

BAB II

LANDASAN TEORI

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

Beberapa penelitian terdahulu yang terkait dengan sistem *ball and beam* diantaranya: Penelitian dengan judul *Fractional Order Feedback Control of a Ball and Beam System*. Hasil penelitian menunjukkan kendali PID dapat diterapkan pada sistem *ball and beam* dan menghasilkan kondisi kestabilan yang diinginkan, tapi terdapat *overshoot* dan lambat mencapai *setpoint*, sistem respon dengan *fractional order PID controller* menghasilkan respon yang lebih baik dari *PID controller classic* dinyatakan dengan *PID fractional* menghasilkan waktu *rise time* pada detik ke 2 dan *overshoot* sebesar 7%, sedangkan *PID controller classic* menghasilkan *rise time* pada detik ke 8 dan *overshoot* sebesar 15% [2].

Kemudian penelitian dengan judul *Modeling and control of ball and beam system using model based and non-linear based control approaches* menggunakan kombinasi *PID-LQR*. Hasil keluran menunjukkan kestabilan yang diinginkan, tetapi masih terdapat *robust dan error* yang belum stabil. Hasil keluaran *PID-LQR Settling Time (5%) 3.958 (sec) steady state error 11%* [3]

Selanjutnya penelitian sistem *ball and beam* menggunakan *Coefficient Diagram Method (CDM) - PID* dan *Ziegler Nicolas (ZN)-PID*. Hasil menunjukkan bahwa kontroler *CDM-PID* dapat mempertahankan stabilitas yang baik dari posisi bola dengan persentase yang lebih rendah dari pada *ZN-PID* kontroler. Tujuan penelitian ini adalah membandingkan metode penalaan *Ziegler-Nichols PID* dengan metode *CDM-PID*. Dari hasil simulasi, dapat diamati bahwa kontroler *CDM-PID* dapat meredam gangguan dalam waktu yang lebih singkat dengan puncak yang lebih sedikit dan tidak terpengaruh oleh variasi gain sedangkan *ZN-PID* dipengaruhi oleh *non linearitas*. Didapatkan analisa *CDM-PID* adalah $T_s = 1.18$ detik dengan analisa $IAE = 121$ sedangkan untuk *ZN-PID* $T_s = 15.38$ detik dengan analisa $IAE = 267$. [4].

Kombinasi *sliding mode control* (SMC) dengan optimasi *Proportional Integral Derivative* (PID) sebelumnya telah dilakukan pada beberapa sistem diantaranya yaitu: Perancangan *sliding mode control* untuk menggerakkan posisi laras meriam pada *platform* yang bergerak [5]. Perancangan kendali SMC untuk mengendalikan posisi laras meriam sehingga gerakan *training* dan *elevasi* meriam dapat dikendalikan pergerakannya dalam pencapaian posisi sudut. *Chattering* terjadi pada fungsi *switching control*. Efek *chattering* ini dieliminasi dengan menghaluskan diskontinuitas kontrol pada sebuah *layer* tipis berbatas (*boundary layer*) yang berda di sekitar *switching surface* yaitu dengan mengganti fungsi *Signum* (Sgn) menjadi *Saturasi* (Sat). Hasil simulasi dan eksperimen, diperoleh bahwa pengaturan posisi laras dengan SMC mampu mengikuti lintasan yang diinginkan dan mampu menekan *error* yang timbul ketika terdapat ketidaktepatan pengukuran parameter meriam dan juga dalam mengatasi gangguan dengan *free chattering*.

Penelitian lain dengan judul Analisis dan simulasi pengendali robot polar derajat kebebasan dua menggunakan *sliding mode control*. Mengatakan SMC merupakan kendali yang stabil dan tahan terhadap gangguan-gangguan baik dari dalam dan luar sistem. Penggunaan SMC dalam penelitiannya untuk menstabilkan gerak robot polar derajat kebebasan dua yang difokuskan pada kontrol posisi pusat lengan dan posisi anguler lengan yang timbul akibat pergerakan manipulator yang bergerak memutar dan memanjang atau memendek. Hasil penelitiannya menunjukkan setelah sistem diberi pengendali, terdapat perbedaan grafik yang merupakan respon dari fungsi gangguan dan perubahan parameter, tetapi lintasan yang dihasilkan stabil dan sesuai dengan lintasan yang diharapkan [6]

Kemudian penelitian dengan judul *Fuzzy sliding mode control of a magnetic ball suspension*. Pada penelitian ini hasil eksperimen menunjukkan *Metode sliding mode control* memiliki *chattering* yang cukup besar. Memperkecil *chattering* yang disebabkan frekuensi tinggi dapat digunakan *metode adaptive fuzzy estimator sliding mode control*. *Adaptive fuzzy estimator sliding mode control* memiliki kinerja yang lebih baik dari pada *sliding mode control* saat diberi gangguan (*disturbance*)[7].

Pengendalian posisi pada *magnetic levitation ball* dengan pengendali *sliding mode* dengan optimasi PID menunjukkan performansi yang baik. Pengendali *sliding mode* dengan optimasi PID berhasil mencapai kestabilan yang baik menggunakan *metode trial and error* dan analisa IAE dengan nilai $\eta = 4$ dan parameter PID $K_p: 0.5; K_i = 0.2; K_d = 0.3$. Hal ini

terbukti dari hasil visualisasi dan analitik yang dilakukan. Didapatkan analisa respon waktu pada saat memasuki daerah stabil yaitu $T_s = 0.1455$ detik dengan analisa $IAE = 0.000505$. [9].

Pada penelitian yang melakukan simulasi desain SMC-PID pada *Rotary Pendulum*. Penggunaan kendali *sliding*-PID menghasilkan respon yang lebih baik dibandingkan dengan kendali PID. Hal tersebut dapat ditunjukkan dengan nilai maksimum *overshoot* pada kendali Sliding-PID (0% untuk sudut α dan sudut θ) lebih kecil dari pada kendali PID (9.4664% untuk sudut α dan 7.7107% untuk sudut θ). Sedangkan waktu yang diperlukan untuk seluruh sistem *rotary pendulum* (untuk menstabilkan sudut α dan sudut θ) dengan kendali Sliding-PID (5.8591 detik) lebih besar jika dibandingkan dengan kendali PID (0.5190 detik). Selain itu *steady state error* dari kendali Sliding-PID (4.94%) lebih besar daripada kendali PID (4.81%) [10]

Penelitian pada pendulum ganda kereta bergerak menggunakan kendali SMC-PID. Dari hasil simulasi yang telah dilakukannya, dapat disimpulkan bahwa penggunaan kendali Sliding-PID menghasilkan respon yang lebih baik dibandingkan menggunakan kendali PID. Untuk menstabilkan sistem pendulum ganda secara keseluruhan, kendali PID memerlukan waktu selama 25,70 detik. Sedangkan kendali SMC-PID hanya memerlukan waktu selama 14,54 detik atau 43.42 % lebih baik daripada kendali PID. Selain itu kendali SMC-PID juga mampu mengurangi *steady state error* serta *overshoot* maksimum yang terjadi pada kendali PID [11].

Berdasarkan dari penelitian dengan judul Pengendali *hybrid* SMC dan PID pada Sistem *Non Linear Multivariabel Continuous Stirred Tank Reactor* (CSTR). Berdasarkan dari hasil simulasi penelitiannya menyatakan bahwa kelemahan pada pengendali SMC adalah *chattering* yang dapat menimbulkan *error steady state*. Sehingga menambahkan pengendali PID dimana aksi kendali *integral* yang mempunyai keunggulan dapat menghilangkan *error steady state* sehingga menghasilkan performansi sistem yang lebih baik [8]

Pada penelitian yang membandingkan pengendali PI, PID, SMC, dan SMC PID menyimpulkan bahwa pengendali SMC tidak dapat mempertahankan frekuensi dengan baik pada waktu yang berbeda-beda. Untuk menangani hal ini maka digunakan DC-DC konverter untuk mempertahankan frekuensi dengan baik dengan cara menggabungkan pengendali PID dan SMC. Dengan menggunakan masing-masing pengendali mendapatkan

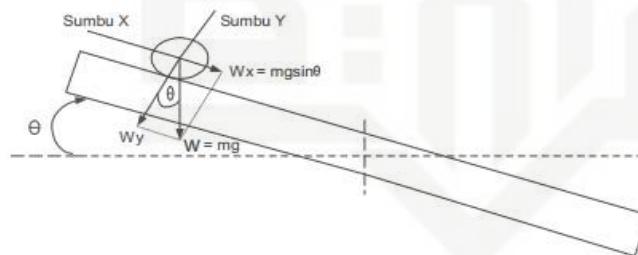
hasil keluaran sistem yang lambat. Setelah melakukan penggabungan kendali SMC PID menghasilkan keluaran sistem yang terbaik. [12]

Berdasarkan kajian pustaka diatas menunjukkan bahwa pengendali pengendali SMC-PID memiliki keunggulan yang telah dibuktikan dari beberapa penelitian-penelitian pada beberapa sistem yang telah menggunakan pengendali SMC-PID. Sehingga penelitian ini diharapkan mampu memberikan nilai *set point* dan keandalan yang lebih baik pada sistem *ball and beam* yang akan dikendalikan menggunakan pengendali SMC dengan optimasi PID. Dengan demikian penulis mengajukan judul tugas akhir **Desain pengendali SMC dengan kombinasi PID untuk mengendalikan posisi pada sistem *ball and beam*.**

2.2. Landasan Teori

2.2.1. Sistem *Ball and Beam*

Sistem *ball and beam* adalah salah satu sistem yang sangat penting dalam mempelajari sistem kontrol. Pergerakan bola pada bidang miring yang merupakan model dari *ball and beam* [1] Pada sistem *ball and beam*, pergerakan bola dipengaruhi oleh besarnya gaya yang ditimbulkan oleh kemiringan batang tempat bola berada, dimana untuk memiringkan batang digunakan motor yang akan menggerakkan batang tersebut sesuai dengan posisi bola dapat dilihat pada Gambar 2.1



Gambar 2.1. Model *Ball and Beam*[1]

2.2.2. DC Servo Motor Dynamic Model

Pemodelan *servomotor* DC dapat dibagi menjadi listrik dan dua subsistem mekanis. Subsistem listrik berbasis pada hukum tegangan Kirchhoff:

$$LmIm + RmIm + Kb\theta = U \tag{2.1}$$

Dimana U adalah tegangan *input*, Im adalah arus, Rm dan Lm adalah resistansi dan induktansi, Kb adalah konstanta emf dan θ adalah kecepatan sudut. Karena dibandingkan

dengan RmI_m dan $K_b \theta$, istilah $L_m I_m$ sangat kecil, oleh karena itu untuk mempermudah pemodelan. Persamaan (2.1) menjadi

$$RmI_m + Kb\theta = U \quad (2.2)$$

Subsistem mekanis diberikan oleh

$$\frac{1}{K_g} (J_m \ddot{\theta} + B_m \dot{\theta}) = \tau_m \quad (2.3)$$

Dimana K_g adalah rasio *gearbox*, J_m adalah momen efektif *inersia*, B_m adalah koefisien gesekan viskos, τ_m adalah torsi di poros motor. Subsistem listrik dan mekanik digabungkan ke masing-masing lainnya melalui persamaan torsi aljabar

$$\tau_m = k_m I_m \quad (2.4)$$

Dimana K_m adalah konstanta torsi motor menggunakan (2.4), (2.3) dapat ditulis sebagai

$$\frac{1}{K_g} (J_m \ddot{\theta} + B_m \dot{\theta}) = k_m I_m$$

$$I_m = \frac{(J_m \ddot{\theta} + B_m \dot{\theta})}{K_m k_g} \quad (2.5)$$

Dengan mensubstitusikan nilai I_m ke (2.2), persamaan *diferensial* untuk model motor DC diperoleh sebagai

$$\frac{Rm + Bm}{K_m k_g} \ddot{\theta} + \left(K_b + \frac{Rm + Bm}{K_m K_g} \right) \dot{\theta} = U \quad (2.6)$$

Mengambil transformasi *Laplace* (2.6), model servomotor DC untuk sistem *ball and beam* diperoleh sebagai

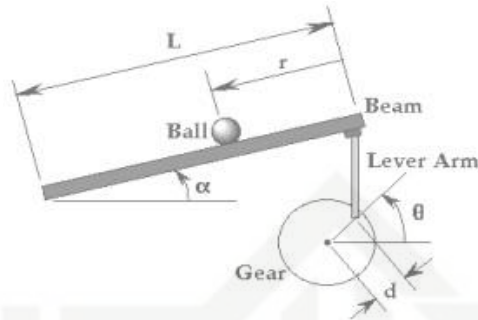
$$\frac{Rm + Jm}{K_m k_g} s^2 \theta + \left(K_b + \frac{Rm + Bm}{K_m K_g} \right) s \theta(s) = U(s)$$

$$\frac{\theta(s)}{U(s)} = \frac{K_m K_g}{Rm + Jm s^2 + (Rm + Bm + K_b K_m K_g) s} \quad (2.7)$$

2.2.3. Model Matematika *Ball And Beam System*

Dalam sistem batang penyeimbang bola, sebuah bola yang diletakkan di atas batang dapat bergerak bebas sepanjang batang tersebut. Batang dihubungkan dengan sebuah

motor yang dilengkapi dengan *gearbox* sehingga jika *gearbox* motor berputar dengan sudut θ , maka sudut batang akan berubah sebesar α . Pada saat posisi batang tidak lagi *vertikal*, gaya gravitasi akan menyebabkan bola bergerak sepanjang batang [2]



Gambar 2.2. Batang Penyeimbang Bola [2]

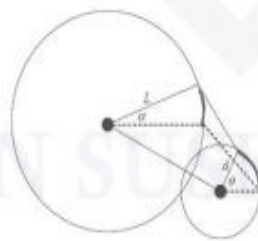
Bila diasumsikan bola bergerak tanpa slip dan gesekan, maka dari gambar di atas diperoleh persamaan untuk gerakan bola yaitu

$$\left(\frac{J}{R^2} + m\right) \ddot{x} + M_g \sin \alpha - m_x (\alpha^2) = 0 \tag{2.8}$$

Persamaan (2.8) dilarikan tentang sudut balok $\alpha = 0$, yang mana memberikan pendekatan linier berikut dari sistem

$$\left(\frac{J}{R^2} + m\right) \ddot{x} = -m g \alpha \tag{2.9}$$

Karena sudut balok α dan sudut gear θ tidak sama, oleh karena itu Gbr.3 digunakan untuk menghitungnya.



Gambar 2.3. Hubungan antara posisi motor dan sudut balok [2]

Jarak di dua lingkaran sama oleh karena itu, persamaan yang menghubungkan sudut balok α dengan sudut gear θ dapat didekati oleh hubungan linier:

$$\alpha L = \theta d$$

$$\alpha = \frac{d}{l} \theta \tag{2.10}$$

Dengan transformasi Laplace maka transfer function dari xposisi bola (X(s)) terhadap sudut gearbox (θ (s)) adalah

$$\left(\frac{J}{R^2} + m\right) \ddot{x} = -m g \frac{d}{l} \tag{2.12}$$

$$\left(\frac{J}{R^2} + m\right) X(s)S^2 = -m g \theta \frac{d}{l}(s)$$

$$\frac{X(s)}{\theta(s)} = \frac{-m g d}{Ls^2 \left(\frac{J}{R^2} + m\right)}$$

Kemudian transfer function (2.12) untuk ball and beam adalah sebagai berikut

$$\frac{X(s)}{\theta(s)} = \frac{-0.11 \times -9.8 \times 0.4}{0.4 \times \left(\frac{2 \times 0.011}{5} + 0.11\right) \times s^2}$$

$$TF = \frac{0.7}{s^2} \tag{2.13}$$

Table 2.1. Ball and beam system parameters [2]

Symbol	Quantity	Value
G	Gravity acceleration	-9.8 (m/s ²)
M	Ball mass	0.011 (kg)
R	Ball radius	0.015 (kg)
L	Beam length	0.4 m
D	Lever length	0.04 m
Δ	Equilibrium point of ball position	0.2 m
A	Beam angle coordinate	-
Θ	Servo gear angle	-

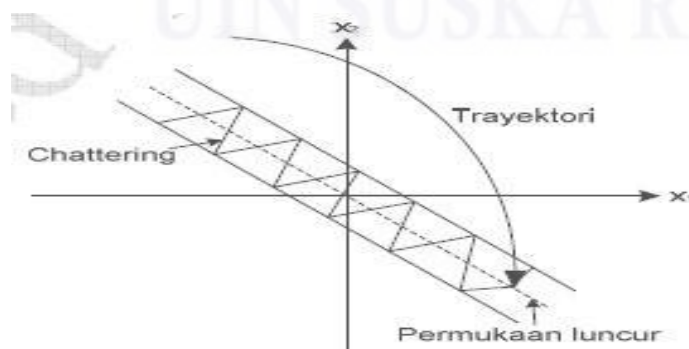
Hak Cipta Dilindungi Undang-Undang
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 b. Pengutipan 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.

2.3. Sliding Mode Control (SMC)

Pengendali Modus Luncur (PML) atau istilah populernya sering disebut dengan *Sliding Mode Controller* (SMC) merupakan sebuah kendali umpan balik pensaklaran berkecepatan tinggi yang efektif dan kokoh dalam mengendalikan sistem *linear* maupun *non-linear*. Sistem kendali ini kokoh karena menyediakan sebuah metoda perancangan sistem yang tidak peka terhadap ketidak pastian parameter lingkungan dan gangguan dari luar [14].

Pada prinsipnya SMC menggunakan sebuah hukum kendali pensaklaran berkecepatan tinggi untuk membawa trayektori status dari sistem linear atau non-linear ke dalam sebuah permukaan tertentu dalam ruang status yang disebut permukaan luncur atau *sliding surface*, kemudian trayektori status tersebut dipelihara agar tetap meluncur pada permukaan tersebut. Proses pemeliharaan trajektori status pada permukaan luncur mengakibatkan terjadinya osilasi pada permukaan luncur. *Osilasi* ini sering disebut dengan *chattering*. Fenomena *chattering* pada permukaan luncur akan berdampak pada stabilitas dari sistem kendali [14].

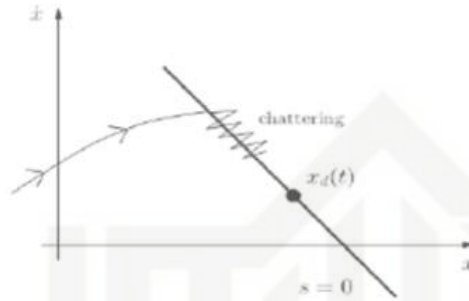
Pada SMC waktu yang dibutuhkan oleh status dalam mencapai permukaan luncur akan berdampak terhadap kecepatan tanggapan sistem terhadap waktu. Waktu yang dibutuhkan oleh status untuk mencapai permukaan luncur sering disebut dengan *hitting time*. Salah satu keuntungan dari SMC adalah ketika sistem mengenai permukaan luncur, maka sistem tersebut tidak akan peka terhadap ketidakpastian parameter lingkungan dan gangguan dari luar. Untuk menjaga perilaku sistem sehingga tidak peka terhadap perubahan lingkungan, maka dibutuhkan *hitting time* yang kecil agar sistem cepat mengenai permukaan luncur. *Hitting time* dan *chattering* yang kecil merupakan dua hal yang sangat penting dalam merancang SMC [14].



Gambar 2.5. Diagram trayektori status[14].

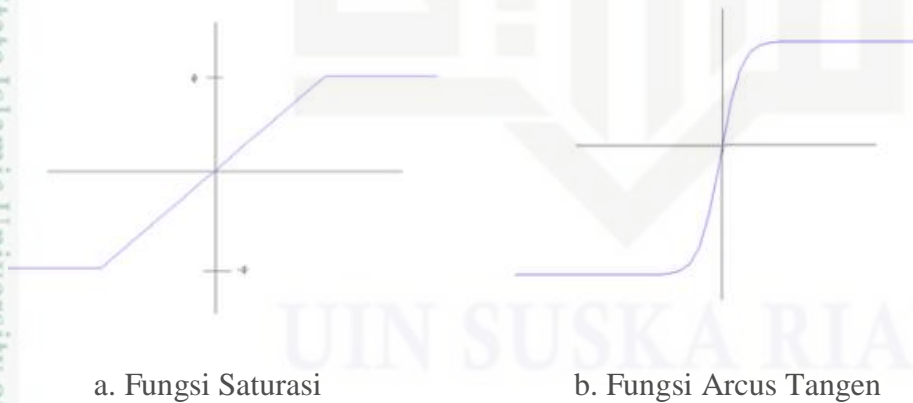
2.3.1. Chattering

SMC merupakan pengendali berumpan balik dengan pensaklaran berkecepatan tinggi (high speed switching feedback), sehingga dalam proses SMC mengalami chattering yang dapat mengganggu kestabilan sistem [14]



Gambar 2.6 Chattering Effect [14]

Chattering adalah sebuah fenomena perubahan kendali dengan frekuensi tinggi ketika trayektori disekitar permukaan bidang luncur dan ketika harga Signum sering berubah-ubah. Beberapa cara untuk mengurangi chattering adalah dengan memperluas fungsi *signum* menjadi *saturasi*, atau mengganti fungsi *signum* dengan arcus tangen [15]. Berikut adalah ilustrasinya:



Gambar 2.7 Fungsi Saturasi dan Arcus Tangen[16]

2.3.2. Perancangan Permukaan Luncur

Slotin (1985) dalam penelitiannya menuliskan bahwa perancangan permukaan luncur dirumuskan dengan persamaan awal sebagai berikut:

$$x^n(t) = f(x) + b(x, t).U + d(t) \tag{2.13}$$

Dimana U merupakan input kendali, x merupakan faktor keadaan, $f(x,t)$ dan $b(x,t)$ berupa fungsi terbatas, $d(t)$ gangguan eksternal. Jika x_d merupakan x maka *tracking error* dapat dinyatakan dengan :

$$e(t) = x(t) - x_d(t)$$

Fungsi *switching* yaitu permukaan $S(x,t)$ didalam ruang keadaan R^n , yang memenuhi persamaan

$$S(x, t) = \left(\frac{d}{dt} + \lambda \right)^{n-1} e \tag{2.14}$$

Dengan λ berupa konstanta positif. Dimana fungsi *switching* ini digunakan untuk menentukan besarnya nilai u agar memenuhi kondisi *sliding*. Permukaan luncur (*sliding surface*) merupakan persamaan yang memenuhi :

$$S(x,t) = 0 \tag{2.15}$$

Berdasarkan nilai kontrol *input* pada SMC bergantung pada nilai S , sehingga memenuhi pertidaksamaan yang disebut kondisi sliding. Kondisi tersebut ditulis dalam bentuk sebagai berikut [7]:

$$S\dot{S} \leq -\eta|S| \tag{2.16}$$

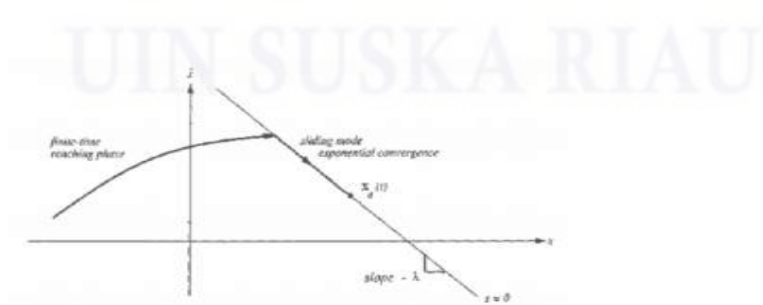
Keterangan:

S = Permukaan luncur

λ = Konstanta positif

n = orde sistem

e = *error*



Gambar 2.8. Permukaan Luncur pada *Sliding Mode*[14]

2.3.3. Pengendali PID

Pengendali PID adalah pengendali otomatis penggabungan dari ketiga metode kendali *proporsional*, *integral*, dan *derivatif*, pada masing-masing memiliki aksi yang khas pada tiap-tiap kekurangan tiap pengendali P,I dan D dapatsaling menutupi dengan menggabungkan ketiganya dan merangkai secara paralel menghasilkan pengendali PID. Pada masing- masing aksi keseluruhan memiliki tujuan untuk mempercepat reaksi sebuah sistem yang mampu menghilangkan offset dan menghasilkan perubahan yang besar.

Untuk menunjukkan hubungan dari ketiga pengendali P,I dan D dapat dinyatakan dengan rumus berikut:

$$u(t) = Kp e(t) + Ki \int_0^t e(t) dt + Kd \frac{de}{dt} \quad (2.17)$$

Persamaan 2.17 dirubah kedalam bentuk *Laplace* menjadi

$$\frac{U(s)}{E(s)} = Kp + \frac{Ki}{s} + Kd \quad (2.18)$$

Dimana:

Kp = *Proporsional Gain*.

Ki = *Integral Gain*.

Kd = *Derivatif Gain*.

Pengendali proporsional (Kp) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus *error steady state*. Pengendali integral (Ki) akan memberikan efek menghapus *error steady state* tetapi berakibat buruk pada saat menanggapi respon. Pengendali derivatif (Kd) berdampak pada stabilitas sistem, mengurangi kesalahan keadaan tunak dan menaikkan respon transien. Untuk hubungan dari ketiga aksi di tunjukkan pada tabel 2.1.

2.3.4. Penalaan parameter PID

Istilah penalaan (*tunning*) untuk menggambarkan metode-metode yang digunakan dalam memilih pengendalian yang cocok agar mendapat unjuk kerja pada nilai respon sistem yang diinginkan, dengan melakukan penalaan dari masing-masing aksi kendali akan didapatkan respon sistem sesuai dengan *setpoint* yanga diberikan sehingga dapat diterapkan pada sistem tertentu. Pada penelitian ini dilakukan penalaan pada pengendali

PID dengan melakukan penalaan pada masing-masing aksi kendali pada parameter PID [15]. Adapun Tabel yang menggambarkan hubungan dari masing-masing aksi kendali pada parameter PID ditunjukkan pada Tabel 2.1 berikut:

Tabel 2.1 Hubungan antara kendali K_p , K_i dan K_d [15].

Respon lup tertutup	Waktu naik	Overshoot	Waktu turun	Kesalahan keadaan tunak
K_p	Menurun	Meningkat	Perubahan kecil	Menurun
K_i	Menurun	Meningkat	Meningkat	Hilang
K_d	Perubahan kecil	Meningkat	Menurun	Perubahan kecil

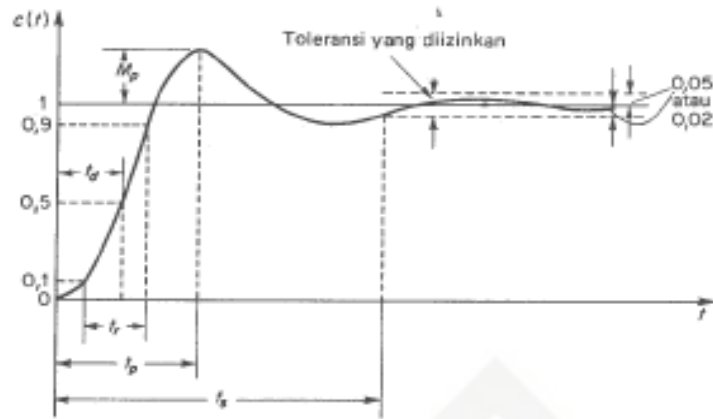
2.3.5. Pengendali PID Metode Heuristik

Metode Heuristik merupakan sebuah metode pemecahan masalah menggunakan ekspolarasi dan cara coba-coba. Heuristik adalah suatu metode untuk bisa menyelesaikan solusi secara penalaan. Rancangan metode Heuristik ini diperoleh dengan cara penambahan parameter yang disesuaikan dengan kinerja plant yang akan dikendalikan. Untuk merancang sistem pengendali PID dilakukan pencarian nilai besarnya K_p , K_i , dan K_d Maka pengujian dilakukan dengan beberapa tahap, dengan penalaan (*Heuristic method*) [16].

- a. Penalaan pengendali dimulai dengan hanya menggunakan pengendali P.
- b. Kemudian baru ditambahkan pengendali I.
- c. Dan terakhir ditambahkan dengan pengendali D.

2.4. Identifikasi Sistem

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 *plant ball and beam* merupakan sistem orde 2 yang ditunjukkan pada Gambar 2.6 berikut:



Gambar 2.9. Respon sistem orde kedua[15]

1. Kontanta waktu (τ)

waktu yang dibutuhkan respon mulai dari $t = 0$ sampai dengan respon mencapai 63.2 % dari respon *steady state*.

2. Waktu tunda (t_d)

Waktu tunda adalah waktu yang diperlukan oleh tanggapan untuk mencapai setengah nilai akhir untuk waktu yang pertama.

$$t_d(3\%) \approx \tau \ln 12 \tag{2.19}$$

3. Waktu naik (t_r)

Waktu naik adalah waktu yang diperlukan oleh tanggapan untuk naik dari 10% menjadi 90%, 5% menjadi 95%, atau 0 menjadi 100% dari nilai akhir yang biasa digunakan.

$$t_r(5\% - 95\%) \approx \tau \ln 19 \tag{2.20}$$

$$t_r(10\% - 90\%) \approx \tau \ln 9 \tag{2.21}$$

4. Waktu puncak (t_p)

Waktu puncak adalah waktu yang diperlukan tanggapan untuk mencapai puncak pertama *overshoot*.

5. *Overshoot* maksimum (M_p)

Overshoot maksimum adalah nilai puncak kurva tanggapan diukur dari satuan. Apabila nilai akhir keadaan tunak tanggapannya jauh dari satu, maka biasanya digunakan persen *overshoot* maksimum dan didefinisikan oleh.

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.

$$Mp = \frac{c(t_p) - c(\infty)}{c(\infty)} 100\% \quad (2.22)$$

Overshoot ini hanya terjadi pada sistem yang kurang teredam

6. Waktu tunak (t_s)

Waktu tunak adalah waktu yang diperlukan untuk menanggapi kurva agar dapat mencapai dan tetap berada dalam gugus nilai akhir ukuran yang disederhanakan dengan presentase mutlak harga akhirnya (biasanya 2% atau 5%) waktu tunak tadi dihubungkan dengan sebesar sistem kendali. Apabila kita menemukan kriteria kesalahan presentase kesalahan presentase untuk sistem, kita boleh menetapkannya dari tujuan desain sistem dalam pertanyaan.

$$t_s(\pm 3\%) \approx 5\tau \quad (2.23)$$

$$t_s(\pm 4\%) \approx 2\tau \quad (2.24)$$

$$t_s(\pm 5\%) \approx 0.5\tau \quad (2.25)$$

7. Error steady state (e_{ss})

Besarnya kesalahan pada keadaan tunak disebut *error steady state* dengan persamaan:

$$e_{ss} = R_{ss} - C_{ss} \quad (2.26)$$

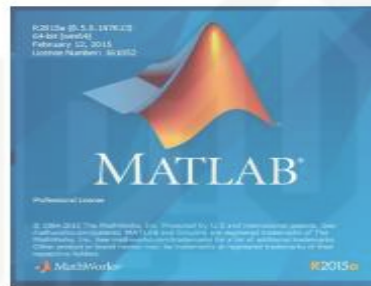
2.4.1. Kriteria Integral Menggunakan Integral of Absolute Error (IAE)

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 serta sangat cocok untuk menekan error yang kecil.

$$IAE = \int_0^{\infty} |e| dt \quad (2.27)$$

2.5. Program Matlab

Matlab merupakan singkatan dari Matrix Laboratory yang berarti bahasa pemrograman level tinggi (semakin tinggi level bahasa pemrograman maka semakin mudah cara menggunakannya) dengan kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan tunggal. Matlab memberikan sistem interaktif yang menggunakan konsep *array/matrix* sebagai variabel elemennya tanpa membutuhkan pendek *larasian array*.



Gambar 2.10 MATLAB R2009

Matlab dikembangkan oleh Mathwork pada tahun 1970. Aplikasi Matlab itu sendiri banyak digunakan dalam bidang yang membutuhkan perhitungan matematika yang rumit, dimana seluruh operasi perhitungan dalam Matlab berupa operasi matrik. Matlab dapat menampilkan hasil perhitungan dalam bentuk plot grafik dan dapat juga dirancang menggunakan GUI (Graphical User Interface) yang kita rancang. Pada software Matlab terdapat beberapa bagian penting yang digunakan dalam menjalankan program yaitu:

1. *Command window* digunakan untuk mengetik fungsi yang diinginkan.
2. *Command history* berfungsi agar fungsi yang telah digunakan sebelumnya dapat digunakan kembali.
3. *Workspace* digunakan untuk membuat variabel yang ada dalam Matlab.

Simulink-Matlab Simulink adalah sebuah kumpulan aplikasi dalam Matlab untuk melakukan modeling, simulasi, dan untuk melakukan analisis dinamik pada suatu sistem. Program simulink memudahkan user untuk membuat suatu simulasi lebih interaktif. Tiruan sistem diharapkan mempunyai perilaku yang sangat mirip dengan sistem fisik. Jika digunakan dengan benar, simulasi akan membantu proses analisis dan desain sistem.