



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

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 yang didapat dari jurnal, paper, buku, dan sumber lainnya. Penelitian yang menggunakan pengendali *hybrid* SMC dan PI pada sistem *non* linier multivariable *Continuous Stirred Tank Reactor* (CSTR) disebutkan pada penelitian tersebut bahwa kelemahan pengendali SMC adalah *chattering* yang menimbulkan *error steady state*. Oleh sebab itu dengan menambahkan pengendali PI dimana aksi kendali integral yang mempunyai keunggulan dapat mengurangi *error steady state* sehingga menghasilkan performansi yang lebih baik. Hasil simulasi menunjukkan pengendali *sliding mode* dengan permukaan lancar PI menunjukkan performansi yang lebih baik[11].

Penelitian yang menggunakan pengendali *hybrid* SMC dan PID pada sistem *coupled tank* untuk mengendalikan *level* membuktikan bahwa respon transien pengendali *hybrid* SMC PID lebih cepat dibandingkan pengendali SMC dan PID pada *coupled tank*[13]. Penelitian yang menggunakan pengendali *sliding mode control* pada *steam drum boiler* untuk mengendalikan *level* dan temperature uap dijelaskan bahwa *sliding mode control* dibedakan menjadi dua jenis metode yaitu metode *static* dan metode *dynamic*. Hasil dari penelitian yang telah dilakukan dijelaskan bahwa dengan metode *static* SMC, ketinggian air sudah stabil diposisi yang diinginkan tetapi waktunya lebih lama 0,5 jam daripada *dynamic* SMC. Dan dengan *static* SMC, temperatur uap lebih cepat daripada *dynamic* SMC yang sudah stabil namun masih belum pada posisi yang diinginkan. Dari hasil analisa yang disimpulkan bahwa *dynamic* SMC dalam desain *controller*  $w$  dan  $Q$  *dynamic* SMC tingkat kesulitannya lebih tinggi dibandingkan *static* SMC tetapi performansi *dynamic* SMC lebih baik dibandingkan *static* SMC[2].

Penelitian yang membandingkan pengendali *static* SMC dan *dynamic* SMC pada *continuous stirred tank reactor* (CSTR) mengatakan bahwa pengendali *dynamic sliding mode* memiliki performansi yang lebih baik dibandingkan *static sliding mode* dalam pencapaian *setpoint*. Keunggulan pengendali *dynamic sliding mode* dapat dilihat dari waktu yang lebih cepat dibandingkan pengendali *static sliding mode*. Hal ini terbukti pada tabel MATLAB yang menunjukkan waktu awal pencapaian awal *setpoint* pengendali *dynamic sliding mode* yaitu pada waktu 1,0452 untuk pengendalian level dan 0,6897 pada pengendalian konsentrasi.

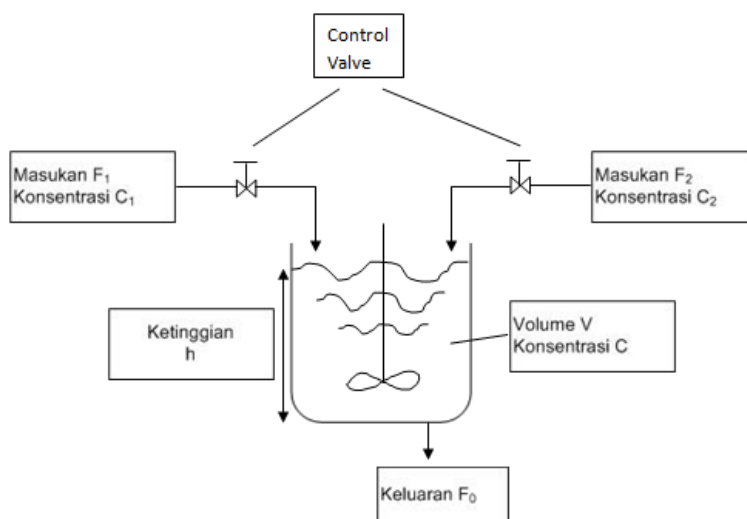
Sedangkan untuk pencapaian awal *setpoint* pengendali *static sliding mode* yaitu pada waktu 10,2278 untuk pengendalian *level* dan 6,2275 pada pengendalian konsentrasi [12].

Dari penelitian terkait inilah penulis ingin mengkombinasikan (*hybrid*) *static* dan *dynamic* untuk mengendalikan *level* dan konsentrasi yang akan diterapkan pada *Continuous Stirred Tank Reactor* (CSTR).

## 2.2 Continuous Stirred Tank Reactor (CSTR)

Reaktor merupakan unit operasi yang banyak digunakan di industri pembuatan bahan-bahan kimia atau industri makanan. Reaktor digunakan sebagai tempat untuk mereaksikan reaktan-reaktan dalam kondisi operasi tertentu sehingga dihasilkan produk dengan spesifikasi yang diinginkan. Dalam pengoperasiannya, reaktor dapat dioperasikan sebagai *batch*, *semi-batch*, dan *kontinyu*. Jenis reaktor yang paling banyak digunakan dalam industri proses adalah *Continuous Stirred Tank Reactor* (CSTR) atau reaktor berpengaduk yang dioperasikan secara *kontinyu*.

CSTR adalah tangki pengaduk yang memerlukan dua fluida atau lebih. CSTR merupakan sistem *non linier multivariable*, dapat dilihat dari beberapa variable yang dapat dikendalikan seperti *pressure*, *temperature*, *level*, *flow*, konsentrasi, dan lain sebagainya. Dengan banyaknya fluida yang diaduk, jika ingin mendapatkan hasil adukan yang sempurna diperlukan pengendalian konsentrasi dalam prosesnya. Berikut adalah ilustrasi sistem pada CSTR :



Gambar 2.1 Sistem CSTR [11]

Pemodelan matematika sistem didapatkan berdasarkan hukum kesetaraan tangki dengan asumsi tidak ada material yang keluar dalam bentuk uap.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.  
 b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.  
 Di larang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan atau menyebutkan sumber:  
 1. Cipta ilmiah yang dilindungi Undang-Undang Hak Cipta Ilmiah di lingkungan perguruan tinggi Islam.  
 UIN SUSKA RIAU



a. Berdasarkan Kesetaraan Volume :

$$V_{in} - V_{out} = \text{Perubahan } V \text{ dalam tangki} \tag{2.1}$$

Diketahui bahwa pada CSTR ada 2 *input* dan 1 *output* , maka :

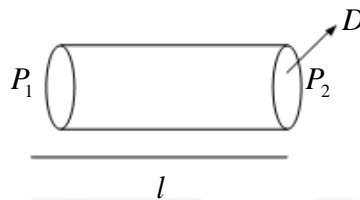
$$F_1 \Delta t + F_2 \Delta t - F_0 \Delta t = \Delta V \tag{2.2}$$

$$\frac{\Delta V}{\Delta t} = F_1 + F_2 - F_0 \tag{2.3}$$

dimana  $\Delta V = A \Delta H$  substitusikan ke persamaan 2.4

$$\frac{A \Delta H}{\Delta t} = F_1 + F_2 - F_0 \tag{2.4}$$

Kecepatan aliran keluaran  $F_0$  tergantung ketinggian permukaan dalam tangki H, dan luar diameter pipa [11].



Gambar 2.2 Pipa Keluaran CSTR [11]

Dari gambar diatas yang merupakan Hukum Bernoulli , maka :

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{P_1 - P_2} \quad P_1 \gg P_2 \tag{2.5}$$

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{P_1} \tag{2.6}$$

Diketahui  $= \rho g H$  , substitusikan ke persamaan 2.7

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{\rho g H} \tag{2.7}$$

Hak Cipta Dilindungi Undang-Undang  
 1. Dilarang mengutip sebagian atau seluruhnya tanpa 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 mengumpukan dan memperbanyak sebagian atau seluruhnya tanpa izin UIN Suska Riau.



$$F_0 = \frac{Dk_t \sqrt{\rho g} \sqrt{H}}{P\ell} \tag{2.8}$$

Dengan demikian, kecepatan aliran keluaran  $F_0$  tergantung dari ketinggian permukaan dalam tangki H, luas diameter pipa (D), dan konstanta celah(k).

Dimana  $\frac{Dk_t \sqrt{\rho g}}{P\ell} = k$  maka:

$$F_0 = k\sqrt{H} \tag{2.9}$$

Substitusikan persamaan (2.9) ke persamaan (2.4) :

$$A \frac{dH}{dt} = (F_1 + F_2) - K\sqrt{H} \tag{2.10}$$

$$\frac{dH}{dt} = \frac{1}{A}(F_1 + F_2) - \frac{k}{A}\sqrt{H} \tag{2.11}$$

**Pengendalian Level**

$$\frac{dH}{dt} = \frac{1}{A}F_1 + \frac{1}{A}F_2 - \frac{k}{A}\sqrt{H} \tag{2.12}$$

b. Berdasarkan Kesetaraan Massa :

$$\Delta C_0 V = C_1 F_1 \Delta t + C_2 F_2 \Delta t - C_0 F_0 \Delta t \tag{2.13}$$

$$\frac{\Delta C_0 V}{\Delta t} = C_1 F_1 + C_2 F_2 - C_0 F_0 \tag{2.14}$$

$$\frac{d}{dt}(C_0 V) = C_1 F_1 + C_2 F_2 - C_0 F_0 \tag{2.15}$$

Dimana  $\frac{dUV}{dt} = U'V + V'U$  , maka:

$$V \frac{dC_0}{dt} + C_0 \frac{dV}{dt} = C_1 F_1 + C_2 F_2 - C_0 F_0 \tag{2.16}$$

Dari persamaan (2.3) maka :

$$V \frac{dC_0}{dt} + C_0(F_1 + F_2 - F_0) = C_1 F_1 + C_2 F_2 - C_0 F_0 \tag{2.17}$$

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

© The Islamic University of Sultan Fariqin Riau

Diketahui  $V = AH$

$$AH \frac{dC_0}{dt} + C_0 F_1 + C_0 F_2 - C_0 F_0 = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.18)$$

$$AH \frac{dC_0}{dt} = (C_1 - C_0) F_1 + (C_2 - C_0) F_2 \quad (2.19)$$

$$\frac{dC_0}{dt} = \frac{1}{AH} \{(C_1 - C_0) F_1 + (C_2 - C_0) F_2\} \quad (2.20)$$

Pengendalian Konsentrasi

$$\frac{dC_0}{dt} = \frac{1}{AH} \{(C_1 - C_0) F_1 + (C_2 - C_0) F_2\} \quad (2.21)$$

Data parameter proses dapat dilihat pada tabel 2.1

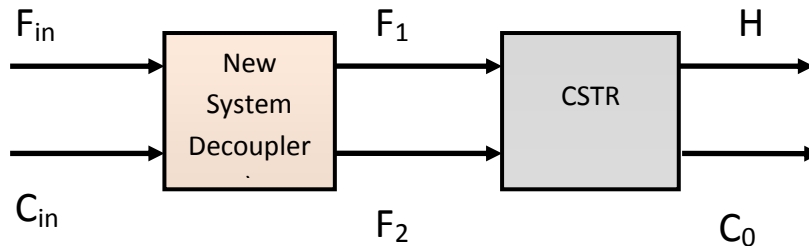
Tabel 2.1 Parameter proses CSTR [11]

Parameter	Simbol	Nilai	Satuan
Laju aliran 1	$F_1$	0.15	$m^3/s$
Laju aliran 2	$F_2$	0.6	$m^3/s$
Konsentrasi 1 (konstan)	$C_1$	1	$kmol^3/s$
Konsentrasi 2 (bervariasi)	$C_2$	1.2 – 1.4	$kmol^3/s$
Volume	$V$	1	$m^3$
Luas permukaan tangki	$A$	1	$m^2$

### 2.3 Metode Decoupler

*Decoupler* dilakukan untuk menyelesaikan permasalahan *coupled system*. Sistem CSTR merupakan sistem non linier sehingga perancangan *decoupler* tidak dapat dilakukan dengan pendekatan *transfer function* maupun pendekatan persamaan *state*. Perancangan *decoupler* pada penelitian ini dilakukan dengan menggunakan metode pendefinisian *state* dengan dirinya sendiri. Dengan demikian, metode *decouple* yang digunakan adalah dengan cara merancang sebuah sistem baru untuk mendapatkan *virtual manipulated variable* yaitu  $F_{in}$  dan  $C_{in}$ . Disain sistem baru ini disebut dengan *decoupler*.

Ilustrasi pengendalian sistem CSTR dengan *decoupler* dapat digambarkan dengan diagram blok. Gambar 2.3 menunjukkan diagram blok dengan disain sistem baru (*decoupler*) yang dirancang.



Gambar 2.3 Diagram Blok dengan *Decoupler* [11]

Bentuk matematis dari *decoupler* :

$$F_{in} = F_1 + F_2 \quad (2.22)$$

$$C_{in} = \frac{C_1 F_1 + C_2 F_2}{F_1 + F_2} \quad (2.23)$$

Dimana  $F_1$  dan  $F_2$ , sebagai fungsi  $F_{in}$  dan  $C_{in}$

Untuk Konsentrasi  $C_1$  :

$$\begin{aligned} C_1 F_1 + C_1 F_2 &= C_1 F_{in} \\ C_1 F_1 + C_2 F_2 &= C_{in} F_{in} \\ \hline F_2 &= \frac{1}{(C_1 - C_2)} \{C_1 F_{in} - C_{in} F_{in}\} \end{aligned} \quad (2.24)$$

Untuk Konsentrasi  $C_2$  :

$$\begin{aligned} C_2 F_1 + C_2 F_2 &= C_2 F_{in} \\ C_1 F_1 + C_2 F_2 &= C_{in} F_{in} \\ \hline F_1 &= \frac{1}{(C_2 - C_1)} \{C_2 F_{in} - C_{in} F_{in}\} \end{aligned} \quad (2.25)$$

## 2.4 Sistem Kendali

### 2.4.1 Pengertian Sistem Kendali

Sistem kendali (*control system*) adalah suatu alat (kumpulan alat) untuk mengendalikan, memerintah, dan mengatur keadaan dari suatu sistem [15]. Teknik kendali berhubungan dengan analisa, disain dan implementasi dari tujuan suatu sistem. Pada umumnya, suatu sistem terdiri dari rangkaian *plant* dan sistem pengendali di dalamnya. *Plant* merupakan seperangkat peralatan (obyek fisik) yang digunakan untuk melakukan suatu operasi tertentu [15].

### 2.4.2 Klasifikasi Sistem Kendali

Secara umum, sistem kendali dapat diklasifikasikan sebagai berikut:



## 1. Sistem Kendali Manual dan Otomatik

### 2. Sistem Lingkar Terbuka (*Open Loop*) dan Lingkar Tertutup (*Closed Loop*)

#### 3. Sistem Kendali Kontiniu dan Diskrit

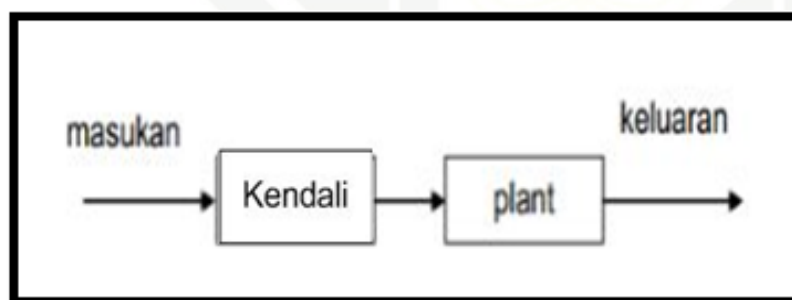
#### 4. Menurut sumber penggerak: Elektrik, Mekanik, Pneumatik, dan Hidraulik

Penjelasan singkat dari jenis-jenis sistem kendali diatas akan dibahas berikut ini:

Sistem Kendali Manual adalah pengendalian yang dilakukan oleh manusia yang bertindak sebagai operator, sedangkan Sistem Kendali Otomatik adalah pengontrolan yang dilakukan oleh peralatan yang bekerja secara otomatis dan operasinya dibawah pengawasan manusia. Sistem Kendali Manual banyak ditemukan dalam kehidupan sehari-hari seperti pada pengaturan suara radio, televisi, cahaya layer televisi, pengaturan aliran air melalui keran, pengendalian kecepatan kendaraan, dan lain-lain. Sedangkan Sistem Kendali Otomatik banyak ditemui dalam proses industri (baik industri proses kimia dan proses otomotif), pengendalian pesawat, pembangkit tenaga listrik dan lain-lain. Untuk sistem kendali terbuka (*Open Loop*) dan sistem kendali tertutup (*Closed Loop*) akan dijelaskan sebagai berikut:

#### 1. Sistem Kendali Terbuka (*Open Loop*)

Sistem kendali terbuka adalah suatu sistem yang keluarannya tidak mempunyai pengaruh terhadap aksi kendali. Artinya, sistem kendali terbuka keluarannya tidak dapat digunakan sebagai umpan balik dalam masukan.



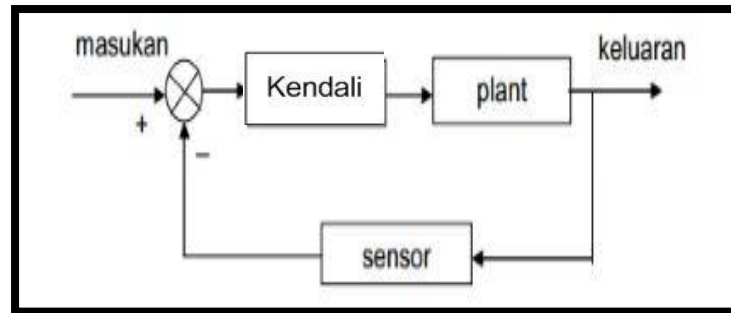
Gambar 2.4 Sistem Kendali Terbuka[16]

Dalam suatu sistem kendali terbuka, keluaran tidak dapat dibandingkan dengan masukan acuan. Jadi, untuk setiap masukan acuan berhubungan dengan operasi tertentu, sebagai akibat ketetapan dari sistem tergantung kalibrasi. Dengan adanya gangguan, sistem kendali terbuka tidak dapat melaksanakan tugas yang sesuai diharapkan. Sistem kendali terbuka dapat digunakan hanya jika hubungan antara masukan dan keluaran diketahui dan tidak terdapat gangguan internal maupun eksternal.

#### 2. Sistem Kendali Tertutup

Sistem kendali tertutup adalah sistem kendali yang sinyal keluarannya mempunyai pengaruh langsung pada aksi pengendalian. Sistem kendali tertutup juga merupakan sistem

kendali berumpan balik. Sinyal kesalahan penggerak, yang merupakan selisih antara sinyal masukan dan sinyal umpan balik (yang dapat berupa sinyal keluaran atau suatu fungsi sinyal keluaran atau turunannya). Diumpankan ke kendali untuk memperkecil kesalahan dan membuat agar keluaran sistem mendekati harga yang diinginkan. Dengan kata lain, istilah “*Closed Loop*” berarti menggunakan aksi umpan balik untuk memperkecil kesalahan sistem.



Gambar 2.5 Sistem Kendali Tertutup[16]

Pada gambar 2.5 menunjukkan hubungan masukan dan keluaran dari sistem kendali tertutup. Jika dalam hal ini manusia bekerja sebagai operator, maka manusia ini akan menjaga sistem agar tetap pada keadaan yang diinginkan, ketika terjadi perubahan pada sistem maka manusia akan melakukan langkah-langkah awal pengaturan sehingga sistem kembali bekerja pada keadaan yang diinginkan.

Berikut ini adalah komponen pada sistem kendali tertutup:

1. *Input* (masukan), merupakan rangsangan yang diberikan pada sistem kendali, merupakan harga yang diinginkan bagi variabel yang dikendali selama pengontrolan. Harga ini tidak tergantung pada keluaran sistem
2. *Output* (keluaran, respons), merupakan tanggapan pada sistem kendali, merupakan harga yang akan dipertahankan bagi variabel yang dikendali, dan merupakan harga yang ditunjukkan oleh alat pencatat
3. *Sistem/Plant*, merupakan sistem fisis yang akan dikendali (misalnya mekanis, listrik, hidraulik ataupun pneumatic) .
4. *Alat kendali* , merupakan peralatan/ rangkaian untuk mengontrol beban (sistem). Alat ini bisa digabung dengan penguat
5. *Elemen Umpan Balik*, menunjukkan/mengembalikan hasil pencatatan ke detector sehingga bisa dibandingkan terhadap harga yang diinginkan (di stel)
6. *Error Detector* (alat deteksi kesalahan), merupakan alat pendeteksi kesalahan yang menunjukkan selisih antara *input* (masukan) dan respons melalui umpan balik (*feedback path*)



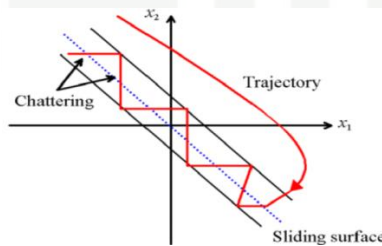


7. Gangguan merupakan sinyal-sinyal tambahan yang tidak diinginkan. Gangguan ini cenderung mengakibatkan harga keluaran berbeda dengan harga masuknya, gangguan ini biasanya disebabkan oleh perubahan beban sistem, misalnya adanya perubahan kondisi lingkungan, getaran ataupun yang lain.

### 2.5 Sliding Mode Controller (SMC)

SMC merupakan sebuah kendali umpan balik pensaklaran berkecepatan tinggi yang efektif dan kokoh dalam mengendalikan sistem *linier* maupun *non-linier*. Sistem kendali ini kokoh karena menyediakan sebuah metode perancangan sistem yang tidak peka terhadap ketidakpastian parameter lingkungan dan gangguan dari luar. SMC merupakan kendali yang didesain dengan menyediakan sebuah pendekatan sistematis, dan memiliki performa yang konsisten di dalam ketidakpastian model parameter. Kendali SMC telah sukses dalam penggunaan manipulasi robot, mesin kendaraan, transmisi otomatis, elektrik motor dan sistem tenaga listrik [14].

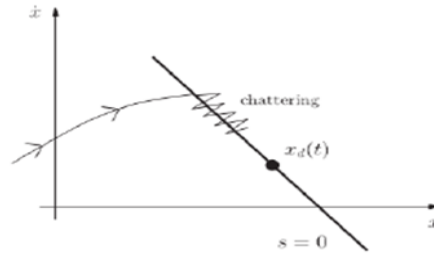
SMC pada prinsipnya menggunakan sebuah hukum kendali pensaklaran berkecepatan untuk membawa trajektori status dari sistem linear maupun non-linear ke dalam sebuah permukaan tertentu dalam ruang status (disebut permukaan luncur/"*sliding surface*"), kemudian trajektori status tersebut dipelihara agar 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].



Gambar 2.6 Diagram Trajektori Status [14]

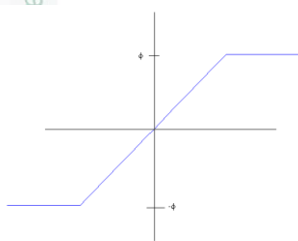
### 2.6 Chattering

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

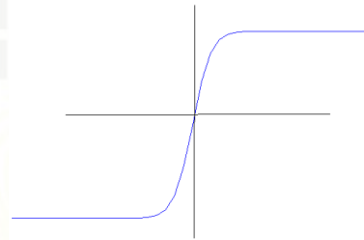


Gambar 2.7 Fenomena *Chattering Effect*[14]

*Chattering* merupakan osilasi keluaran pengendali dengan frekuensi tinggi yang disebabkan oleh *switching* yang sangat cepat untuk membentuk *slidingmode*. *Chattering* yang besar akan mengakibatkan akurasi menjadi rendah, rusaknya sistem mekanis aktuator, hilangnya energi pada *power, error steady state* dan lain sebagainya. Berbagai cara dapat dilakukan untuk mereduksi *chattering*, namun yang paling efektif adalah mengubah fungsi diskontinu *signum* menjadi fungsi kontinu saturasi / histeresis / *arcus tangen*.



a. Fungsi Saturasi



b. Fungsi *Arcus Tangen*

Gambar 2.8 Fungsi Saturasi dan *Arcus Tangen*[12]

## 2.7 Sinyal Kendali

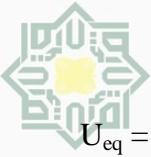
Untuk mendapatkan sinyal kendali yang mampu membawa status trayektori menuju permukaan luncur dan mempertahankan status trayektori agar tetap berada di sekitar permukaan luncur, maka diperlukan dua macam sinyal kendali. Sinyal kendali yang pertama adalah sinyal kendali ekivalen yang berfungsi untuk membawa status trayektori menuju permukaan luncur ( $U_{eq}$ ) dan sinyal kendali yang kedua adalah sinyal kendali natural yang berfungsi untuk mempertahankan status trayektori agar tetap berada di sekitar permukaan luncur ( $U_n$ )[14].

Sinyal kendali total merupakan penjumlahan dari dua sinyal kendali tersebut dan dapat ditulis:

$$U = U_{eq} + U_N \tag{2.23}$$

Dimana:

$U$  = Sinyal kendali total

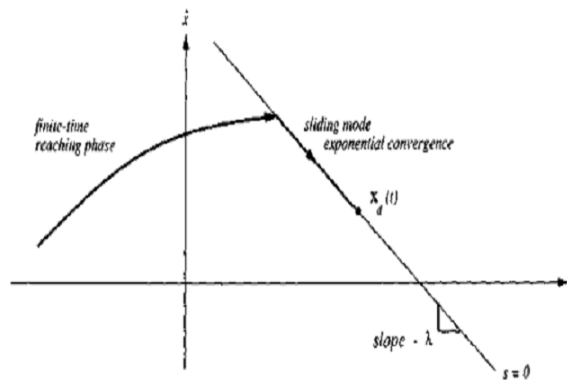


$U_{eq}$  = Sinyal kendali ekivalen

$U_N$  Sinyal kendali natural

### 2.9 Permukaan Luncur

Permukaan luncur adalah suatu batasan yang dibuat oleh perancang dimana nantinya variable status tersebut akan dipaksa untuk menuju ke permukaan luncur dan meluncur (*sliding*) menuju ke kondisi yang diinginkan. Pada saat status dari sistem sudah berada pada permukaan luncur maka sistem menjadi kebal dari gangguan luar maupun perubahan parameter yang terjadi pada sistem. Sehingga *Variabel* status dari sistem dapat dengan cepat menuju permukaan luncur agar sistem menjadi tidak peka akan perubahan parameter maupun gangguan dari luar.



Gambar 2.9 Permukaan luncur pada *sliding mode*[14]

#### 2.9.1 Perancangan Permukaan Luncur

Pandang suatu sistem dinamis[14] :

$$\dot{x}^n(t) = f(x) + b(x, t).U + d(t) \tag{2.24}$$

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

$$e(t) = x(t) - x_d(t) \tag{2.25}$$

Fungsi *swicthing* yaitu permukaan  $S(x,t)$  didalam ruang keadaan  $R^n$  untuk *static sliding mode* pengendali menggunakan

$$S(x, t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \tag{2.26}$$



sedangkan untuk *dynamic sliding mode* pengendali menggunakan

$$S(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n+1-1} e \quad (2.27)$$

Dengan  $\lambda$  berupa konstanta positif. Dimana fungsi *swicthing* ini digunakan untuk menentukan besarnya nilai  $u$  agar memenuhi kondisi *sliding*.

Permukaan luncur (*sliding surface*) merupakan persamaan yang memenuhi :

$$s(x, t) = 0 \quad (2.28)$$

Besar nilai pengendali *input* pada SMC bergantung pada nilai  $S$ , sehingga memenuhi pertidaksamaan yang disebut kondisi *sliding*. kondisi tersebut ditulis dalam bentuk sebagai berikut :

$$S\dot{S} \leq -\eta|S| \quad (2.29)$$

Keterangan :

$S$  = Permukaan luncur

$\lambda$  = Konstanta Positif

$n$  = Orde Sistem

$e$  = *Error*

## 2.10 Analisa Kestabilan Lyapunov

Alexander Mikhailovitch Lyapunov, seorang ilmuwan dari Rusia, telah mengembangkan metode untuk menentukan stabilitas dari sistem proses didasarkan dari penghematan energi yang disimpan. Menurut Lyapunov, hubungan antara kestabilan dan energi adalah sebagai berikut: "Sistem dikatakan stabil apabila energi yang disimpan makin lama makin kecil, maka osilasi yang terjadi juga semakin lama semakin kecil. Sebaliknya, sistem dikatakan tidak stabil apabila energi yang disimpan makin lama makin besar, maka osilasi yang terjadi semakin lama semakin besar juga". Agar sistem dapat dianalisis kestabilannya, maka perlu dibuat model matematis yang menghubungkan antara masukan, proses dan *output*.

Pada sistem pengaturan, diharapkan agar keluaran/respon dari sistem akan menuju nilai yang sama dengan masukan/*set point*. Hal ini identik dengan mengharapkan agar nilai *error* dapat menuju nilai nol serta tetap berada pada nilai nol. Mengacu pada teori *state-*

space, dimana nilai error dapat diambil sebagai variabel state sistem, maka dapat juga

diambil suatu fungsi skalar [11] :

$$V(x) = \frac{1}{2}(S)^2 \tag{2.30}$$

yang memenuhi :

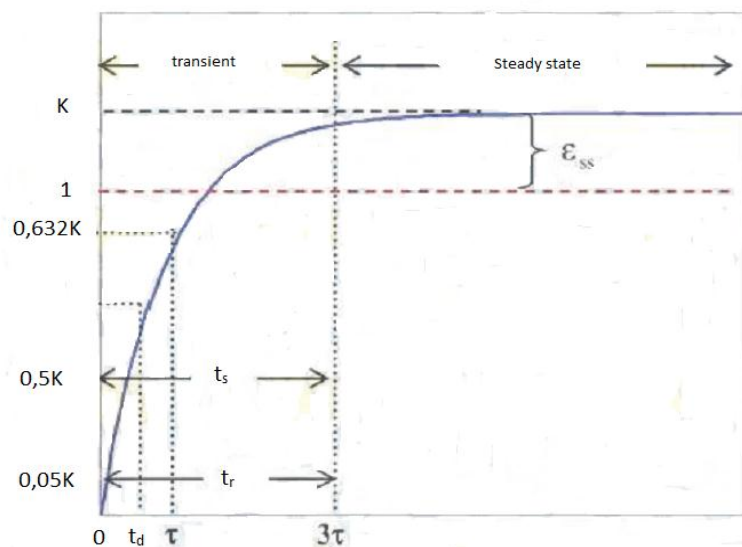
$$\dot{V}(x) = \dot{S}S < 0 \tag{2.31}$$

Sebagai fungsi Lyapunov, dimana  $S$  adalah suatu fungsi skalar dari  $x$  dan state  $x$  adalah state error. Fungsi skalar  $S$  selanjutnya akan disebut sebagai fungsi permukaan luncur ( $S$ ).

### 2.11 Metode Identifikasi Sistem

CSTR merupakan sistem berorde satu, oleh sebab itu metode identifikasi yang dilakukan adalah metode identifikasi statis. Metode identifikasi dilakukan dengan pendekatan grafis, dimana sinyal uji diberikan pada sistem untuk mengetahui respon *open loop* sistem. Dari respon sistem, dapat diketahui karakteristik-karakteristik penting dari sistem.

Salah satu metode identifikasi statis adalah metode pengamatan respon waktu. Identifikasi sistem dengan metode ini bekerja berdasarkan pengamatan grafis terhadap masukan *step*. Karakteristik respon waktu untuk sistem orde pertama diberikan berdasarkan respon sistem terhadap masukan sinyal *step*. Karakteristik respon waktu sistem orde pertama dibedakan menjadi karakteristik respon transien dan keadaan tunak (*steady state*). Grafik respon sistem orde pertama untuk  $X_{ss} = 1$  dan  $Y_{ss} = K$  ditunjukkan pada Gambar 2.10



Gambar 2.10 Respon Sistem Orde 1 [11]

Karakteristik respon transien sistem orde pertama , terdiri dari :

1. Spesifikasi Teoritis

Konstanta waktu ( $\tau$ ) adalah waktu yang dibutuhkan respon mulai dari  $t=0$  sampai respon mencapai 63,2% dari respon *steadystate*. Konstanta waktu menyatakan kecepatan respon sistem. Konstanta waktu yang lebih kecil akan mempercepat respon sistem.

2. Spesifikasi Praktis

a. Waktu tunak atau *settling time* ( $t_s$ ) adalah ukuran waktu yang menyatakan bahwa respon sistem telah masuk pada daerah stabil. Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_s$  dapat diformulasikan sebagai berikut :

1.  $t_s (\pm 3\%) \approx 5\tau$
2.  $t_s (\pm 4\%) \approx 2\tau$
3.  $t_s (\pm 5\%) \approx 0,5\tau$

b. 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 :

1.  $t_r (5\% - 95\%) \approx \tau \ln 19$
2.  $t_r (10\% - 90\%) \approx \tau \ln 19$

c. 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 :

$$t_d (\pm 3\%) \approx \tau \ln 12$$

d. Karakteristik respon keadaan tunak (*steady state*) sistem orde pertama diukur berdasarkan kesalahan pada keadaan tunak atau *error steady state* ( $e_{ss}$ ).

$$e_{ss} = R_{ss} - C_{ss}$$

dengan  $C_{ss}$  dan  $R_{ss}$  masing-masing adalah keluaran dan masukan sistem pada keadaan tunak.

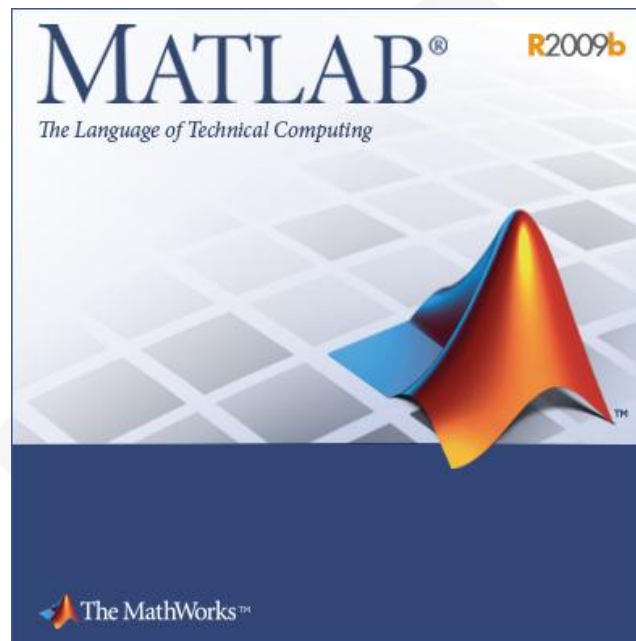
2.12 MATLAB

MATLAB adalah singkatan dari *MATrix LABORatory*. Pertama kali dibuat untuk mempermudah penggunaan dua koleksi subrutin pada pustaka FORTRAN yaitu: *LINPACK*

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.



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. Oleh karena itu bahasa MATLAB menyelesaikan masalah tersebut memerlukan waktu lebih cepat dibanding waktu yang dibutuhkan bahasa pemrograman tingkat tinggi lain.



Gambar 2.11 Tampilan MATLAB R2009

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. Berdasar 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:

- a. Adanya *toolbox* yang disebut MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer

Komputasi matematik dan komputasi teknik :

- a. Analisis data.
- b. Simulasi dan pemodelan
- c. Grafik-grafik perhitungan.
- d. Manipulasi matriks
- e. Pem-plot-an fungsi dan data
- f. Implementasi algoritma
- g. Pembuatan antarmuka pengguna / *interface* (visualisasi )
- h. Peng-antarmuka-an dengan program dalam bahasa lainnya

Fungsi-fungsi yang digunakan MATLAB:

- a. Fungsi matematika
- b. Fungsi fisika
- c. Fungsi statistik
- d. Fungsi visualisasi.

MATLAB merupakan *software* yang digunakan untuk:

- a. Para pendidikan (untuk melakukan riset dalam bentuk komputasi, pemodelan, simulasi dan demonstrasi / visualisasi)





- b. Para ilmuwan
  - c. Matematikawan
  - d. Pelajar dan mahasiswa terutama para pendidik matematika
  - e. Praktisi
  - f. Insinyur.
- dan untuk siapa saja tanpa ada batasannya

Sebagai sebuah sistem, 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 *MATLABdesktop* 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. *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 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.

#### 4. *MATLAB Language*

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



## 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 saja yang 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.