Kendali Kecepatan dan Posisi Motor DC

Priyova M. Rafief[1], Karunia Dini F.[1], Octsana Dhiyaa W.[1], Bodhi Setiawan[1]

Universitas Gadjah Mada[1]

Praktikum Sistem Kendali Lanjut

UNIVERSITAS
GADJAH MADA

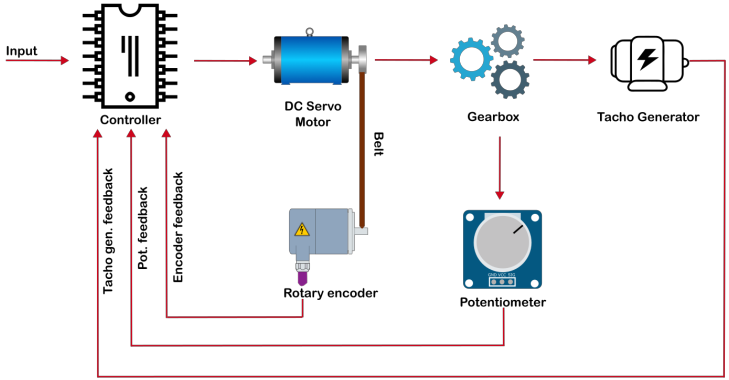
Anggota Kelompok 1

1. Priyoya M. Rafief (20/457197/SV/17644)
2. Karunia Dini F. (20/464248/SV/18567)
3. Octsana Dhiyaa W. (20/464253/SV/18572)
4. Bodhi Setiawan (20/464239/SV/18558)

Isi Pembahasan

Gambaran Umum Proyek Sistem Kendali Motor DC

Diagram Sistem

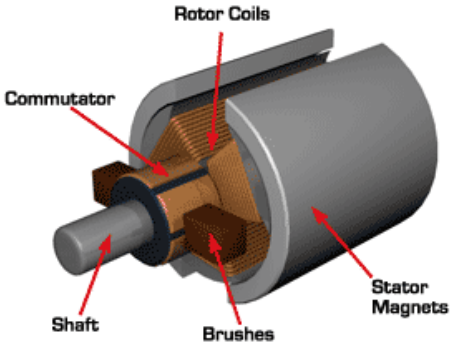


Pemodelan Sistem Motor DC

Pengertian Motor DC

Motor DC atau dalam bahasa Indonesia disebut motor arus searah adalah jenis motor listrik yang mengubah energi listrik arus searah menjadi energi mekanis. Bentuk energi yang dihasilkan berupa putaran. Prinsip kerja motor aru searah berdasarkan pada interaksi antara dua fluks magnetik yang disebut dengan kumparan medan dan kumparan jangkar. Kumparan medan menghasilkan fluks magnet dengan arah dari kutub utara ke kutub selatan, sedangkan kumparan jangkar menghasilkan fluks magnetik yang melingkar.
Sumber: Wikipedia

Ilustrasi Motor DC:



Struktur Fisik

Parameter:

J	:	Momen inersia rotor $(Kg.m^2)$
b	:	Koefisien gaya gesek viskos $(N.m.s)$
Ke	:	Koefisien gaya elektromotif $(V/rad/sec)$
Kt	:	Koefisien torsi motor $(N.m/Amp)$
R	:	Resistansi kumparan (Ohm)
L	:	Induktansi kumparan (H)

Struktur:

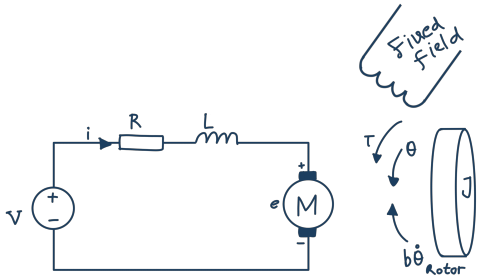
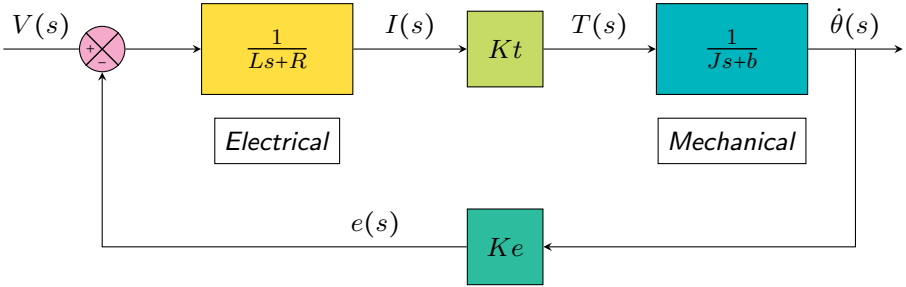


Diagram Blok Plant Motor DC

Struktur motor DC dengan parameter-parameter sebelumnya memiliki diagram blok sebagai berikut:



Fungsi Alih

Diagram blok *plant* motor DC menghasilkan persamaan fungsi alih berikut:

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{Kt}{(Js+b)(Ls+R)+KtKe} \qquad \left[\frac{rad/sec}{V}\right] \tag{1}$$

Persamaan di atas merupakan fungsi alih kecepatan motor DC. Dengan mengintegralkan fungsi alih tersebut, maka diperoleh fungsi alih untuk posisi motor DC:

$$\frac{\theta(s)}{V(s)} = \frac{Kt}{s((Js+b)(Ls+R)+KtKe)} \qquad \left[\frac{rad}{V}\right] \tag{2}$$

State Space

Kecepatan:

$$\frac{\delta}{\delta t}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}-\frac{b}{J} & \frac{Kt}{J}\\ -\frac{Ke}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}V \qquad (3)$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix}$$

Struktur:

$$\frac{\delta}{\delta t}\begin{bmatrix}\theta\\ \dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & -\frac{b}{J} & \frac{Kt}{J}\\ 0 & -\frac{Ke}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{1}{L}\end{bmatrix}V \qquad (4)$$

$$y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ i\end{bmatrix}$$

Perancangan Kendali PID Motor DC

Apa Itu Kendali PID?

- PID=**Proportional-Integral-Derivative**
- Kendali mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri
- Secara kontinu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur. Persamaan:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{5}$$

Mengapa Kendali PID?

Kendali PID berfungsi untuk meminimalkan nilai kesalahan (*error*) setiap waktu dengan penyetelan variabel kontrol, seperti posisi, kecepatan, damper, daya, dan lain sebagainya.
Contoh perbandingan sistem dengan dan tanpa PID:

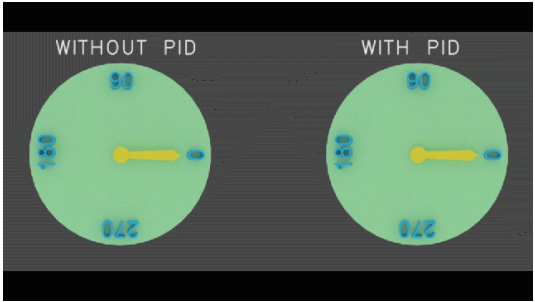
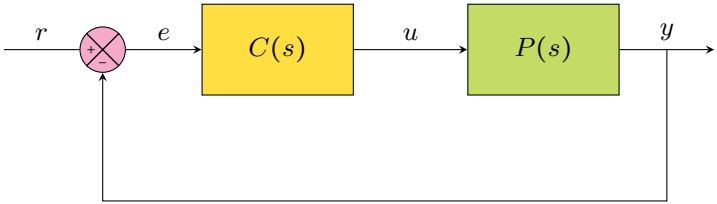


Diagram Blok Kendali



Keterangan:

$C(s)$	:	*Controller*
$P(s)$	:	*Plant*
$r(s)$	:	*Output* yang diinginkan
$e(s)$	:	Nilai *error*
$u(s)$	:	Sinyal kendali
$y(s)$	:	*Output* sesungguhnya

Diagram Blok Kendali PID Motor DC: Kecepatan

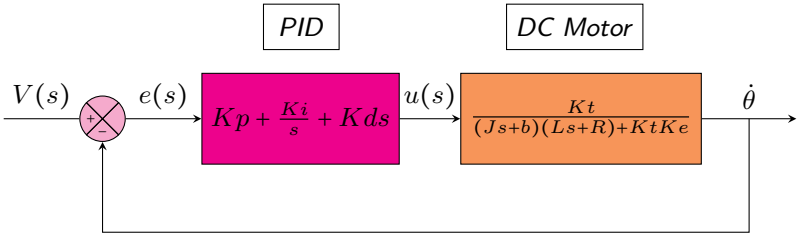
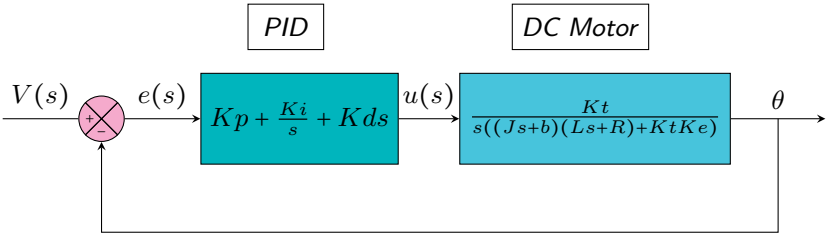


Diagram Blok Kendali PID Motor DC: Posisi



Simulasi Kendali PID

Uji Perbandingan Sistem *Open-loop* dengan *Closed-loop* Motor DC

Program *Open-loop*:

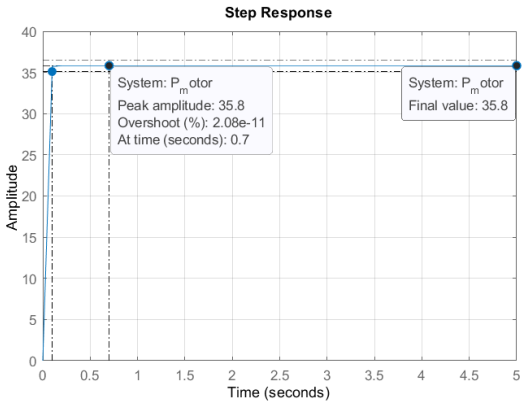
```
J = 3.2284E-6;
b = 3.5077E-6;
Kt = 0.0274;
Ke = 0.0274;
R = 4;
L = 2.75E-6;;
s = tf('s');
P_motor = Kt/((J*s+b)*(L*s+R)+Kt*Ke);
rP_motor = 0.1/(0.5*s+1)
ltiview('step', P_motor, 0:0.1:5);
```

Program *Closed-loop*:

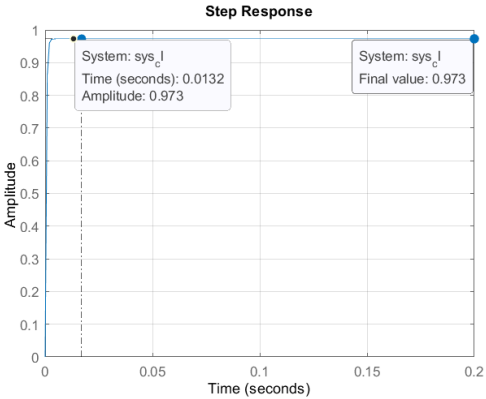
```
J = 3.2284E-6;
b = 3.5077E-6;
Kt = 0.0274;
Ke = 0.0274;
R = 4;
L = 2.75E-6;
s = tf('s');
P_motor = Kt/((J*s+b)*(L*s+R)+Kt*Ke);
t = 0:0.001:0.2;
sys_cl = feedback(P_motor,1)
step(sys_cl,t)
```

Uji Perbandingan Sistem *Open-loop* dengan *Closed-loop* Motor DC

Hasil *Open-loop*:



Hasil *Closed-loop*:

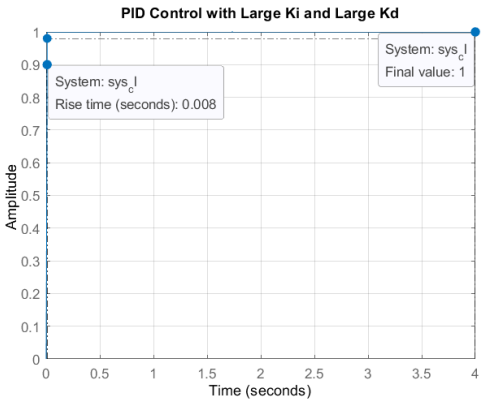


Kendali PID: Kecepatan

Program:

```
J = 3.2284E-6;
b = 3.5077E-6;
Kt = 0.0274;
Ke = 0.0274;
R = 4;
L = 2.75E-6;
s = tf('s');
P_motor = Kt/((J*s+b)*(L*s+R)+Kt*Ke);

Kp = 100;
Ki = 200;
Kd = 10;
C = pid(Kp,Ki,Kd);
sys_cl = feedback(C*P_motor,1);
step(sys_cl, 0:0.01:4)
grid
title('PID Control with Large Ki and Large Kd')
```

Hasil:

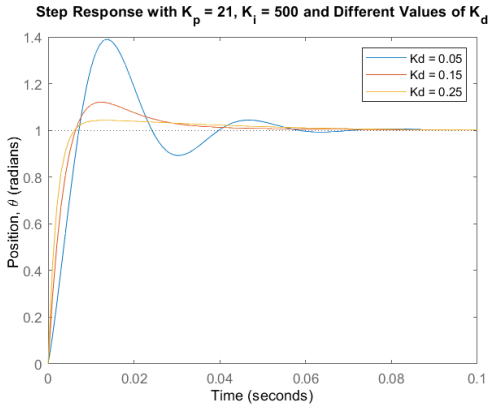


Kendali PID: Posisi

Program:

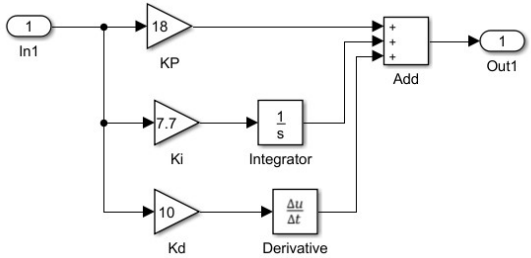
```
J = 3.2284E-6;
b = 3.5077E-6;
K = 0.0274;
R = 4;
L = 2.75E-6;
s = tf('s');
P_motor = K/(s*((J*s+b)*(L*s+R)+K^2));
Kp = 21;
Ki = 500;
Kd = 0.05;

for i = 1:3
  C(:,:,i) = pid(Kp,Ki,Kd);
  Kd = Kd + 0.1;
end

sys_cl = feedback(C*P_motor,1);
t = 0:0.001:0.1;
step(sys_cl(:,:,1), sys_cl(:,:,2), sys_cl(:,:,3), t)
ylabel('Position, \theta (radians)')
title('Step Response with K_p = 21, K_i = 500 and
      Different Values of K_d')
legend('Kd = 0.05', 'Kd = 0.15', 'Kd = 0.25')
```

Hasil:

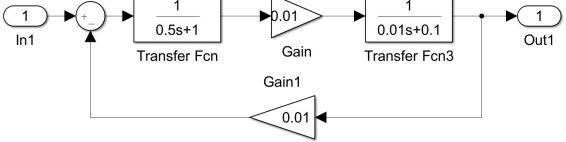


Simulink

Blok PID:

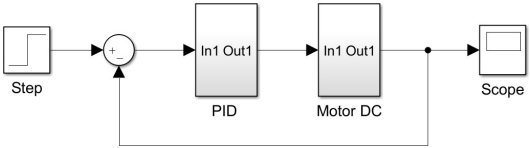


Blok Plant Motor DC

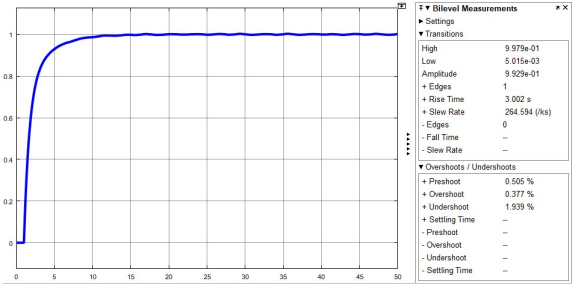


Simulink

Blok Simulink



Hasil



Simulasi LQR

Simulasi LQR: Kecepatan

Klik untuk menjalankan animasi scroll:

△ △ △

```
%% Mendefinisikan Parameter-parameter yang diperlukan
= 0.01;
= 0.1;
= 0.01;
= 1;
= 0.5;

 Matriks State-Space

= [-b/J  K/J
/L  -R/L];
= [0
L];
= [1 0];
0;

 Cek Controllability Sistem
 = ctrb(A,B)
nk(Co)

 LQR
= [1  0
  1];
= 1;
```

Simulasi LQR: Posisi

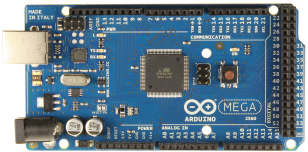
Klik untuk menjalankan animasi scroll:

△ △ △

```
%% Mendefinisikan Parameter-parameter yang diperlukan
J = 3.2284E-6;
b = 3.5077E-6;
K = 0.0274;
R = 4;
L = 2.75E-6;

%% Matriks State-Space

A = [0 1    0
0 -b/J  K/J
0 -K/L  -R/L];
B = [0
0
1/L];
C = [1 0  0];
D=0;

%% Cek Controllability Sistem
Co = ctrb(A,B)
rank(Co)

%% LQR
Q = [1  0  0
0  1  0
```

Sistem Mekanikal dan Elektrikal

Komponen-Komponen Utama



Arduino Mega

Motor DC

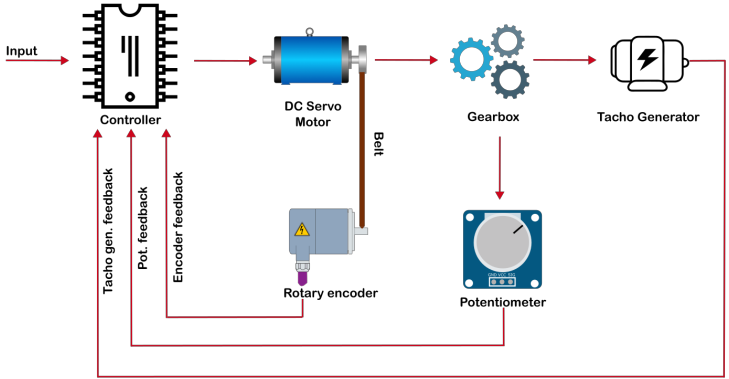
Tachogenerator

Potensiometer 360^o

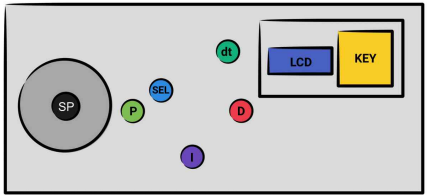
Rotary Encoder

Diagram Sistem

Komponen-komponen utama kemudian disusun menjadi suatu sistem yang ditunjukkan pada diagram sistem seperti yang sudah ditunjukkan di awal.



Konfigurasi Pin-Pin Arduino Mega



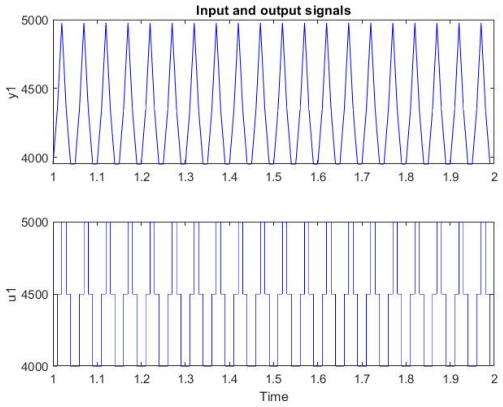
Kontroler

Input-Output	**Fungsi**	**Pin**
SP	Set Poin	A7
P	Proporsional	A5
I	Integral	A8
D	Derivatif	A6
SEL	*Selector*	34, 32, 30
dt	*Time Sampling*	A4
LCD	LCD	0, 1
KEY	Keypad	52, 50, 48, 46, 44, 42, 40, 38

Desain GUI

Percobaan Identifikasi Sistem

Grafik Input dan Output

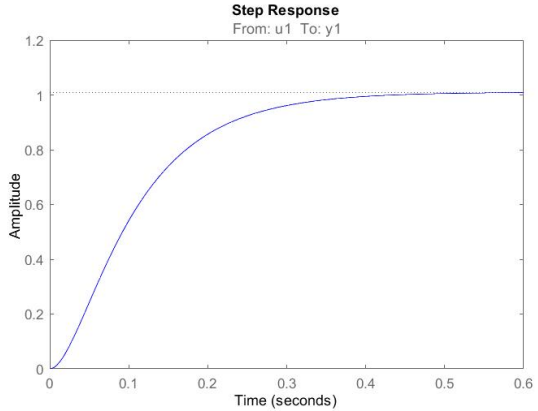


Transfer Function

```
tf1 =

  From input "u1" to output "y1":
          386.7
  ---------------------
  s^2 + 44.73 s + 382.8
```

Percobaan Identifikasi Sistem

Step Response



State Space

```
A =
          x1        x2        x3        x4        x5        x6        x7
x1     328.2     487.8     24.99     -3364  -1.031e-30  4.167e-31 -1.444e-76
x2    -997.6    -948.2    -244.3      1556   4.623e-31 -1.869e-31  8.794e-77
x3     591.9     342.8     195.8      3612   6.666e-31 -2.694e-31  1.058e-76
x4     117.9     98.34     19.18    -28.59  -3.142e-32  1.27e-32  3.932e-78
x5         0         0         0         0        -200         0         0
x6         0         0         0         0         0      -200         0
x7         0         0         0         0         0         0      -200

B =
          u1
x1  -5.354e+31
x2   4.768e+31
x3   5.667e+31
x4  -4.095e+30
x5  -3.525e+20
x6  -9.512e+19
x7  -2.824e+19

C =
          x1        x2        x3        x4        x5        x6        x7
y1  1.319e-28  9.893e-29  3.302e-29  1.962e-28  3.728e-61 -1.507e-61  8.734e-108

D =
    u1
y1   0

K =
          y1
x1  4.147e+31
x2  1.061e+31
x3  -3.895e+31
x4  -3.638e+30
x5   1.243e-32
x6   2.517e-79
x7          0
```

GUI Python TKinter