



## BAB II

### TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian terkait yang telah dilakukan sebelumnya dan terdapat penjelasan tentang landasan teori yang dipakai dalam mengerjakan Tugas Akhir ini. Landasan teori tersebut meliputi teori dalam sistem kendali posisi, *magnetic levitation ball*, permodelan matematis *plant*, perangkat lunak MATLAB, dan pengendali LQR.

#### 2.1. Penelitian terkait

Dalam Penelitian Tugas Akhir ini dilakukan studi literatur yang merupakan pencarian teori serta referensi yang relevan dengan kasus dan permasalahan yang akan diselesaikan, teori dan referensi didapat dari jurnal, paper, buku dan sumber lainnya.

Penelitian yang dilakukan pada pengontrolan stabilitas profil tegangan generator pada *Automatic Voltage Regulator (AVR)* menggunakan pengendali *Linear Quadratic Regulator (LQR)* yang bertujuan untuk menjaga kestabilan sistem. Pada sistem AVR ini waktu yang di butuhkan cukup lama untuk menstabilkan tegangan. Pengendali LQR dapat mempercepat waktu kestabilan sistem. Hal ini telah dibuktikan sistem AVR ini akan stabil pada saat detik ke 19.0812, sedangkan jika sistem AVR ini menggunakan pengendali LQR maka waktu yang di perlukan untuk mencapai kestabilan hanya sistem menggunakan pengendali LQR waktu yang di butuhkan untuk mencapai kestabilan sangat cepat yaitu 0.8866 detik [4].

Penelitian yang dilakukan untuk mengendalikan posisi sudut angguk untuk roket RKX-300 dengan metode kendali LQR dan *pole placement*. Pergerakan pada roket ditentukan oleh sirip – sirip roket. Mengendali sirip angguk atau defleksi *elevator* roket untuk mengatur posisi sudut angguk roket dengan menggunakan metode *Linear Quadratic Regulator (LQR)* dan *pole placement* yang diterapkan bersama-sama pada *plant*. Penggunaan metode LQR saja menghasilkan hasil yang tidak optimal, kemudian digunakan metode *pole placement* yang memperlihatkan perbaikan pada respon transien sistem terhadap masukan *step*, yang selanjutnya dengan metode LQR sistem dapat mencapai nilai referensi yang diinginkan. Pada kendali optimal LQR dengan *tracking system*, variabel yang perlu dicari adalah nilai konstanta penguatan umpan balik K dan konstanta F. Nilai K dan F yang diperoleh kemudian digunakan sebagai parameter kendali. Saat sistem dipasangkan pengendali LQR sistem dapat mencapai *setpoint* yang diinginkan dengan waktu yang cepat untuk mencapai kestabilannya, namun masih terdapat *over shoot*



yang cukup besar yaitu 45% dan saat ditambah metode *pole placement* dihasilkan *overshoot* 0%. Pada saat sistem diberikan gangguan sistem masih *robust* menghadapi gangguan yang terjadi [9].

Penelitian yang dilakukan menggunakan teknik kendali optimal LQR untuk mengendalikan kecepatan pada motor DC. Setelah mendapatkan nilai yang tepat untuk Q dan R pada sistem, penelitian ini juga membandingkan hasil performansi sistem menggunakan metode LQR dan PID. Didapatkan hasil bahwa menggunakan pengendali LQR respon yang di hasilkan lebih baik dari pada respon PID. Pada PID sedikit lambat untuk mencapai *setpoint* yang ditentukan, selisihnya adalah  $63 \times 10 \text{ms} = 0.63$  detik [10].

Penelitian yang melakukan pengendalian pada sistem non-linear menggunakan pengendali *Fuzzy Gain Scheduling*. Sistem yang digunakan adalah sistem *magnetic levitation ball*. Yang mana sistem sebelum menggunakan pengendali sistem ini tidak stabil dan berosilasi. Namun, menggunakan pengendali *Fuzzy Gain Scheduling* sistem dapat mencapai *setpoint* dan stabil [11].

Penelitian yang melakukan pemodelan dan simulasi pada sistem *magnetic levitation ball* menggunakan pengendali PID. Hasil yang didapat dengan menggunakan pengendali konvensional PID ini sistem dapat mencapai keadaan stabil, namun pada saat sistem belum menggunakan pengendali sistem tidak dapat mencapai keadaan stabil. Namun, menggunakan PID konvensional ini sistem masih mengalami *overshoot* dan osilasi [5].

Berdasarkan referensi yang telah dikumpulkan, dapat disimpulkan bahwa pengendali LQR memiliki beberapa keunggulan dibandingkan pengendali PID, sehingga pengendali ini layak untuk digunakan dan dikembangkan. Berdasarkan hasil pada penelitian sebelumnya diketahui bahwa menggunakan pengendali LQR ini respon sistem yang dihasilkan dapat mencapai *setpoint* lebih cepat dan optimal. Oleh karena itu, penulis tertarik untuk mengembangkan penelitian tentang pengendalian posisi dengan menggunakan pengendali LQR pada sistem *magnetic levitation ball*.

## 2.2 *Magnetic Levitation Ball*

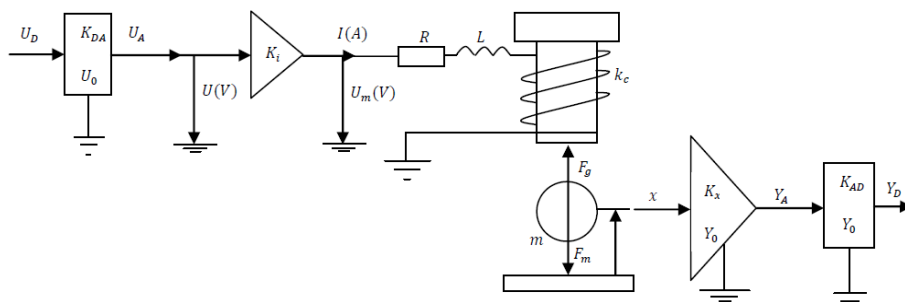
*Magnetic levitation Ball* adalah sebuah sistem yang terdiri atas bola baja *ferromagnetic* yang disuspensi dalam sebuah medan magnet oleh tegangan listrik dengan cara mengendalikan arus pada saat mengalir pada kumparan elektromagnet yang terdiri dari sejumlah lilitan tembaga dan posisi kumparan tepat berada diatas bola baja. Pada

sistem ini bola baja diasumsikan bergerak secara vertikal yaitu naik turun dan berhenti tepat pada posisi melayang dari posisi bola saat diletakkan dalam kondisi awal [2].

Cara termudah untuk melayangkan objek elektromagnetik dilihat dari sistem kendali adalah magnetik suspensi. Obyek yang akan dilayangkan ditempatkan di bawah sebuah elektromagnet. Dengan kekuatan medan magnet yang dihasilkan oleh elektromagnet yang dikendalikan dengan tepat akan melawan gaya gravitasi pada objek yang dilayangkan. Metode ini disebut teorema *circumvents Earnshaw* dengan menggunakan kendali umpan balik [1]. Dengan demikian sistem hanya akan melawan gaya gravitasi, sistem ini bekerja pada gaya tarik antara gaya elektromagnetik dan benda. Selain itu objek yang akan dilayangkan adalah bola baja biasa. Untuk mencegah bola baja menempel pada elektromagnet maka posisi benda harus bisa diperhitungkan posisi. Informasi dari sensor akan masuk pada rangkaian kendali yang akan mengatur arus dalam elektromagnet. Jika benda terlalu dekat dengan elektromagnet maka benda akan menempel maka arus yang masuk di electromagnet harus dikurangi, tetapi jika objek terlalu jauh maka arus pada electromagnet harus ditambah [1].

### 2.3 Sistem Modeling *Magnetic Levitation Ball*

Menurut tujuan dan kondisi operasi, sebuah pemodelan sangat penting untuk menemukan keseimbangan antara kesederhanaan dan kerumitan model, namun harus mempertimbangkan relevansi sub-proses dalam sistem. Dalam pemodelan sistem dinamis biasanya banyak perhitungan yang harus dilakukan sebelum model tersebut diperoleh, oleh karena itu diperlukan pendekatan lain yang dipertimbangkan. Pada bagian berikut, prosedur pemodelan akan dijelaskan.



Gambar 2.1 Sistem *magnetic levitation ball* [11]



Gambar 2.1 memperlihatkan model dan struktur pelayangan magnet pada bola baja (*magnetic levitation ball*). Adapun komponen dan struktur *Magnetic Levitation Ball* ini terdiri dari:

1. *Digital to Analog Converter* (DAC) yang berfungsi sebagai pengubah sinyal digital menjadi sinyal analog.
2. *Power amplifier* berfungsi sebagai sumber arus yang konstan.
3. Solenoida berfungsi sebagai magnet untuk melawan gaya gravitasi. Dan bola baja yang berfungsi sebagai benda yang akan dilayangkan.
4. Sensor Posisi berfungsi untuk menentukan jarak benda dengan elektromagnet.
5. *Analog to Digital Converter* (ADC) berfungsi untuk mengubah sinyal analog menjadi sinyal digital

### 2.3.1 D/A Converter

D/A *converter* adalah perangkat yang digunakan untuk mengkonversi sinyal masukan dalam bentuk digital menjadi sinyal keluaran dalam bentuk analog (tegangan) [9]. Tegangan keluaran yang dihasilkan D/A *converter* sebanding dengan nilai digital yang masuk ke dalam D/A *converter*. Sebuah D/A *converter* menerima informasi digital dan mentransformasikannya ke dalam bentuk suatu tegangan analog. Informasi digital adalah dalam bentuk angka biner dengan jumlah digit yang pasti.

D/A *converter* dapat ditunjukkan dengan fungsi linier, dimana model *output voltage*  $U_A$  , D/A *converter input*  $U_D$ , D/A *converter gain*  $K_{DA}$ , D/A *converter offset*  $U_0$  sebagai berikut :

$$U_A = K_{DA} * U_D + U_0 \tag{2.1}$$

Dari persamaan (2.1)  $U_0 = 0$

$$U_A = K_{DA} * U_D \tag{2.2}$$

Dimana:

$U_A$  = *output voltage* (V)

$U_D$  = *converter input*

$K_{DA}$  = *converter gain*

$U_0$  = *converter offset*

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hal-hal yang dihindari dalam penulisan karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

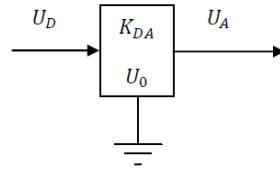
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

© Saiful M. Kasim, UIN Suska Riau

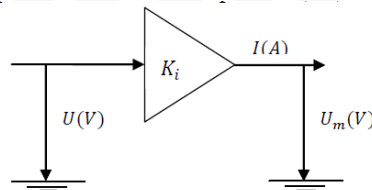
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.2 D/A converter [11]

### 2.3.2 Power Amplifier

Power amplifire dirancang sebagai sumber arus konstan dengan stabilisasi umpan balik arus untuk mengatasi perubahan arus yang terjadi pada magnet. Dimana arus  $I$ , amplifier gain  $K_i$ , input voltage  $U_A$ .



Gambar 2.3 Power Amplifier [11]

$$I = K_i * U_A \quad (2.3)$$

$$K_i = K_{am} / ((R_C + R_s) + R_s * K_s * K_{am}) \quad (2.4)$$

Dimana:

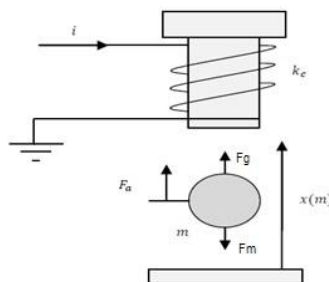
$I$  = arus (A)

$U_A$  = input voltage (V)

$K_i$  = amplifier gain

$K_{am}$  = power amplifier gain

### 2.3.3 Ball and Coil Subsystem



Gambar 2.4 Ball and Coil Subsystem [11]

Persamaan gerak didasarkan pada keseimbangan dari semua gaya yang bekerja pada bola baja, yang dimana ada tiga gaya yaitu gaya gravitasi  $F_g$ , gaya elektromagnetic  $F_m$  dan gaya percepatan  $F_a$ . Pada gambar 2.4 persamaan akan diturunkan dimana lilitan konstan  $k_c$ , position offset  $x_{L0}$ , dan redaman konstanta  $k_{fv}$ . Menurut hukum newton 2 yang berbunyi “ percepatan suatu benda akan sebanding dengan jumlah gaya (resultan gaya) yang bekerja pada benda tersebut dan berbanding terbalik dengan massanya” maka

Gaya total:

$$F_a = F_m - F_g \quad (2.5)$$

Dimana, gaya magnetic:

$$F_m = \frac{i^2 k_c}{(x - x_{L0})^2} \quad (2.6)$$

Gaya gravitasi:

$$F_g = mg \quad (2.7)$$

Dan gaya percepatan:

$$F_a = m\ddot{x} \quad (2.8)$$

Substitusikan persamaan (2.6),(2.7),dan (2.8), kedalam persamaan (2.5) :

$$m\ddot{x} = \frac{i^2 k_c}{(x - x_{L0})^2} - mg \quad (2.9)$$

Ketika bola redaman diperhitungkan, maka  $(k_{fv}\dot{x})$  dimasukkan kedalam persamaan (2.9) yang menjadi,

$$m\ddot{x} + k_{fv}\dot{x} = \frac{i^2 k_c}{(x - x_{L0})^2} - mg \quad (2.10)$$

Karena  $i = k_i U_A$ , maka substitusikan persamaan (2.3) ke dalam persamaan (2.10) sehingga;

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$m\ddot{x} = \frac{k_c Ki^2 U_A^2}{(x - x_{L0})^2} - mg - k_{fv} \dot{x} \quad (2.11)$$

$$\ddot{x} = \frac{k_c Ki^2 U_A^2}{m(x - x_{L0})^2} - \frac{k_{fv} \dot{x}}{m} - \frac{mg}{m} = \frac{k_c Ki^2 U_A^2}{m(x - x_{L0})^2} - \frac{k_{fv} \dot{x}}{m} - g \quad (2.12)$$

Dimana :

$F_a$  = gaya percepatan (N)

$F_m$  = gaya elektromagnetik (N)

$F_g$  = gaya gravitasi (N)

$F_d$  = gaya peredam (N)

$K_c$  = lilitan konstan

$U_A$  = input voltage (V)

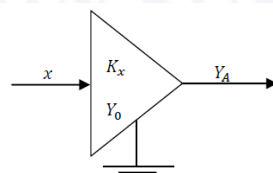
$K_{fv}$  = redaman konstan

$x_{L0}$  = position offset

### 2.3.4 Sensor Posisi

Dalam sistem *magnetic levitation ball* ini dibutuhkan pengukuran posisi bola, pada gambar 2.5 yang digunakan untuk mengukur bola adalah tegangan output  $Y_A$ , posisi bola  $x$ , posisi sensor gain  $K_x$ , dan posisi sensor *offset*  $Y_0$ . Sensor posisi dapat didefinisikan dalam persamaan berikut

$$Y_A = K_x x + Y_0 \quad (2.13)$$



Gambar 2.5 Sensor posisi [11]

Dimana:

$Y_A$  = tegangan output (V)

$x$  = posisi bola (m)

$K_s$  = posisi sensor gain

$Y_0$  = posisi sensor *offset*

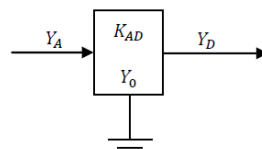
### 2.3.5 A/D Converter

A/D converter ditampilkan jika gambar 2.6, mempunyai A/D converter output  $Y_D$ , A/D converter input  $Y_A$ , A/D converter gain  $K_{AD}$ , dan A/D converter offset  $Y_0$ . A/D converter dapat didefinisikan dalam persamaan (2.12)

$$Y_D = K_{AD}Y_A + Y_0 \quad (2.14)$$

Dimana  $Y_0 = 0$

$$Y_D = K_{AD}Y_A \quad (2.15)$$



Gambar 2.6 A/D Converter [11]

### 2.4 Model Matematika *Magnetic Levitation*

Dari persamaan (2.10), kita dapat menyimpulkan sistem yang digunakan adalah sistem non-linier. Dengan menetapkan  $x_1 = x$  dan  $x_2 = \dot{x}$  sebagai variabel state dan model *state space* berasal dari persamaan (2.10) sebagai berikut

$$\begin{bmatrix} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{k_c K_i^2 U_A^2}{m(x_1 - x_{L0})^2} - \frac{k_{fv} x_2}{m} - g \end{bmatrix} \quad (2.16)$$

Dimana  $k_c K_i^2 = k_F$ , maka

$$\begin{bmatrix} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{k_F U_A^2}{m(x_1 - x_{L0})^2} - \frac{k_{fv} x_2}{m} - g \end{bmatrix} \quad (2.17)$$



Sekarang, kita ingin menlinierisasikan sistem diatas dengan keseimbangan  $(x_{ss}, u_{ss})$

dimana  $x_{ss} = \begin{pmatrix} x_{1,ss} \\ x_{2,ss} \end{pmatrix}$

Dalam keseimbangan, persamaannya sebagai berikut:

$$\dot{x}_1 = x_2 = 0 \tag{2.18}$$

$$\dot{x}_2 = \frac{k_F U_A^2}{m(x_1 - x_{L0})^2} - \frac{k_{fv}}{m} x_2 - g = 0 \tag{2.19}$$

$$\dot{x}_2 m = \frac{k_F U_A^2}{(x_1 - x_{L0})^2} - k_{fv} x_2 - mg = 0 \tag{2.20}$$

Dari persamaan (2.18) dan (2.20) kita dapat :

$$x_2 m = 0 \tag{2.21}$$

$$\frac{k_F U_A^2}{(x_1 - x_{L0})^2} - k_{fv}(0) - mg = 0 \tag{2.22}$$

Dari persamaan (2.21) dan (2.22) kita dapat :

$$\frac{k_F U_{ss}^2}{(x_{1,ss} - x_{L0})^2} - mg = 0 \tag{2.23}$$

$$\frac{k_F U_{ss}^2}{(x_{1,ss} - x_{L0})^2} = mg \tag{2.24}$$

Dari persamaan (2.24) maka:

$$U_{ss} = \sqrt{\frac{mg}{k_F}} (x_{1,ss} - x_{L0}) \tag{2.25}$$

Dengan menggunakan *taylor series*, untuk persamaan linierisasi diatas  $(x_{ss}, u_{ss})$  dapat ditulis sebagai berikut

$$f(x, u) = f(x_{ss}, u_{ss}) + \left( \frac{\partial(f(x_{ss}, u_{ss}))}{\partial(x_{ss}) * 1!} x(t) + \frac{\partial(f(x_{ss}, u_{ss}))}{\partial(u_{ss}) * 1!} u(t) \right) \tag{2.26}$$

Dengan menerapkan persamaan (2.26) untuk persamaan non linier  $\frac{k_F U_A^2}{m(x_1 - x_{L0})^2}$  pada persamaan (2.19) kita dapat :

$$k_F \left( \frac{U_A}{(x_{1ss} - x_{L0})} \right)^2 = \left( \frac{k_F u_{ss}^2}{(x_{1ss} - x_{L0})^2} \right) + \left( \frac{-2k_F u_{ss}^2}{(x_{1ss} - x_{L0})^3} \right) x_1(t) + \left( \frac{2k_F u_{ss}}{(x_{1ss} - x_{L0})} \right) u(t) \quad (2.27)$$

Substitusikan dari persamaan (2.24) ke persamaan (2.27) sehingga kita dapat :

$$k_F \left( \frac{U_A}{(x_{1ss} - x_{L0})} \right)^2 = mg + \left( \frac{-2k_F u_{ss}^2}{(x_{1ss} - x_{L0})^3} \right) x_1(t) + \left( \frac{2k_F u_{ss}}{(x_{1ss} - x_{L0})} \right) u(t) \quad (2.28)$$

Substitusikan persamaan (2.28) ke persamaan (2.20), maka

$$\ddot{x}_2 = \left( \frac{-2k_F u_{ss}^2}{m(x_{1ss} - x_{L0})^3} \right) x_1 - \left( \frac{k_{fv}}{m} \right) \dot{x}_2 + \left( \frac{2k_F u_{ss}}{m(x_{1ss} - x_{L0})^2} \right) u(t) \quad (2.29)$$

Karena dari persamaan (2.2)  $U_A = K_{DA} * U_D$  dan dari persamaan (2.17) dan (2.29), model *state space* dapat ditulis sebagai berikut

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-2k_F u_{ss}^2}{m(x_{1ss} - x_{L0})^3} & -\frac{k_{fv}}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_F K_{DA} u_{ss}}{m(x_{1ss} - x_{L0})^2} \end{bmatrix} U_D \quad (2.30)$$

$$y = \begin{bmatrix} k_x * K_{AD} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Setelah mensubstitusikan nilai  $u_{ss}$  dari persamaan (2.25) ke model *state space* (2.30), model *state space* menjadi :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{2g}{m(x_{L0} - x_{1ss})^3} & -\frac{k_{fv}}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2\sqrt{\frac{k_F g}{m}} \frac{K_{DA}}{(x_{L0} - x_{1ss})} \end{bmatrix} U_D \quad (2.31)$$

$$y = \begin{bmatrix} k_x * K_{AD} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Tabel 2.1 Nilai-nilai parameter dari *magnetic ball levitation* [11]

Parameter	Symbol	Value
Ball diameter	$D_k$	12.7e-3m
Ball mass	$M$	0.0084kg
Jarak dari dasar	$T_d$	0.0019 m
Batas jarak = 0.0019 - $D_k$	$L$	0.0063 m
Viscose friction	$k_{fv}$	0.02 N.s/m



Gravity Acceleration constant	$G$	$9.81\text{m/s}^2$
Aggregated coil constant	$k_F$	$0.606 \times 10^{-6} \text{N/V}$
Converter gain	$K_{DA}$	10
Coil bias	$x_{L0}$	$8.26 \times 10^{-3} \text{m}$
Position sensor constant	$k_x$	797.4603
Analog to digital converter gain	$K_{AD}$	0.2
Maximum DA converter output voltage	$U_{\text{Dam}}$	5 V
Coil resistance	$R_c$	$3.5 \Omega$
Coil inductance	$L_c$	$30 \times 10^{-3} \text{H}$
Current sensor gain	$K_s$	13.33
Power amplifier gain	$K_{am}$	100

Setelah mensubstitusikan nilai-nilai parameter dari *magnetic ball levitation* dari tabel 2.1, *state space* (2.31) menjadi

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{19.62}{(0.00826 - x_{1ss})} & -2.381 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.53206}{(0.00826 - x_{1ss})} \end{bmatrix} U_D \quad (2.32)$$

$$y = \begin{bmatrix} 159.49206 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

## 2.5 Sistem Kendali Optimal

Istilah optimal mengandung maksud hasil yang baik yang dapat dicapai dengan memperhatikan kondisi dan kendala dari suatu sistem [10]. Dalam sistem kendali optimal, istilah optimal sering kali merujuk pada minimal, misalnya meminimalkan bahan bakar, waktu dan kesalahan. Pada dasarnya, solusi kendali optimal dititik beratkan pada pencarian nilai sinyal kendali  $u(t)$  sehingga indeks performansi dapat dioptimasi. Nilai indeks performansi yang terbaik tidak akan di ketahui sampai dengan proses pengendalian selesai. Hal ini tergantung pada kondisi plant yang dikendalikan dan spesifik indeks performansi yang dioptimalkan.

Secara umum perancangan sistem kendali optimal mempunyai 2 tujuan (fungsi), yaitu: sebagai regulator (menstabilkan sistem dengan variabel/output agar tetap kecil) dan sebagai *Tracker/Servomechanism* (mengontrol sistem agar mengikuti trayektori dan keadaan selalu dalam batas-batas tertentu) [12]. Melihat hal tersebut kendali optimal menggunakan metode *Linear Quadratic Regulator* (LQR) dapat diterapkan pada sistem gerakan training laras meriam untuk menyelesaikan permasalahan regulator sistem pada



gerakan sudut yang berubah-ubah dan mendapatkan nilai optimal indeks performansi yang terbaik.

Sistem kendali yang baik adalah sistem kendali yang memiliki daya tanggap yang cepat dan stabil, tetapi tidak memerlukan energi yang berlebihan. Sistem yang demikian dapat dicapai melalui indeks performansi yang tepat. Berdasarkan optimasi indeks performansi inilah disebut sebagai sistem kendali optimal [10].

Pada suatu sistem indeks performansi dipilih sesuai dengan bagian yang akan dioptimalkan. Bentuk umum dari indeks adalah sebagai berikut :

$$J = \int_{t_0}^T L(x, u, t) dt \tag{2.33}$$

$J$  = indeks performansi

$L(x, u, t)$  = fungsi dari  $x$ ,  $u$  dan  $t$

$T$  = waktu

Salah satu metode yang biasa digunakan untuk meminimalkan indeks performansi yaitu persamaan *Aljabar Riccati*, yang digunakan untuk mengoptimalkan sistem proses yang berbentuk *linear*. Suatu sistem kendali akan optimal pada indeks unjuk kerja yang diberikan, tetapi tidak optimal lagi pada indeks unjuk kerja yang lain [10].

### 2.5.1 Linear Quadratic Regulator (LQR)

Pokok bahasan dalam kendali optimal adalah menentukan sinyal kendali yang akan di proses untuk memenuhi batasan fisik sesuai dengan kriteria performansi yang diinginkan. *Linear Quadratic Regulator* (LQR) merupakan salah satu metode kendali optimal dengan kriteria kuadratic untuk menyelesaikan permasalahan regulator. Disebut linier karena model dan bentuk kendalinya berupa sistem linear sedangkan disebut kuadratik karena memiliki *Cost Function* yaitu kuadrat dan karena referensi sistem bukan fungsi waktu maka disebut regulator [13]. Dari hasil linierisasi didapatkan suatu *plant* linear dalam bentuk :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.34}$$

Dimana,

$A$  : matriks sistem

$B$  : matriks *input*

$C$  : matriks *output*

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

© Himpunan Matematika UIN Suska Riau

- $y$  : State output
- $x$  : State sistem
- $u$  : State input

Pada perancangan pengendali optimal LQR, terlebih dahulu menentukan matriks Q dan R yang selanjutnya digunakan untuk menentukan indeks performansi sistem, harga matriks Q dan R ditentukan sesuai dengan kriteria yang diinginkan dengan menggunakan indeks performansi [14].

$$J(t_0) = \frac{1}{2} S(T)x^2(T) + \frac{1}{2} \int_{t_0}^T (Qx^2 + Ru^2) \quad (2.35)$$

Dengan syarat :

$$S(T) \geq 0, \quad Q \geq 0, \quad R > 0$$

dimana

- $t_0$  = waktu awal
- $T$  = waktu akhir
- $x$  = matriks State akhir
- $Q$  = matriks semi definit positif
- $R$  = matriks definit positif
- $S$  = matriks semi definit positif

Sehingga diperoleh persamaan Hamilton

$$H(x, u, \lambda, t) = L(x, u, t) + \lambda^T f(x, u, t)$$

$$H(x, u, \lambda, t) = \frac{1}{2} (x^T Qx + u^T Ru) + \lambda^T (Ax + Bu) \quad (2.36)$$

Dari fungsi Hamilton tersebut dapat diperoleh syarat perlu dan syarat batas sebagai berikut:

1. Syarat perlu

a) Persamaan State

$$\dot{x} = \left( \frac{\partial x}{\partial x} \right)^T Qx + \left( \frac{\partial Qx}{\partial \lambda} \right)^T u = Ax + Bu \quad (2.37)$$

b) Persamaan Costate

$$\dot{\lambda} = \frac{1}{2} \left[ \left( \frac{\partial x}{\partial x} \right)^T Qx + \left( \frac{\partial Qx}{\partial x} \right)^T x + \left( \frac{\partial u}{\partial x} \right)^T Ru + \left( \frac{\partial Ru}{\partial x} \right)^T u \right] + \left( \frac{\partial \lambda}{\partial x} \right)^T (Ax + Bu) + \left( \frac{\partial (Ax + Bu)}{\partial x} \right)^T \lambda$$

$$\dot{\lambda} = Qx + A^T \lambda \quad (2.38)$$

c) Kondisi stasioner

$$\frac{\partial H}{\partial u} = 0$$

$$\frac{1}{2} \left[ \left( \frac{\partial x}{\partial x} \right)^T Qx + \left( \frac{\partial Qx}{\partial x} \right)^T x + \left( \frac{\partial u}{\partial x} \right)^T Ru + \left( \frac{\partial Ru}{\partial x} \right)^T u \right] + \left( \frac{\partial x}{\partial u} \right)^T (Ax + Bu) + \left( \frac{\partial (Ax + Bu)}{\partial u} \right)^T \lambda = 0$$

$$Ru + B^T \lambda = 0$$

$$u = -R^{-1} B^T \lambda \tag{2.39}$$

2. Syarat batas

a. Batas awal

$$t = 0 \quad x(0) = 0$$

b. Batas akhir

$$(\phi_x + \psi_x^T v - \lambda)^T dx|_T + (\phi_t + \psi_t^T v - H) dt|_{t=T} = 0 \tag{2.40}$$

Karena  $\psi = 0$  dan

$$\phi(T) = \frac{1}{2} x^T(T) S(T) x(T)$$

$$\phi_x = \frac{\partial \phi}{\partial x} = S(T) x(T)$$

sehingga diperoleh persamaan

$$(S(T) x(T) + v \cdot 0 - \lambda)^T dt|_{t=T} = 0$$

$$S(t) x(t) = \lambda(t) \tag{2.41}$$

Dari persamaan *Costate* dan (2.36), keduanya mengandung variabel  $\lambda$ . Bila keduanya disubstitusikan maka akan menghasilkan persamaan:

$$S(t) x(t) = \lambda(t)$$

$$\dot{\lambda}(t) = \dot{S}(t) x(t) + S(t) \dot{x}(t)$$

$$\dot{\lambda}(t) = -Qx - A^T \lambda$$

$$\dot{S}(t) x(t) + S(t) \dot{x}(t) = -Qx - A^T \lambda$$

$$\dot{S}(t) x(t) + S(t) \{Ax + Bu\} = -Qx - A^T \lambda$$

$$\dot{S}(t) x(t) + S(t) \{Ax + B(-R^{-1} B^T \lambda)\} = -Qx - A^T \lambda$$

$$-\dot{S}x = (A^T S + SA - SBR^{-1} B^T S + Q)x$$

$$-\dot{S} = A^T S + SA - SBR^{-1}B^T S + Q \quad (2.42)$$

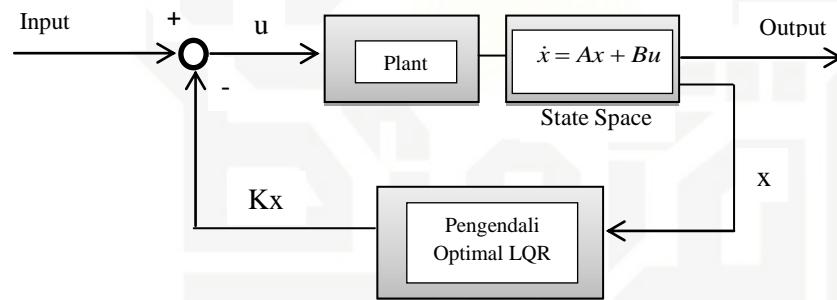
Persamaan (2.42) disebut persamaan differensial Riccati. Untuk  $\dot{S} = 0$  persamaannya disebut *Algebraic Riccati Equation* (ARE) [14]. Dengan di dapatkannya matriks S dari persamaan ARE ini sehingga persamaan sinyal kendali pun dapat di hitung:

$$\bar{u} = -R^{-1}B^T \lambda = -R^{-1}B^T Sx = -Kx \quad (2.43)$$

Maka konstanta umpan balik keadaan K dapat dicari sebelum menghasilkan sinyal kendali pada persamaan (2.34) [14].

$$K = R^{-1}B^T S \quad (2.44)$$

Dimana nilai S adalah unik, solusi semi definit positif untuk persamaan ARE harus memenuhi syarat persamaan (2.42). Selanjutnya dalam perancangan teknik kendali optimal LQR, Setelah nilai S diketahui maka, nilai S tersebut disubstitusikan kedalam persamaan (2.44) sehingga didapatkan nilai matriks optimal K, dengan K adalah *Gain State Feedback*, sehingga sistem kendali optimal LQR hasil desain dapat di perlihatkan pada gambar 2.7 berikut:



Gambar 2.7 Sistem Kendali dengan Skema Kendali Optimal LQR [15]

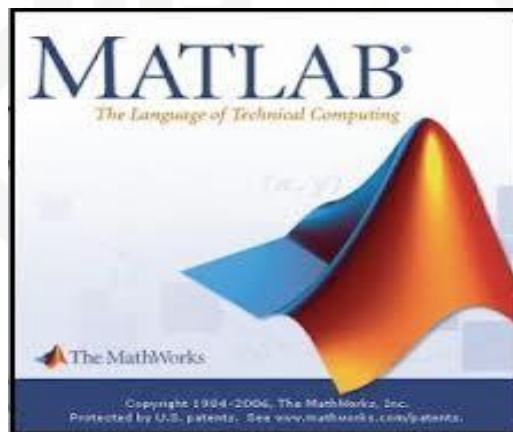
Dari penjabaran di atas dapat disimpulkan bahwa langkah-langkah untuk merancang kendali LQR adalah sebagai berikut :

1. Jika persamaan matematis masih dalam bentuk *transfer function*, ubah bentuk *transfer function plant* nonlinier menjadi bentuk *state space* dan dapatkan matriks A, B, C dan D dari *plant* tersebut.
2. Tentukan matriks pembobot Q dan R.
3. Selesaikan persamaan aljabar riccati hingga menghasilkan suatu matriks S yang definit positif
4. Hitung optimal *gain feedback* K



## 2.6 MATRIX LABORATORY (MATLAB)

MATLAB adalah singkatan dari MATrix LABoratory. Pertama kali dibuat untuk mempermudah penggunaan dua koleksi subrutin pada pustaka FORTRAN yaitu: LINPACK dan EISPACK, dalam menangani komputasi matriks. Sejak itu, MATLAB berkembang menjadi sebuah sistem yang interaktif sekaligus sebagai bahasa pemrograman untuk keperluan-keperluan ilmiah, komputasi teknis, dan visualisasi. Elemen data dasar MATLAB adalah matriks. Perintah-perintah diekspresikan dalam bentuk yang sangat mirip dengan bentuk yang digunakan dalam matematika dan bidang teknik. Contoh persamaan  $b=Ax$ , dengan A, b, dan x matriks, ditulis:  $b=A*x$ . Untuk mendapat solusi x dari A dan b, tulis:  $x=A\b$ . Tidak diperlukan penulisan program khusus untuk operasi-operasi matriks seperti perkalian matriks atau invers matriks. Oleh karena itu bahasa MATLAB menyelesaikan masalah tersebut memerlukan waktu lebih cepat dibanding waktu yang dibutuhkan bahasa pemrograman tingkat tinggi lain.



Gambar 2.8 Tampilan Matlab

Pada pertengahan tahun 1970, Cleve Moler dan beberapa rekan tergabung dalam suatu team pengembangan *software* yang dibiayai oleh *The National Science Foundation* untuk tujuan membuat subrutin-subrutin dalam pustaka FORTRAN yang dinamai LINPACK dan EISPACK. LINPACK berisi koleksi subrutin untuk penyelesaian persamaan linear, sementara EISPACK adalah koleksi subrutin untuk penyelesaian masalah nilai pribadi (*eigenvalue*). Baik LINPACK maupun EISPACK pada prinsipnya merupakan program untuk komputasi matriks.

Dipenghujung tahun 1970, Cleve ingin dapat mengajarkan kepada mahasiswa materi aljabar linear di Universitas New Mexico menggunakan LINPACK dan EISPACK tanpa harus menulis rutin-rutin program dalam bahasa FORTRAN. Berdasarkan keinginan tersebut, Cleve mulai menulis program untuk memberikan kemudahan akses interaktif





pada LINPACK dan EISPACK. Cleve menamakan programnya dengan MATLAB yang merupakan singkatan dari MATrix LABoratory. Beberapa tahun kemudian, ketika Cleve berkunjung ke universitas lain untuk berbicara, atau sebagai *Visiting Professor*, Cleve meninggalkan duplikasi MATLABnya pada komputer di universitas tersebut. Hanya dalam satu atau dua tahun, MATLAB versi pertama ini telah menjadi buah bibir pembicaraan orang, terutama yang berada dalam komunitas matematika terapan.

Dari hasil kunjungan Cleve di Universitas Stanford, sekitar awal tahun 1983, John Little, seorang *engineer*, menampilkan MATLAB dengan memperkenalkan penerapan MATLAB yang potensial dalam bidang-bidang keteknikan. Karena itu, dalam tahun 1983, Little, Moler, dan Steve Bangert membentuk team untuk mengembangkan MATLAB generasi kedua. MATLAB versi ini dibuat menggunakan bahasa C dan terintegrasi dengan grafik. *The MathWorks, Inc.* didirikan tahun 1984 untuk memasarkan dan melanjutkan pengembangan MATLAB.

Penggunaan MATLAB:

1. Adanya *toolbox* yang disebut MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer
2. Komputasi matematik dan komputasi teknik
3. Analisis data.
4. Simulasi dan pemodelan
5. Grafik-grafik perhitungan.
6. Manipulasi matriks
7. Pem-plot-an fungsi dan data
8. Implementasi algoritma
9. Pembuatan antarmuka pengguna / *interface* (visualisasi )
10. Peng-antarmuka-an dengan program dalam bahasa lainnya

Fungsi-fungsi yang digunakan MATLAB:

1. Fungsi matematika
2. Fungsi fisika
3. Fungsi statistik
4. Fungsi visualisasi.



MATLAB merupakan *software* yang digunakan untuk:

1. Para pendidikan (untuk melakukan riset dalam bentuk komputasi, pemodelan, simulasi dan demonstrasi / visualisasi)
2. Para ilmuwan
3. Matematikawan
4. Pelajar dan mahasiswa terutama para pendidik matematika
5. Praktisi
6. Insinyur.
7. dan untuk siapa saja tanpa ada batasannya

Sebagai sebuah system, MATLAB tersusun dari 5 bagian utama:

1. *Development Environment*.

Merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah *graphical user interfaces* (GUI). Termasuk didalamnya adalah MATLAB desktop dan *Command Window*, *command history*, sebuah *editor* dan *debugger*, dan *browsers* untuk melihat *help*, *workspace*, *files*, dan *search path*.

2. *MATLAB Mathematical Function Library*

Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: *sum*, *sin*, *cos*, dan *complex arithmetic*, sampai dengan fungsi-fungsi yang lebih kompleks seperti *matrix inverse*, *matrix eigenvalues*, *Bessel functions*, dan *fast Fourier transforms*.

3. *MATLAB Language*

Merupakan suatu *high-level matrix/array language* dengan *control flow statements*, *functions*, *data structures*, *input/output*, dan fitur-fitur *object-oriented programming*. Inimemungkinkan bagi kita untuk melakukan kedua hal baik “pemrograman dalam lingkupsederhana ” untuk mendapatkan hasil yang cepat, dan “pemrograman dalam lingkup yang lebihbesar” untuk memperoleh hasil-hasil dan aplikasi yang kompleks.

4. *Graphics*

MATLAB memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatu grafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, *image processing*, *animation*, dan



*presentation graphics*. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi anda untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan *graphical user interfaces* pada aplikasi MATLAB anda.

### 5. MATLAB Application Program Interface (API)

Merupakan suatu *library* yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinteraksi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan *routines* dari MATLAB (*dynamic linking*), pemanggilan MATLAB sebagai sebuah *computational engine*, dan untuk membaca dan menuliskan *MAT-files*. Kehadiran Matlab memberikan jawaban sekaligus tantangan. Disini kita bisa memilih apa aja yg ingin kita pelajari, mempelajari metoda visualisasi saja, pemrograman saja, atau keduanya di sini intinya Matlab memang dihadirkan bagi mereka yang tidak ingin disibukkan dengan rumitnya sintak dan alur logika pemrograman sementara pada saat yang sama membutuhkan hasil komputasi dan visualisasi yang maksimal untuk mendukung pekerjaannya. Selain itu Matlab juga memberikan keuntungan bagi *programmer* dan *developer* program yaitu menjadi program pembanding yang sangat handal, hal tersebut karena kekayaannya akan fungsi matematika, fisika, statistik dan visualisasi.

## 2.7 Identifikasi Sistem

Identifikasi sistem yang digunakan dalam penelitian ini adalah metode pengamatan respon waktu. Metode ini mengidentifikasi sistem berdasarkan pengamatan grafis terhadap masukan *step*. Dimana sinyal uji diberikan kepada sistem untuk mengetahui respon sistem *open loop*. Dari respon sistem, dapat diketahui karakteristik-karakteristik penting dari sistem. Sistem yang digunakan pada pengaturan posisi pada *magnetic levitation ball* merupakan sistem orde 2 yang ditunjukkan pada gambar 2.9 sebagai berikut :

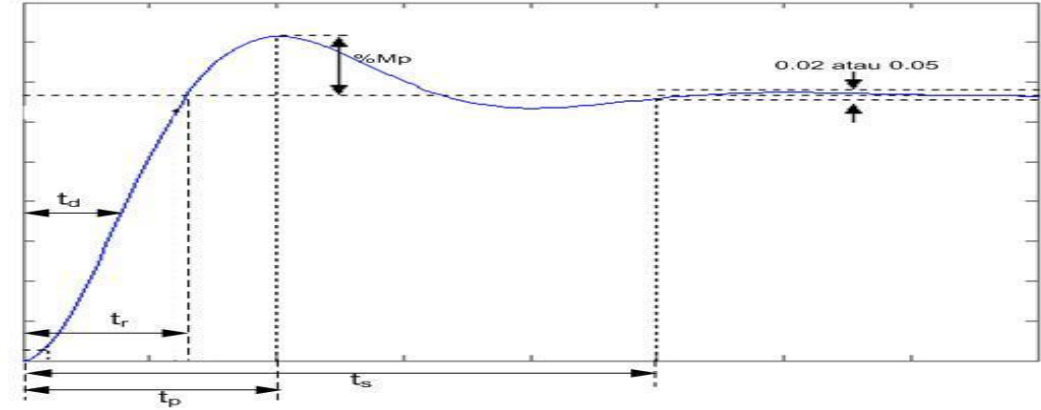
### Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber.

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.9 Respon Sistem Orde Kedua [15]

**Karakteristik Orde 2:**

Respon transien terdiri dari:

1. Waktu tunak atau *settling time* ( $t_s$ ), adalah ukuran waktu yang menyatakan bahwa respon sistem telah masuk pada daerah stabil yang ukurannya ditentukan dengan persentase mutlak dari harga akhir (5% atau 2%)
2. Waktu naik atau *rise time* ( $t_r$ ), adalah ukuran waktu yang menyatakan bahwa respon sistem telah naik dari 5% ke 95% atau 10% ke 90% dari nilai respon pada keadaan tunak (*steady state*). Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_r$  dapat diformulasikan sebagai berikut :
3. Waktu tunda atau *delay time* ( $t_d$ ), adalah waktu yang dibutuhkan respon mulai  $t=0$  sampai respon mencapai 50% dari nilainya pada keadaan tunak (*steady state*). Waktu tunda menyatakan besarnya faktor keterlambatan respon akibat proses *sampling*. Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_d$  dapat diformulasikan sebagai berikut :  $t_d (\pm 3\%) \approx \tau \ln 12$
4. Waktu puncak ( $t_p$ ): Waktu yang diperlukan respon untuk mencapai puncak pertama overshoot.
5. Maksimum *overshoot* ( $M_p$ ): nilai yang menyatakan perbandingan antara nilai maksimum respon (*overshoot*) yang melampaui nilai *steady state* dibanding dengan nilai *steady state*. Besarnya persen *overshoot* dinyatakan sebagai berikut:

$$\% M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \tag{2.47}$$



**2.8 Kriteria Integral Menggunakan IAE (*Integral of Absolute Error*)**

Kriteria integrasi membutuhkan data tanggapan mulai dari  $t = 0$  hingga mencapai keadaan tunak. Dengan demikian kriteria ini didasarkan pada seluruh tanggapan dari proses yang bersangkutan. Kriteria integral yang digunakan pada penelitian ini adalah kriteria IAE (*Integral of Absolute Error*) yang bertujuan untuk mendapatkan nilai IAE sekecil mungkin dengan pemilihan kriteria tergantung pada karakteristik sistem proses dan beberapa syarat tambahan yang diperoleh dari tanggapan lup tertutup. Kriteria IAE lebih dipopuler dikalangan praktisi industri sebab mudah dalam pemakaiannya sertasangat cocok untuk menekan error yang kecil.

$$IAE = \int_0^{\infty} |e| dt \tag{2.48}$$

Hak Cipta Dilindungi Undang-Undang  
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.