

MODUL 5 SISTEM KENDALI PID Kasus – P dan D



Mata Kuliah : Sistem Kendali

Dosen : MHI

Kelas Kelompok : D3TK-43-03

Kelompok 7 A2R :

- | | |
|-----------------------|------------|
| 1. Anggi Angga Reksa | 6702190033 |
| 2. Nauval Zaki Ishomi | 6702194027 |

**PROGRAM STUDI D3 TEKNOLOGI KOMPUTER
FAKULTAS ILMU TERAPAN
UNIVERSITAS TELKOM
BANDUNG
2021**

MODUL 5 Sistem Kendali PID : Kasus P dan D

- **JUDUL PRAKTIKUM**

- Sistem Kendali PID Kasus P dan D

1. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa dapat memahami fungsi dan cara kerja PID pada motor DC
2. Mahasiswa dapat membuat program sistem kendali berbasis PID dengan error yang dihubungkan dengan konstanta proporsional dan derivatif

3. PARAMETER PENILAIAN

No.	Parameter	Persentase (%)
1.	Lembar Penilaian Praktikum	40%
2.	Jurnal/Laporan Praktikum	60%

4. PERALATAN DAN BAHAN

Alat dan Bahan :

1. Robot Kit Line Follower
2. Baterai LiPo 2-Cell 1300 mAh
3. Kabel Mini-USB
4. Arduino Nano
5. Battery Checker
6. Battery Balancer

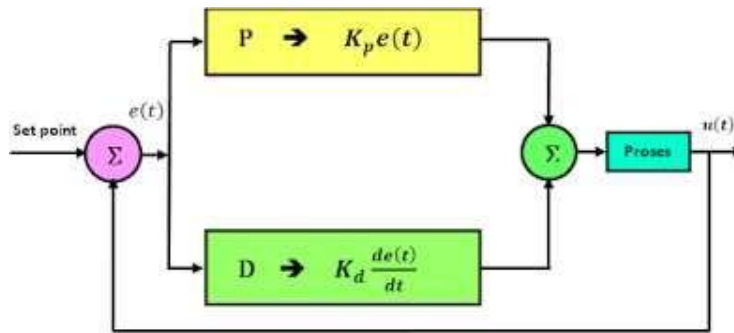
Perangkat Lunak :

1. Software IDE Arduino
2. Software Proteus (untuk simulasi)

5. TEORI DASAR

5.1. Sistem Kendali PD

Teknik kendali proporsional-derivatif (PD) adalah pengendali yang merupakan gabungan antara teknik kendali proporsional (P) dengan teknik kendali derivatif (D). Gambar 1 merupakan gambar diagram blok sistem kendali PD.



Gambar 1 Diagram blok sistem kendali PD.

Persamaan hubungan antara keluaran sistem dengan sinyal error pada kombinasi aksi kendali proporsional-derivative adalah sebagai berikut.

$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t) \quad \text{..... (1)}$$

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad \text{.....(2)}$$

Dalam penerapannya di software, kondisi ideal pada robot adalah bergerak maju lurus mengikuti garis, dengan kata lain error = 0 . Dari sini dapat diasumsikan bahwa Set Point (SP) / kondisi ideal adalah saat SP = 0. Nilai sensor yang dibaca oleh sensor disebut *Process Variable* (PV) / nilai aktual pembacaan. Menyimpangnya posisi robot dari garis disebut sebagai *error* (e), yang didapat dari $e = SP - PV$. Dengan mengetahui besar *error*, mikrokontroler dapat memberikan nilai PWM motor kiri dan kanan yang sesuai agar dapat menuju ke posisi ideal (SP = 0). Besarnya nilai PWM ini dapat diperoleh dengan menggunakan kontrol Proporsional (P), dimana $P = e \times K_p$ (K_p adalah konstanta proporsional yang nilainya diset sendiri dari hasil *tuning/trial and error*).

Jika pergerakan robot masih terlihat bergelombang, dapat ditambahkan parameter kontrol Derivatif (D). Kontrol D digunakan untuk mengukur seberapa cepat robot bergerak dari kiri ke kanan atau dari kanan ke kiri. Semakin cepat bergerak dari satu sisi ke sisi lainnya, maka semakin besar nilai D. Konstanta D (K_d) digunakan untuk menambah atau mengurangi imbas dari derivatif. Dengan mendapatkan nilai K_d yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional dapat diminimalisasi. Dengan mendapatkan nilai K_d yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional bisa diminimalisasi. Nilai D didapat dari $D = K_d / T_s \times \text{rate}$, dimana T_s adalah time sampling atau waktu cuplik dan $\text{rate} = e(n) - e(n-1)$. Dalam program, nilai error (SP – PV) saat itu menjadi nilai *last_error*, sehingga *rate* didapat dari $\text{error} - \text{last_error}$

Ringkasan efek perubahan dari masing-masing parameter PID terhadap keluaran dari robot *line follower* dapat dilihat pada tabel berikut.

Tabel 1 Efek Perubahan Konstanta terhadap Parameter PID terhadap Keluaran Sistem

Parameter	Rise Time	Overshoot	Settling Time	Steady State Error
K_p	Menurun	Meningkat	Perubahan kecil	Menurun
K_i	Menurun	Meningkat	Meningkat	Menurun signifikan
K_d	Sedikit turun	Sedikit turun	Sedikit turun	Tidak ada efek

PROSEDUR PRAKTIKUM

A. Percobaan dalam praktikum

1. Peralatan dan Bahan Praktikum



Gambar 1 Contoh susunan dan urutan sensor pada robot line follower.

Code Program :

```
// PRAKTIKUM 1 SISTEM KENDALI ON/OFF
// KELOMPOK 7 A2R
// ANGGOTA : Anggi Angga Reksha (6702190033)
// ANGGOTA : Nauval Zaki Ishomi (6702194027)

//Deklarasi Pin yang digunakan untuk sensor
int sensor1 = A0; int sensor2 = A1; int
sensor3 = A2; int sensor4 = A3; int sensor5
= A4; int sensor6 = A5; int baca_sensor[6];
int sendat[6]; int sensorBit; int but1 = 8;
int but2 = 9;

//Variable bantuan untuk menyimpan memory a.k.a Error Detection Memory (EDM)
//pin penggerak Motor int ENA =
4; //Pin 1&2 harus HIGH int ENB =
2; //Pin 3&4 harus HIGH

//pin Motorkiri
int motorInput1 = 5; //input motor driver int
motorInput2 = 6; //input motor driver
```

```

//pin Motorkanan int
motorInput3 = 3; int
motorInput4 = 11;

//Initial Speed of Motor int
kecepatanSetPoint = 150;;

//Konstanta PID float
Kp = 15 ; float Ki =
0 ; float Kd = 5 ;

float error = 0, P = 0, I = 0 , D = 0 , PID_value = 0; float
lastError = 0;

void setup(){
// Keenam Sensor Photodiode sebagai INPUT yaitu sensor cahaya
pinMode(sensor1, INPUT); pinMode(sensor2, INPUT);
pinMode(sensor3, INPUT); pinMode(sensor4, INPUT);
pinMode(sensor5, INPUT); pinMode(sensor6, INPUT);

//Motor sebagai OUTPUT penggerak
pinMode(motorInput1, OUTPUT); pinMode(motorInput2,
OUTPUT); pinMode(motorInput3, OUTPUT);
pinMode(motorInput4, OUTPUT);

pinMode(ENA, OUTPUT); pinMode(ENB,
OUTPUT);

Serial.begin(9600);
} void loop(){
read_sensor_values();
hitungPID();
motor_control();
}

//Membaca sinyal analog dari sensor void
read_sensor_values(){ baca_sensor[0]
= analogRead(sensor1);
baca_sensor[1] = analogRead(sensor2);
baca_sensor[2] = analogRead(sensor3);
baca_sensor[3] = analogRead(sensor4);
baca_sensor[4] = analogRead(sensor5);
baca_sensor[5] = analogRead(sensor6);

// 1 = Gelap 0 = Terang
// Case 1 0 0 0 0 0
if (baca_sensor[0] < 34 && baca_sensor[1] > 34 &&
baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34)
{
error = 5;
Serial.print("\n");
}

// Case 1 1 0 0 0 0
else if (baca_sensor[0] < 34 && baca_sensor[1] < 34 &&
baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34){

```

```

        error =
4;
    Serial.print("\n");
}

    // Case 0 1 0 0 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] < 34 &&
baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34){
    error =
3;
    Serial.print("\n");
}

    // Case 0 1 1 0 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] < 34 &&
baca_sensor[2] < 34 && baca_sensor[3] > 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34){
    error =
2;
    Serial.print("\n");
}

    // Case 0 0 0 1 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] < 34 && baca_sensor[3] > 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34){
    error =
1;
    Serial.print("\n");
}

    // Case 0 0 1 1 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] < 34 && baca_sensor[3] < 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34){
error = 0;
    Serial.print("\n");
}

    // Case 0 0 0 1 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] > 34 && baca_sensor[3] < 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34){

    error = -1;
    Serial.print("\n");
}

    // Case 0 0 0 1 1 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] > 34 && baca_sensor[3] < 34 &&
baca_sensor[4] < 34 && baca_sensor[5] > 34){

    error = -2;
    Serial.print("\n");
}

    // Case 0 0 0 0 1 0

```

```

else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
baca_sensor[4] < 34 && baca_sensor[5] > 34){

    error = -3;
    Serial.print("\n");
}

// Case 0 0 0 0 1 1
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
baca_sensor[4] < 34 && baca_sensor[5] < 34){

    error = -4;
    Serial.print("\n");
}

// Case 0 0 0 0 0 1
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
baca_sensor[4] > 34 && baca_sensor[5] < 34){

    error = -5;
    Serial.print("\n");
}

/*
// Belokan kiri tajam
else if (baca_sensor[0] < 34 && baca_sensor[1] < 34 &&
baca_sensor[2] < 34 && baca_sensor[3] < 34 &&
baca_sensor[4] < 34 && baca_sensor[5] > 34){
    error =
10;
    Serial.print("\
n");
}

//Belokan kanan tajam
    else if (baca_sensor[0] > 34 && baca_sensor[1] < 34 &&
baca_sensor[2] < 34 && baca_sensor[3] < 34 &&
baca_sensor[4] < 34 && baca_sensor[5] < 34){

        error = 11;
        Serial.print("\n");
    }

//Putar Balik
    else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
baca_sensor[4] > 34 && baca_sensor[5] > 34){

        error = 12;
        Serial.print("\n");
    }

//Garis Finish diam
    else if (baca_sensor[0] < 34 && baca_sensor[1] < 34 &&
baca_sensor[2] < 34 && baca_sensor[3] < 34 &&
baca_sensor[4] < 34 && baca_sensor[5] < 34){

```

```

        error = 13;
        Serial.print("\n");
    }

    // Menampilkan data sensor ke Serial Monitor
    // Data sensor 1-6
    // Formatting tampilkan sensor

    for(int i=0; i<=5; i++){
    Serial.print("Sensor");
        Serial.print(i+1);
        Serial.print(": ");
        Serial.print(baca_sensor[i]);
    Serial.print("\n");        delay(1000);
    }
    */
}

// Menghitung Nilai PID void
hitungPID(){
    P = error;
    I = error + lastError;
    D = error - lastError;

    PID_value = (Kp * P) + (Ki * I) + (Kd * D);

    lastError = error;
}

// Mengatur Kecepatan Motor void
motor_control()
{
    int kecepatanMotorKiri = kecepatanSetPoint - PID_value;
    int kecepatanMotorKanan = kecepatanSetPoint + PID_value;

    // Kecepatan Motor agar tidak melebihi batas pwm
    kecepatanMotorKiri = constrain(kecepatanMotorKiri, 0, 255);
    kecepatanMotorKanan = constrain(kecepatanMotorKanan, 0, 255);

    digitalWrite(ENA, HIGH);
    analogWrite(motorInput1, kecepatanMotorKiri);
    analogWrite(motorInput2, 0);

    digitalWrite(ENB, HIGH);
    analogWrite(motorInput3, kecepatanMotorKanan);
    analogWrite(motorInput4, 0);

    Serial.print(lastError);
    Serial.print(" ");
    Serial.print(kecepatanMotorKiri);
    Serial.print(" ");
    Serial.println(kecepatanMotorKanan);
}

```

Hasil Percobaan:

Circuit design tugas 3 | Tinkercad x VKI213-D3TK-43-03: Assignment x Microsoft Word - Modul 5 - Siste x +

tinkercad.com/things/h4eL5mDZ3Sy-tugas-3/editel

tugas 3

All changes saved

Simulator time: 00:00:20.463

Code Stop Simulation Export Share

Anggi Angga Rekxa Nauval Zaki Ishomi

```

188 P = error;
189 I = error + lastError;
190 D = error - lastError;
191 PID_value = (Kp * P) + (Ki * I) + (Kd * D);
192 lastError = error;
193 }
194 // Mengatur Kecepatan Motor
195 void motor_control()
196 {
197   int kecepatanMotorKiri = kecepatanSetPoint - PID_value;
198   int kecepatanMotorKanan = kecepatanSetPoint + PID_value;
199   // Kecepatan Motor agar tidak melebihi batas pwm
200   kecepatanMotorKiri = constrain(kecepatanMotorKiri, 0, 255);
201   kecepatanMotorKanan = constrain(kecepatanMotorKanan, 0, 255);
202   digitalWrite(ENA, HIGH);
203   analogWrite(motorInput1, kecepatanMotorKiri);
204 }

```

Serial Monitor

```

0.00 150 150
0.00 150 150
0.00 150 150
0.00 150 150
0.00 150 150
0.00 150 150
0.00 150 150
0.00 150 150
0.00

```

Send Clear

Type here to search

89% 11:23 29/04/2021

Tabel 1Tabel Kebenaran Sistem Kendali

Sensor						Error	Nilai Setpoint	Analog Value	
1	2	3	4	5	6			Motor Kiri	Motor Kanan
1	0	0	0	0	0	5	150	75(2313 Rpm)	225 (8348 Rpm)
1	1	0	0	0	0	4	150	90(3240 Rpm)	210(7856 Rpm)

0	1	0	0	0	0	3	150	105(3969 Rpm)	195(7181 Rpm)
0	1	1	0	0	0	2	150	120(4294 Rpm)	180(6576 Rpm)
0	0	1	0	0	0	1	150	135(4853 Rpm)	165(6098 Rpm)
0	0	1	1	0	0	0	150	150(5876 Rpm)	150(5876 Rpm)
0	0	0	1	0	0	-1	150	165(6098 Rpm)	135(4853 Rpm)
0	0	0	1	1	0	-2	150	180(6576 Rpm)	120(4294 Rpm)
0	0	0	0	1	0	-3	150	195(7181 Rpm)	105(3969 Rpm)
0	0	0	0	1	1	-4	150	210(7856 Rpm)	90(3240 Rpm)
0	0	0	0	0	1	-5	150	225(8348 Rpm)	75(2313 Rpm)

Kesimpulan Praktikum :

Sistem kendali PID digunakan untuk mengoreksi error dari pengukuran variabel input (sensor) agar output sistem sesuai dengan nilai set point untuk menghasilkan error yang sekecil mungkin.

Berdasarkan hasil percobaan yang dilakukan, didapatkan nilai parameter kontrol PID $K_p = 5$, $K_d = 0$, $K_i = 0$, Parameter tersebut digunakan dalam melakukan perhitungan PID dan mengatur kecepatan Motor kiri dan Kanan., Teknik kendali proporsional-derivatif (PD) adalah pengendali yang merupakan gabungan antara teknik kendali proporsional (P) dengan teknik kendali derivatif (D).