

LAPORAN PRAKTIKUM MODUL 3
KONTROL POSISI MOTOR DC
TF4022 – Sistem Kontrol Diskrit

Hari, Tanggal Praktikum : Selasa, 31 Oktober 2023

Kelompok: 10

Nama Anggota:

13320026 Yosep Putra Setiyanto

13320032 Dewi Ashiila Hikmawan

13320045 Zayna Mosa Kalila

13320052 Daffa Auliya Ulhaq Sulistiyani

1. Latar Belakang

Kontrol posisi tidak kalah penting dari kontrol kecepatan. Hal ini dibuktikan dari banyaknya aplikasi dari pengendalian posisi di industri, seperti pengendalian bukaan *valve*, pintu otomatis, printer dan 3-D *printing*, CNC, dan lain-lain. Pada modul sebelumnya, praktikan melakukan kontrol pada sistem kecepatan motor DC dengan metode penalaan Ziegler Nichols, Cohen Coon, hingga penalaan halus, lalu diperoleh model dari sistem kecepatan.

Pada praktikum modul 3 ini, praktikan menurunkan model sistem posisi dari sistem kecepatan yang ada untuk kemudian dilakukan implementasi pengontrolan posisi dan pengujian sistem kontrol posisi dalam keadaan tidak diberi gangguan ataupun diberi gangguan.

2. Tujuan Praktikum

Tujuan yang ingin dicapai pada modul praktikum ini adalah sebagai berikut:

1. Menurunkan model sistem posisi dari sistem kecepatan yang telah didapat dari percobaan modul sebelumnya.
2. Melakukan kontrol pada sistem posisi motor DC pada keadaan tanpa gangguan dan dengan gangguan.

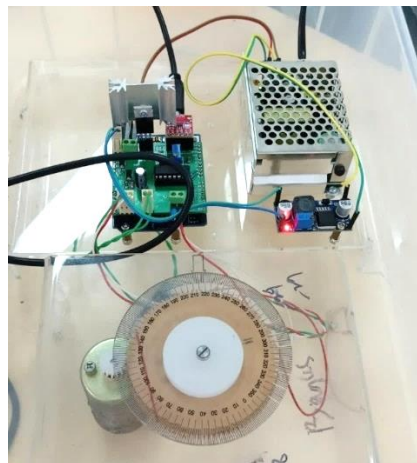
3. Alat dan Bahan

Tabel 3-1 Alat dan Bahan

No	Alat dan bahan	Jumlah	Keterangan
1	Laptop	1	Disiapkan sendiri
2	Kit Praktikum	1	Disediakan asisten
3	Modul Praktikum Ketiga	1	Disediakan, dapat diunduh di kuliah2021 dan Edunex
4	Dokumentasi Kit Praktikum	1	Disediakan, dapat diunduh di kuliah2021 dan Edunex
5	Software Arduino IDE	1	Instal sendiri, unduh di https://www.arduino.cc/en/Main/Software
6	Software MATLAB	1	Dapat menggunakan Matlab <i>online</i>
7	Program SKD23.ino	1	Disediakan, dapat diunduh di kuliah2021 dan Edunex
8	Program Simulasi.m	1	Disediakan, dapat diunduh di kuliah2021 dan Edunex
9	Colokan	1	Untuk menambahkan cabang listrik

4. Prosedur Praktikum

Kit praktikum berupa rangkaian motor DC dengan mikroprosesor Arduino sebagai pengendali. Terdapat juga potensiometer dan tachometer. Kit praktikum disambungkan dengan sumber daya listrik sendirinya. Arduino kemudian disambungkan dengan laptop untuk pengambilan data dan pemrograman. Kit praktikum ini menggunakan *library* Arduino Adafruit_MCP4725. Kit yang dipakai adalah kit nomor 02 milik kelompok 2 karena kit kami (kit 10) rusak.



Gambar 4-4-1 Kit Praktikum Motor DC

Praktikum Sistem Kontrol Diskrit modul ke-3 dibagi menjadi 3 bagian, yaitu pemodelan sistem dan simulasi, persiapan sistem kontrol, dan implementasi kontrol pada motor.

Percobaan 1 : Pemodelan Sistem dan Simulasi

Pemodelan sistem posisi dilakukan dengan cara menurunkan model sistem kecepatan serta hubungan antara sensor kecepatan (tachometer) dengan sensor posisi (potensiometer) pada kit praktikum. Beberapa hal yang perlu diperhatikan dalam tahap ini adalah sebagai berikut:

- Perbandingan roda gigi motor terhadap roda gigi potensiometer adalah 16:100.
- Bila kecepatan motor dalam rotasi per menit (rpm) tidak diketahui, asumsikan kecepatan putar motor 100% adalah 6400 rpm.
- Sistem diasumsikan tidak ada slip antara as dan roda gigi.

Berikut model sistem kecepatan dan posisi kit praktikum Sistem Kontrol Diskrit:

Tabel 4-1 Model Sistem Kecepatan dan Posisi Kit Praktikum 10 Sistem Kontrol Diskrit

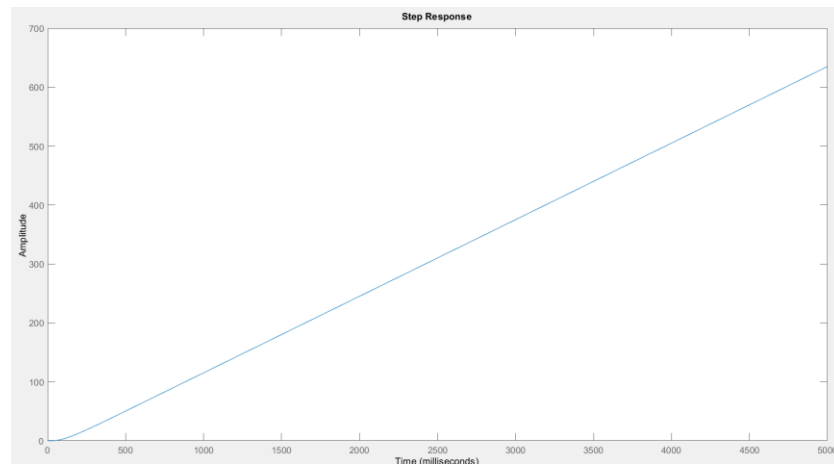
Parameter	Nilai	Satuan	Model Kecepatan	Model Posisi
Waktu cacah	0,0250	detik	$\frac{0,8123 e^{-0,029 s}}{0,0849s + 1}$	$\frac{16 * 0,8123 e^{-0,029 s}}{100(0,0849s^2 + s)}$
Gain	0,8123			
Konstanta waktu	0,0849			
Waktu tunda	0,0291	detik		

Karena bacaan potensiometer pada kit 10 memiliki performa yang buruk, oleh karena itu diputuskan untuk menggunakan kit nomor yang lain, yaitu kit nomor 2. Kit tersebut dilakukan kembali identifikasi sistem dari awal. Tahap-tahap identifikasi dilakukan seperti pada modul 1 dan modul 2 sebelumnya. Berikut model sistem kecepatan dan posisi Kit Praktikum 2 Sistem Kontrol Diskrit.

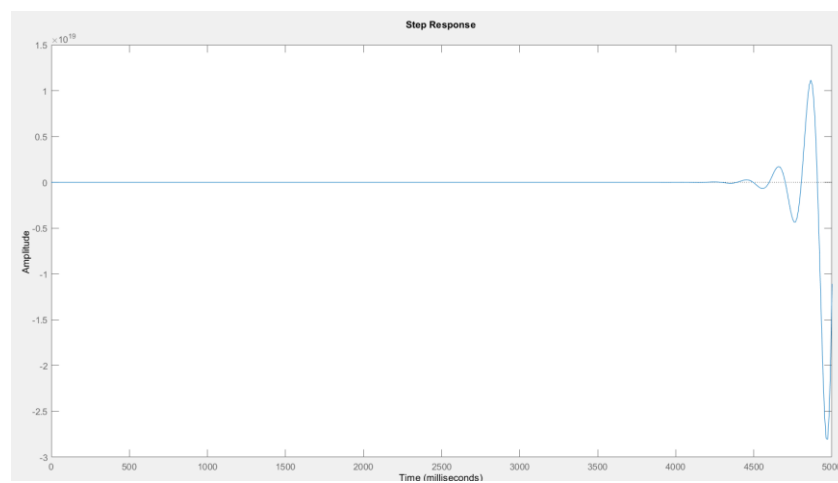
Tabel 4-2 Model Sistem Kecepatan dan Posisi Kit Praktikum 02 Sistem Kontrol Diskrit

Parameter	Nilai	Satuan	Model Kecepatan	Model Posisi
Waktu cacah	0,0100	detik	$\frac{0,8618 e^{-0,0193 s}}{0,0797s + 1}$	$\frac{16 \times 0,8618 e^{-0,0193 s}}{100(0,0797s^2 + s)}$
Gain	0,8618			
Konstanta waktu	0,0797			
Waktu tunda	0,0193	detik		

Program simulasi.m dimodifikasi sehingga program tersebut mensimulasikan sistem kalang terbuka dan tertutup dari sistem posisi.



Gambar 4-2 Simulasi respons step sistem awal kalang terbuka



Gambar 4-3 Simulasi respons step sistem awal kalang tertutup

Berdasarkan gambar di atas, terlihat respon step sistem posisi telah berhasil disimulasikan. Pada kalang terbuka, dengan diberikan respon step yang merupakan kumpulan respon impuls, maka respon akan menunjukkan hasil yang terus meningkat seiring waktu. Sedangkan pada kalang tertutup respon sesuai dengan gambar di atas.

Penalaan kontrol menggunakan metode kalang tertutup Ziegler Nichols dilakukan berdasarkan rumusan berikut ini.

Table 4-1 Rumus parameter kontrol kalang tertutup Ziegler Nichols

Controller	P	T_m	D	Controller	P	I_m	D
P only	$0.5 P_{crit}$	–	–	PID original	$0.6 P_{crit}$	$0.5 \tau_{crit}$	$0.125 \tau_{crit}$
PI	$0.45 P_{crit}$	$0.833 \tau_{crit}$	–	PID some overshoot	$0.33 P_{crit}$	$0.5 \tau_{crit}$	$0.33 \tau_{crit}$
PID	$0.6 P_{crit}$	$0.5 \tau_{crit}$	$0.125 \tau_{crit}$	PID no overshoot	$0.2 P_{crit}$	$0.3 \tau_{crit}$	$0.5 \tau_{crit}$

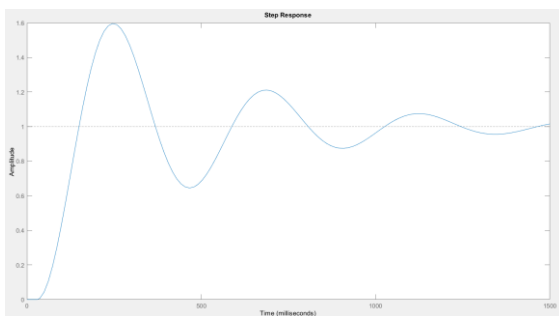
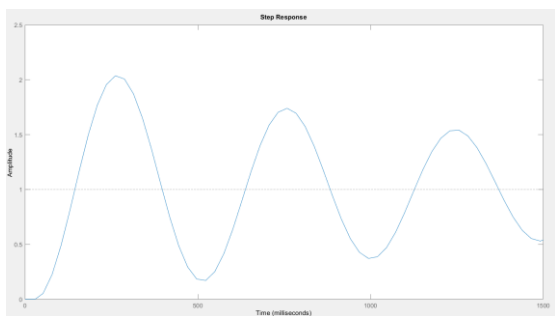
Penentuan P_{crit} dan juga T_{crit} akan dijelaskan lebih lanjut pada bagian analisis. Berikut ini parameter Kontrol Kalang Tertutup Ziegler Nichols yang didapatkan untuk disimulasikan :

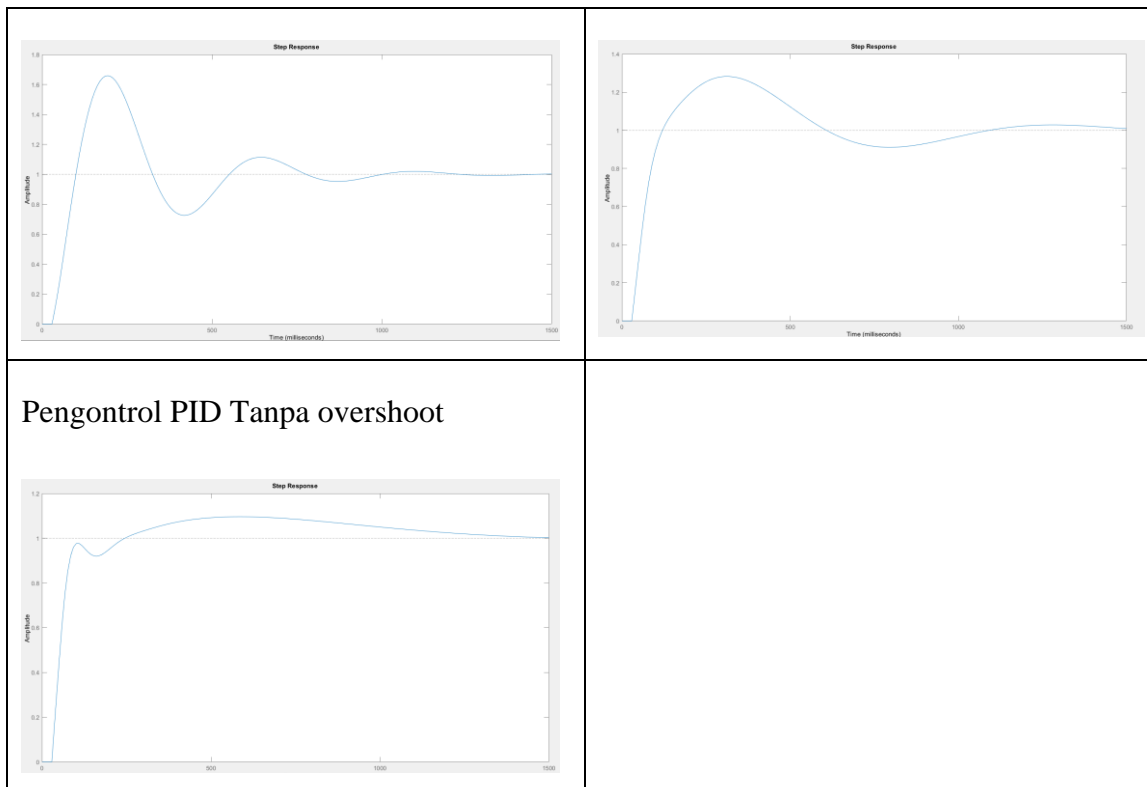
Tabel 4-3 Parameter Kontrol Kalang Tertutup Ziegler Nichols yang digunakan

No	Kontrol	K_p	T_i	T_d	K_i	K_d
1	P	0.152	-	-	-	-
2	PI	0.1368	260.729	-		-
3	PID	0.1824	156.5	39.125	0.001165	7.1364
4	PID <i>Overshoot</i> kecil	0.10032	156.5	103.29	0.000641	10.36205
5	PID tanpa <i>Overshoot</i>	0.0608	93.9	156.5	0.000647	9.5152
	P_{crit}	0.304	0.304	0.304	0.304	0.304
	T_{crit}	313	313	313	313	313

Berikut ini merupakan hasil simulasi respon step sistem yang telah diberi sistem kontrol sesuai dengan parameter yang digunakan di atas.

Table 4-2 Hasil Simulasi Kontrol untuk berbagai jenis pengontrol

<p>Pengontrol P</p> 	<p>Pengontrol PI</p> 
<p>Pengontrol PID</p>	<p>Pengontrol PID <i>Overshoot</i> Kecil</p>



Berikut hasil identifikasi performa kontrol untuk tiap parameter kontrol:

Tabel 4-4 Performa Simulasi Kontrol Posisi

No	Kontrol	Rise Time (ms)	Overshoot (%)	Settling Time (ms)	peak
1	P	80.2192	59.5923	1613.7000	1.5959
2	PI	77.3723	103.6400	6631.7000	2.0364
3	PID	54.6749	65.8900	958.9792	1.6589
4	PID <i>Overshoot</i> kecil	65.1556	28.2461	1405.3000	1.2825
5	PID tanpa <i>Overshoot</i>	48.4035	9.6149	1242.6000	1.0961

Berdasarkan performa hasil simulasi tersebut, Pengontrol P masih memiliki offset yang relatif besar, sehingga masih kurang efektif untuk digunakan. Untuk pengontrol PI justru hasil simulasi menunjukkan adanya osilasi yang semakin besar hingga 103.64% overshoot. Hal tersebut dikarenakan pada sistem telah memiliki komponen integrator ($1/s$) untuk posisi, sehingga dengan ditammbahkannya pengontrol Integral akan menimbulkan 2 kali integrasi sehingga osilasi semakin besar. Untuk Pengontrol PID, osilasi dari

pengontrol integral diredam sedikit oleh pengontrol derivatif, sehingga overshoot menjadi lebih rendah.

Untuk pengontrol PID dengan overshoot kecil, nilai overshoot semakin dioptimasi, kali ini menunjukkan hasil overshoot hanya 28%. Dan yang terakhir PID tanpa overshoot berusaha menghilangkan overshoot, namun sepertinya dengan tetap adanya pengontrol Integral yang menimbulkan dua kali integrasi, overshoot masih tetap ada. namun begitu, dari hasil didapatkan overshoot yang cukup rendah yaitu hanya 9% saja.

Percobaan 2: Persiapan Sistem Kontrol

Pada percobaan ini, kami memeriksa pembacaan sensor posisi serta melakukan percobaan untuk kalang terbuka. Tab *compensation* dimodifikasi dengan tujuan untuk mengamati respon posisi. Fungsi diatur agar masukan rentang -1 hingga 1 motor tidak akan bergerak:

```
float deadzoneCompensator(float inputSerial){
    float diamTolerance = 3;
    float inputCompensated;
    if(inputSerial <= 1 & inputSerial >= -1){
        inputCompensated = 0;
    }
    if(inputSerial < ((-1)*diamTolerance)){
        inputCompensated = (minimumTerdeteksiNegatif)+((inputSerial-
(diamTolerance))/((-100)-0))*((-100)-(minimumTerdeteksiNegatif));
    }
    else if (inputSerial > diamTolerance){
        inputCompensated = (minimumTerdeteksiPositif)+((inputSerial-
(diamTolerance))/((100)-0))*((100)-(minimumTerdeteksiPositif));
    }
    else{
        inputCompensated = 0;
    }
    return inputCompensated;
}
```

Program disiapkan dalam kondisi kalang terbuka terkompensasi untuk mengamati posisi. Kami memberikan nilai masukan -100 hingga 100:

Tabel 4-5 Klasifikasi Pengamatan Posisi Terhadap Masukan

No	Rentang Masukan	Klasifikasi Pengamatan
1	$-40 \leq \text{Masukan} \leq -5$	Hasil pembacaan posisi stabil dan posisi mudah diamati
2	$-100 \leq \text{Masukan} < -40$	Hasil pembacaan posisi tidak stabil dan posisi sulit diamati
3	$5 \leq \text{Masukan} \leq 40$	Hasil pembacaan posisi stabil dan posisi mudah diamati
4	$40 < \text{Masukan} \leq 100$	Hasil pembacaan posisi tidak stabil dan posisi sulit diamati

Terlihat bahwa saat putaran motor semakin cepat, semakin sulit untuk mengamati posisi. Tab *control* dimodifikasi untuk mengendalikan posisi dengan memperhatikan dua batasan berikut:

- Nilai error berada dalam rentang -50 hingga 50, karena apabila pembacaan posisi lebih dari 100 maka akan kembali ke 0 dan sebaliknya, apabila pembacaan posisi kurang dari 0 akan langsung ke 100.
- Berikan batas pada aksi kontrol (hasil perhitungan kontrol) pada rentang yang diinginkan sesuai dengan klasifikasi pada tabel 4-6.

Berikut kode tab control kami:

```
int controlPID(float feedback, float setPoint){
    Error = setPoint - feedback;
    if(Error > 50){
        Error -= 100;
    }
    else if (Error < -50){
        Error += 100;
    }
    else{
        Error = Error;
    }
    sumError = sumError + (float)(Error*(float)((float)TimeSampling/1));

    P = kp*Error;
    I = ki*sumError;
    D = kd*(Error-LastError)/(float)((float)TimeSampling/1);

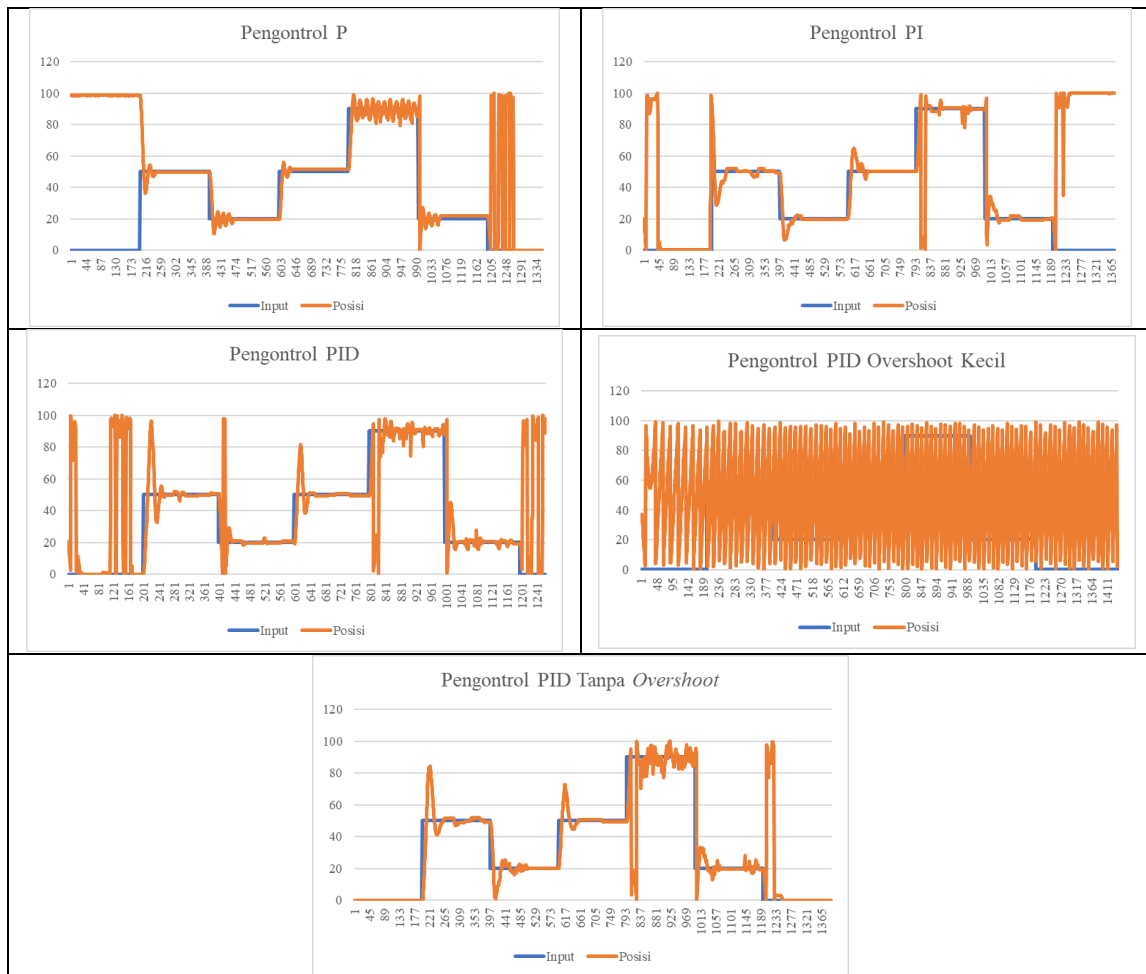
    LastError = Error;

    float output = gain*(P+I+D);
    if (output>40.0) {
        output = 40.0;
    }
    else if (output<-40.0) {
        output = -40.0;
    }
    return deadzoneCompensator(output);
}
```

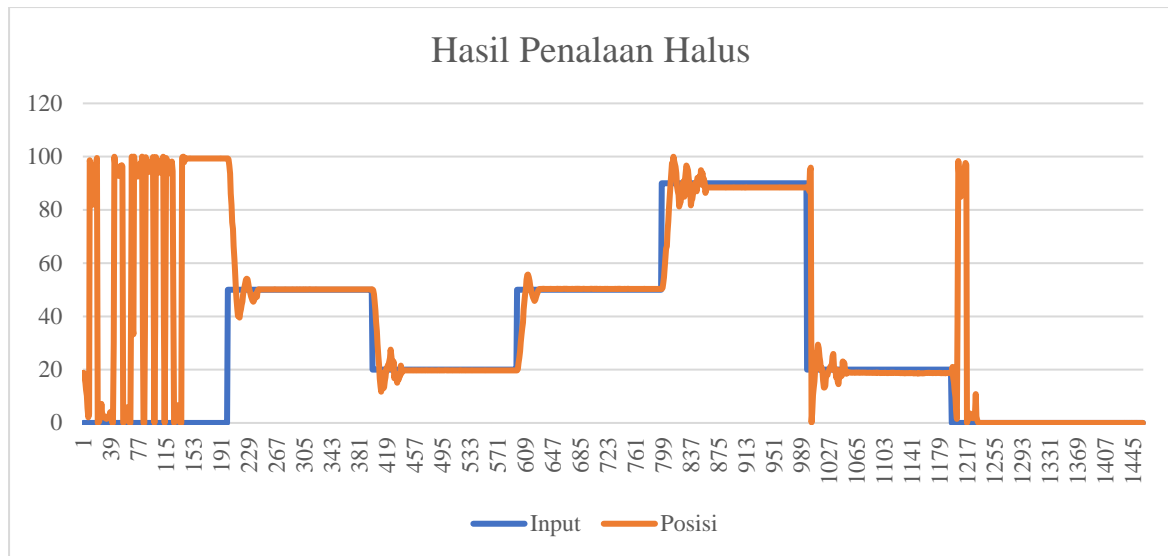

Percobaan 3: Implementasi Kontrol

Pada bagian ini, implementasi kontrol dilakukan. Sistem posisi pada kit praktikum Sistem Kontrol Diskrit sangat sensitif terhadap putaran motor. Pertama, kontrol posisi dilakukan dengan *set point* 50, 20, 50, 90, dan 20. Di antar *set point* diberikan jeda sekitar dua detik lebih. Berikut grafik dari percobaan untuk kontroler P, PI, PID, PID dengan *overshoot* rendah, serta PID tanpa *overshoot*:

Tabel 4-6 Grafik Hasil Implementasi Kontrol P-PI-PID Posisi pada Kit

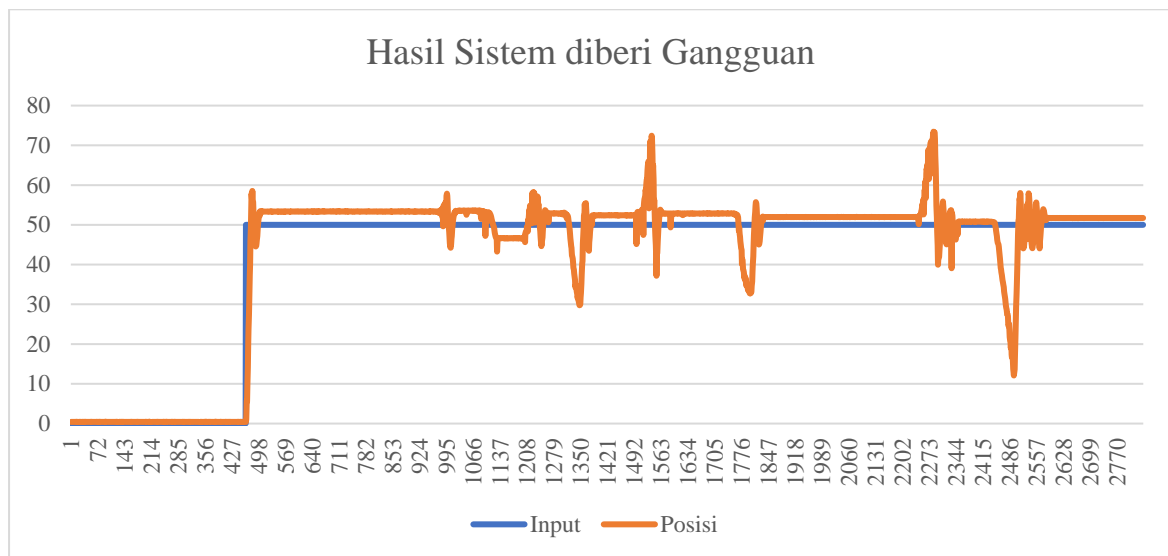


Kami mengubah nilai salah satu jenis kontrol yang menghasilkan performa terbaik untuk melakukan penalaan halus. Hasil dari penalaan halus sudah cukup baik. Berikut grafik hasil dari penalaan halus:



Gambar 4-4 Grafik Hasil Implementasi Kontrol Posisi Penalaan Halus pada Kit

Setelah itu, kami memberikan gangguan pada kit dengan cara memutar potensiometer secara manual saat sedang berjalan pada *set point* 50:



Gambar 4-5 Grafik Hasil Implementasi Kontrol Posisi dengan Gangguan pada Kit

Data yang didapatkan diolah di MATLAB. Berikut hasil performa implementasi kontrol posisi yang dilakukan:

Tabel 4-7 Performa Implementasi Kontrol Posisi

No	Kontrol	Delay Time	Rise Time	Overshoot	Settling Time	Offset
1	P	0,0344	0,0551	19,1416	0,6224	18,81
2	PI	0,0279	0,0446	34,2944	0,9867	17,62
3	PID	0,0551	0,0882	44,8306	1,5017	13,40
4	PID Overshoot kecil	0,0165	0,0264	664,5202	NaN	28,79

5	PID tanpa <i>Overshoot</i>	0,0308	0,0493	43,6322	1,0705	3,11
6	Penalaan Halus	0,0481	0,0769	17,3740	0,6154	11,46

5. Analisis Hasil Percobaan

Percobaan 1: Pemodelan Sistem dan Simulasi

Program simulasi.m dimodifikasi sehingga program tersebut mensimulasikan sistem kalang terbuka dan tertutup dari sistem posisi. Dalam penurunan model sistem posisi dari sistem kecepatan, pembacaan 100% tachometer adalah 6400 RPM. 1% dari 6400 RPM adalah 64 RPM atau 64/60 *rotary per second*. Karena rasio roda gigi adalah 16:100, pembacaan potensiometer sekitar 100/16 kali. Sehingga faktor konversi sebesar 0,0066. Persamaan sudut adalah sebagai berikut:

$$\Delta\theta = 0,0066 \omega_{tachometer}$$

Kemudian karena pada modul sebelumnya telah dibuat model kecepatan dari sistem, maka untuk membuat model posisi kita perlu melakukan integral dari model kecepatan. Pada sistem diskrit integral dapat dilakukan dengan sumasi, sedangkan pada laplace domain, integral dapat dilakukan dengan mengalikan model dengan 1/s.

Untuk menentukan Pcrit dan juga Tcrit, dilakukan karakteristik sistem yang akan dikontrol. Salah satu cara untuk melakukan karakteristik sistem adalah dengan menggunakan Root Locus.

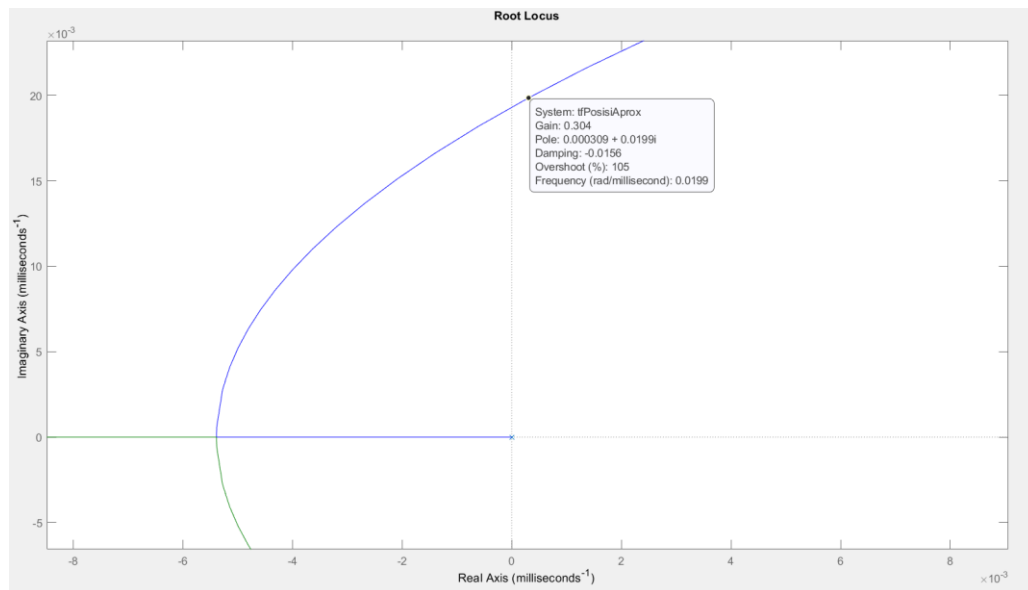
```

4   gain = 0.8123*1;
5   tau = 84.9/1;
6   timeDelay = 29.1/1;
7   timeSampling = 10/1;
8
9   tfPosisi = tf(gain*16/100,[tau 1], 'InputDelay',timeDelay)*tf(1,[1 0]);
10  tfPosisi.TimeUnit = 'milliseconds';
11
12  figure()
13  step(tfPosisi)
14  figure()
15  tfPosisiAprox = pade(tfPosisi);
16  tfPosisiAprox.TimeUnit = 'milliseconds';
17  rlocus(tfPosisiAprox)

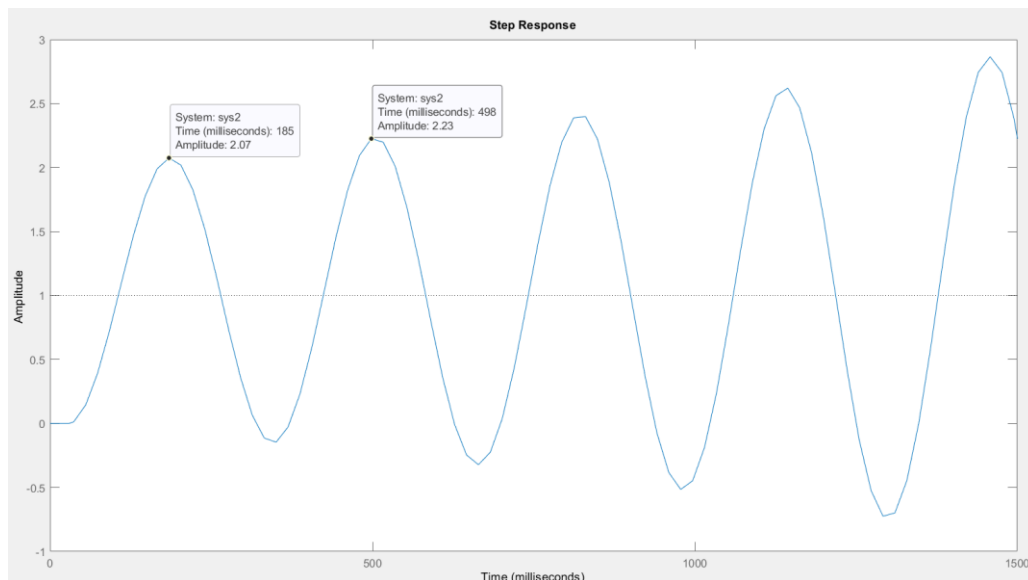
```

Berdasarkan kode yang digunakan seperti pada gambar di atas, pada baris 1-7 akan dilakukan input identifikasi sistem. Pada baris 9-10 dilakukan pemodelan sistem untuk simulasi. Kemudian pada baris 15-16 dilakukan aproksimasi sistem orde satu dengan time delay menjadi sistem dengan orde lebih tinggi tanpa time delay. Dan terakhir pada baris 17 dilakukan plot root locus dari pemodelan sistem yang telah dibuat.

Untuk menentukan P_{crit} , dapat dilakukan dengan mencari gain yang dapat membuat sistem mengalami osilasi 100%. Berdasarkan gambar dibawah ini, didapatkan P_{crit} sebesar 0.304



Untuk mendapatkan T_{crit} (critical period), periode dari satu gelombang akan dihitung pada kondisi osilasi berkelanjutan (pada saat nilai critical gain). Dapat dilihat gambar berikut untuk respon sistem pada kondisi osilasi berkelanjutan (nilai critical gain). Didapatkan nilai T_{crit} sebesar $498 - 185 = 313$ ms.



Nilai P_{crit} dan T_{crit} tersebut, selanjutnya pada kode akan digunakan untuk parameter simulasi sistem kontrol sesuai kode berikut ini.

```

% P
kp = 0.5*Pcrit;
ti = 0;
ki = 0;
td = 0;
kd = 0;

% PI
kp = 0.45*Pcrit;
ti = 0.833*tcrit;
ki = kp/ti;
td = 0;
kd = kp*td;

% PID
kp = 0.6*Pcrit;
ki = kp/(0.5*tcrit);
kd = kp*0.125*tcrit;

% PID Overshoot Kecil
kp = 0.33*Pcrit;
ki = kp/(0.5*tcrit);
kd = kp*0.33*tcrit;

% PID Tanpa Overshoot
kp = 0.2*Pcrit;
ki = kp/(0.3*tcrit);
kd = kp*0.5*tcrit;

```

Dilakukan simulasi kontrol menggunakan kode berikut ini.

```

%Fungsi Pengontrol
PID = kp+ki/s+kd*s;
PID.TimeUnit = 'milliseconds';
sys2 = (feedback(sys1*PID,1));
sys2.TimeUnit = 'milliseconds';
setpoint = 1;
hold off;
figure(2);
step(sys2);
xlim([0 1500]);
stepinfo(sys2)
hold off;

```

Hasil pengontrol dan identifikasi performa kontrol diberikan pada bagian 3 sebelumnya.

Percobaan 2: Persiapan Sistem Kontrol

Untuk mengubah tab kompensasi, variabel bacaan fungsi diubah dari keluaran pin tachometer menjadi keluaran pin potensiometer. Potensiometer adalah bagian yang mengatur putaran motor pada kit praktikum. Oleh karena itu, untuk melakukan

pengontrolan posisi yang membutuhkan konversi kinerja kode program, nilai keluaran potensiometer harus dimasukkan ke dalam fungsi kontrol. Selain itu, modifikasi dilakukan untuk mengkompensasi karakteristik zona mati, sehingga motor tidak berputar pada rentang nilai -1 hingga 1 dan masukan dalam rentang nilai tersebut dapat dibaca dengan baik dan menghasilkan hasil yang diinginkan.

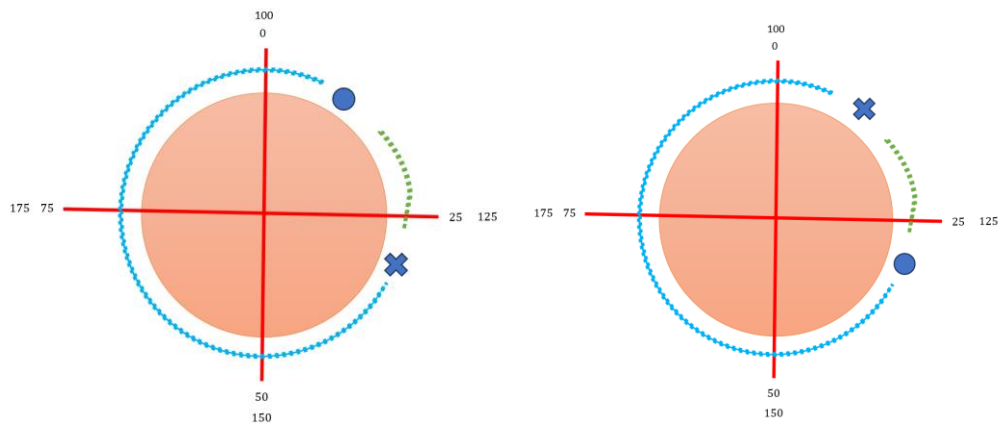
Karena komponen untuk kontrol sistem kecepatan berbeda dengan komponen untuk kontrol sistem posisi, pengamatan posisi harus dilakukan dengan mengambil data transien dari komponen potensiometer. Dengan rentang masukan -100 hingga 100 dan melihat grafik hasilnya. Hal ini diperlukan untuk mengkompensasi zona mati komponen potensiometer dengan nilai masukan tegangan mikrokontroler.

Selain melakukan berbagai modifikasi, ada juga pengamatan pada rentang nilai masukan yang diberikan dan klasifikasi berdasarkan pengamatan terhadap keluaran potensiometer dengan melihat grafik yang dihasilkan oleh Serial Plotter. **Tabel 4-6** menampilkan klasifikasi pengamatan.

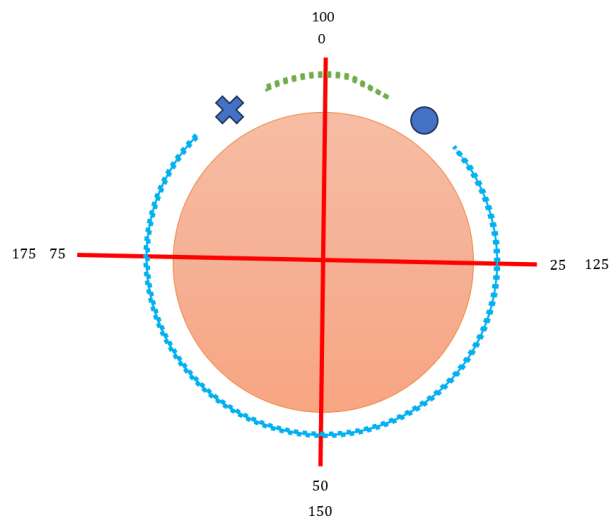
Hasil pengamatan menunjukkan bahwa rentang masukan -40 sampai -5 memiliki posisi yang stabil dan mudah diamati, masukan -100 sampai -40 memiliki posisi tidak stabil dan sulit diamati, sedangkan rentang 5 sampai 40 pembacaan posisi stabil dan mudah diamati, dan masukan 40 sampai 100 memiliki pembacaan posisi tidak stabil dan sulit diamati.

Modifikasi yang dilakukan pada tab control bertujuan mengendalikan posisi. pemodifikasian yang dilakukan berupa nilai error berada dalam rentang -50 hingga 50, karena apabila pembacaan posisi lebih dari 100 maka akan kembali ke 0 dan sebaliknya, apabila pembacaan posisi kurang dari 0 akan langsung ke 100.

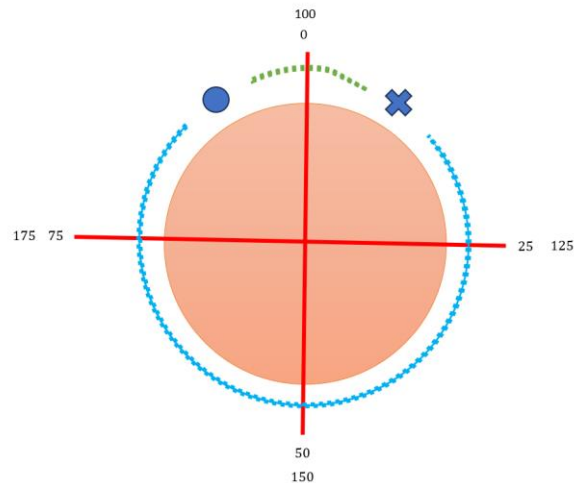
Perhatikan gambar berikut ini, tanda X merupakan set point atau tujuan yang diinginkan, sedangkan tanda O merupakan posisi saat ini atau feedback yang diberikan sensor.



Pada gambar tersebut, ketika (setpoint lebih besar dari feedback) dan (setpoint dikurangi feedback dibawah 50), atau ketika (feedback lebih besar dari setpoint) dan (feedback dikurangi setpoint masih dibawah 50), maka error nya agar motor bergerak mengikuti lintasan terpendek adalah $\text{Error} = \text{Setpoint} - \text{Feedback}$.



Pada gambar tersebut, ketika (setpoint lebih besar dari feedback) dan (setpoint dikurangi feedback ternyata lebih dari 50), maka error nya agar motor bergerak mengikuti lintasan terpendek adalah $\text{Error} = - (100 - (\text{Setpoint} - \text{Feedback}))$.



Pada gambar tersebut, ketika (feedback lebih besar dari setpoint) dan (feedback dikurangi setpoint ternyata lebih dari 50), maka error nya agar motor bergerak mengikuti lintasan terpendek adalah $\text{Error} = 100 - (\text{Setpoint} - \text{Feedback})$.

Percobaan 3: Implementasi Kontrol

Hasil pengontrolan menggunakan parameter pengontrol dari metode kalang terbuka Ziegler Nichols menunjukkan hasil yang cukup baik. Namun sistem tidak dapat mencapai posisi dengan nilai 0 dengan baik, *offset* pada nilai tersebut sangat tinggi serta berosilasi. Hal ini dapat disebabkan oleh batasan fisik pada aktuator (motor) yang mencegahnya mencapai *set point* yang diinginkan atau pemodelan posisi yang tidak cocok dengan kondisi aktual. Jika respons sistem saat *set point* = 0 tersebut diabaikan, dapat dikatakan rata-rata *offset* sistem hanya sekitar 1-2.

PID dengan *overshoot* kecil sangat agresif, tidak stabil, serta tidak mencapai *set point* sama sekali. Hal ini dapat disebabkan oleh nilai K_p , T_i , dan T_d yang masih belum sesuai dengan karakteristik dinamika sistem kit praktikum atau disebabkan oleh nilai *sampling time* yang digunakan tidak cocok dengan PID dengan *overshoot* kecil. Kit yang digunakan memiliki respons lebih sensitif saat diimplementasikan metode ini. Sedangkan dari kelima pengontrol, PID tanpa *overshoot* memiliki parameter performa paling baik namun memiliki settling time paling lama (1,84 detik). Metode *overshoot* kecil cenderung menghasilkan K_p yang lebih tinggi dibandingkan dengan metode tanpa *overshoot*. K_p yang lebih tinggi membuat sistem kontrol menjadi lebih agresif, yang meningkatkan osilasi dan *overshoot*. Metode *overshoot* kecil juga menghasilkan *damping ratio* lebih rendah dibandingkan dengan metode tanpa *overshoot*. Dalam metode Ziegler-

Nichols *overshoot* kecil, K_i dibuat menjadi lebih tinggi untuk mencapai tingkat *overshoot* yang diinginkan. Peningkatan ini dapat menyebabkan osilasi.

Saat dilakukan penalaan halus (Gambar 4-4), osilasi pada sistem berkurang, hal ini dapat dilihat pada grafik dan *settling time* yang paling rendah dibandingkan dengan lima kontroler sebelumnya. Namun penalaan halus ini meningkatkan nilai *delay time* dan *rise time*. Sedangkan dari percobaan kit diberikan gangguan manual (Gambar 4-5), respons sistem cukup baik dan selalu balik ke set point setelah diberikan gangguan walaupun berosilasi terlebih dahulu.

Secara keseluruhan, implementasi kontrol posisi sudah cukup baik. Kontroler yang memiliki performa terbaik adalah kontroler dengan penalaan halus. Namun sebaiknya tetap dilakukan modifikasi tab control lebih lanjut untuk mengurangi error pada set point = 0.

Tabel 5-1 Performa Terbaik Hasil Implementasi Kontrol Posisi

<i>Delay Time</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Offset</i>
0,0481	0,0769	17,3740	0,6154	11,46

6. Kesimpulan

Pada percobaan 1, model kecepatan sistem, hasil identifikasi dari modul 1 dan modul 2, sebelumnya dikonversikan menjadi modul posisi dengan menggunakan 2 langkah, yaitu integrasi dan faktor konversi. Integrasi dilakukan karena posisi merupakan integrasi dari kecepatan. Integrasi dapat dilakukan dengan sumasi ppada sistem diskrit atau mengalikan $1/s$ pada domain laplace. Faktor konversi ditinjau dari perbandingan roda gigi yang digunakan antara motor dan sensor potensiometer, serta kecepatan (RPM) dari motor. Simulasi respon step kalang terbuka dan tertutup untuk sistem awal menunjukkan kelakuan yang sesuai. Untuk simulasi kontrol, digunakan metode ziegler nichols dengan variasi pengontrol P, PI, PID, PID dengan overshoot kecil, dan PID tanpa overshoot. Untuk menentukan parameter dicari P_{crit} yaitu gain dimana osilasi sistem terbilang besar (tidak stabil) ditandai dengan overshoot yang mendekati 100%, sedangkan T_{crit} merupakan periode osilasi saat sistem tidak stabil. Hasil simulasi kontrol yang dilakukan menunjukkan bahwa pengontrol P masih menimbulkan offset, pengontrol PI memiliki osilasi yang besar, hal tersebut dikarenakan terjadinya dua kali integrasi, pengontrol PID sedikit mengompensasi osilasi pengontrol PI sebelumnya, pengontrol PID dengan sedikit

overshoot semakin mengompensasi osilasi, dan PID tanpa overshoot berusaha mengoptimasi agar sistem tidak memiliki overshoot, namun tetap saja karena adanya dua kali integrasi tetap adanya overshoot namun sangat kecil dibawah 10% saja.

Untuk percobaan 2, pada tab kompensasi, variabel bacaan fungsi diubah dari keluaran pin tachometer menjadi keluaran pin potensiometer, yang mengatur putaran motor. Untuk pengontrolan posisi yang membutuhkan konversi kinerja kode program, nilai keluaran potensiometer dimasukkan ke dalam fungsi kontrol, dengan modifikasi untuk mengkompensasi karakteristik zona mati. Observasi terhadap posisi dilakukan dengan mengambil data transien dari komponen potensiometer, dengan rentang masukan -100 hingga 100. Hasil pengamatan dan klasifikasi menunjukkan bahwa rentang masukan tertentu memiliki posisi stabil atau tidak stabil, yang menjadi dasar untuk modifikasi pada tab control dengan tujuan mengendalikan posisi. Pemodifikasian tersebut melibatkan pengaturan nilai error, batas aksi kontrol, dan penyesuaian rentang nilai masukan, dengan hasil pengamatan yang dijelaskan dalam Tabel 4-6.

Pada percobaan 3, hasil pengontrolan menggunakan parameter pengontrol dari metode kalang terbuka Ziegler Nichols menunjukkan hasil yang cukup baik. Namun sistem tidak dapat mencapai posisi dengan nilai 0 dengan baik, *offset* pada nilai tersebut sangat tinggi serta berosilasi. PID dengan *overshoot* kecil sangat agresif, tidak stabil, serta tidak mencapai *set point* sama sekali. PID tanpa *overshoot* memiliki parameter performa paling baik namun memiliki *settling time* paling lama. Saat dilakukan penalaan halus, osilasi pada sistem berkurang dan *delay time* dan *rise time* bertambah. Saat kit diberikan gangguan manual, respons sistem cukup baik walaupun berosilasi terlebih dahulu. Secara keseluruhan, implementasi kontrol posisi sudah cukup baik. Kontroler yang memiliki performa terbaik adalah kontroler dengan penalaan halus (dengan *delay time* = 0,05 detik, *rise time* = 0,08 detik, *overshoot* = 17%, *settling time* = 0,6 detik, serta *offset* = 12). Namun sebaiknya tetap dilakukan modifikasi tab control lebih lanjut untuk mengurangi error pada *set point* = 0.

7. Tugas Tambahan

Kontrol Diskrit selain PID Konvensional : Linear Quadratic Regulator (LQR)

Pada praktikum modul 1-3, telah dicoba pengontrol konvensional PID (*Proportional, Integral, Derivative*) pada sistem motor DC, namun hasilnya masih belum memiliki respons yang maksimal. Pengontrol PID tidak dirancang untuk optimalitas, tetapi memberikan kontrol yang kuat dan efektif dalam berbagai aplikasi praktis. Untuk itu, sebaiknya dicari metode kontrol yang lebih optimal. Salah satu metode yang menggunakan representasi sistem dengan ruang keadaan (*state space*) adalah metode *Linear Quadratic Regulator* (LQR). *Tuning* pengendali PID menjadi rumit ketika berhadapan dengan beberapa variabel yang saling berinteraksi. Sementara pengendali LQR sangat cocok untuk sistem multivariabel karena menangani sistem ruang keadaan yang saling berhubungan. Dengan demikian, kami melakukan kontrol kecepatan menggunakan metode kontrol diskrit *Linear Quadratic Regulator* (LQR) untuk kit Motor DC Sistem Kontrol Diskrit.

Ruang keadaan adalah persamaan untuk mendeskripsikan dinamika sistem dalam persamaan diferensial atau persamaan diferensi orde 1 (tidak melibatkan turunan kedua). Dengan menggunakan kontroler yang menggunakan ini, diharapkan kontroler dapat menyesuaikan dengan dinamika sistem. LQR dapat efektif menyelesaikan masalah pengendalian pada sistem Multi Input Multi Output (MIMO) sehingga berguna dalam mengatasi kompleksitas sistem modern. LQR dirancang untuk mengatur atau membuat keluaran (y) mendekati nol dengan input minimum. Kontrol LQR ini kami lakukan pada tugas besar SKD 2023.

Pendapat dan Saran Praktikum SKD

Penjelasan modul praktikum sudah jelas dan pembagian tahap materi setiap modul sudah baik. Modul-modul memberikan bayangan implementasi ilmu dan teori yang dipelajari di kelas. Praktikum sangat efektif dalam mendorong mahasiswa untuk belajar. Asisten sangat antusias dan memberikan feedback saat pasca pengumpulan laporan yang menjadi pembelajaran dan wawasan untuk mengembangkan diri. Namun pembagian kelompok pada praktikum ini sebaiknya lebih memerhatikan topik tugas akhir praktikan serta nilai akhir TF3205 Kontrol Otomatik (sebaiknya pembagian mahasiswa yang mengambil topik kontrol dan nilai TF3205 yang tinggi dibagi rata pada setiap kelompok

agar kemampuan kelompok lebih rata dan adil). Kendala pada praktikum adalah banyaknya waktu diperlukan untuk melakukan pemograman dan beban yang lumayan besar (tidak sebanding dengan sks).

Awalnya, kit praktikum cukup memadai untuk pengambilan data dalam praktikum. Namun kit sering terjadi masalah dan menghambat pengambilan data dan menghabiskan banyak waktu dalam *troubleshooting*. Dan jika sudah rusak, banyak waktu terbuang karena harus mengambil data dinamika sistem dari nol. Kerusakan kit yang muncul ini mungkin terjadi akibat percobaan implementasi kontrol yang dilakukan oleh praktikan sendiri namun sebaiknya tersedia lebih banyak kit cadangan sehingga dapat langsung menukar kit jika rusak.

8. Kontribusi

Kelompok 10 SKD 2023/2024 memohon maaf atas keterlambatan dan ketidakdisiplinan kelompok kami dalam mengerjakan modul 3. Sebagian besar dari kelompok kami mengalami kendala pribadi yang menghambat keberjalanan pembelajaran praktikum kelompok kami. Namun kami belajar bahwa hal tersebut tidaklah menjadi alasan untuk menjadi lalai dan seharusnya kami lebih bisa mengatur waktu dan meningkatkan komunikasi kami.

Tabel 8-1 Kontribusi Anggota Kelompok

Nama	NIM	Kontribusi Anggota
Yosep Putra S	13320026	Pemograman Matlab dan Arduino Prosedur percobaan 1, 2, 3 Analisis percobaan 1,2 Kesimpulan percobaan 1, 2 Pengambilan data percobaan 3
Dewi Ashiila H	13320032	Prosedur percobaan 1, 2, 3 Pengolahan data percobaan 3 Analisis percobaan 1, 3 Kesimpulan percobaan 2, 3 Tugas tambahan
Zayna Mosa Kalila	13320045	Latar belakang dan tujuan
Daffa Auliya U S	13320052	Latar belakang dan tujuan

9. Referensi

- [1] K. Ogata , *Modern Control Engineering*, Prentice Hall, 2010.
- [2] N. S. Nise, *Control Systems Engineering*, John Wiley & Sons, 2020.
- [3] E. Ekawati, F. Mukhlis, N. Habibullah, N. Diyah, *Modul 3 : Kontrol Posisi Motor DC*, Praktikum TF4022 Sistem Kontrol Diskrit, 2023.

10. Lampiran

[Data Praktikum TF4022_Kel10_Modul3.xlsx](#)