BAB 2
DASAR TEORI

2.1 Heat Exchanger

Proses *Heat Exchanger* atau pertukaran panas antara dua fluida dengan temperatur yang berbeda, baik bertujuan memanaskan atau mendinginkan fluida banyak diaplikasikan secara teknik dalam berbagai proses thermal dalam dunia industri.

Berdasarkan arah aliran fluida, *Heat Exchanger* dapat dibedakan menjadi:

1. *Heat Exchanger* dengan aliran searah (*co-current/parallel flow*)

Pertukaran panas jenis ini, kedua fluida (dingin dan panas) masuk pada sisi *Heat Exchanger* yang sama, mengalir dengan arah yang sama, dan keluar pada sisi yang sama. Karakter *Heat Exchanger* jenis ini, temperatur fluida dingin yang keluar dari *Heat Exchanger* (Tco) tidak dapat melebihi temperatur fluida panas yang keluar (Tho), sehingga diperlukan media pendingin atau media pemanas yang banyak.

Pertukaran panas yang terjadi:

$$Mc.Cc(Tco - Tci) = Mh.Ch(Tho - Thi) \qquad (2.1)$$

Dimana:

Mc = Massa air (Kilogram)

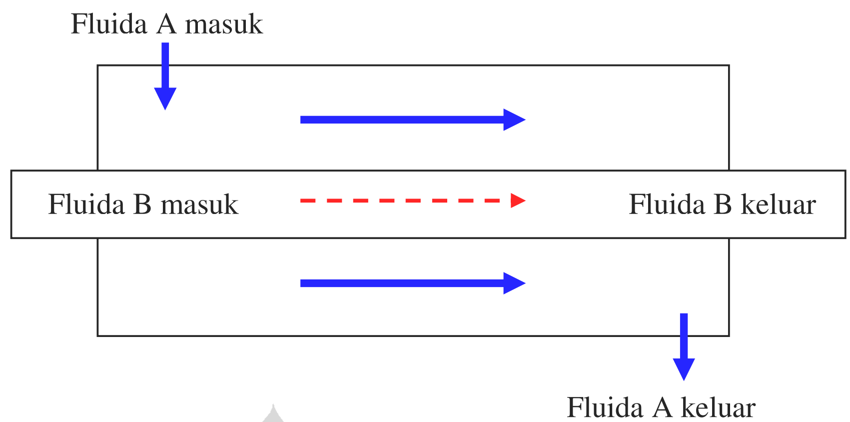
$Cc = Cp$ = Kapasitas panas/dingin (Kcal/Kg.K)

Tco = Suhu air dingin yang keluar dari *heat exchanger* (K)

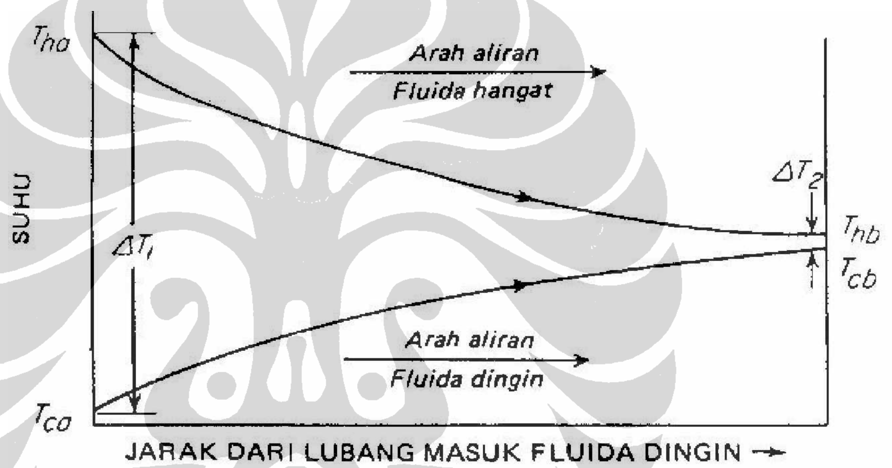
Tci = Suhu air dingin yang masuk ke *heat exchanger* (K)

Tho = Suhu air panas yang keluar dari *heat exchanger* (K)

Thi = Suhu air panas yang masuk ke *heat exchanger* (K)



Gambar 2.1 Sketsa *Heat Exchanger co-current/parallel flow*



Gambar 2.2 Profil temperatur pada *Heat Exchanger co-current/parallel flow*

Dengan asumsi nilai kapasitas spesifik fluida dingin (Cc) dan panas (Ch) konstan, tidak ada kehilangan panas ke lingkungan serta keadaan *steady state*, maka panas yang dipindahkan :

$$q = U..A.T_{AMTD} \tag{2.2}$$

Dimana:

q = perubahan panas (K)

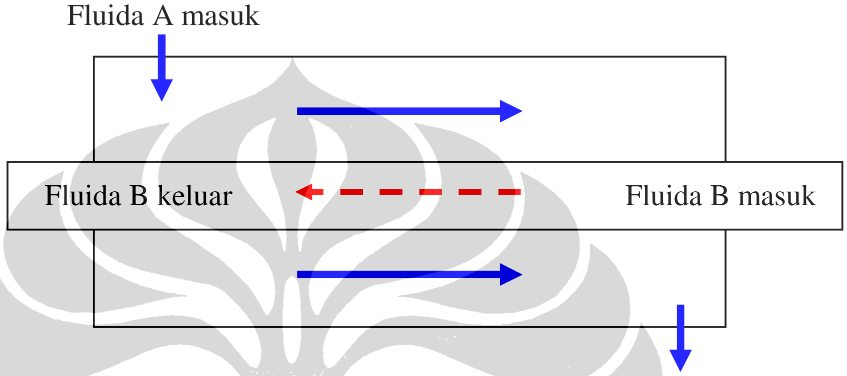
U= koefisien panas secara keseluruhan (Kcal/s.m^{2}K)

A= luas perpindahan panas (m^2)

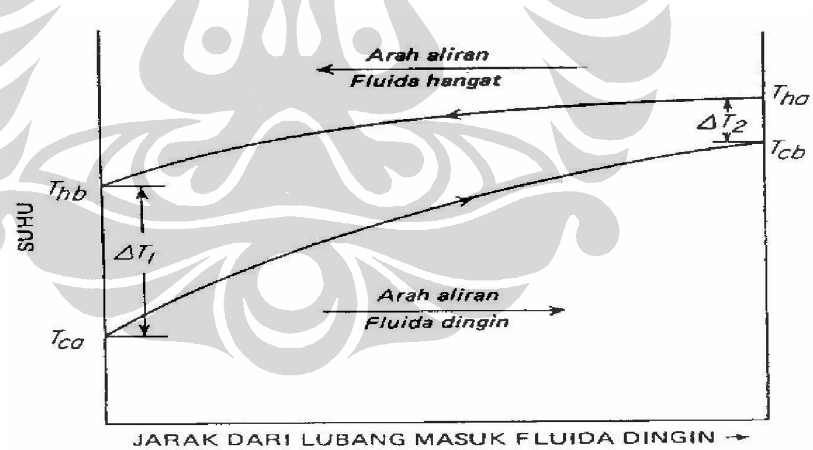
$T_{AMTD} = \frac{(Thi + Tho)}{2} - \frac{(Tci + Tco)}{2}$ (*Arithmetic Mean Temperature Difference*)

2. *Heat Exchanger* dengan aliran berlawanan arah (*counter-current flow*)

Heat Exchanger jenis ini memiliki karakteristik; kedua fluida (panas dan dingin) masuk ke *Heat exchanger* dengan arah berlawanan, mengalir dengan arah berlawanan dan keluar *Heat exchanger* pada sisi yang berlawanan.



Gambar 2.3 Sketsa *Heat Exchanger counter-current flow*



Gambar 2.4 Profil temperatur pada *Heat Exchanger counter-current flow*

Panas yang dipindahkan pada aliran *counter current* mempunyai persamaan yang sama dengan persamaan (2.2)

2.2 Konsep Dasar MPC

Model Predicive Control (MPC) atau sistem kendali prediktif termasuk dalam konsep perancangan pengendali berbasis model proses, dimana model proses digunakan secara eksplisit untuk merancang pengendali dengan cara meminimumkan suatu fungsi kriteria. Ide yang mendasari pada setiap jenis MPC adalah [3]:

1. Penggunaan model proses secara eksplisit untuk memprediksi keluaran proses yang akan datang dalam rentang waktu tertentu (horizon).
2. Perhitungan rangkaian sinyal kendali dengan meminimasi suatu fungsi kriteria.
3. Strategi surut; pada setiap waktu pencuplikan (pada waktu k) horizon dipindahkan menuju waktu pencuplikan berikutnya (pada waktu $k+1$) dengan melibatkan pemakaian sinyal kendali pertama (yaitu $u(k)$) untuk mengendalikan proses, dan kedua prosedur diatas diulang dengan menggunakan informasi terakhir.

Model MPC memiliki beberapa keunggulan dibandingkan dengan metode pengendali lainnya, diantaranya adalah:

1. Konsepnya sangat intuitif serta penalaannya mudah.
2. Dapat digunakan untuk mengendalikan proses yang beragam, mulai dari proses yang sederhana, hingga proses yang kompleks, memiliki waktu tunda yang besar. *Non-minimum phase* atau proses yang tidak stabil.
3. Dapat menangani sistem multivariabel.
4. Mempunyai kompensasi terhadap waktu tunda.
5. Mempunyai kemampuan dari pengendali *feed forward* untuk mengkompensasi gangguan yang terukur.
6. Mudah untuk mengimplementaikan pengendali yang diperoleh.

7. Dapat memperhitungkan batasan atau *constraints* dalam merancang pengendali.

8. Sangat berguna jika sinyal acuan untuk masa yang akan datang diketahui.

Selain beragam keuntungan yang dimiliki, metode MPC juga mempunyai kelemahan, yaitu masalah penurunan aturan sinyal kendali yang cukup kompleks dan keperluan akan model proses yang baik.

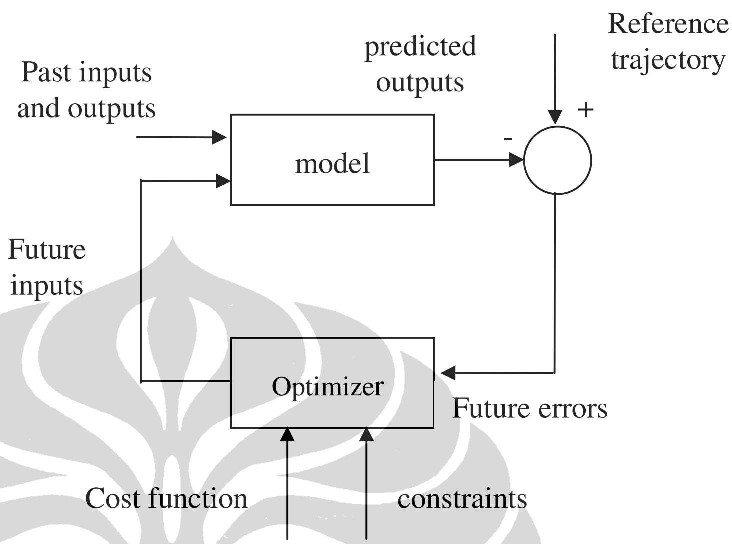
Struktur dasar dari pengendali MPC dapat dilihat pada gambar 2.1. metodologi semua jenis pengendali yang termasuk kedalam kategori MPC dapat dikenali oleh strategi berikut [3]:

1. Keluaran proses yang akan datang untuk rentang *horizon Hp* yang ditentukan yang dinamakan sebagai *prediction horizon*, diprediksi pada setiap waktu pencuplikan dengan menggunakan model proses. Keluaran proses terprediksi ini $y(k+i\,|\,k)$ untuk *i=1........Hp* bergantung pada nilai masukan dan keluaran lampau dan kepada sinyal kendali yang akan datang $u(k+i\,|\,k)$, *i*=0......Hp-1, yang akan digunakan sistem harus dihitung.

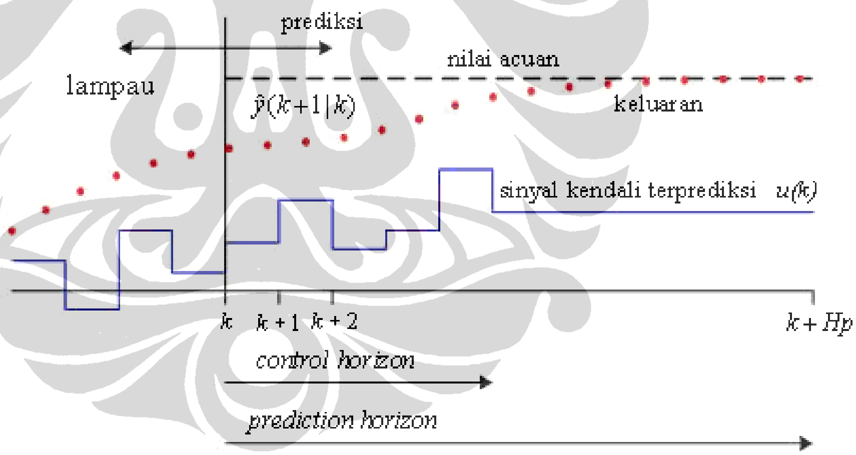
2. Serangkaian sinyal kendali dihitung dengan mengoptimasi suatu fungsi kriteria yang ditetapkan sebelumnya, dengan tujuan untuk menjaga proses sedekat mungkin terhadap trayektori acuan $r(k+i)$. Fungsi kriteria tersebut umumnya berupa suatu fungsi kuadratik dari kesalahan antara sinyal keluaran terprediksi dengan trayektori acuan. Solusi eksplisit dapat diperoleh jika fungsi kriteria adalah kuadratik, model linear, dan tidak ada *constraints*, jika tidak optimasi iteratif harus digunakan untuk memecahkannya. Langkah pertama dan kedua dapat diilustrasikan pada gambar 2.2.

3. Sinyal kendali $u(k\,|\,k)$ dikirim ke proses, sedangkan sinyal kendali terprediksi berikutnya dibuang, karena pada pencuplikan berikutnya $y(k+1)$ sudah diketahui nilainya. Maka langkah pertama diulang dengan nilai keluaran proses yang baru dan semua prosedur perhitungan yang

diperlukan diperbaiki. Sinyal kendali yang baru $u(k+1|k+1)$ (nilainya berbeda dengan $u(k+1|k)$) dihitung dengan menggunakan konsep *receding horizon*.



Gambar 2.5 Struktur Pengendali MPC



Gambar 2.6 Kalkulasi Keluaran Proses dan Pengendali Terprediksi

2.3 Fungsi Kriteria Pada Model Predictive Control

Seperti yang telah dinyatakan sebelumnya bahwa perhitungan sinyal kendali pada MPC dilakukan dengan meminimumkan suatu fungsi kriteria. Fungsi kriteria yang digunakan dalam algoritma MPC berbentuk kuadratik seperti berikut:

$$V(k)=\sum_{i=1}^{Hp}\| \underline{\hat{y}}(k+i\mid k)-\underline{r}(k+i\mid k)\|^2_{\underline{Q}(i)}+\sum_{i=0}^{Hu-1}\|\Delta\underline{\hat{u}}(k+i\mid k)\|^2_{\underline{R}(i)} \qquad (2.1)$$

Dengan:

$\underline{\hat{y}}(k+i\mid k)=$ keluaran terprediksi untuk i-langkah kedepan saat waktu k

$\underline{r}(k+1\mid k)=$ nilai trayektori acuan (*reference trajectory*)

$\Delta\underline{\hat{u}}(k+i\mid k)=$ perubahan nilai sinyal kendali terprediksi untuk i-langkah kedepan saat waktu k

$Q(i)$ dan $\underline{R}(i)$ = faktor bobot

Hp = prediction horizon

Hu = control horizon

Dari persamaan fungsi kriteria tersebut, selalu dibuat asumsi bahwa nilai $Hu<Hp$ dan $\Delta\underline{\hat{u}}(k+i\mid k)=0$ untuk $i\geq Hu$, sehingga nilai masukan terprediksi $\underline{\hat{u}}(k+i)=\underline{\hat{u}}(k+Hu-i\mid k)$ untuk semua $i\geq Hu$ seperti yang terlihat pada gambar 2.6.

Bentuk dari fungsi kriteria pada persamaan (2.5) menyatakan bahwa vektor kesalahan $\underline{\hat{y}}(k+i\mid k)-\underline{r}(k+i\mid k)$ dibebankan pada setiap rentang *prediction horizon*. Walaupun demikian tetap ada kemungkinan untuk menghitung vektor kesalahan pada titik-titik tertentu saja dengan cara mengatur matriks faktor bobot $Q(i)$ bernilai nol pada langkah yang diinginkan. Selain vektor kesalahan, fungsi kriteria pada persamaan (2.5) juga memperhitungkan perubahan vektor masukan dalam rentang *control horizon*. Pemilihan penggunaan $\Delta\underline{\hat{u}}(k+i\mid k)$ pada fungsi kriteria bertujuan untuk meminimumkan perubahan sinyal kendali yang masuk ke *plant*.

2.4 Model Proses

Pada pembahasan tesis ini, model proses yang digunakan berupa model ruang keadaan diskrit linear seperti berikut:

$$\underline{x}(k+1) = \underline{\Delta x}(k) + \underline{Bu}(k) \tag{2.2}$$

$$\underline{y}(k) = \underline{Cx}(k) \tag{2.3}$$

Dengan:

$\underline{u}(k) =$ vektor masukan berdimensi-*l*

$\underline{x}(k) =$ vektor keadaan berdimensi -*n*

$\underline{y}(k) =$ vektor keluaran berdimensi –*m*

$\underline{A} =$ matriks keadaan berdimensi *n* x *n*

$\underline{B} =$ matriks masukan berdimensi *n* x *l*

$\underline{C} =$ matriks keluaran berdimensi *m* x *n*

Persamaan ruang keadaan ini merupakan kondisi ideal dan sederhana untuk sebuah sistem, karena tidak adanya *disturbance* serta *direct feed trough* pada keluaran sistem. Pada perancangan MPC di Bab selanjutnya akan diuraikan penurunan sinyal kendali untuk model yang lebih kompleks yang digunakan.

2.5 Prediksi

Sinyal masukan yang digunakan dalam perhitungan prediksi keluaran adalah perubahan nilai sinyal masukan $\Delta u(k)$ pada setiap waktu pencuplikan k . Dimana perubahan tersebut merupakan selisih antara nilai sinyal masukan saat k atau $\underline{u}(k)$ dan sinyal masukan satu langkah sebelumnya $\underline{u}(k-1)$. Dalam menyelesaikan masalah pengendali prediktif, nilai keluaran terprediksi $\underline{\hat{y}}(k+i\,|\,k)$ harus dapat dihitung dengan menggunakan estimasi terbaik dari variabel keadaan saat ini $\underline{x}(k)$, nilai masukan yang lampau $\underline{u}(k-1)$, dan nilai perkiraan dari perubahan masukan yang akan datang $\underline{\Delta\hat{u}}(k+i\,|\,k)$. Sebelum melangkah lebih jauh, hal pertama yang harus dilakukan adalah memprediksi nilai variabel keadaan dengan melakukan iterasi model ruang keadaan pada persamaan (2.2)dan (2.3). Perhitungan prediksi variabel keadaan adalah sebagai berikut;

$$\underline{\hat{x}}(k+1 \mid k) = \underline{A}\underline{x}(k) + \underline{B}\underline{\hat{u}}(k \mid k) \tag{2.4}$$

$$\begin{aligned}\underline{\hat{x}}(k+2 \mid k) &= \underline{A}\underline{\hat{x}}(k+1 \mid k) + \underline{B}\underline{\hat{u}}(k+1 \mid k) \\ &= \underline{A}^2\underline{x}(k) + \underline{A}\underline{B}\underline{\hat{u}}(k \mid k) + \underline{B}\underline{\hat{u}}(k+1 \mid k)\end{aligned} \tag{2.5}$$

$$\vdots$$

$$\begin{aligned}\underline{\hat{x}}(k+Hp \mid k) &= \underline{A}\underline{\hat{x}}(k+Hp-1 \mid k) + \underline{B}\underline{\hat{u}}(k+Hp-1 \mid k) \\ &= \underline{A}^{Hp}\underline{x}(k) + \underline{A}^{Hp-1}\underline{B}\underline{\hat{u}}(k \mid k) + \ldots + \underline{B}\underline{\hat{u}}(k+Hp-1 \mid k)\end{aligned} \tag{2.6}$$

Pada setiap langkah prediksi digunakan $\underline{\hat{u}}(k \mid k)$ bukan $\underline{u}(k)$, karena besarnya nilai $\underline{u}(k)$ belum diketahui ketika menghitung prediksi.

Sekarang, diasumsikan bahwa nilai masukan hanya berubah pada waktu k, k+1, …, k+Hu–1, dan setelah itu menjadi konstan, sehingga didapatkan bahwa $\underline{\hat{u}}(k+i \mid k) = \underline{\hat{u}}(k+Hu-1 \mid k)$ untuk $Hu \leq i \leq Hp$-1.

Selanjutnya, perhitungan prediksi diubah sehingga mengandung $\underline{\Delta\hat{u}}(k+i \mid k)$ daripada $\underline{\hat{u}}(k+i \mid k)$, dengan

$$\underline{\Delta\hat{u}}(k+i \mid k) = \underline{\hat{u}}(k+i \mid k) - \underline{\hat{u}}(k+i-1 \mid k) \tag{2.7}$$

dan pada setiap waktu pencuplikan k nilai yang sudah diketahui hanya $\underline{u}(k$-1), maka

$$\underline{\hat{u}}(k \mid k) = \underline{\Delta\hat{u}}(k \mid k) + \underline{u}(k-1 \mid k) \tag{2.8}$$

$$\underline{\hat{u}}(k+1 \mid k) = \underline{\Delta\hat{u}}(k+1 \mid k) + \underline{\Delta\hat{u}}(k \mid k) + \underline{u}(k-1 \mid k) \tag{2.9}$$

$$\vdots$$

$$\underline{\hat{u}}(k+Hu-1 \mid k) = \underline{\Delta\hat{u}}(k+Hu-1 \mid k) + \ldots + \underline{\Delta\hat{u}}(k \mid k) + \underline{u}(k-1 \mid k) \tag{2.10}$$

Dengan mensubstitusikan persamaan (2.8) – (2.10) ke persamaan (2.4) – (2.6), diperoleh persamaan:

$$\underline{\hat{x}}(k+1 \mid k) = \underline{A}\underline{x}(k) + \underline{B}\left[\underline{\Delta\hat{u}}(k \mid k) + \underline{u}(k-1)\right] \tag{2.11}$$

$$\begin{aligned}\underline{\hat{x}}(k+2 \mid k) &= \underline{A}^2\underline{x}(k) + \underline{A}\underline{B}\left[\underline{\Delta\hat{u}}(k \mid k) + \underline{u}(k-1)\right] \\ &\quad + \underline{B}\underbrace{\left[\underline{\Delta\hat{u}}(k+1 \mid k) + \underline{\Delta\hat{u}}(k \mid k) + \underline{u}(k-1)\right]}_{\underline{\hat{u}}(k+1|k)}\end{aligned}$$

$$= \underline{A}^2\underline{x}(k) + \left(\underline{A} + \underline{I}\right)\underline{B}\underline{\Delta\hat{u}}(k \mid k) + \underline{B}\underline{\Delta\hat{u}}(k+1 \mid k) + \left(\underline{A} + \underline{I}\right)\underline{B}\underline{u}(k-1) \tag{2.12}$$

$$\vdots$$

$$\underline{\hat{x}}(k+Hu\mid k)=\underline{A}^{Hu}\,\underline{x}(k)+\left(\underline{A}^{Hu-1}+\ldots+\underline{A}+\underline{I}\right)\underline{B}\underline{\Delta\hat{u}}(k\mid k)+\ldots$$
$$+\underline{B}\underline{\Delta\hat{u}}(k+Hu-1\mid k)+\left(\underline{A}^{Hu-1}+\ldots+\underline{A}+\underline{I}\right)\underline{B}\underline{u}(k-1) \tag{2.13}$$

Dengan mengacu pada persamaan $\hat{u}(k+i\mid k)=\hat{u}(k+Hu-i\mid k)$ untuk $i>Hu$, maka perhitungan prediksi untuk $i>Hu$ adalah;

$$\underline{\hat{x}}(k+Hu+1\mid k)=\underline{A}^{Hu+1}\,\underline{x}(k)+\left(\underline{A}^{Hu}+\ldots+\underline{A}+\underline{I}\right)\underline{B}\underline{\Delta\hat{u}}(k\mid k)+\ldots$$
$$+\left(\underline{A}+\underline{I}\right)\underline{B}\underline{\Delta\hat{u}}(k+Hu-1\mid k)+\left(\underline{A}^{Hu}+\ldots+\underline{A}+\underline{I}\right)\underline{B}\underline{u}(k-1) \tag{2.14}$$

$$\vdots$$

$$\underline{\hat{x}}(k+Hp\mid k)=\underline{A}^{Hp}\,\underline{x}(k)+\left(\underline{A}^{Hp-1}+\ldots+\underline{A}+\underline{I}\right)\underline{B}\underline{\Delta\hat{u}}(k\mid k)+\ldots$$
$$+\left(\underline{A}^{Hp-Hu}+\ldots+\underline{A}+\underline{I}\right)\underline{B}\underline{\Delta\hat{u}}(k+Hu-1\mid k)$$
$$+\left(\underline{A}^{Hp-1}+\ldots+\underline{A}+\underline{I}\right)\underline{B}\underline{u}(k-1) \tag{2.15}$$

Persamaan (2.11) – (2.15) dapat disusun ke dalam bentuk vektor matriks sebagai berikut:

$$\begin{bmatrix}\underline{\hat{x}}(k+1\mid k)\\ \vdots\\ \underline{\hat{x}}(k+Hu\mid k)\\ \underline{\hat{x}}(k+Hu+1\mid k)\\ \vdots\\ \underline{\hat{x}}(k+Hp\mid k)\end{bmatrix}=\underbrace{\underbrace{\begin{bmatrix}\underline{A}\\ \vdots\\ \underline{A}^{Hu}\\ \underline{A}^{Hu+1}\\ \vdots\\ \underline{A}^{Hp}\end{bmatrix}}_{\Psi}\underline{x}(k)+\underbrace{\begin{bmatrix}\underline{B}\\ \vdots\\ \sum_{i=0}^{Hu-1}\underline{A}^i\underline{B}\\ \sum_{i=0}^{Hu}\underline{A}^i\underline{B}\\ \vdots\\ \sum_{i=0}^{Hp-1}\underline{A}^i\underline{B}\end{bmatrix}}_{\Gamma}\underline{u}(k-1)}_{\text{Lampau}}$$
$$\underbrace{+\underbrace{\begin{bmatrix}\underline{B} & \cdots & 0_{nxl}\\ \underline{A}\underline{B}+\underline{B} & \cdots & 0_{nxl}\\ \vdots & \ddots & \vdots\\ \sum_{i=0}^{Hu-1}\underline{A}^i\underline{B} & \cdots & \underline{B}\\ \sum_{i=0}^{Hu}\underline{A}^i\underline{B} & \cdots & \underline{A}\underline{B}+\underline{B}\\ \vdots & \vdots & \vdots\\ \sum_{i=0}^{Hp-1}\underline{A}^i\underline{B} & \cdots & \sum_{i=0}^{Hp-Hu}\underline{A}^i\underline{B}\end{bmatrix}}_{\Theta}\begin{bmatrix}\underline{\Delta\hat{u}}(k)\\ \vdots\\ \underline{\Delta\hat{u}}(k+Hu-1)\end{bmatrix}}_{\text{Prediksi}} \tag{2.16}$$

Selain itu, persamaan prediksi keluaran $\underline{\hat{y}}(k+i \mid k)$ dapat ditulis seperti berikut ini;

$$\underline{\hat{y}}(k+1 \mid k) = \underline{C}\,\underline{\hat{x}}(k+1 \mid k) \qquad (2.17)$$

$$\underline{\hat{y}}(k+2 \mid k) = \underline{C}\,\underline{\hat{x}}(k+2 \mid k) \qquad (2.18)$$

$$\vdots$$

$$\underline{\hat{y}}(k+Hp \mid k) = \underline{C}\,\underline{\hat{x}}(k+Hp \mid k) \qquad (2.19)$$

Persamaan (2.17) – (2.19) kemudian dapat ditulis kedalam vektor matriks sebagai berikut:

$$\begin{bmatrix} \underline{\hat{y}}(k+1 \mid k) \\ \vdots \\ \underline{\hat{y}}(k+Hp \mid k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{C} & \underline{0}_{mxn} & \cdots & \underline{0}_{mxn} \\ \underline{0}_{mxn} & \underline{C} & \cdots & \underline{0}_{mxn} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{mxn} & \underline{0}_{mxn} & \cdots & \underline{C} \end{bmatrix}}_{\underline{C}_y} \begin{bmatrix} \underline{\hat{x}}(k+1 \mid k) \\ \vdots \\ \underline{\hat{x}}(k+Hp \mid k) \end{bmatrix} \qquad (2.20)$$

2.6 Strategi Pengendali *Model Predictive Control* tanpa *Constraints*

Fungsi kriteria yang akan diminimumkan adalah fungsi kuadratik seperti pada persamaan (2.1) dan dapat ditulis sebagai berikut

$$V(k) = \left\| \underline{Y}(k) - \underline{T}(k) \right\|_{\underline{Q}}^2 + \left\| \underline{\Delta U}(k) \right\|_{\underline{R}}^2 \qquad (2.21)$$

Dimana:

$V(k)$ = fungsi kriteria

$\underline{Y}(k)$ = matriks keluaran terprediksi

$\underline{T}(k)$ = matriks sinyal acuan (trajectory)

$\underline{\Delta U}(k)$ = perubahan sinyal kendali

$$\underline{Y}(k) = \begin{bmatrix} \underline{\hat{y}}(k+1 \mid k) \\ \vdots \\ \underline{\hat{y}}(k+Hp \mid k) \end{bmatrix}, \ \underline{T}(k) = \begin{bmatrix} \underline{r}(k+1 \mid k) \\ \vdots \\ \underline{r}(k+Hp \mid k) \end{bmatrix},$$

$$\underline{\Delta U}(k) = \begin{bmatrix} \underline{\hat{u}}(k \mid k) \\ \vdots \\ \underline{\hat{u}}(k+Hu-1 \mid k) \end{bmatrix}$$

dan matriks faktor bobot $\underline{Q}$ dan $\underline{R}$ adalah sebagai berikut

$$\underline{Q} = \begin{bmatrix} Q(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q(Hp) \end{bmatrix} \tag{2.22}$$

$$\underline{R} = \begin{bmatrix} R(0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R(Hu-1) \end{bmatrix} \tag{2.23}$$

Bentuk fungsi kriteria menunjukkan bahwa vektor kesalahan (error) $\underline{Y}(k)-\underline{T}(k)$ diperhitungkan pada tiap pencuplikan dalam rentang *prediction horizon*, namun jika perhitungan error hanya diinginkan pada rentang waktu tertentu, hal ini dapat dilakukan dengan mengatur nilai faktor bobot Q bernilai 0 pada waktu tersebut. Selain vektor kesalahan, fungsi kriteria juga memperhitungkan perubahan dari vektor masukan yang hanya terjadi dalam rentang waktu *control horizon*.

Berdasarkan pada persamaan ruang keadaan (2.16) dan (2.20), maka matriks $\underline{Y}(k)$ dapat ditulis dalam bentuk;

$$\underline{Y}(k) = \underline{C}_Y \underline{\Psi}\, \underline{x}(k) + \underline{C}_Y \underline{\Gamma}\, \underline{u}(k-1) + \underline{C}_Y \underline{\Theta} \underline{\Delta U}(k) \tag{2.24}$$

Selain matriks-matriks di atas, didefinisikan juga suatu matriks penjejakan kesalahan $\underline{E}(k)$, yaitu selisih antara nilai trayektori acuan yang akan datang dengan tanggapan bebas dari sistem. Tanggapan bebas adalah tanggapan yang akan terjadi pada rentang *prediction horizon* jika tidak ada perubahan nilai masukan ($\Delta\underline{U}(k) = \underline{0}$) [3]. Persamaan matematis dari matriks $\underline{E}$ (k) adalah sebagai berikut ;

$$\underline{\mathcal{E}}(k) = \underline{T}(k) - \underline{C}_Y \underline{\Psi}\, \underline{x}(k) - \underline{C}_Y \underline{\Gamma}\, \underline{u}(k-1) \tag{2.25}$$

Persamaan (2.21) kemudian dapat ditulis kembali dalam bentuk yang mengandung matriks $\underline{E}(k)$ dan $\Delta\underline{U}(k)$ sebagai berikut ;

$$\begin{aligned} V(k) &= \left\| \underline{C}_y \underline{\Theta} \underline{\Delta U}(k) - \underline{\mathcal{E}}(k) \right\|_{\underline{Q}}^2 + \left\| \underline{\Delta U}(k) \right\|_{\underline{R}}^2 \\ &= \left[\underline{\Delta U}^T(k) \underline{\Theta}^T \underline{C}_y^T - \underline{\mathcal{E}}^T(k) \right] \underline{Q} \left[\underline{C}_y \underline{\Theta} \underline{\Delta U}(k) - \underline{\mathcal{E}}(k) \right] + \underline{\Delta U}^T(k) \underline{R} \underline{\Delta U}(k) \end{aligned}$$

$$= \underbrace{\underline{\mathcal{E}}^T(k)\underline{Q}\underline{\mathcal{E}}(k)}_{\underline{c_1}} - \underline{\Delta U}^T(k)2\underbrace{\underline{\Theta}^T \underline{C}_y^T \underline{Q}\underline{\mathcal{E}}(k)}_{\underline{\mathcal{G}}} + \underline{\Delta U}^T(k)\underbrace{\left[\underline{\Theta}^T \underline{C}_y^T \underline{Q}\underline{C}_y \underline{\Theta} + \underline{R}\right]}_{\underline{\mathcal{H}}}\underline{\Delta U}(k)$$

(2.26)

Pada persamaan (2.26), bagian $\underline{\mathcal{E}}^T(k)\underline{Q}\underline{\mathcal{E}}(k)$ tidak mengandung unsur $\Delta\underline{U}(k)$ sehingga bagian tersebut bisa dianggap konstan sehingga bagian tersebut tidak diikutsertakan dalam proses optimasi untuk menghitung nilai $\Delta\underline{U}(k)$. Persamaan (2.26) kemudian dapat ditulis kembali menjadi

$$V(k) = \underline{c_1} - \underline{\Delta U}^T(k)\underline{\mathcal{G}} + \underline{\Delta U}^T(k)\underline{\mathcal{H}}\,\underline{\Delta U}(k) \tag{2.27}$$

dimana

$$\underline{\mathcal{G}} = 2\underline{\Theta}^T \underline{C}_y^T \underline{Q}\underline{\mathcal{E}}(k) \tag{2.28}$$

dan

$$\underline{\mathcal{H}} = \underline{\Theta}^T \underline{C}_y^T \underline{Q}\underline{C}_y \underline{\Theta} + \underline{R} \tag{2.29}$$

Nilai optimal $\Delta\underline{U}(k)$ dapat dihitung dengan membuat gradien dari $V(k)$ bernilai nol [3]. Gradien $V(k)$ dari persamaan (2.27) adalah

$$\nabla_{\underline{\Delta U}(k)}\underline{V}(k) = -\underline{\mathcal{G}} + 2\underline{\mathcal{H}}\,\underline{\Delta U}(k) \tag{2.30}$$

Dengan membuat nol nilai $\nabla_{\underline{\Delta U}(k)}\underline{V}(k)$ pada persamaan (2.30), maka didapatkan nilai optimal dari perubahan sinyal kendali sebagai berikut:

$$\underline{\Delta U}(k)_{opt} = \frac{1}{2}\underline{\mathcal{H}}^{-1}\underline{\mathcal{G}} \tag{2.31}$$

Setelah nilai matriks $\Delta\underline{U}(k)$ didapatkan, maka nilai yang digunakan untuk mengubah sinyal kendali hanya nilai dari baris pertama matriks $\Delta\underline{U}(k)$ sedangkan nilai dari baris yang lain dari matriks $\Delta\underline{U}(k)$ dibuang [3].

2.7 Strategi Pengendali *Model Predictive Control* dengan *Constraints*

Pada setiap kendali proses, pasti terdapat batasan atau *constraints* pada amplitudo sinyal kendali. Selain itu, besarnya *slew rate* sinyal kendali juga dapat menjadi batasan. Persamaan *constraints* untuk amplitudo dan *slew rate* sinyal kendali secara berturut-turut adalah sebagai berikut

$$\underline{F}\underline{U}(k) \le \underline{f} \qquad (2.32)$$

$$\underline{E}\underline{\Delta U}(k) \le \underline{e} \qquad (2.33)$$

Pada algoritma MPC, yang akan dihitung adalah nilai optimal perubahan sinyal kendali $\Delta\underline{U}(k)$ sehingga sangat perlu untuk mengubah bentuk *constraints* yang belum mengandung $\Delta\underline{U}(k)$ menjadi bentuk *constraints* yang mengandung $\Delta\underline{U}(k)$. Sebagai contoh adalah pertidaksamaan (2.32), karena pada pertidaksamaan (2.32) belum mengandung $\Delta\underline{U}(k)$ maka bentuk pertidaksamaan (2.32) harus diubah terlebih dahulu menjadi bentuk yang mengandung $\Delta\underline{U}(k)$.

Untuk *constraints* yang berupa batasan nilai maksimum dan minimum sinyal kendali, maka pertidaksamaannya dapat ditulis sebagai berikut

$$\underline{u}_{\min} \le \underline{u}(k) \le \underline{u}_{\max} \qquad (2.34)$$

Pertidaksamaan (2.34) dapat ditulis menjadi dua bentuk yang terpisah seperti berikut ini

$$-\underline{u}(k) \le -\underline{u}_{\min} \qquad (2.35)$$

$$\underline{u}(k) \le \underline{u}_{\max} \qquad (2.36)$$

Pertidaksamaan (2.35) dan (2.36) masing-masing dapat ditulis dalam bentuk yang mengandung $\Delta\underline{U}(k)$ menjadi

$$-\underline{F'}\underline{\Delta U}(k) \le -\underline{u}_{\min} + \underline{F_1}u(k-1) \qquad (2.37)$$

$$\underline{F'}\underline{\Delta U}(k) \le \underline{u}_{\max} - \underline{F_1}u(k-1) \qquad (2.38)$$

dimana

$$\underline{F'} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{HuxHu} \qquad (2.39)$$

dan

$$\underline{F_1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{Hux1} \qquad (2.40)$$

Untuk pertidaksamaan (2.33), bentuknya tidak perlu diubah lagi karena pada pertidaksamaan tersebut sudah mengandung unsur $\underline{\Delta U}(k)$.

Pertidaksamaan (2.33), (2.37), dan (2.38) kemudian dapat disusun menjadi sebuah vektor matriks sebagai berikut

$$\underbrace{\begin{bmatrix} -\underline{F'} \\ \underline{F'} \\ \underline{E} \end{bmatrix}}_{\underline{\Omega}} \underbrace{\underline{\Delta U}(k)}_{\underline{\delta}} \leq \underbrace{\begin{bmatrix} -\underline{u}_{\min} + \underline{F_1}u(k-1) \\ u_{\max} - \underline{F_1}u(k-1 \\ \underline{e} \end{bmatrix}}_{\underline{\omega}} \qquad (2.41)$$

Vektor matriks pada pertidaksamaan (2.41) digunakan pada perhitungan nilai optimal perubahan sinyal kendali $\underline{\Delta U}(k)_{opt}$.

2.8 Metode *Quadratic Programming*

Fungsi kriteria pada pengendali MPC dengan *constraints* sama dengan fungsi kriteria pada pengendali MPC tanpa *constraints* (persamaan (2.35). Permasalahan utama proses optimasi ini adalah meminimalkan fungsi kriteria

$$\underline{\Delta U}^T(k)\underline{\mathcal{H}}\,\underline{\Delta U}(k) - \underline{\Delta U}^T(k)\underline{\mathcal{G}} \qquad (2.42)$$

berdasarkan pada pertidaksamaan *constraints* (2.43) atau

$$\min_{\underline{\theta}} \frac{1}{2}\underline{\delta}^T\,\underline{\Phi}\underline{\delta} + \underline{\phi}\underline{\delta} \qquad (2.43)$$

berdasarkan pada *constraints*

$$\underline{\Omega}\underline{\delta} \leq \underline{\omega} \qquad (2.44)$$

Bentuk (2.42) dan (2.44) adalah masalah optimasi standar yang disebut sebagai permasalahan *Quadratic Programming* (QP). Bila ada bagian yang aktif di dalam himpunan *constraints* pada persamaan (2.44), maka bagian aktif tersebut akan membuat pertidaksamaan (2.44) menjadi suatu persamaan

$$\underline{\Omega}_a \underline{\delta} = \underline{\omega}_a \tag{2.45}$$

dengan matriks $\underline{\Omega}_a$ adalah bagian yang aktif dari matriks pertidaksamaan (2.44). Persamaan (2.45) kemudian dijadikan sebagai *constraints* dari fungsi kriteria pada persamaan (2.43).

Permasalahan optimasi persamaan (2.43) dengan subyek terhadap persamaan (2.45) dapat diselesaikan dengan teori pengali *Lagrange*

$$\min_{\underline{\delta},\underline{\lambda}} L(\underline{\delta},\underline{\lambda}) \tag{2.46}$$

dengan

$$L(\underline{\delta},\underline{\lambda}) = \frac{1}{2}\underline{\delta}^T \underline{\Phi}\underline{\delta} + \underline{\phi}\underline{\delta} + \underline{\lambda}(\underline{\Omega}_a \underline{\delta} - \underline{\omega}_a) \tag{2.47}$$

Selanjutnya dengan melakukan diferensiasi parsial terhadap $\underline{\delta}$ dan $\underline{\lambda}$ dari persamaan (2.47), maka didapatkan kondisi *Karush-Kuhn-Tucker* sebagai berikut

$$\nabla_{\underline{\delta}} L(\underline{\delta},\underline{\lambda}) = \underline{\Phi}\underline{\delta} + \underline{\phi} + \underline{\Omega}_a^T \underline{\lambda} \tag{2.48}$$

$$\nabla_{\underline{\lambda}} L(\underline{\delta},\underline{\lambda}) = \underline{\Omega}_a \underline{\delta} - \underline{\omega}_a \tag{2.49}$$

atau

$$\nabla L(\underline{\delta},\underline{\lambda}) = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix} - \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \tag{2.50}$$

Selanjutnya dengan membuat $\nabla L(\underline{\delta},\underline{\lambda}) = 0$, maka didapatkan solusi optimal untuk $\underline{\delta}$ dan $\underline{\lambda}$ sebagai berikut

$$\begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix}_{opt} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix}^{-1} \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \tag{2.51}$$

Solusi pada *Quadratic Programming* pada kondisi normal menghasilkan nilai yang *feasible*, yaitu nilai yang memenuhi pertidaksamaan *constraints* yang ada dan dapat menghasilkan nilai fungsi kriteria minimum. Masalah yang paling sering muncul pada optimasi dengan *constraints* adalah solusi yang *infeasible*, dimana nilai yang dihasilkan tidak memenuhi pertidaksamaan *constraints* yang ada. *QP solver* akan menghentikan proses perhitungan jika terjadi solusi yang *infeasible*. Hal ini tentu tidak dapat diterima karena sinyal kendali hasil komputasi harus selalu ada untuk digunakan sebagai masukan bagi *plant*, sehingga sangat penting untuk membuat metode cadangan dalam menghitung sinyal masukan ketika algoritma MPC diterapkan.