

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Dalam perkembangannya penelitian CSTR telah banyak dilakukan. Dimulai dengan pengendalian CSTR menggunakan pengendali konvensional PID untuk mengendalikan *temperature*. *Tuning* parameter P, I dan D dilakukan dengan metode PSO (*Particle Swarm Optimization*). Pada penelitian ini kita harus mencari parameter pengendali  $K_p$ ,  $K_i$  dan  $K_d$ . Namun, hal tersebut telah banyak dipecahkan oleh para peneliti dewasa ini. Dengan demikian, pengendali konvensional PID kurang menarik untuk dikembangkan lebih lanjut (Ali, 2010).

Sanposh (2008) mendesain *feedback linearization controller* dalam mengendalikan *level* pada CSTR. Pada penelitian ini kestabilan sistem tidak dapat dijamin karena kehadiran *error steady state*. Selain itu, *feedback linearization controller* mahal dalam implementasi. Teknik *gain scheduling* telah diterapkan pada CSTR untuk mengendalikan *level*. Pada penelitian ini terdapat waktu *transient* yang lambat, dikarenakan *gain* penjadwalan pengendali yang dipilih untuk titik operasi adalah konstan, tetapi faktanya lintasan *transient* dari sistem CSTR beroperasi lebih jauh atau bahkan lebih dekat dari titik operasi sebenarnya.

Banu (2007) mengkombinasikan *gain scheduling* dengan pengendali *fuzzy logic* untuk mengendalikan temperatur pada sistem CSTR. Pengendali cerdas *fuzzy logic* digunakan untuk mengatasi masalah waktu *transient* sistem CSTR dan *gain schedule* yang tepat untuk tiap – tiap kondisi operasi yang berbeda. Kelemahan pada penelitian ini adalah pengendali *fuzzy logic* terbatas dalam analisis.

Leonid dan Jeffrey (2003) melakukan penelitian untuk mengendalikan konsentrasi pada sistem CSTR menggunakan *Sliding Mode Controller*. Kelemahan pada penelitian ini adalah kehadiran *chattering* sebagai akibat dari penggunaan pengendali *sliding mode*. Kehadiran *chattering* menyebabkan kestabilan sistem terganggu dan *error steady state* yang cukup besar.

Herlambang (2010) mendisain pengendali *Sliding Mode (SM)* untuk mengendalikan *level* dan temperatur dalam *Steam Drum Boiler*. Pada penelitian ini teguh membagi SMC menjadi *static SMC* dan *dynamic SMC*.

Dian (2013) merancang pengendali pada CSTR dengan metode *decouple sliding mode control* untuk memisahkan *level* dan konsentrasi agar tidak saling mempengaruhi. Pada penelitian ini terdapat *chattering* yang bias menyebabkan *error steady state*.

Berdasarkan kajian pustaka dapat disimpulkan, CSTR merupakan sistem *non* linier multivariabel. Penelitian pada CSTR umumnya hanya mengendalikan satu variabel saja sementara variabel yang lain dianggap konstan. Namun, merujuk pada kajian pustaka pengendalian CSTR dengan mengikutsertakan dua variabel misalnya *level* dan konsentrasi adalah hal yang mungkin dilakukan. Permasalahan lain yang muncul pada CSTR adalah kekokohan sistem dalam mengatasi gangguan sehingga terjadi *error steady state* yang cukup besar. Untuk mengatasi kekokohan sistem dipilih pengendali *sliding mode*.

Pemilihan pengendali *sliding mode* didasari karena sifatnya yang kokoh dalam mengatasi gangguan. Pengendali *sliding mode* termasuk dalam kendali umpan balik yang memiliki pensaklaran berkecepatan tinggi sehingga menimbulkan osilasi dengan frekuensi tinggi yang dikenal dengan fenomena *chattering*. *Chattering* adalah fenomena merugikan yang dapat menyebabkan akurasi menjadi rendah, merusak mekanis sistem (*actuator*), hilangnya energi pada *power*, dan menimbulkan *error steady state*. Menghilangkan *chattering* juga tidak berdampak baik pada sistem karena kestabilan sistem tidak terjaga ketika *chattering* dihilangkan (Dian, 2013).

Pada penelitian ini akan dikembangkan pengendalian sistem *non* linier multivaribel CSTR menggunakan pengendali *decouple sliding mode* untuk mengendalikan *level* dan konsentrasi. Desain pengendali PID pada permukaan luncur SMC dilakukan untuk meminimalisir *error steady state* yang disebabkan oleh *chattering* namun kestabilan sistem tetap terjaga.

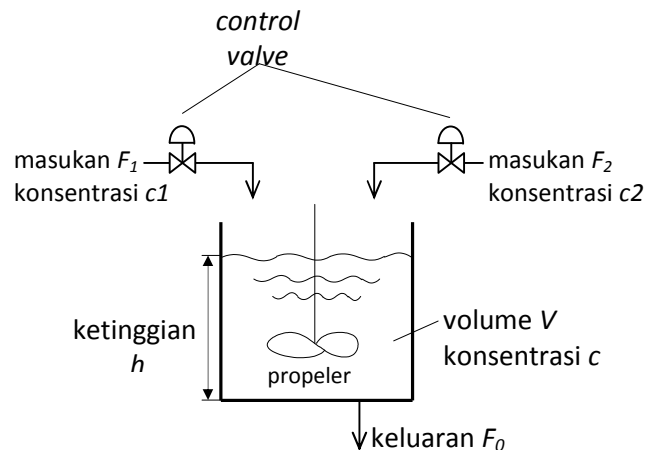
## **2.2 Dasar Teori**

### **2.2.1 *Continous Stirred Tank Reactor (CSTR)***

CSTR adalah tangki pengaduk yang umumnya digunakan pada industri-indusri kimia yang memerlukan percampuran dua atau lebih fluida, aplikasinya dapat ditemukan pada industri minuman, industri *bioteknologi*, dan sebagainya. Dengan banyaknya fluida yang

diaduk, untuk mendapatkan hasil adukan yang sempurna diperlukan pengendalian konsentrasi dalam prosesnya.

Pengendalian konsentrasi menjadi hal yang penting guna mendapatkan kualitas fluida yang dihasilkan agar sesuai dengan yang diharapkan misalnya saja rasa. Pengendalian konsentrasi ini berkaitan dengan pengendalian *level* dalam tangki. Karena untuk mendapatkan konsentrasi dengan perbandingan yang tepat tentu saja kita harus menentukan berapa *level* fluida dalam tangki untuk perbandingan konsentrasi yang telah ditentukan tersebut. Misalnya untuk perbandingan konsentrasi 60% : 40% maka *level* tangki nya adalah 50 meter. Sehingga jika *level* diatur pada nilai 60 meter tentu saja konsentrasi yang dihasilkan tidak akan sesuai dengan *set point* yang telah diatur (Luyben,1999). Ilustrasi untuk CSTR dapat dilihat pada Gambar 2.1.



Gambar 2.1. Sistem CSTR  
(sumber: Dian, 2012)

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

$$V_{in} - V_{out} = V \text{ dalam tangki} \quad (2.1)$$

Dengan mengasumsikan pada CSTR ada 2 input dan 1 output dan  $v = F \cdot t$ , maka didapatkan:

$$F_1 \Delta t + F_2 \Delta t - F_0 \Delta t = \Delta V \quad (2.2)$$

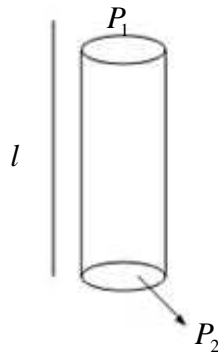
$$\frac{\Delta V}{\Delta t} = F_1 + F_2 - F_0 \quad (2.3)$$

dimana  $\Delta V = A \Delta H$  sehingga;

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

$$\frac{dH}{dt} = \frac{F_1}{A} + \frac{F_2}{A} - \frac{F_0}{A} \quad (2.5)$$

Kecepatan aliran keluaran  $F_0$  tergantung dari ketinggian permukaan dalam tangki  $H$ , dan luas diameter pipa (Dian,2013).



Gambar 2.2. Pipa keluaran CSTR  
(sumber: Dian,2013)

Dari gambar 2.2 didapatkan:

$$F_0 = \frac{Dk_t}{P_l} \sqrt{P_1 - P_2} \quad (2.6)$$

dimana;  $P_1 \gg P_2$

$$P = \rho gh$$

$$F_0 = \text{laju aliran (m}^3/\text{s)}$$

$$D = \text{diameter pipa (m)}$$

$$l = \text{panjang pipa (m)}$$

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{P_1} \quad (2.7)$$

$$F_0 = \frac{Dk_t}{P\ell} \sqrt{\rho g H} \quad (2.8)$$

$$F_0 = \frac{Dk_t \sqrt{\rho g} \sqrt{H}}{P\ell} \quad (2.9)$$

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

Dimana  $\frac{Dk_t \sqrt{\rho g}}{P\ell} = k$  sehingga;

$$F_0 = k\sqrt{H} \quad (2.10)$$

Substitusikan ke persamaan (2.5)

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

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

Untuk persamaan pengendalian konsentrasi:

$$\Delta C_0 V = C_1 F_1 \Delta t + C_2 F_2 \Delta t - C_0 F_0 \Delta t \quad (2.13)$$

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

$$\frac{dC_0 V}{dt} = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.15)$$

Dimana  $\frac{dUV}{dt} = U'V + UV'$  maka;

$$\frac{dC_0 V}{dt} = V \frac{dC_0}{dt} + C_0 \frac{dV}{dt} \quad (2.16)$$

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

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

$$V \frac{dC_0}{dt} + C_0 F_1 + C_0 F_2 - C_0 F_0 = C_1 F_1 + C_2 F_2 - C_0 F_0 \quad (2.19)$$

$$V \frac{dC_0}{dt} = C_1 F_1 + C_2 F_2 - C_0 F_0 - C_0 F_1 - C_0 F_2 + C_0 F_0 \quad (2.20)$$

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

Karena  $V = AH$  maka;

$$AH \frac{dC_0}{dt} = ((C_1 - C_0)F_1 + (C_2 - C_0)F_2) \quad (2.22)$$

$$\frac{dC_0}{dt} = \frac{1}{AH} ((C_1 - C_0)F_1 + (C_2 - C_0)F_2) \quad (2.23)$$

Jadi persamaan untuk pengendalian *level* dan konsentrasi pada CSTR adalah sebagai berikut:

$$\frac{dh}{dt} = \frac{1}{A} (F_1 + F_2) - \frac{k}{A} \sqrt{h} \quad (2.24)$$

$$\frac{dC_0}{dt} = \frac{1}{AH} ((C_1 - C_0)F_1 + (C_2 - C_0)F_2) \quad (2.25)$$

Dimana:

$H = level (m^3)$

$F_1 = laju aliran fluida1 (m^3/detik)$

$C_0 = konsentrasi fluida (kmol^3/detik)$

$F_2 = laju aliran fluida2 (m^3/detik)$

$A = luas tangki (m^2)$

$C_1 = konsentrasi fluida1 (kmol^3/detik)$

$k = konstanta celah$

$C_2 = konsentrasi fluida2 (kmol^3/detik)$

dari persamaan tersebut dapat dilihat bahwa dinamika fungsi CSTR adalah non linier. Data parameter proses dapat dilihat pada berikut.

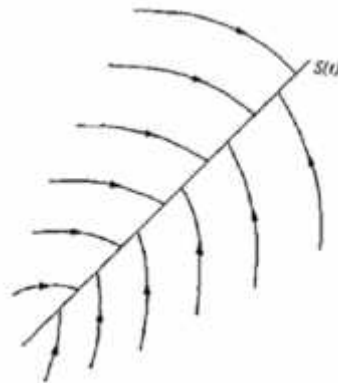
Tabel 2.1. Parameter proses CSTR

Parameter	Nilai	Parameter	Nilai
<i>Flow rate 1</i>	$F_1 = 0.6m^3/sec$	<i>Concentration 2</i>	$1.2 - 1.4 kmol^3/sec$
<i>Flow rate 2</i>	$F_2 = 0.15m^3/sec$	<i>Volume</i>	$V = 1m^2$
<i>Concentration 1</i>	$1 kmol^3/sec$	<i>Cross section</i>	$A = 1m^2$

Sumber: Dian (2013)

### 2.2.2 Pengendali *Sliding Mode*

Pengendali *sliding mode* adalah salah satu pengendali yang dapat diterapkan pada sistem linier maupun *non* linier. Prinsip kerja dari pengendali *sliding mode* adalah memaksa trajektori *state* suatu sistem menuju permukaan luncur tertentu dan mempertahankannya pada permukaan luncur tersebut. Pengendali *sliding mode* terkenal dengan kekokohnya dalam mengatasi gangguan (Stephanopoulos,1994). Ilustrasi *sliding mode* ditunjukkan pada gambar berikut:



Gambar 2.3. ilustrasi pengendali *sliding mode*  
(sumber: Teguh,2010)

Pengendali *sliding mode* terkenal dengan kekokohnya dalam mengatasi gangguan yang datang. Telah sukses diterapkan dalam berbagai sistem seperti : *electric power, robot manipulator, power konverter*, proses industri, dan lain sebagainya (Teguh,2010).

Jika terdapat suatu sistem dinamis

$$\dot{x} = f(x,t) + b(x,t)u + dt \quad (2.26)$$

di mana  $u$  adalah *control input*,  $x$  merupakan vektor keadaan,  $f(x,t)$  dan  $b(x,t)$  berupa fungsi terbatas,  $dt$  adalah gangguan eksternal. Jika  $x_{ref}$  merupakan nilai yang diinginkan maka *error* dapat dinyatakan dengan :

$$e(t) = x(t) - x_{ref}(t) \quad (2.27)$$

fungsi *switching* yaitu permukaan luncur  $S(x,t)$  untuk *static sliding mode* :

$$S(x,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \quad (2.28)$$

permukaan luncur untuk *dynamic sliding mode* :

$$S(x,t) = \left(\frac{d}{dt} + \lambda\right)^{n+1-1} e \quad (2.29)$$

Dengan  $\lambda$  merupakan suatu konstanta positif. Dimana fungsi *switching* ini digunakan untuk menentukan besarnya nilai  $u$  agar memenuhi kondisi *sliding*. Permukaan *sliding* ini membagi bidang menjadi dua bagian, yaitu  $S > 0$  dan  $S < 0$ . Agar *trajectory* keadaan dapat meluncur, maka sistem harus berada dalam kondisi *sliding*. Dengan demikian, besar nilai *input* kontrol bergantung pada nilai  $S(x,t)$ . Perancangan *control law* (aturan kontrol) pada SMC dilakukan hingga  $e(t)$  bergerak menuju permukaan *sliding* dan meluncur pada permukaan tersebut untuk semua  $t \geq 0$ .

Permukaan luncur merupakan persamaan yang memenuhi :

$$S(x,t) = 0 \quad (2.30)$$

besarnya nilai *control input* pada pengendali *sliding mode* bergantung pada nilai  $S$ , sehingga disebut kondisi *sliding*. Kondisi tersebut harus memenuhi persamaan Lyapunov.

$$S\dot{S} \leq -\eta|S| \quad (2.31)$$

dengan  $\eta$  suatu konstanta positif

Berdasarkan control law yang memenuhi kondisi sliding adalah :

$$\hat{F}_{in} = \frac{1}{a_1} \left(-b\sqrt{H}\right) - k \text{sign}(s)$$

Dimana nilai  $k$  harus memenuhi persamaan  $S\dot{S} \leq -\eta|S|$

### 2.2.3 Persamaan Lyapunov

Alexander Mikhailovitch Lyapunov, seorang ilmuwan dari Rusia, telah mengembangkan metode untuk menentukan stabilitas dari sistem proses didasarkan dari penghematan energi yang disimpan. 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 keluaran (Stephanopoulos,1994).

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*, di mana nilai *error* dapat diambil sebagai variabel *state* sistem, maka dapat juga diambil suatu fungsi skalar :

$$V(x) = \frac{1}{2} [S]^2 \quad (2.32)$$

yang memenuhi :

$$\dot{V}(x) = S\dot{S} < 0 \quad (2.33)$$

#### 2.2.4 Permukaan Luncur

Permukaan luncur adalah trayektori/lintasan yang dipilih dengan pertimbangan status trayektori sistem dapat menuju permukaan tersebut di manapun kondisi awalnya dalam waktu yang terbatas, serta status trayektori sistem dapat dipertahankan di sekitar permukaan luncur tersebut. Permukaan luncur merupakan komponen penting dari pengendali sliding mode sebagai tempat trayektori keadaan meluncur dari kondisi awal (*initial condition*) menuju keadaan yang diinginkan (*reference point*). Permukaan luncur secara umum didisain dengan menggunakan kombinasi linier *variable state* (Stephanopoulos,1994).

Permukaan luncur didefinisikan sebagai  $S(x)$  atau  $S$  :

$$S(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} e \quad (2.34)$$

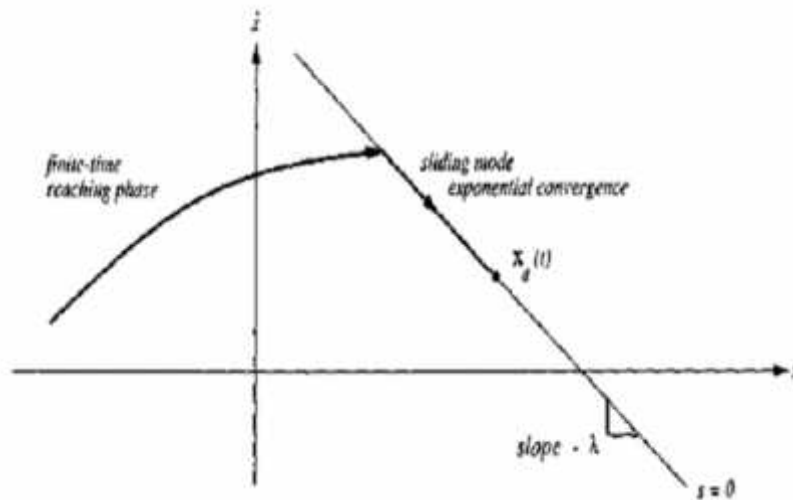
Dimana:

$n$  : orde sistem

$e$  : *error*

$\lambda$  : konstanta positif

ilustrasi permukaan luncur seperti gambar berikut:



Gambar 2.4. Permukaan luncur pada *sliding mode*  
(sumber: Dian, 2012)

### 2.2.5 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 ( $u_n$ ) (Ruey, 2004).

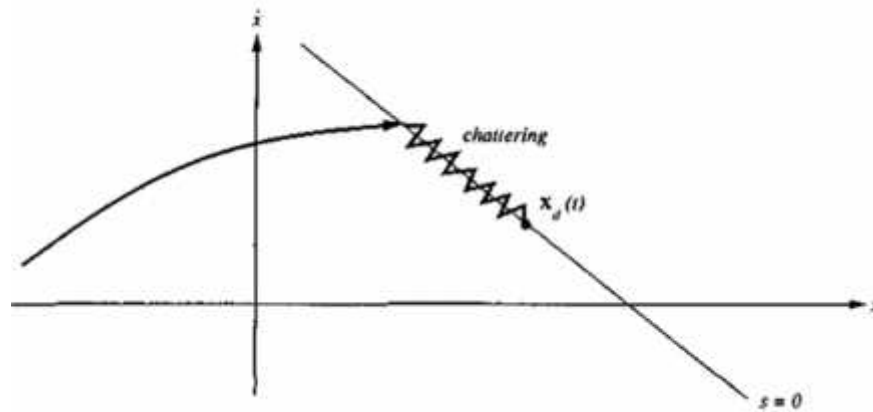
Sinyal kendali total merupakan penjumlahan dari dua sinyal kendali tersebut dan dapat ditulis :

$$U = U_{eq} + U_n \quad (2.35)$$

### 2.2.6 Kondisi Chattering

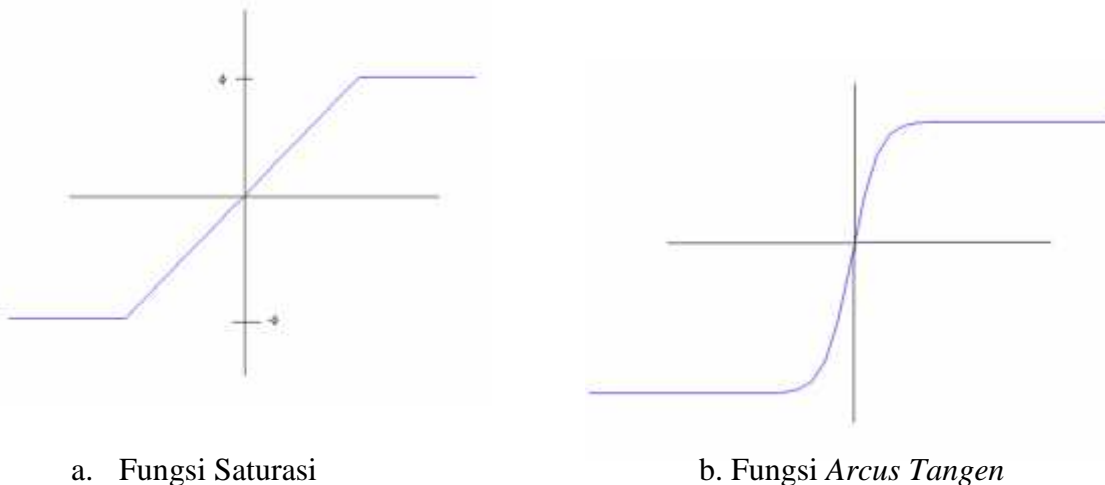
Pengendali *sliding mode* termasuk dalam pengendali berumpan balik dengan pensaklaran berkecepatan tinggi (*high speed switching feedback*), sehingga dalam

penerapannya *sliding mode controller* memiliki kelemahan yaitu kondisi *chattering* yang dapat mengganggu kestabilan sistem. *Chattering* merupakan osilasi keluaran pengendali dengan frekuensi tinggi yang disebabkan oleh *switching* yang sangat cepat untuk membentuk *sliding mode*. Osilasi yang sangat tinggi pada sinyal kendali ini menyebabkan ketidakstabilan pada sistem (Stephanopoulos,1994).



Gambar 2.5. *Chattering Phenomenon*  
(sumber: Dian, 2012)

*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 diskontinyu *signum* menjadi fungsi kontinyu saturasi / histeresis / *arcus tangen*.



a. Fungsi Saturasi

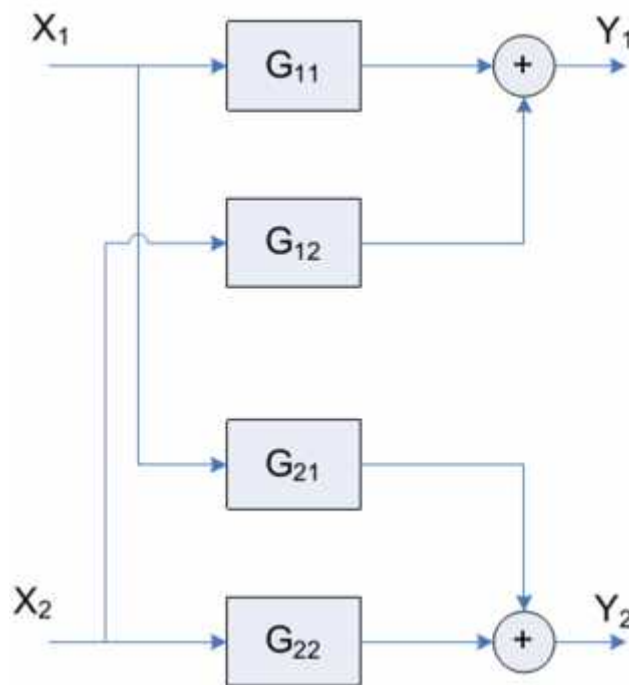
b. Fungsi Arcus Tangen

Gambar 2.6. Fungsi Saturasi dan Arcus Tangen

(sumber: Dian, 2011)

### 2.2.7 Metode *Decouple*

Pada sistem CSTR memiliki beberapa variabel masukan dan keluaran yang saling berkaitan satu sama lain. Perubahan suatu masukan kadang tidak hanya berpengaruh pada satu keluaran saja, melainkan berpengaruh juga pada keluaran lainnya yang disebut dengan *Multi Input dan Multi Output (MIMO)*. Penggambaran sistem ini dapat dilihat pada Gambar 2.6.



Gambar 2.6. Interaksi input dan output pada Sistem MIMO  
(sumber: Dian, 2012)

Hubungan suatu masukan berpengaruh pada keluaran loop yang lain. Penulisan model untuk Gambar 2.6 adalah :

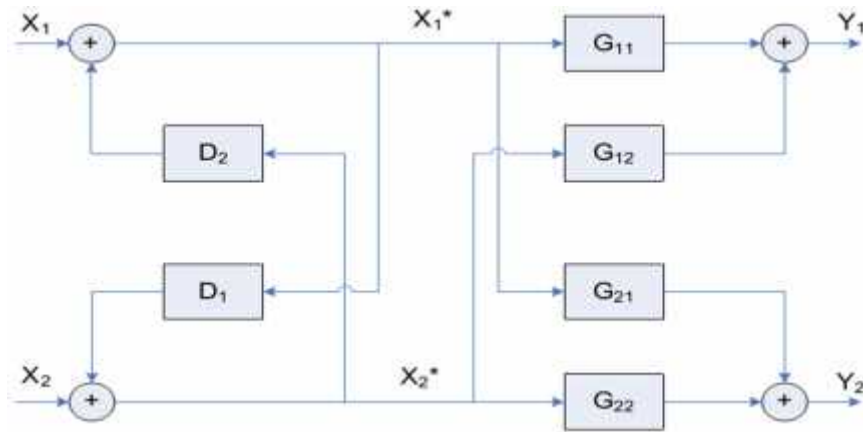
$$\text{Loop 1 : } Y_1 = G_{11} X_1 + G_{12} X_2 \quad (2.36)$$

$$\text{Loop 2 : } Y_2 = G_{21} X_1 + G_{22} X_2 \quad (2.37)$$

Ada beberapa penyelesaian *decouple* yang dapat dilakukan, yaitu :

1. *Decouple* dengan pendekatan *transfer function*

Untuk melakukan *decouple* dengan metode ini, langkah pertama adalah dengan cara mendapatkan *transfer function* dari sistem, yaitu hubungan antara  $Y_1$  dengan  $X_1$ ,  $Y_1$  dengan  $X_2$ , dan  $Y_2$  dengan  $X_1$ ,  $Y_2$  dengan  $X_2$ . Metode *decouple* dengan pendekatan *transfer function* dapat dilihat pada Gambar 2.7



Gambar 2.7. *Decouple* dengan pendekatan *transfer function*  
(sumber: Dian, 2012)

*Decoupler* dirancang untuk mengkompensasi interaksi proses yang tidak diinginkan. Untuk memperoleh persamaan pada *decoupler* maka dilakukan analisa terpisah pada tiap masukan - keluaran sehingga diperoleh nilai  $D_2$  sebagai berikut :

$$Y_1 = G_{11} X^*_1 + G_{12} X^*_2 \quad (2.38)$$

$$X^*_1 = X_1 + D_2 X^*_2 \quad (2.39)$$

$$Y_1 = G_{11} X_1 + G_{11} D_2 X^*_1 + G_{12} X^*_2 = G_{11} X_1 + (G_{11} D_2 + G_{12}) X^*_2 \quad (2.40)$$

Agar output  $Y_1$  hanya dipengaruhi nilainya dari input  $X_1$ , nilai input  $X^*_2$  harus sama dengan nol ( $X^*_2 = 0$ ), sehingga persamaan menjadi :

$$(G_{11}D_2 + G_{12}) X^*_2 = 0 \quad (2.41)$$

$$G_{11} D_2 + G_{12} = 0 \quad (2.42)$$

$$G_{11} D_2 = - G_{12} \quad (2.43)$$

$$D_2 = -\frac{G_{12}}{G_{11}} \quad (2.44)$$

Untuk  $D_1$  diperoleh dari persamaan berikut:

$$Y_2 = G_{21} X^*_1 + G_{22} X^*_2 \quad (2.45)$$

$$X^*_2 = X_2 + D_1 X^*_1 \quad (2.46)$$

$$Y_2 = G_{22} X_2 + G_{21} D_1 X^*_1 + G_{21} X^*_1 = G_{22} X_2 + (G_{22} D_1 + G_{21}) X^*_1 \quad (2.47)$$

Agar *output*  $Y_2$  hanya dipengaruhi nilainya dari input  $X_2$ , nilai input  $X^*_1$  harus sama dengan nol ( $X^*_1 = 0$ ), sehingga persamaan menjadi :

$$(G_{22}D_1 + G_{21}) X^*_1 = 0 \quad (2.48)$$

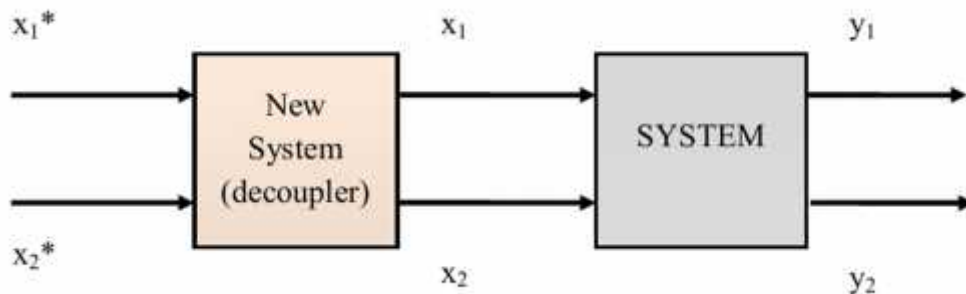
$$G_{22} D_1 + G_{21} = 0 \quad (2.49)$$

$$G_{22} D_1 = -G_{21} \quad (2.50)$$

$$D_1 = -\frac{G_{21}}{G_{22}} \quad (2.51)$$

## 2. Direct Method Decouple

*Direct method decouple* adalah metode *decouple* secara langsung, yang mana mendefinisikan *state* dengan dirinya sendiri untuk mendapatkan virtual *manipulated variable*. *Direct method decouple* adalah metode yang paling sederhana dibandingkan metode lainnya. *Direct method decouple* juga dapat langsung dirancang untuk menyelesaikan persoalan *coupled system* pada sistem *non* linier tanpa harus melinierkan sistem terlebih dahulu. Ilustrasi menggunakan *direct method decoupler* ditunjukkan pada Gambar 2.9.



Gambar 2.8. *Direct Method Decouple*

(sumber: Dian, 2012)

Tujuannya adalah untuk mendapatkan virtual manipulated variable  $X_1^*$  dan  $X_2^*$  sebagaimana fungsi  $X_1$  dan  $X_2$ .

### 2.2.7 Pengendali Proporsional Integral Derivative (PID)

PID adalah salah satu pengontrol otomatis yang banyak digunakan di industri. Kontroler PID adalah kontroler berumpanbalik yang paling populer di dunia industri. Selama lebih dari 50 tahun, kontroler PID terbukti dapat memberikan performa kontrol yang baik meski mempunyai algoritma sederhana yang mudah dipahami. Hal krusial dalam desain kontroler PID ialah tuning atau pemberian parameter P, I, dan D agar didapatkan respon sistem yang diinginkan (Iwan, 2008).

Adapun bentuk umum dari aksi kendali PID sebagai berikut:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \quad (2.52)$$

Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan. Efek dari perubahan parameter PID pada sistem control ditunjukkan pada Table 2.2.

Tabel 2.2. Tanggapan sistem kontrol PID terhadap perubahan parameter

Tanggapan loop tertutup	Waktu naik	<i>overshoot</i>	Waktu turun	Kesalahan keadaan tunak
Proporsional ( $K_p$ )	Menurun	Meningkat	Perubahan kecil	Menurun
Integral ( $K_i$ )	Menurun	Meningkat	Meningkat	Hilang
Derivative ( $K_d$ )	Perubahan kecil	Menurun	Menurun	Perubahan kecil

Sumber: Ali (2004)

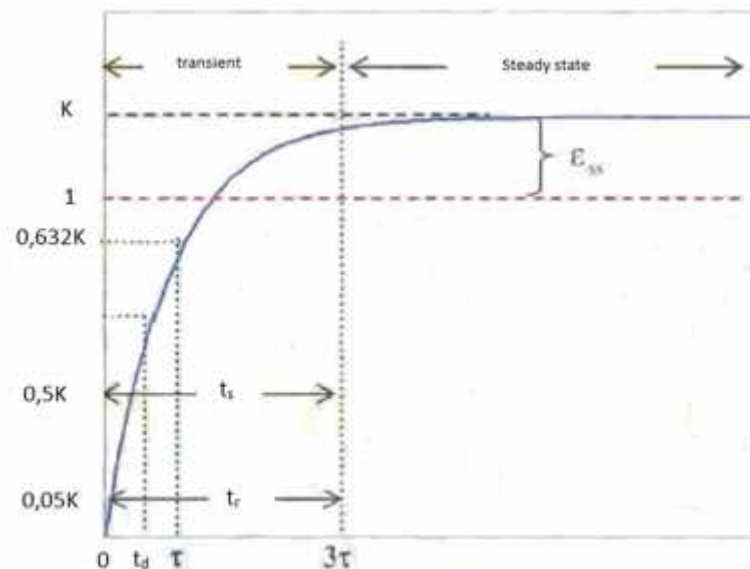
Untuk merancang sistem kontrol PID, kebanyakan dilakukan dengan metoda coba-coba atau (trial & error). Hal ini disebabkan karena parameter  $K_p$ ,  $K_i$  dan  $K_d$  tidak independent. Untuk

mendapatkan aksi kontrol yang baik diperlukan langkah coba-coba dengan kombinasi antara P, I dan D sampai ditemukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  seperti yang diinginkan (Ali, 2004).

### 2.2.8 Sistem Pengendalian

CSTR merupakan sistem berorde satu, oleh sebab itu metode identifikasi yang dilakukan adalah metode identifikasi statis. Metode identifikasi dilakukan dengan pendekatan grafis, di mana 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 2.2.



Gambar 2.2. Respon Sistem Orde 1  
(sumber: Dian, 2011)



Respon transien terdiri dari :

a) Spesifikasi teoritis :

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

b) Spesifikasi praktis :

1. 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 :

$$t_s (\pm 3\%) = 5\tau$$

$$t_s (\pm 4\%) = 2\tau$$

$$t_s (\pm 5\%) = 0,5\tau$$

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 :

$$t_r (5\% - 95\%) = \tau \ln 19$$

$$t_r (10\% - 90\%) = \tau \ln 9$$

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 :

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

4. 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.