum haruslah bernilai 0 untuk K_p , K_i , dan K_d . Setelah memberikan nilai 0 pada konfigurasi blok PID pendulum barulah nilai PID motor DC bisa dicari dengan cara tidak memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberinilai hingga grafik respon menunjukkan pergerakan *osilasi*.

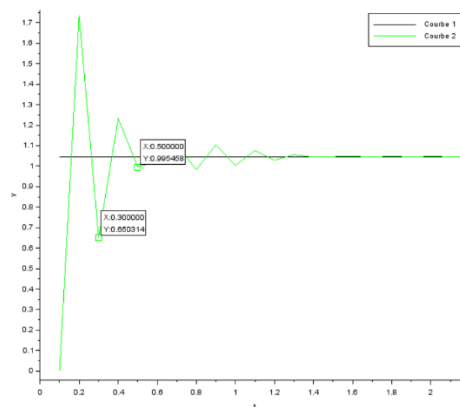


Gambar 4-28 Konfigurasi blok PID pada pendulum ayun

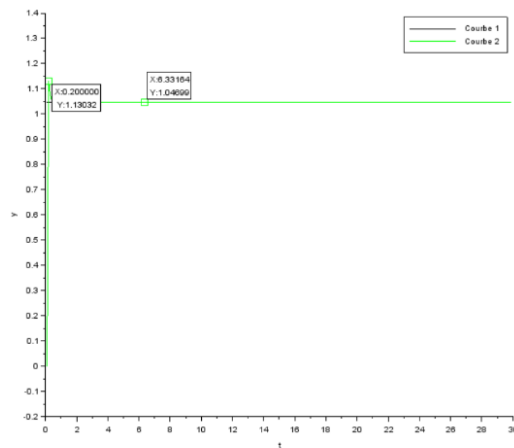


Gambar 4-29 Konfigurasi dan skema mencari nilai PID menggunakan *Ziegler-Nichols*

Gambar 4-28 dan Gambar 4-29 menunjukkan konfigurasi pada blok Xcos untuk PID pada *feedback* pendulum dan blok Xcos PID untuk kendali motor DC. Dari hasil iterasi didapatkan nilai K_u untuk mencari PID ialah 9.

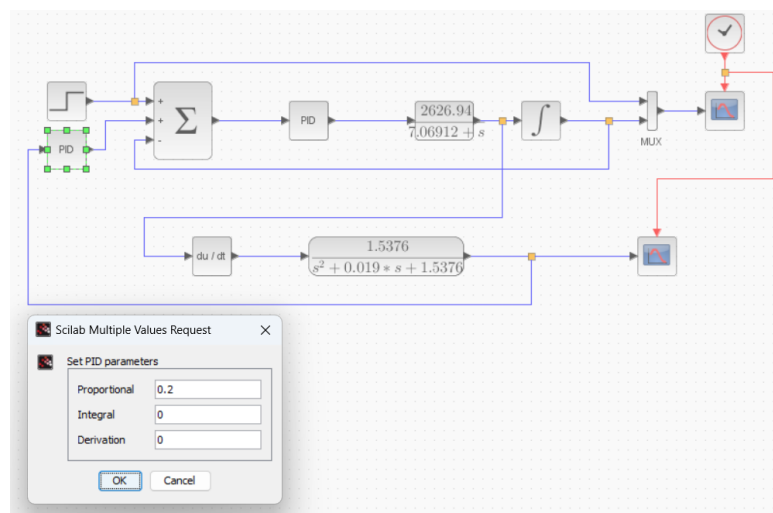


Gambar 4-30 Grafik respon fungsi *transfer* motor DC setelah diberi nilai $K_p = 9$. Gambar 4-30 menunjukkan grafik respon dari fungsi *transfer* motor DC setelah diberi nilai $K_p = 9$. Dari hasil grafik respon diatas bisa dapatkan nilai dari T_u , nilai T_u yang didaapat adalah 0,2. Tabel 2-3 dapat digunakan untuk mencari nilai dari PID menggunakan metode *Ziegler-Nichols* dengan K_u adalah 9 dan T_u adalah 0,2. Hasil yang didapat dari Tabel 2-3 ialah $K_p = 5,4$, $K_i = 54$, dan $K_d = 0,13$.

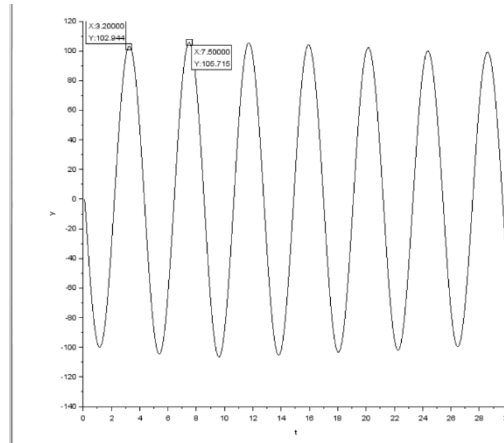


Gambar 4-31 Respon fungsi *transfer* motor DC setelah diberi nilai PID menggunakan metode *Ziegler-Nichols*

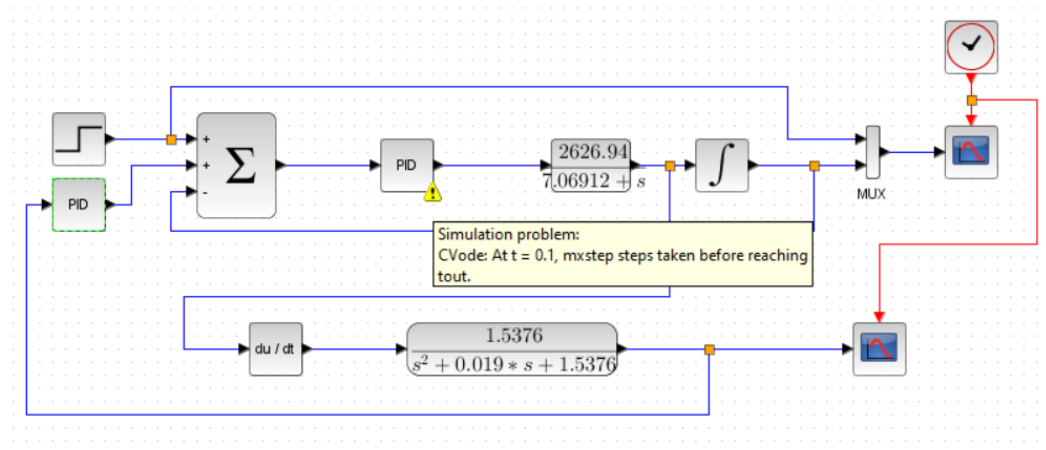
Gambar 4-31 menunjukkan respon fungsi *transfer* motor DC. Dari gambar diatas dilihat bahwa grafik respon fungsi *transfer* motor DC (kiri) menunjukkan hasil yang bagus, hal ini dikarenakan *feedback* daripada pendulum ayun tidak dimasukkan kedalam sistem kendali. Pada grafik respon pendulum ayun (kanan), pendulum ayun belum terlihat redaman yang terjadi karena *feedback* dari pendulum ayun belum diberi kendali . Untuk mengendalikan Pendulum ayun perlu ditambahkan pengendali PD. Caranya adalah dengan tidak memasukkan nilai konstanta *integral* dan *derivatif* pada kendali *proportional* diberinilai hingga grafik menunjukkan pergerakan *osilasi*. Gambar 4-32 menunjukkan konfigurasi blok diagram PID. Dari hasil iterasi didapatkan nilai K_u untuk mencari PD ialah 0,2.



Gambar 4-32 Konfigurasi mencari nilai PD metode *Ziegler-Nichols*

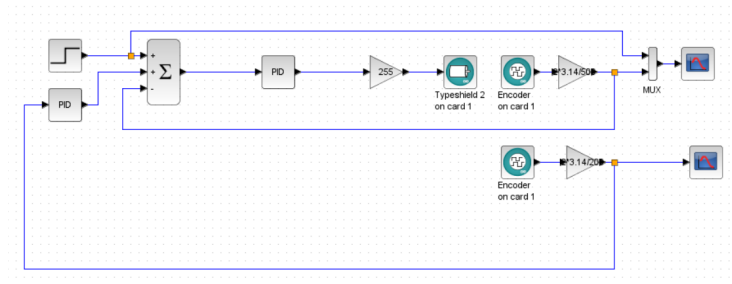


Gambar 4-33 Grafik respon fungsi *transfer* pendulum ketika diberikan nilai $K_p = 0,2$. Gambar 4-33 menunjukkan grafik respon fungsi *transfer* pendulum ketika hanya diberi nilai $K_p = 0,2$. Berdasarkan grafik respon diatas, nilai T_u untuk mencari nilai PD menggunakan metode *Ziegler-Nichols* adalah 4,3. Setelah mendapatkan nilai dari K_u dan T_u , selanjutnya dapat mencari nilai dari PD menggunakan Tabel 2-3. Hasil yang didapatkan untuk nilai PD adalah $K_p = 0,16$ dan $K_d = 0,86$.

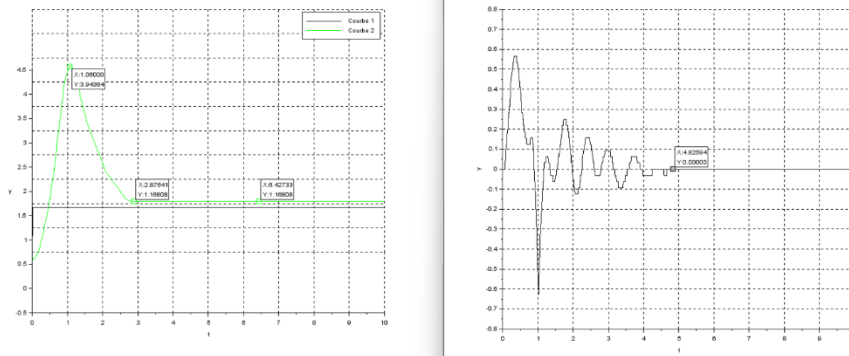


Gambar 4-34 *Error* pada simulasi Xcos

Gambar 4-34 menunjukkan *error* yang terjadi pada simulasi Xcos. Adapun hasil simulasi yang didapatkan tidak memunculkan grafik respon yang disebabkan oleh adanya komputasi yang tidak bisa terbaca atau tidak terselesaikan oleh Scilab Xcos. Setelah mendapatkan nilai dari kendali PID+PD, selanjutnya nilai kendali tersebut bisa diujikan pada alat *rotary pendulum*.



Gambar 4-35 Skema kendali *delayed feedback* pada alat *rotary pendulum*



(a) Grafik respon posisi motor DC (b) Grafik respon pendulum ayun

Gambar 4-36 Grafik respon kendali alat *rotary pendulum* dengan kendali PID+PD *Ziegler-Nichols* menggunakan skema *delayed feedback*

Gambar 4-35 dan Gambar 4-36 menunjukkan skema kendali *delayed feedback* dan grafik respon dari alat *rotary pendulum*. Dari gambar diatas didapatkan respon objek kendali *rotary pendulum* memiliki parameter spesifikasi respon sistem sebagai berikut:

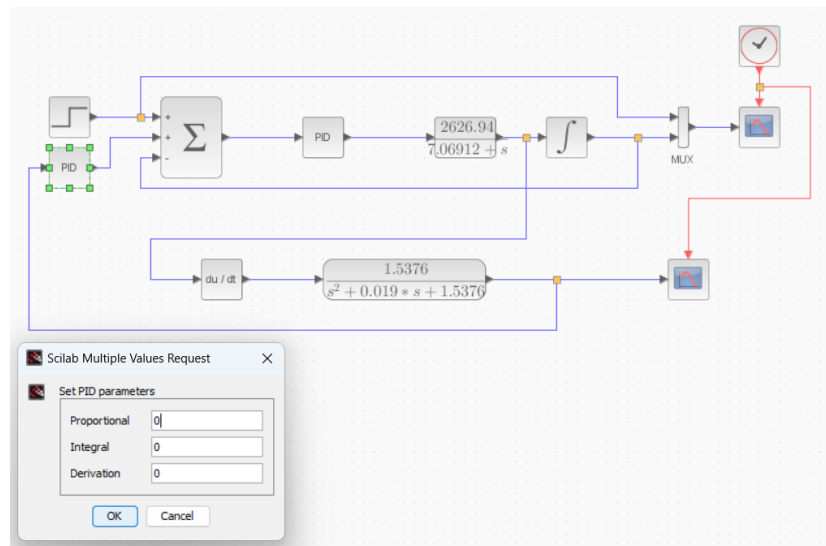
- *Error steady state* = 0,633 rad
- *Input* = 1,047 rad
- *Settling time* = 2,87 s
- *Overshoot* = 276,31%

Pada pendulum ayun objek kendali *rotary pendulum* berayun selama 4,82 detik dan setelah itu pendulum ayun berhenti berayun.

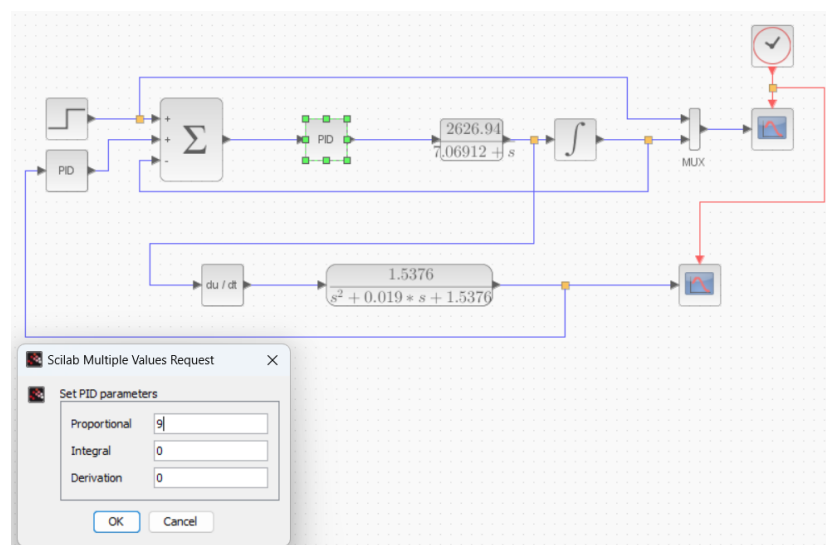
4.2.6 Pengujian menggunakan *delayed feedback control* dengan PID *Tyurus-Luyben* + PD *Ziegler Nichols*

Pada pengujian ini munggunakan skema *delayed feedback control* dengan kendali PID *Tyurus-Luyben* + PD *Ziegler-Nichols*. Pada pengujian ini kendali yang

pertama dicari ialah nilai dari PID untuk mengendalikan motor DC. Pada pengujian sama seperti sub sub bab 4.3.5 yang dimana mencari nilai dari PID motor DC kendali konfigurasi dari blok PID pendulum haruslah bernilai 0 untuk K_p , K_i , K_d . Setelah memberikan nilai 0 pada konfigurasi blok PID pendulum barulah nilai PID motor DC bisa dicari dengan cara tidak memasukkan nilai konstanta *integral* dan *derivativ* pada kendali *proportional* diberinilai hingga grafik menunjukkan pergerakan *osilasi*.

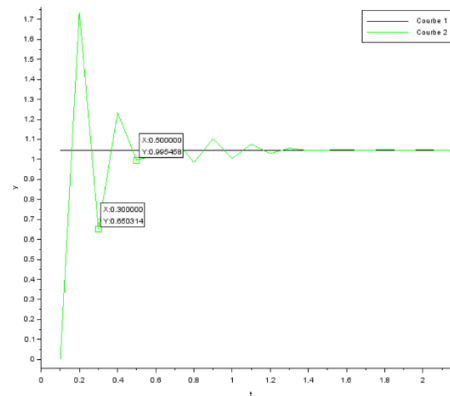


Gambar 4-37 Konfigurasi blok PID pada pendulum ayun



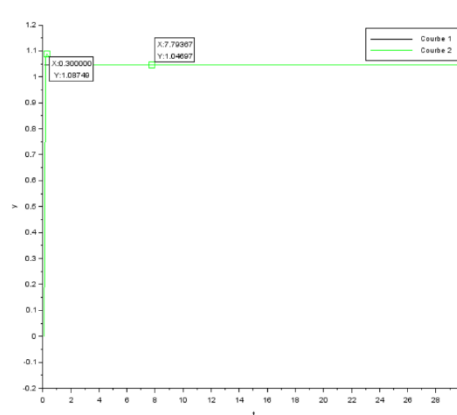
Gambar 4-38 Konfigurasi dan skema mencari nilai PID menggunakan *Tyreus-Luyben*

Gambar 4-37 dan Gambar 4-38 menunjukkan konfigurasi pada blok Xcos untuk PID pada *feedback* pendulum dan blok Xcos PID untuk kendali motor DC. Dari hasil iterasi didapatkan nilai K_{cu} untuk mencari PID ialah 9.



Gambar 4-39 Grafik respon fungsi *transfer* motor DC (kiri) setelah diberi nilai K_p
= 9

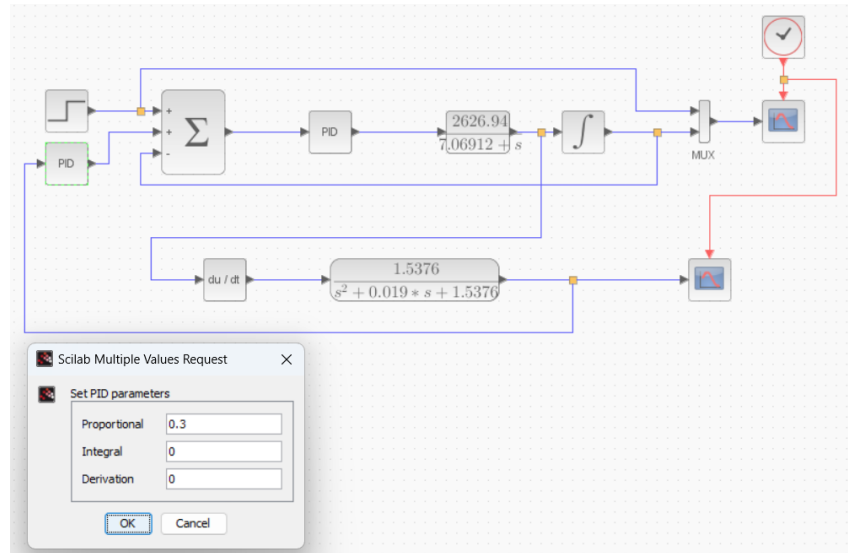
Gambar 4-39 menunjukkan grafik respon dari fungsi *transfer* motor DC setelah diberi nilai K_p . Dari hasil grafik respon diatas bisa dapatkan nilai dari P_u , nilai P_u yang didapat adalah 0,2. Tabel 2-4 dapat digunakan untuk mencari nilai dari PID menggunakan metode *Tyres-Luyben* dengan K_{cu} adalah 9 dan P_u adalah 0,2. Hasil yang didapat dari Tabel 2-4 ialah $K_p = 2,81$, $K_i = 6,38$, dan $K_d = 0,084$.



Gambar 4-40 Grafik respon fungsi *transfer* motor DC setelah diberi nilai PID
menggunakan metode *Tyres-Luyben*

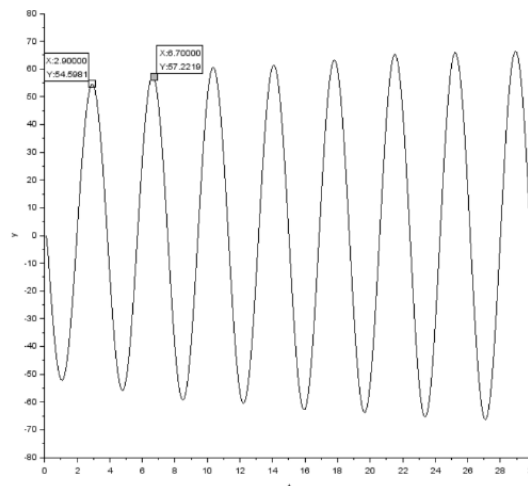
Gambar 4-40 menunjukkan respon fungsi *transfer* motor DC. Dari gambar diatas dilihat bahwa grafik respon fungsi *transfer* motor DC (kiri) menunjukkan hasil yang bagus, hal ini dikarenakan *feedback* daripada pendulum ayun tidak dimasukkan kedalam sistem kendali. Pada grafik respon pendulum ayun (kanan),

pendulum ayun belum terlihat redaman yang terjadi karena *feedback* dari pendulum ayun belum diberi kendali. Untuk mengendalikan pendulum ayun perlu ditambahkan pengendali PD. Caranya adalah dengan tidak memasukkan nilai konstanta *integral* dan *derivativ* kemudian pada konstanta *proportional* diberi nilai hingga grafik respon menunjukkan pergerakan *osilasi*.



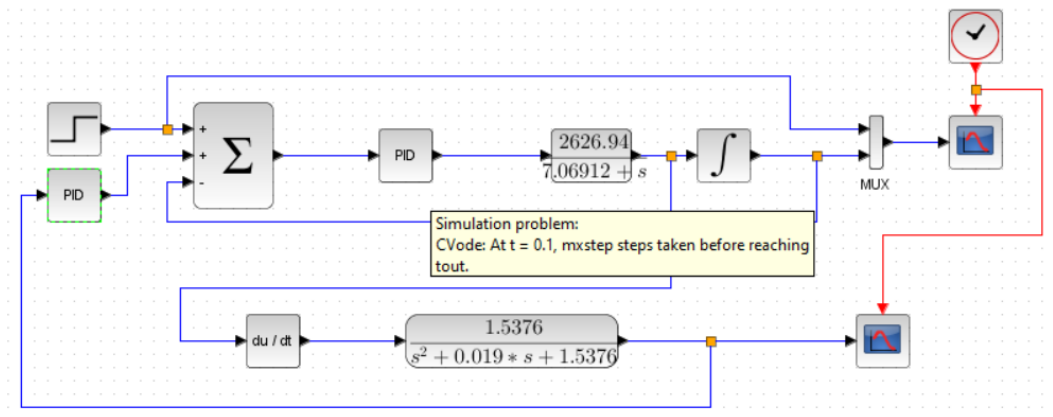
Gambar 4-41 Konfigurasi mencari nilai PD metode *Tyreus-Luyben*

Gambar 4-41 menunjukkan konfigurasi blok diagram PID. Dari hasil iterasi didapatkan nilai K_u untuk mencari PD ialah 0,3.



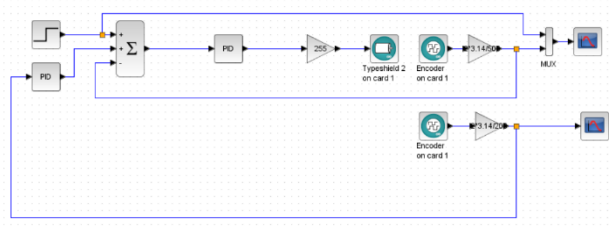
Gambar 4-42 Grafik respon fungsi *transfer* pendulum ketika diberinilai $K_p = 0,3$
 Gambar 4-42 menunjukkan hasil respon pendulum ayun ketika diberikan nilai $K_p = 0,3$. Berdasarkan grafik respon diatas, nilai T_u untuk mencari nilai PD menggunakan metode *Ziegler-Nichols* adalah 3,8. Setelah mendapatkan nilai dari

K_u dan T_u , selanjutnya dapat mencari nilai dari PD menggunakan Tabel 2-3. Hasil yang didapatkan untuk nilai PD adalah $K_p = 0,24$ dan $K_d = 0,11$.

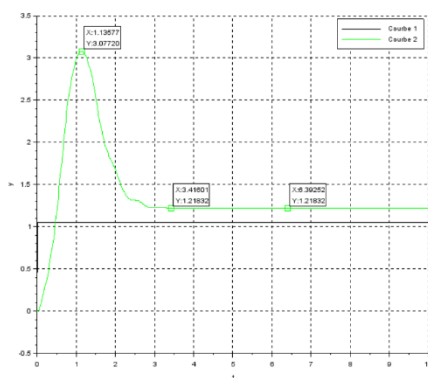


Gambar 4-43 Error pada simulasi Xcos

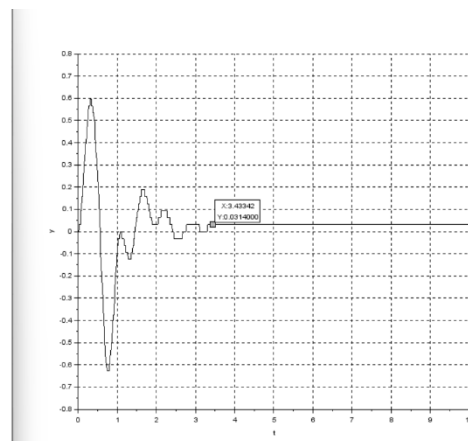
Gambar 4-43 menunjukkan *error* Xcos saat melakukan simulasi. Adapun hasil simulasi yang didapatkan tidak memunculkan respon yang disebabkan oleh adanya komputasi yang tidak bisa terbaca atau tidak terselesaikan oleh Scilab Xcos. Setelah mendapatkan nilai dari kendali PID+PD, selanjutnya nilai kendali tersebut bisa diujikan pada alat *rotary pendulum*.



Gambar 4-44 Skema kendali *delayed feedback* pada alat *rotary pendulum*



(a) Grafik respon posisi motor DC



(b) Grafik respon pendulum ayun

Gambar 4-45 Grafik respon kendali alat *rotary pendulum* dengan kendali PID Tyreus-Luyben + PD Ziegler-Nichols menggunakan skema *delayed feedback* Gambar 4-44 dan Gambar 4-45 menunjukkan skema kendali *delayed feedback* dan grafik respon dari alat *rotary pendulum*. Dari gambar diatas didapatkan respon objek kendali *rotary pendulum* memiliki parameter spesifikasi respon sistem sebagai berikut:

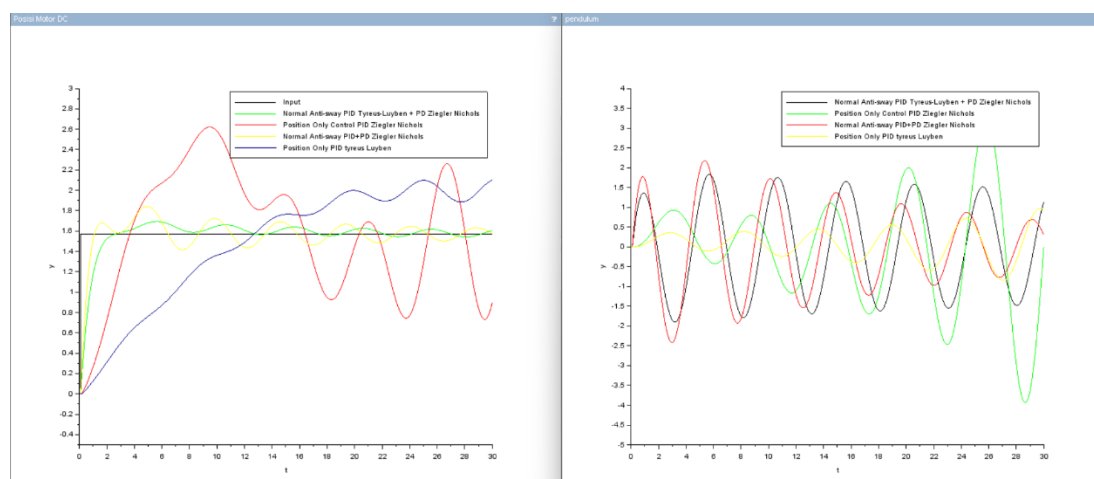
- *Error steady state* = 0,074 rad
- *Input* = 1,047 rad
- *Sattling Time* = 3,4 s
- *Overshoot* = 193,22%

Pada pendulum ayun objek kendali *rotary pendulum* berayun selama 6,6 detik dan setelah itu pendulum ayun berhenti berayun.

4.3 Perbandingan

4.3.1 Perbandingan fungsi *transfer* sistem kendali

Setelah mendapatkan respon-respon sistem dari setiap skema kontroler yang digunakan, proses selanjutnya ialah membandingkan respon-respon dari sistem tersebut.

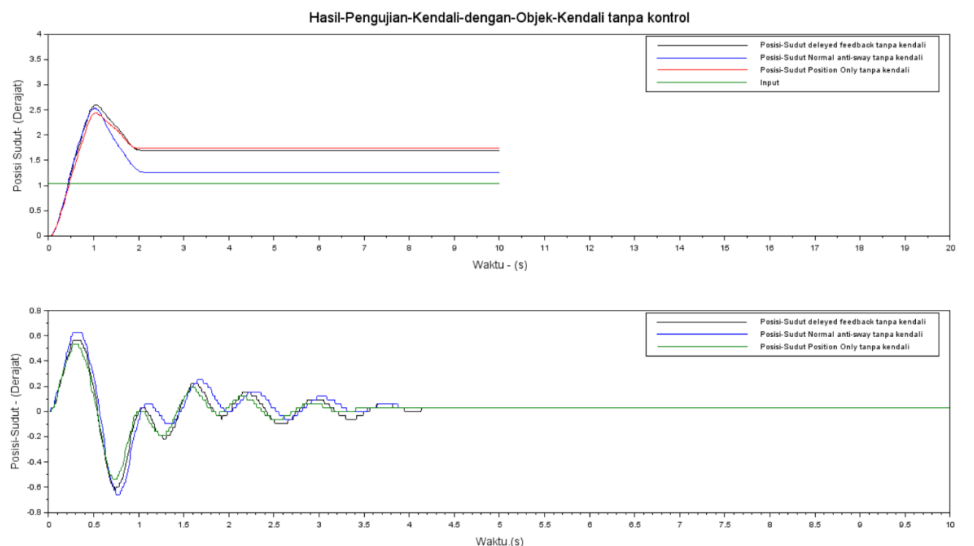


Gambar 4-46 Grafik respon perbandingan fungsi *transfer* sistem kendali Gambar 4-46 menunjukkan perbandingan respon pada fungsi *transfer* sistem kendali. Pada perbandingan untuk fungsi *transfer* sistem *anti-sway* skema kendali *delayed feedback* tidak dimasukkan karena pada metode ini grafik kendali tidak

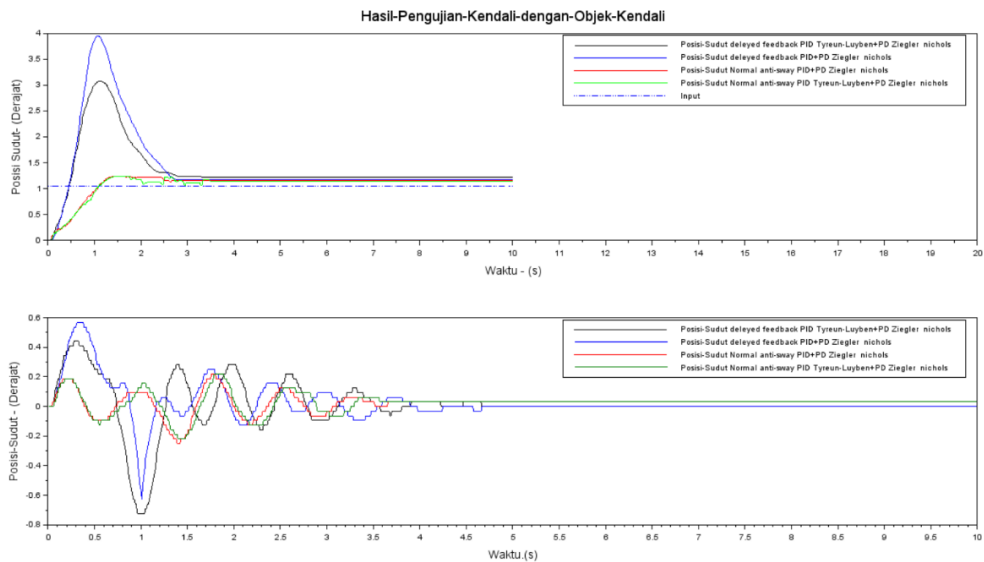
muncul pada tampilan *display* dan mengalami *error* pada proses simulasi. Hasil pada simulasi fungsi *transfer* dan hasil pada simulasi objek kendali memiliki perbedaan nilai pada grafik respon yang dihasilkan, hal ini dikarenakan pada fungsi *transfer* terdapat nilai-nilai yang diabaikan seperti nilai gaya gesek yang dimana pada objek kendali *rotary pendulum* terdapat gaya gesek tetapi pada fungsi transfer nilai tersebut diabaikan dan pada fungsi *transfer* motor DC pembebanan yang terjadi adalah pembebanan tanpa beban yang dimana pada objek kendali *rotary pendulum* terdapat pembebanan yang terjadi pada motor DC.

4.3.2 Perbandingan sistem *anti-sway* pada objek kendali

Setelah melakukan perbandingan pada fungsi *transfer* sistem kendali, selanjutnya adalah membandingkan sistem *anti-sway* pada objek kendali *rotary pendulum*, terdapat dua perbandingan yang dilakukan pada alat objek kendali *rotary pendulum*. Perbandingan pertama dilakukan menggunakan skema yang digunakan tanpa kontroler pengendali, perbandingan kedua dilakukan menggunakan skema yang digunakan dan menggunakan pengendali PID+PD dengan metode yang berbeda.



Gambar 4-47 Objek kendali tanpa kontrol pengendali



Gambar 4-48 Perbandingan sistem *anti-sway* pada objek kendali

Gambar 4-48 menunjukkan perbandingan respon alat *rotary pendulum* dengan skema yang berbeda dan menggunakan skema yang berbeda, Gambar 4-47 menunjukkan perbandingan dari setiap skema kendali yang digunakan tanpa kontroler. Pada perbandingan sistem *anti-sway* ini *input* yang masuk kedalam sistem adalah 1,047 rad. Pada gambar 4-48, perbandingan ini skema kendali *position control only* tidak dimasukkan karena pada skema tersebut nilai dari kendali PID sangatlah kecil hingga bisa dianggap nol, hal ini membuat objek kendali tidak bergerak.

4.4 Pembahasan

Gambar 4-47 menunjukkan perbandingan respon objek kendali *rotary pendulum* tanpa diberi kendali. Berdasarkan gambar tersebut diketahui nilai *input* yang diberikan adalah 1,047 rad. Respon dari motor DC pada objek kendali *rotary pendulum* menunjukkan hasil yang sangat jauh berbeda dengan nilai yang diinginkan. Hasil tersebut menunjukkan bahwa alat *rotary pendulum* memerlukan kendali agar hasil keluaran sesuai dengan nilai masukan yang diberikan.

Gambar 4-48 menunjukkan hasil objek kendali setelah diberikan nilai kendali PID+PD. Ketika menggunakan skema *normal anti-sway control* dan diberi kendali PID+PD dengan metode *Ziegler-Nichols* grafik respon motor DC menunjukkan perbaikan tetapi masih memiliki sedikit *overshoot*. Begitu juga

dengan kendali PID *Tyres-Luyben* +PD *Ziegler-Nichols* menunjukkan perbaikan pada respon motor DC tetapi masih memiliki *overshoot*. Pada kedua metode tersebut hasil keluaran dari motor DC sudah mendekati nilai *input*. Ketika menggunakan skema *delayed feedback* dan diberikan kendali PID+PD dengan metode *Ziegler-Nichols* terlihat pada grafik respon memiliki *overshoot* yang sangat besar tetapi nilai *steady state error* dari metode ini sudah mendekati nilai dari input yang diberikan, hal yang serupa terjadi juga ketika skema *delayed feedback* diberikan kendali PID *Tyres-Luyben* +PD *Ziegler-Nichols*. Untuk skema *position control only* objek kendali *rotary pendulum* tidak bergerak dikarenakan nilai kontroler PID pada skema ini sangatlah kecil.

Pada Gambar 4-48 juga menunjukkan respon pendulum ayun setelah diberikan nilai kendali PD. Pendulum ayun memiliki ayunan yang sangat besar ketika belum diberikan nilai kendali PD seperti ditunjukkan pada Gambar 4-47. Ketika menggunakan skema *normal anti-sway control* dan berikan nilai PID+PD *Ziegler-Nichols*, nilai dari ayunan pendulum menjadi mengecil dan tidak sebesar sebelum diberi nilai kendali dan juga pendulum sudah berhenti sebelum 4 detik, hal serupa juga terjadi pada skema *normal anti-sway control* ketika diberi kendali PID *Tyres-Luyben* + PD *Ziegler Nichols*. Ketika menggunakan skema *delayed feedback* kendali PID+PD *Ziegler-Nichols* dan PID *Tyres-Luyben* +PD *Ziegler Nichols* ayunan pendulum masih besar seperti saat belum diberi nilai kendali.

BAB 5

PENUTUP

5.1 Kesimpulan

Kesimpulan dari penelitian ini adalah:

1. Simulasi objek kendali telah dilakukan dengan merancang blok-blok fungsi pada Xcos, dengan melakukan simulasi dapat diketahui respon dari objek kendali *rotary pendulum*.
2. Perancangan sistem *anti-sway* dapat dilakukan dengan mencari nilai kendali PID + PD menggunakan simulasi model matematis objek kendali terlebih dahulu dengan konfigurasi *normal anti-sway control*, *position control only* dan *delayed feedback control*. Rancangan sistem *anti-sway* yang telah dirancang sudah dapat disimulasikan pada alat objek kendali *rotary pendulum* dan grafik respon dari setiap konfigurasi yang digunakan menunjukkan hasil yang lebih baik setelah diberi nilai pengendali.
3. Berdasarkan hasil implementasi sistem kendali pada alat, disimpulkan bahwa alat memang membutuhkan pengendali untuk mencapai respon yang diinginkan. Penggunaan metode kendali PID *Tyres-Luyben* + PD *Ziegler Nichols* dengan konfigurasi *normal anti-sway control* menunjukkan hasil yang lebih baik dibandingkan dengan konfigurasi *position control only* dan *delayed feedback control* dengan nilai *overshoot* yang lebih kecil yaitu sebesar 13,38% dan nilai *error steady state* sebesar 0,12 rad dengan *input* 1,57 rad kemudian nilai dari ayunan pendulum menjadi kecil daripada sebelum diberikan nilai kendali PD.

5.2 Saran

Penerapan kendali *anti-sway* selanjutnya dapat menggabungkan dua metode kendali yang berbeda, seperti penggunaan metode kendali PID+PD ditambah fuzzy ataupun metode lainnya

DAFTAR PUSTAKA

- Ahmed, & Alnaib, I. (2019). DC Motors.
- Akbar, M. J., Rahmdhani, M., & Wibowo, A. S. (2016). Sistem Kendali Anti Sway Overhead Crane. *Jurnal Penelitian dan Pengembangan Telekomunikasi, Kendali, Elektrik, dan Elektronika (TEKTRIKA)*, 160-166.
- Hadiyan, Faza. (2021). Perancangan Dan Implementasi Sistem Kendali Untuk Rotary Pendulum dengan Kontroller PID Menggunakan SCILAB-ARDUINO. Yogyakarta: Jurusan Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia.
- Handayani, D. N., Pramudya, Y., Suparwoto, & Muchlas. (2018). The Application of Scilab Software in Frequency Mode Simulation on the Circular Membrane. *Journal of Physics: Theories and Applications*, 83-94.
- Kurniawan, E., Simbolon, R. S., & MN, N. (2020). Analysis and Simulation of PI and PID Control System Using Xcos Scilab. *Journal of Technomaterials*, 108-116.
- Masoud, Z. N., & Nayfeh, A. H. (2002). Sway reduction on container cranes using delayed feedback controller. *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*.
<https://doi.org/10.2514/6.2002-1279>
- Muntari, & Nurhadi, H. (2013). Desain Sistem Kendali Rotary Pendulum dengan Sliding-PID. *JURNAL TEKNIK POMITS*, F243-F249
- Nise, N. S. (2015). Control System Engineering. John Willey & Sons, Inc.
- Omar, H.M. Control of gantry and tower cranes. M.S. *Virginia Tech. Thesis*, Blacksburg, VA. 2003.
- Shahrokhi, M., & Zomorodi, A.R. (2012). Comparison of PID Controller Tuning Methods.
- Solihin, M. I., & Wahyudi, W. (2006). Anti-Sway Control of Automatic Gantry Crane with Minimum Sensor.
- Solihin, M. I., & Wahyudi, W. (2009). Sensorless anti-swing control of automatic gantry crane using Dynamic Recurrent Neural Network-based soft sensor.

International Journal of Intelligent Systems Technologies and Applications. <https://doi.org/10.1504/ijista.2009.022691>

Solihin, M. I., Legowo, A., & Wahyudi, W. (2010). Fuzzy-tuned PID Anti-swing Control of Automatic Gantry Crane. *Journal of Vibration and Control*.

Toochinda, V. (2016). Feedback Control with Scilab and Arduino. Bangkok: Varodom Toochinda, Ph,D

Triyogi, Nandang. (2017). Perancangan dan Pembuatan Alat Peraga Sistem Kendali Ayunan pada Simulator Crane Menggunakan Arduino Mega 2560 Berbasis PC. Yogyakarta: Jurusan Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia.

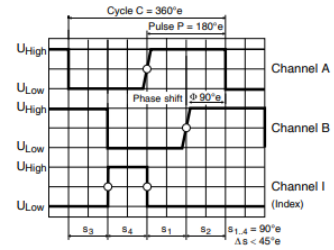
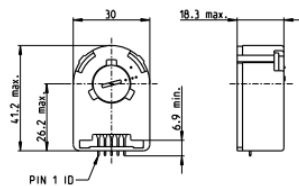
Yong-Seok Kim, Han-Suk Seo and Sul, S.-K. (2001) 'A new anti-sway control scheme for trolley crane system', in *Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No.01CH37248)*. *Proceedings of 2001 IEEE Industry Applications Society 36th Annual Meeting - IAS'01*, Chicago, IL, USA: IEEE, pp. 548–552 vol.1. Available at: <https://doi.org/10.1109/IAS.2001.955474>.

Zheng, D., Zhang, S., Wang, S., Hu, C., & Zhao, X. (2015). *A Capacitive Rotary Encoder Based on Quadrature Modulation and Demodulation*.

LAMPIRAN-LAMPIRAN

Datasheet sensor encoder HEDS 5540

Encoder HEDS 5540 500 CPT, 3 channels



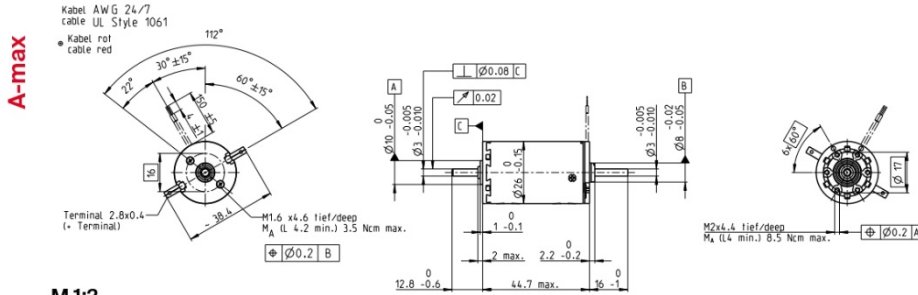
Direction of rotation cw (definition cw p. 78)

- Stock program
- Standard program
- Special program (on request)

Type	Part Numbers				
	110511	110513	110515	110517	X drives
Counts per turn	500	500	500	500	500
Number of channels	3	3	3	3	3
Max. operating frequency (kHz)	100	100	100	100	100
Max. speed (rpm)	12000	12000	12000	12000	12000
Shaft diameter (mm)	3	4	6	8	2-4

Datasheet Motor DC Maxon A-max 26

A-max 26 Ø26 mm, precious metal brushes CLL, 4.5 watt



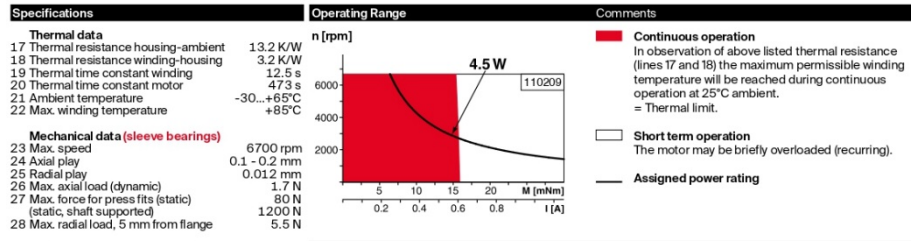
M 1:2

- Stock program
- Standard program
- Special program (on request)

Part Numbers

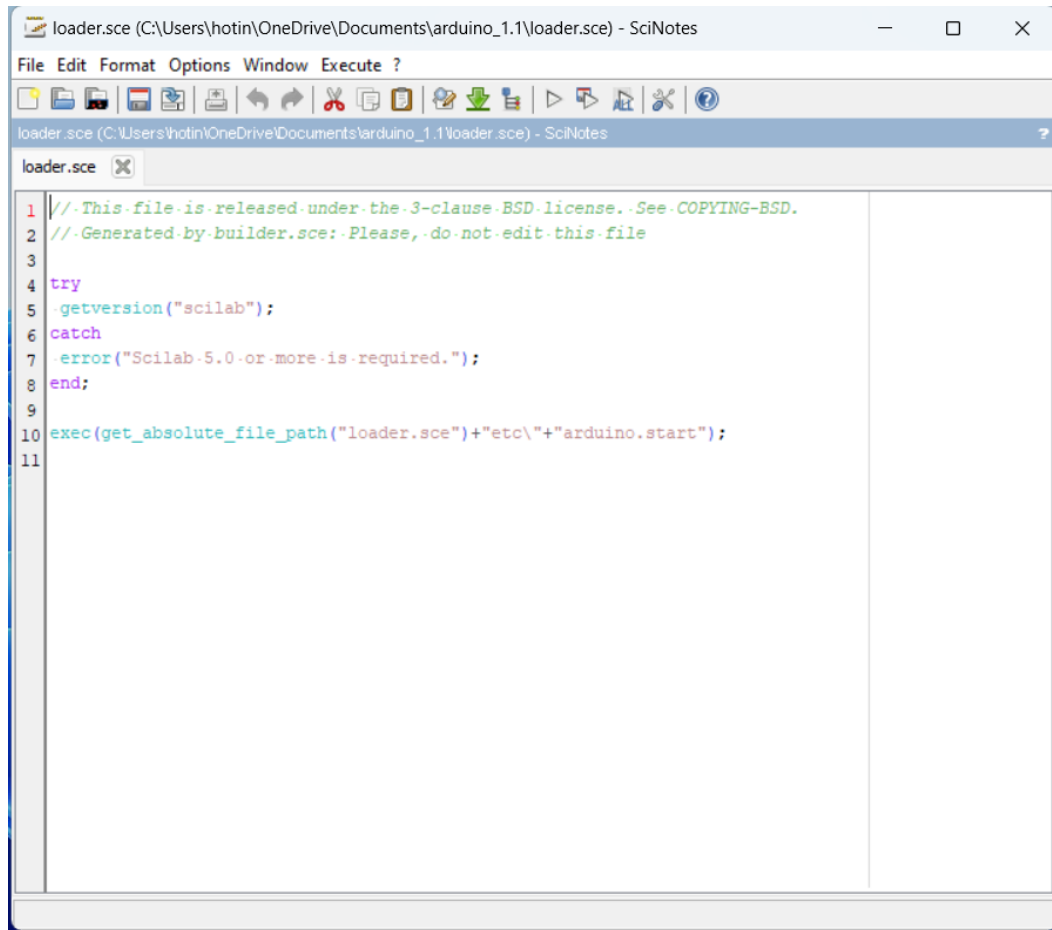
with terminals	110204	110205	110206	110207	110208	110209	110210	110211	110212	110213	110214
with cables	353109	353110	353111	353112	353113	353114	353115	353116	353117	353118	353119

Motor Data												
Values at nominal voltage												
1 Nominal voltage	V	2.4	3.6	6	7.2	9	12	15	18	24	30	36
2 No load speed	rpm	3890	5190	4090	4060	4020	4440	3530	3640	4510	4680	4520
3 No load current	mA	677	69.9	29.2	24	19	16.5	9.41	8.2	8.45	7.16	5.67
4 Nominal speed	rpm	3460	4640	2940	2650	2620	3030	2070	2180	3060	3210	3050
5 Nominal torque (max. continuous torque)	mNm	4.53	5.08	11.3	13.3	13.4	13.2	12.9	12.9	12.8	12.6	12.5
6 Nominal current (max. continuous current)	A	0.84	0.84	0.84	0.814	0.647	0.529	0.33	0.284	0.262	0.214	0.171
7 Stall torque	mNm	35.9	44.1	39.2	36.1	38.2	41.4	31.4	32.5	40.1	40.3	38.5
8 Stall current	A	6.15	6.71	2.83	2.27	1.8	1.62	0.783	0.697	0.797	0.665	0.513
9 Max. efficiency	%	81	81	81	81	81	81	80	80	81	81	81
Characteristics												
10 Terminal resistance	Ω	0.39	0.536	2.12	3.17	4.99	7.41	19.2	25.8	30.1	45.1	70.2
11 Terminal inductance	mH	0.0402	0.0509	0.227	0.332	0.528	0.77	1.9	2.57	2.99	4.34	6.68
12 Torque constant	mNm/A	5.84	6.57	13.9	16.8	21.2	25.5	40.1	46.7	50.3	60.6	75.2
13 Speed constant	rpm/V	1640	1450	689	569	451	374	238	205	190	158	127
14 Speed/torque gradient	rpm/mNm	109	119	105	108	106	108	114	113	114	117	119
15 Mechanical time constant	ms	16.6	16.1	15	14.9	14.9	14.9	14.9	14.9	14.9	15	15
16 Rotor inertia	gcm ²	14.4	12.9	13.6	13.2	13.3	13.1	12.6	12.6	12.5	12.2	12.1



maxon Modular System		Details on catalog page 38	
Planetary Gearhead Ø26 mm 0.75 - 4.5 Nm Page 390 Spur Gearhead Ø30 mm 0.07 - 0.2 Nm Page 391 Planetary Gearhead Ø32 mm 0.75 - 6.0 Nm Page 392/393/396 Spur Gearhead Ø38 mm 0.1 - 0.6 Nm Page 404 Screw Drive Ø32 mm Page 426-433		Encoder MR 128 - 1000 CPT, 3 channels Page 478 Encoder Enc 22 mm 100 CPT, 2 channels Page 483 Encoder HED_5540 500 CPT, 3 channels Page 487/489	Recommended Electronics: Page 38 Notes ESCON Module 24/2 500 ESCON 36/2 DC 500 EPOS4 Micro 24/5 509 EPOS4 Mod./Comp. 50/5 510 EPOS4 Comp. 24/5 3-axes 511 EPOS4 Mod./Comp. 24/1.5 512 EPOS4 50/5 515 EPOS2 P 24/5 520

Loader Scilab



The image shows a screenshot of a SciNotes application window. The title bar reads "loader.sce (C:\Users\hotin\OneDrive\Documents\arduino_1.1\loader.sce) - SciNotes". The menu bar includes "File", "Edit", "Format", "Options", "Window", "Execute", and "?". The toolbar contains icons for file operations (new, open, save, print, copy, paste, delete) and execution (run, stop, refresh). The main text area contains the following code:

```
1 // This file is released under the 3-clause BSD license. See COPYING-BSD.
2 // Generated by builder.sce: Please, do not edit this file
3
4 try
5   getversion("scilab");
6 catch
7   error("Scilab 5.0 or more is required.");
8 end;
9
10 exec(get_absolute_file_path("loader.sce")+"etc"+"arduino.start");
11
```