

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sejenis

Penelitian sebelumnya berjudul “Simulasi Pengaturan Kecepatan Motor DC Berbasis PID Menggunakan LabVIEW” [1]. Penelitian ini merancang pengendali PID untuk mengatur kecepatan Motor DC. Simulasi Motor DC dengan sistem pengendali PID dan dibuat dengan LabVIEW. *Interface* simulasi pada *front panel* menampilkan grafik tegangan dan kecepatan. Terdapat input *set point* kecepatan dalam RPM. Berdasarkan hasil percobaan terhadap berbagai parameter K_p , T_i dan T_d dimana saat $K_p = 0,001$, $T_i = 0,001$ dan $T_d = 0,001$ respon sistem memiliki *rise time* 0,348 detik, *overshoot* 0,354% dan *settling time* 0,570 detik.

Penelitian selanjutnya berjudul “Aplikasi *Hardware In Loop* Pada Pengendalian Motor DC Menggunakan MATLAB” [2]. *Hardware In the Loop* (HIL) sering digunakan untuk mengidentifikasi suatu objek atau suatu sistem secara virtual yang berguna secara langsung untuk mengurangi kesalahan-kesalahan yang mengakibatkan kerusakan sistem yang akan dijalankan, memberikan perspektif baru dalam pengembangan suatu sistem dan juga sebagai sarana pembelajaran dalam bidang kendali dan instrumentasi. Penelitian yang dilakukan mengambil tema pengendalian kecepatan Motor DC dengan metode PID dengan menggabungkan Mikrokontroler Arduino Uno sebagai objek fisik

dan MATLAB sebagai objek virtual untuk model Motor DC yang akan dikendalikan. Pengujian dilakukan untuk mengetahui respon sistem pada model motor DC dan tuning PID dengan cara mengirimkan instruksi program ke dalam plant model di MATLAB serta pengujian dengan variasi set point dengan parameter kendali PID yang telah optimal. Berdasarkan hasil yang diperoleh, maka konsep Hardware In The Loop telah berjalan dengan baik dan sistem telah sinkron antara MATLAB dan Arduino Uno dan juga mendapatkan parameter PID terbaik dengan $K_p = 26$, $K_i = 1$ dan $K_d = 0.30$ kemudian *rise time* dibawah 0.5 detik, *overshoot* dibawah 10% dan *steady state error* dibawah 5% untuk diterapkan pada pengendalian Motor DC. Kemudian konsep dan teori kendali PID juga dapat dijalankan dengan menggunakan pemrograman pada Mikrokontroler Arduino Uno.

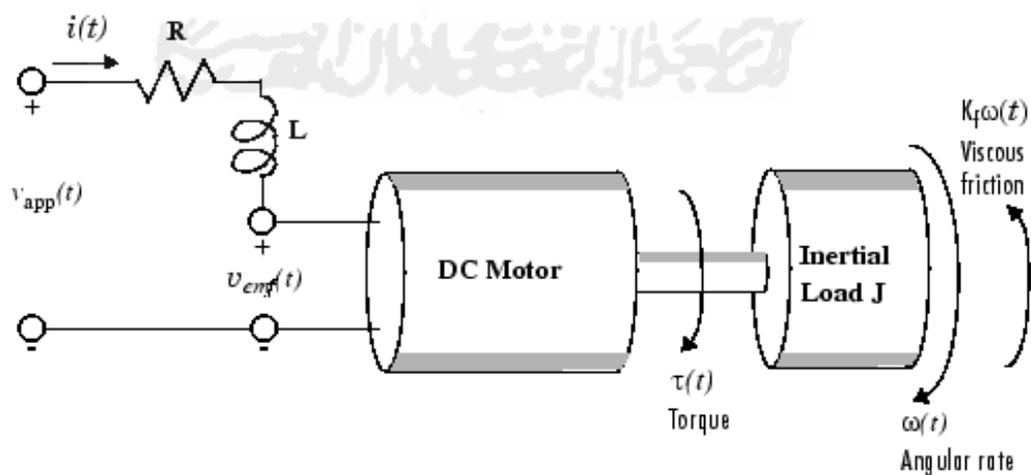
Penelitian lainnya berjudul “Aplikasi Fuzzy Logic Sebagai Kontrol Posisi Motor DC” [3]. Pada artikel ini membahas kendali posisi dari suatu motor DC menggunakan Logika Fuzzy, Kendali fuzzy logic diimplementasikan pada komputer dan algoritma kendali dirancang menggunakan program LabVIEW dan melihat kondisi yang riil dari posisi kendali motor DC. Pengaturan posisi gerak motor dilakukan dengan mengatur tegangan motor dan menggunakan metode *Pulse Width Modulation* (PWM). Feedback sistemnya adalah sinyal frekuensi dari tacho dan sinyal frekuensi ini diubah menjadi tegangan untuk diinputkan pada komputer melalui ADC. Sistem fuzzy logic mempunyai 2 crisp input yaitu *error* dan perubahan *error* posisi motor dan mempunyai 1 crisp output yaitu perubahan tegangan. Metode defuzzifikasi yang digunakan adalah metode *center of area*. Jumlah label dari membership function 7 label. Respon sistem ditampilkan dalam

bentuk grafik sudut posisi motor terhadap waktu. Hasil pengujian menunjukkan membership function dengan bentuk segitiga dan defuzzifikasi *center of area* dapat memberikan hasil yang baik. Pada percobaan membandingkan kendali PID dengan FLC dan hasilnya memperlihatkan bahwa tanggapan motor DC dengan Control Fuzzy Logic (FLC) mampu memberikan peforma yang baik.

Kemudian penelitian yang kini dilakukan dengan menggunakan blok *simulink* MATLAB dengan persamaan model matematis motor DC dengan *state feedback* sebagai pengendali.

2.2 Motor DC

Motor DC merupakan suatu perangkat elektronis pertama yang digunakan dalam sejarah industri perkembangan sistem transmisi daya untuk mengubah besaran listrik menjadi besaran mekanik [4]. Kemudian motor DC juga merupakan suatu aktuator yang umum digunakan pada sistem kendali. Prinsip kerjanya apabila motor diberi daya maka arus akan mengalir ke dalam motor dan menghasilkan torsi putar yang sebanding dengan arusnya [5].



Gambar 2.1 Model motor DC

Motor DC sendiri memiliki karakteristik sesuai dengan persamaan 2.1 sebagai berikut.

$$\tau(t) = K_m \cdot i(t) \quad (2.1)$$

Torsi (τ) yang dihasilkan motor sebanding dengan arus *armature* yang diinduksikan oleh tegangan masukan pada motor DC (i). Konstanta *armature* dituliskan dengan (K_m) yang dihubungkan dengan sifat fisik motor seperti banyaknya lilitan koil penghantar, kekuatan medan magnet, dan lain sebagainya [6]. Gaya gerak listrik induksi v_{emf} sebanding dengan kecepatan sudut ω pada keluaran motor.

$$v_{emf} = K_b \cdot \omega(t) \quad (2.2)$$

Parameter K_b merupakan konstanta gaya gerak listrik yang tergantung pada sifat fisik motor. Kemudian bagian mekanik persamaan motor diturunkan dengan beban inersia (J) dikalikan turunan kecepatan sudut sama dengan jumlah semua torsi pada keluaran motor dengan $K_f \omega(t)$ sebagai gaya gesek.

$$\begin{aligned} \tau(t) - f_k(t) &= J \cdot \alpha \\ K_m i(t) - K_f \omega(t) &= J \frac{d\omega(t)}{dt} \end{aligned} \quad (2.3)$$

Bagian elektrik persamaan motor dapat ditunjukkan sebagai berikut.

$$v_{app}(t) = v_R(t) + v_L(t) + v_{emf}(t) \quad (2.4)$$

Dengan mensubstitusi parameter $v_{emf}(t) = K_b \omega(t)$ dapat diperoleh :

$$v_{app}(t) = R \cdot i(t) + L \cdot \frac{di(t)}{dt} + K_b \omega(t) \quad (2.5)$$

Kemudian persamaan mekanik 2.3 dan elektris 2.5 akan diubah menjadi bentuk *laplace* menjadi persamaan 2.6 dan 2.7 sebagai berikut :

$$K_m I(s) - K_f W(s) = J.s.W(s) \quad (2.6)$$

Pada persamaan 2.7 parameter $I(s)$ dieliminasi untuk memperoleh :

$$\begin{aligned} V(s) &= R.I(s) + L.s.I(s) + K_b W(s) \\ V(s) - K_b W(s) &= I(s).(R + L.s) \\ I(s) &= \frac{V(s) - K_b W(s)}{R + L.s} \end{aligned} \quad (2.7)$$

Setelah itu persamaan 2.6 pada bagian mekanik motor DC akan digabungkan dengan persamaan 2.7 bagian elektrik motor yang sudah diubah ke *laplace* dengan persamaan:

$$K_m \cdot \left(\frac{V(s) - K_b W(s)}{R + L.s} \right) = J.s.W(s) - K_f W(s) \quad (2.8)$$

Dengan memindahruaskan persamaan 2.8 kemudian didapatkan persamaan dengan masukan tegangan $V(s)$ dan keluaran kecepatan sudut $\omega(s)$ sehingga pada akhirnya diperoleh persamaan

$$W(s) = \left(\frac{K_m}{(Js + K_f)(R + Ls) + K_m K_b} \right) V(s) \quad (2.9)$$

Persamaan 2.9 diisi dengan parameter dari tabel 2.1, sehingga didapatkan persamaan 2.10

$$W(s) = \frac{1.5}{s^2 + 14s + 40.2} V(s) \quad (2.10)$$

Dalam percobaan ini, bentuk fungsi alih persamaan 2.9 yang kemudian akan menjadi model untuk *plant* yang akan dikendalikan dengan metode *pole placement* dengan parameter yang akan ditunjukkan pada tabel 2.1 sebagai berikut.

Tabel 2.1 Karakteristik motor DC

Parameter	Nilai
Tahanan (R)	2,0 Ohm
Induktansi (L)	0,5 H
Konstanta Torsi (K_m)	0,15
Konstanta EMF (K_b)	0,15
Gaya Gesek (K_f)	0,2 <i>N.ms</i>
Momen Inersia (J)	0,02 <i>kg.m² / s²</i>

2.3 *State Space*

Sistem kompleks modern dapat mempunyai beberapa masukan dan keluaran, dan ini mungkin dihubungkan dengan berbagai macam cara [7]. Untuk menganalisis sistem seperti ini, adalah penting untuk mengurangi kompleksitas pernyataan matematika, seperti yang dilakukan pada komputer. Pendekatan ruang keadaan terhadap analisis sistem adalah yang terbaik.

Sedangkan teori kontrol didasarkan pada hubungan masukan-keluaran, atau fungsi pemindahan, teori kontrol modern didasarkan pada sistem persamaan dalam bentuk persamaan diferensial n-orde pertama, yang mungkin dikombinasikan ke dalam persamaan diferensial matriks-vektor orde pertama. Penggunaan notasi matriks-vektor sangat mempermudah pernyataan matematika dalam sistem persamaan. Bertambahnya jumlah variabel, jumlah masukan, atau jumlah keluaran tidak menambah kompleksitas persamaan. Kenyataannya, analisis sistem dengan masukan dan keluaran yang kompleks dapat diselesaikan dengan menggunakan cara ini yang hanya sedikit lebih kompleks dari yang dibutuhkan untuk analisis sistem persamaan diferensial skalar orde pertama.

Analisis ruang keadaan yang mencakup pembahasan ketidak-unikatan himpunan variabel keadaan, nilai eigen matriks $n \times n$, ketidak-samaan nilai eigen,

pendiagonalan matriks $n \times n$, penyajian ruang keadaan sistem orde ke- n dengan fungsi gaya r , teorema Cayley-Hamilton dan perhitungan e^{At} .

Ketidak-unikan himpunan variabel keadaan telah dinyatakan bahwa himpunan variabel keadaan suatu sistem adalah tidak unik. Misalnya bahwa x_1, x_2, \dots, x_n adalah suatu himpunan variabel keadaan. Selanjutnya sebagai himpunan variabel keadaan yang lain kitadapat menggunakan setiap himpunan fungsi

$$\hat{x}_1 = X_1(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$$

$$\hat{x}_2 = X_2(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$$

.

.

.

$$\hat{x}_n = X_n(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$$

dengan syarat bahwa untuk setiap himpunan nilai $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$ terdapat suatu himpunan nilai $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$ yang unik dan sebaliknya. Jadi, jika x merupakan suatu vektor keadaan, maka \hat{x} yang memenuhi hubungan

$$\hat{x} = Px$$

juga merupakan suatu vektor keadaan, dengan syarat bahwa matriks P non-singular. Vektor-vektor keadaan yang berbeda membawa informasi yang sama mengenai perilaku sistem.

2.4 Sistem Servo dengan Integrator

Diasumsikan sebuah *plant* didefinisikan oleh Persamaan (2.11) dan (2.12).

$$\dot{x} = Ax + Bu \quad (2.11)$$

$$y = Cx \quad (2.12)$$

Dengan $x = \text{state vector}$ untuk *plant* (n -vector)

$u = \text{sinyal kontrol}$ (skalar)

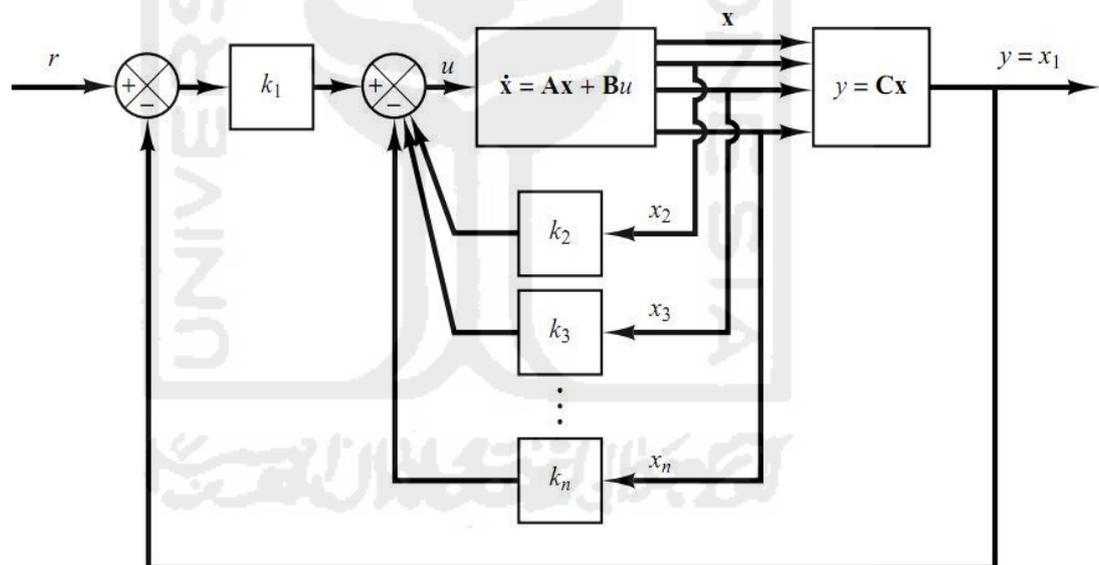
$y = \text{sinyal output}$ (skalar)

$A = \text{matriks } n \times n$

$B = \text{matriks } n \times 1$

$C = \text{matriks } 1 \times n$

Gambar 2.2 merupakan konfigurasi umum dari sebuah sistem servo tipe 1 dengan *plant* yang mempunyai integrator [8]. Diasumsikan $y = x_1$ dan input referensi adalah sinyal *step*. Pada sistem ini digunakan skema *state-feedback control* seperti yang ditunjukkan oleh Persamaan (2.13).



Gambar 2.2 Sistem Servo Tipe 1 dengan *Plant* yang mempunyai Integrator

$$u = -[0 \ k_2 \ k_3 \ \dots \ k_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + k_1(r - x_1)$$

$$= -K_x + k_1 r \quad (2.13)$$

dengan $K = [k_1 k_2 \dots k_n]$.

Diasumsikan bahwa input referensi (fungsi *step*) diberikan pada saat $t = 0$. Selanjutnya untuk $t > 0$, dinamika sistem dapat dideskripsikan oleh Persamaan (2.1) dan (2.3), atau pula oleh Persamaan (2.14).

$$\dot{x} = Ax + Bu = (A - BK)x + Bk_1r \quad (2.14)$$

Selanjutnya dapat didesain sistem servo tipe 1 sedemikian hingga *closed-loop poles* ditempatkan pada lokasi yang diinginkan. Sistem yang didesain dapat merupakan sebuah sistem yang stabil, $y(\infty)$ akan mencapai nilai konstan dari r , dan $u(\infty)$ akan mencapai nilai nol. (r merupakan input *step*).

Pada kondisi *steady state* dinyatakan dengan Persamaan (2.15).

$$\dot{x}(\infty) = (A - BK)x(\infty) + Bk_1r(\infty) \quad (2.15)$$

Perlu dicatat bahwa $r(t)$ merupakan *input step*, sehingga diperoleh $r(\infty) = r(t) = r$ (konstan) untuk $t > 0$. Dengan mengurangkan Persamaan (2.15) dari Persamaan (2.14), sehingga diperoleh Persamaan (2.16).

$$\dot{x}(t) - \dot{x}(\infty) = (A - BK)x(\infty) + Bk_1r \quad (2.16)$$

Dengan mendefinisikan $x(t) - x(\infty) = e(t)$, Persamaan (2.16) dapat diubah kedalam bentuk Persamaan (2.17).

$$\dot{e} = (A - BK)e \quad (2.17)$$

Persamaan (2.7) menggambarkan dinamika *error* sistem.

Desain sistem servo tipe 1 dalam hal ini dikonversi menjadi desain *stable regulator system* sedemikian hingga $e(t)$ menuju nol, dengan kondisi inisial $e(0)$. Jika sistem didefinisikan sebagai Persamaan (2.11) yang merupakan *state controllable*, maka dengan menentukan *eigen values* yang diinginkan $\mu_1, \mu_2, \dots, \mu_n$ untuk matriks $A - BK$, matriks K bisa dihitung menggunakan metode *pole-placement*.

Nilai *steady state* dari $x(t)$ dan $u(t)$ bisa dicari dengan cara sebagai berikut:

Pada saat *steady state* $t = (\infty)$, diperoleh Persamaan (2.18) dari Persamaan (2.14).

$$\dot{x}(\infty) = 0 = (A - BK)x(\infty) + Bk_1r \quad (2.18)$$

Karena *eigen values* yang diinginkan dari matriks $A - BK$ semuanya berada di sebelah kiri dari *s-plane*, maka terdapat *inverse* dari matriks $A - BK$. Akibatnya, $x(\infty)$ dapat dihitung dengan menggunakan Persamaan (2.19).

$$x(\infty) = -(A - BK)^{-1}Bk_1r \quad (2.19)$$

Dengan $u(\infty)$ dapat ditentukan dengan Persamaan (2.20).

$$u(\infty) = -Kx(\infty) + k_1r = 0 \quad (2.20)$$

selanjutnya *state feedback gain* matriks K dan *gain constant* k_1 dapat dihitung menggunakan metode *pole-placement*.