# BAB 2

# MODEL PREDICTIVE CONTROL

## 2.1. Sejarah dan Konsep Dasar MPC

Sejarah MPC berawal dari pelopornya yaitu *Richalet*, *Cutler* dan *Ramarker* (1979). Suksesnya pengaplikasian *predictive control* dan kesederhanaan konsep serta teorinya menciptakan ketertarikan banyak akademi/universitas di seluruh dunia. *Predictive control* ini memungkinkan *self-tuning control* juga memiliki *variance control* yang minimum dan menggunakan algoritma *Generalize Predictive Control* (GPC). Pengaplikasian *predictive control* kemudian meluas antara lain pada *distillation column*, *hydrocracker*, *fluidized bed catalytic cracker*, *utility boiler*, *chemical reactor, transonic wind tunnel*, *robot arm*, *servo mechanism*, *mechatronic servo system*, *pulp and paper plant*.

Algoritma *Generalized Predictive Control* merupakan salah satu metode sistem kendali prediktif yang mampu beradaptasi dengan perubahan parameter sistem (pengendali adaptif). Sistem kendali prediktif termasuk ke dalam kategori konsep perancangan pengendali berbasis model proses di mana model proses nantinya digunakan untuk merancang pengendali *plant* dengan cara meminimasi *objective function* (fungsi kriteria).

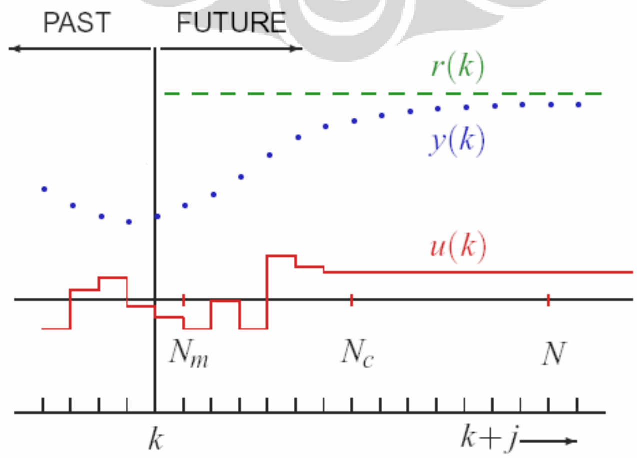
Pada pengendali prediktif, model internal yang merupakan pemodelan linier dari proses digunakan untuk mengetahui perilaku sistem. Dengan model internal ini perilaku perilaku sistem diprediksi dalam kurun waktu yang terbatas (yang disebut *preceding horizon*). Hasil dari prediksi ini kemudian digunakan pada tiap waktu pencuplikan, untuk mengoptimalkan keluaran sistem melalui sinyal masukan. Solusi dari permasalahan optimalisasi ini merupakan nilai masukan optimal bagi sistem untuk waktu tertentu. Terdapat banyak variasi metode pengendali prediktif dengan nama yang berbeda pula. Namun ide yang mendasari dari semua pengendali prediktif pada prinsipnya sama, yang nenbedakan satu sama lain di antaranya terletak pada model proses yang digunakan untuk mendiskripsikan sistem, model derau, serta fungís kriteria yag diminimisasi. Nama lain dalam literatur yang banyak digunakan untuk pengendali prediktif adalah *Model Predictive Control* (MPC).

Keuntungan MPC dibandingkan metode pengendali konvensional lainnya di antaranya adalah :

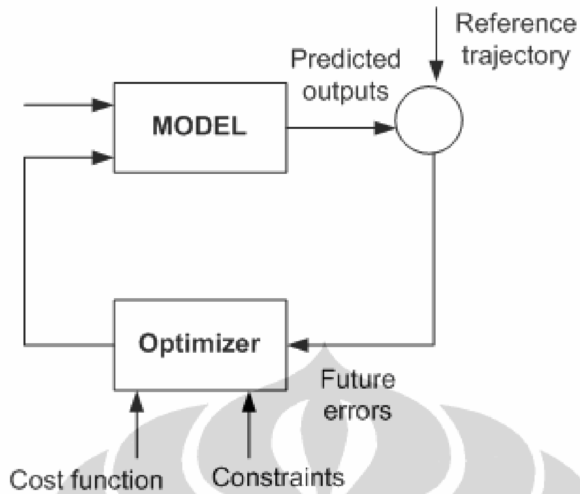
1. Dapat memperhitungkan batasan pada sistem (*constraints*) dalam perancangan pengendali.
2. Konsepnya sangat intuitif serta penalaannya sangat mudah.
3. Dapat digunakan untuk mengendalikan proses yang beragam, mulai dari proses yang sederhana sampai proses yang kompleks, mempunyai waktu tunda besar, *non-minimum phase* atau proses yang tidak stabil.
4. Dapat menangani sistem multivariabel.
5. Mempunyai kompensasi terhadap waktu tunda.
6. Mempunyai kemampuan pengendali *feed forward* untuk mengkompensasi gangguan yang terukur.
7. Mudah untuk mengimplementasikan pengendali yang diperoleh.
8. Sangat berguna jika sinyal referensi untuk masa datang diketahui.

Prinsip yang mendasari pada setiap jenis pengendali prediktif antara lain :

1. Menggunakan model proses untuk memprediksi keluaran yang akan datang dalam rentang waktu yang telah ditentukan (*horizon*).
2. Menghitung sinyal kendali dengan meminimasi *objective function* (fungsi kriteria) yang ditetapkan sebelumnya dengan tujuan untuk menjaga keluaran proses agar sedekat mungkin dengan trayektori acuan.
3. Sinyal kendali $u(k \mid k)$ dikirim ke proses sedangkan sinyal kendali terprediksi berikutnya dibuang, karena pada pencuplikan berikutnya, keluaran $y(k+1)$ sudah diketahui nilainya. Maka langkah pertama diulang dengan nilai keluaran proses yang baru dan semua prosedur perhitungan yang diperlukan diperbaiki. Sinyal kendali yang baru $u(k+1 \mid k+1)$ nilainya berbeda dengan $u(k+1 \mid k)$, diperoleh dengan menggunakan konsep *receding horizon*. Konsep *receding horizon* dapat dilihat pada gambar 2.1




**Gambar 2.1.** Strategi *receding horizon*




**Gambar 2.2.** Struktur dasar pengendali MPC

Lima konsep yang dikenal di dalam MPC yaitu :

1. Model proses dan *disturbance*.
2. *Performance index*.
3. Pengendalian/penanganan *constraints*.
4. Optimalisasi.
5. *Receding horizon principle*.

MPC menyediakan teknologi yang mendukung operasi industri dengan kemampuan antara lain :

1. Batasan (*constraints*) selalu dicukupi.
2. Memberikan gambaran awal operasi yang memungkinkan/memenuhi.
3. Memaksimalkan produktivitas modal.
4. Memaksimalkan sisa masa produktif plant.

Di dalam MPC dibutuhkan pengontrolan yang ketat untuk menjamin :

1. Keprediksian dan reproduksi proses produksi.
2. Fleksibilitas proses produksi.

## 2.2. Formulasi Model Predictive Control

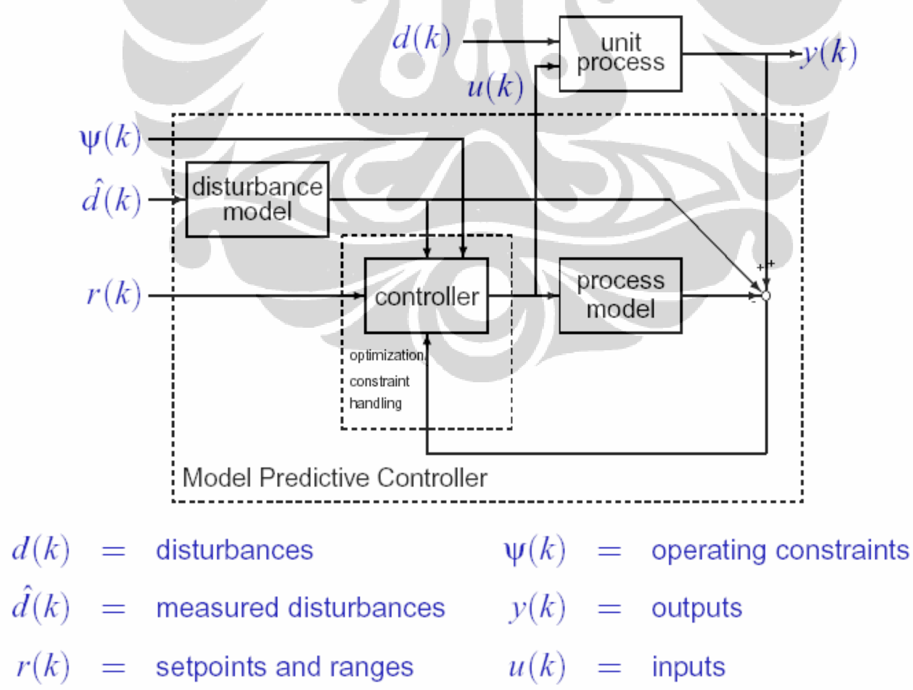
Dalam formulasi dasar MPC ada beberapa asumsi yang dibuat yaitu model acuan bersifat linier, fungsi kriteria merupakan fungsi kuadratik, dan *constraints* berbentuk pertidaksamaan linier.

### 2.2.1. Model Proses

Dalam pemodelan ini digunakan ruang keadaan diskrit dan linier. Persamaan ruang keadaan diskrit yang digunakan adalah sebagai berikut :

$$\underline{x}(k+1\,|\,k) = \underline{A}\underline{x}(k) + \underline{B}\underline{u}(k) \tag{2.1}$$

$$\underline{y}(k) = \underline{C}\underline{x}(k) \tag{2.2}$$




**Gambar 2.3.** Blok diagram proses MPC

di mana :

$x(k)$ = vektor ruang keadaan berdimensi-n

$y(k)$ = vektor keluaran terukur berdimensi-m

$u(k)$ = vektor masukan berdimensi-r

$\underline{A}$ = matriks keadaan berdimensi nxn

$\underline{B}$ = matriks masukan berdimensi nxr

$\underline{C}$ = matriks keluaran berimensi mxn

Persamaan ruang keadaan di atas merupakan kondisi ideal, di mana tidak ada gangguan (*disturbance*). Pada pembahasan penulisan ini sistem yang digunakan mendapat gangguan (*disturbance*) dari glukosa yang terdapat dalam makanan yang dikonsumsi si penderita DM. Sehingga nantinya terdapat matriks *disturbance*.

Dalam perhitungan prediksi keluaran dengan MPC, sinyal masukan yang digunakan adalah perubahan nilai sinyal masukan saat ini $\Delta u(k)$. Di mana perubahan tersebut merupakan selisih antara nilai sinyal masukan saat $k$ atau $u(k)$ dan satu langkah sebelumnya $u(k-1)$. Oleh karena itu persamaan ruang keadaan 2.1 harus harus diubah bentuknya supaya terdapat unsur $\Delta u(k)$ di dalamnya. Hal pertama yang dilakukan adalah mencari prediksi dari persamaan ruang keadaan 2.1 dengan melakukan iterasi terhadap persamaan tersebut sebagai berikut :

$$\underline{x}(k+1 \mid k) = \underline{A}x(k \mid k) + \underline{B}u(k \mid k) \tag{2.3}$$

$$\begin{aligned}\underline{x}(k+2 \mid k) &= \underline{A}x(k+1 \mid k) + \underline{B}u(k \mid k) \\ &= \underline{A^2 x}(k) + \underline{AB}u(k \mid k) + \underline{B}u(k+1 \mid k) \end{aligned} \tag{2.4}$$

$$\underline{x}(k+Hp \mid k) = \underline{A}\underline{x}(k+Hp-1 \mid k) + \underline{B}u(k+Hp-1 \mid k)$$

$$= \underline{A}^2 \underline{x}(k) + \underline{A}^{Hp-1} \underline{B}u(k \mid k) + \ldots + \underline{B}u(k+Hp-1 \mid k) \tag{2.5}$$

Perubahan sinyal masukan $\Delta u(k+i \mid k)$ merupakan selisih antara nilai sinyal masukan prediksi saat *i*-langkah ke depan yang dilakukan saat waktu *k* atau $u(k+i \mid k)$, dengan sinyal masukan satu langkah sebelum $i$ atau $u(k+i-1 \mid k)$, atau secara matematis dapat ditulis sebagai berikut :

$$\Delta u(k+i \mid k) = u(k+i \mid k) - u(k+i-1 \mid k) \tag{2.6}$$

Sedangkan pada tiap langkah *k*, informasi yang dapat diketahui hanya nilai $u(k-1)$, sehingga :

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

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

$$\vdots$$

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

Dengan mensubtitusi persamaan (2.7)-(2.9) ke persamaan (2.3)-(2.5) maka didapat bentuk persamaan sebagai berikut :

$$\underline{x}(k+1 \mid k) = \underline{A}\underline{x}(k \mid k) + \underline{B}\left[\Delta u(k \mid k) + u(k-1)\right] \tag{2.10}$$

$$x(k+2 \mid k) = \underline{A}^2 \underline{x}(k) + \underline{A}\underline{B}\left[\Delta u(k \mid k) + u(k-1)\right] + \underline{B}\left[u(k+1 \mid k) + \Delta u(k \mid k) + u(k-1)\right]$$

$$= \underline{A}^2 \underline{x}(k) + \left(\underline{A} + \underline{I}\right)\underline{B}\Delta u(k \mid k) + \underline{B}\Delta u(k+1 \mid k) + \left(\underline{A} + \underline{I}\right)\underline{B}u(k-1) \tag{2.11}$$

$$\underline{x}(k+Hu \mid k) = \underline{A}^{Hu} \underline{x}(k) + (A^{Hu-1} + \ldots + \underline{A} + \underline{I})\underline{B}\Delta u(k \mid k)\ldots + \underline{B}\Delta u(k+Hu-1 \mid k)$$

$$+ (A^{Hu-1} + \ldots + \underline{A} + \underline{I})\underline{B}u(k \mid k) \tag{2.12}$$

Untuk $i \geq Hu$ nilai perubahan sinyal masukan $\Delta u(k+i \mid k)$ selalu sama sehingga :

$$\underline{x}(k+Hu+1 \mid k) = \underline{A}^{Hu+1} \underline{x}(k) + (A^{Hu} + \ldots + \underline{A} + \underline{I})\underline{B}\Delta u(k \mid k)\ldots + (\underline{A} + \underline{I})\underline{B}\Delta u(k+Hu-1 \mid k)$$

$$+ (A^{Hu} + \ldots + \underline{A} + \underline{I})\underline{B}u(k \mid k) \tag{2.13}$$

$$\underline{x}(k+Hp\,|\,k)=\underline{A}^{Hp}\,\underline{x}(k)+(A^{Hp-1}+\ldots+\underline{A}+\underline{I})\underline{B}\Delta u(k\,|\,k)\ldots+(\underline{A}^{Hp-Hu}+\ldots+\underline{A}+\underline{I})\underline{B}\Delta u(k+Hu-1\,|\,k)$$

$$+(A^{Hp-1}+\ldots+\underline{A}+\underline{I})\underline{B}u(k\,|\,k) \qquad (2.14)$$

Persamaan (2.10)-(2.14) dapat ditulis k dalam bentuk vektor matriks sebagai berikut :

$$\begin{bmatrix} x(k+1\,|\,k) \\ \vdots \\ x(k+Hu\,|\,k) \\ x(k+Hu-1\,|\,k) \\ \vdots \\ x(k+Hp\,|\,k) \end{bmatrix} = \underbrace{\underbrace{\begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^{Hu} \\ \underline{A}^{Hu+1} \\ \vdots \\ \underline{A}^{Hp} \end{bmatrix}}_{\underline{\psi}} x(k) + \underbrace{\begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\Gamma}} u(k-1)}_{\text{Lampau}} + \underbrace{\underbrace{\begin{bmatrix} \underline{B} & \cdots & 0 \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} & \cdots & \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} & \cdots & \underline{A}\underline{B}+\underline{B} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^{Hp-Hu} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\Theta}} \begin{bmatrix} \Delta u(k\,|\,k) \\ \vdots \\ \Delta u(k.Hu-1\,|\,k) \end{bmatrix}}_{\text{Presiksi}} \qquad (2.15)$$

Sedangkan untuk persamaan keluaran 2.1. dapat dilakukan prediksi sebagai berikut :

$$\underline{y}(k+1\,|\,k)=\underline{C_y}\,\underline{x}(k+1\,|\,k) \qquad (2.16)$$

$$\underline{y}(k+2\,|\,k)=\underline{C_y}\,\underline{x}(k+2\,|\,k) \qquad (2.17)$$

$$\vdots$$

$$\underline{y}(k+Hp\,|\,k)=\underline{C_y}\,x(k+Hp\,|\,k)$$

(2.18)

Persamaan 2.18 dapat ditulis ke dalam bentuk vektor mariks sebagai berikut :

$$\begin{bmatrix} y(k+1\,|\,k) \\ \vdots \\ y(k+Hp\,|\,k) \end{bmatrix} = \begin{bmatrix} \underline{C_y} & 0 & \cdots & 0 \\ 0 & C_y & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_y \end{bmatrix} \begin{bmatrix} \underline{x}(k+1\,|\,k) \\ \vdots \\ \underline{x}(k+Hp\,|\,k) \end{bmatrix} \qquad (2.19)$$

**<u>Model Proses Non Linier</u>**

Nonlinier MPC lebih kompleks daripada linier MPC. Nonlinier MPC dapat mengendalikan sistem/model nonlinier dengan tingkat gangguan yang besar. Selain itu pada nonlinier MPC titik pengoperasian dapat berubah-ubah, atau dengan kata lain rentang pengontrolannya lebar dan dinamis. Secara umum, dimodelkan sebagai berikut :

$$\begin{aligned} x(k+1) &= f\left(x(k), u(k), u_{MD}(k), w(k)\right), \\ y(k) &= g(x(k)) + \xi(k), \end{aligned} \tag{2.20}$$

di mana w(k) adalah *unmeasured disturbance* dan ξ(k) adalah *measurement noise*. Di dalam perancangan ini, keduanya diabaikan. Permasalahan di dalam nonlinier MPC adalah terdapat beberapa *local optima* dan salah satunya tidak dapat dipastikan apakah optimum yang ditemukan merupakan optimum global atau bukan. Tujuan faktor pembatasan di dalam nonlinier MPC adalah untuk mempercepat dan memastikan solusi yang sebenarnya dalam *real time*.

### 2.2.2. Pembentukan Constraints

Pada dasarnya sinyal dengan *range* yang tidak terbatas sangat tidak realistis, karena dalam kondisi nyata semua proses memiliki batasan (*constraints*). *Constraints* yang berlaku pada proses dapat berasal dari amplitudo sinyal kendali dan *slew rate* pada aktuator. Secara berturut-turut *constraints* tersebut dapat dinyatakan dalam pertidaksamaan berikut ini :

$$\underline{F}U(k) \leq \underline{f} \tag{2.21}$$

$$E\Delta U(k) \leq \underline{e} \tag{2.22}$$

Karena dalam penyelesaian masalah pada MPC akan digunakan nilai $\Delta u$, maka pertidaksamaan 2.21 dan 2.22 diubah ke dalam bentuk pertidaksamaan yang mengandung unsur $\Delta u$. Konversi dilakukan sebagai berikut :

1. Untuk *constraints* pada amplitudo sinyal kendali, bila batasannya merupakan nilai maksimum dan minimum maka pertidaksamaannya akan berbentuk seperti di bawah ini :

$$u_{\min} \leq u(k) \leq u_{\max} \tag{2.23}$$

Dengan membagi pertidaksamaan di atas masing-masing ke dalam sebuah pertidaksamaan, seperti berikut :

$$-\underline{u}(k) \leq -\underline{u}_{\min} \tag{2.24}$$

$$\underline{u}(k) \leq \underline{u}_{\max} \tag{2.25}$$

Mengingat $u(k) = \Delta u(k) + u(k-1)$ maka pertidaksamaan 2.24 dan 2.25 dapat diubah ke dalam bentuk sebagai berikut :

$$u_{\min} \leq \Delta u(k) + u(k-1) \tag{2.26}$$

$$u_{\max} \leq -\Delta u(k) - u(k-1) \tag{2.27}$$

Dengan membawa faktor $\Delta u(k)$ ke sebelah kiri tanda pertidaksamaan dan memasukkan matriks $F$ maka petidaksamaan 2.26 dan 2.27 dapat ditulis seperti di bawah ini :

$$-\underline{F'}\underline{\Delta U}(k) \leq -\underline{u}_{\min} + \underline{F_1}u(k-1) \tag{2.28}$$

$$\underline{F'}\underline{\Delta U}(k) \leq \underline{u}_{\max} - \underline{F_1}u(k-1) \tag{2.29}$$

di mana :

Simulasi pengendalian..., Nur Hanifah Yuninda, FT UI, 2008

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

$$\underline{F_1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{Hux1} \tag{2.30}$$

2. Untuk *constraints slew rate* pada aktuator tidak diperlukan konversi karena pertidaksamaan sudah mengandung nilai $\Delta u(k)$. Sehingga secara umum semua *constraints* dapat diturunkan ke dalam sebuah pertidaksamaan berikut :

   $\underline{\Omega}\Delta U(k) \leq \underline{\omega}$

   di mana : $\underline{\Omega} = \begin{bmatrix} -\underline{F}' \\ \underline{F}' \\ \underline{E} \end{bmatrix}$ dan $\underline{\omega} = \begin{bmatrix} -\underline{u}_{\min} + \underline{F_1}u(k-1) \\ u_{\max} - \underline{F_1}u(k-1 \\ \underline{e} \end{bmatrix}$

   atau dapat ditulis sebagai berikut :

$$\underbrace{\begin{bmatrix} -\underline{F}' \\ \underline{F}' \\ \underline{E} \end{bmatrix}}_{\underline{\Omega}} \underbrace{\underline{\Delta U}(k)}_{\underline{\theta}} \leq \underbrace{\begin{bmatrix} -\underline{u}_{\min} + \underline{F_1}u(k-1) \\ u_{\max} - \underline{F_1}u(k-1 \\ \underline{e} \end{bmatrix}}_{\underline{\omega}} \tag{2.31}$$

### 2.2.3. Fungsi Kriteria

Seperti telah dijelaskan sebelumnya tujuan dari algoritma MPC adalah meminimasi fungsi kriteria untuk mendapatkan sinyal kendali. Dalam algoritma MPC digunakan fungsi kriteria kuadratik sebagai berikut :

$$V(k) = \sum_{i=1}^{Hp} \left\| \hat{y}(k+i\,|\,k) - r(k+i\,|\,k) \right\|_{Q(i)}^{2} + \sum_{i=0}^{Hu-1} \left\| \Delta u(k+i\,|\,k) \right\|_{R(i)}^{2} \tag{2.32}$$

di mana :

$\hat{y}(k+i\,|\,k)$ = keluaran terprediksi untuk *i*-langkah ke depan

$r(k+i\,|\,k)$ = nilai acuan (*reference trajectory*)

$Q(i), R(i)$ = faktor bobot

$\Delta u(k+i\,|\,k)$ = perubahan nilai sinyal masukan

$Hu$ = rentang waktu pengendalian (*control horizon*)

$Hp$ = rentang waktu prediksi (*prediction horizon*)

Diasumsikan bahwa nilai $Hu \le Hp$ dan nilai $\Delta u(k+i\,|\,k) = 0$ untuk $i \ge Hu$ sehingga :

$$\Delta u(k+i\,|\,k) = \Delta u(k+Hu-i\,|\,k) \tag{2.33}$$

untuk semua $i \ge Hu$. Bentuk fungsi kriteria 2.32 menunjukkan bahwa vektor kesalahan (eror) $\hat{y}(k+i\,|\,k) - r(k+i\,|\,k)$ diperhitungkan pada tiap pencuplikan dalam rentang *prediction horizon*. Walau demikian tetap ada kemungkinan untuk memperhitungkan kesalahan tersebut hanya pada langkah-langkah tertentu dengan cara mengatur nilai faktor bobot $Q$ bernilai 0 pada langkah yang diinginkan. Selain vektor kesalahan, fungsi kriteria 2.32 juga memperhitungkan perubahan dari vektor masukan, yang hanya terjadi dalam rentang waktu *control horizon*. Pemilihan penggunaan operator $\Delta u(k+i\,|\,k)$ daripada $u(k+i\,|\,k)$ dalam fungsi kriteria disebabkan karena $\Delta u(k+i\,|\,k)$ merupakan perubahan yang terjadi pada sinyal masukan bagi *plant*, yang tidak diharapkan terjadi pada sistem. Perubahan sinyal kendali diminimalkan sehingga bernilai mendekati atau sama dengan nol.

## 2.3. Strategi Penurunan Sinyal Kendali MPC tanpa *constraints*

Pada kasus MPC tanpa *constraints*, fungsi kriteria diminimasi dengan mengatur nilai gradien sama dengan nol dari fungsi kriteria tersebut, untuk mendapatkan nilai perubahan sinyal $\Delta u$ optimal. Fungsi kriteria 2.32 dapat ditulis ke dalam bentuk berikut :

$$V(k) = \left\| \underline{Y}(k) - \underline{T}(k) \right\|_Q^2 + \left\| \underline{\Delta U}(k) \right\|_R^2 \tag{2.34}$$

di mana

$$\underline{Y}(k) = \begin{bmatrix} \hat{y}(k+1 \mid k) \\ \vdots \\ \hat{y}(k+Hp \mid k) \end{bmatrix}, \quad \underline{T}(k) = \begin{bmatrix} r(k+1 \mid k) \\ \vdots \\ r(k+Hp \mid k) \end{bmatrix}, \quad \underline{\Delta U}(k) = \begin{bmatrix} \Delta u(k \mid k) \\ \\ \Delta u(k+Hu-1 \mid k) \end{bmatrix}$$

dan matriks bobot ***Q = I*** dan ***R = 0***

Dari persamaan ruang keadaan 2.15 maka nilai keluaran Y adalah :

$$\underline{Y}(K) = \underline{C}_y \underline{\Psi} x(k) + \underline{C}_y \Gamma u(k-1) + \underline{C}_y \underline{\Theta} \Delta \underline{U}(k) \tag{2.35}$$

Didefinisikan suatu matriks penjejakan kesalahan $\underline{E}$, di mana matriks ini merupakan perbedaan antara nilai trayektori dengan respons bebas dari sistem. Respon bebas merupakan respon yang terjadi selama *prediction horizon* dan tidak terjadi perubahan sinyal masukan $(\Delta u = 0)$.

$$\underline{\mathrm{E}}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi} x(k) - \underline{C}_y \underline{\Gamma} u(k-1) \tag{2.36}$$

Dengan mensubtitusi persamaan 2.35 dan 2.36 ke dalam persamaan fungsi kriteria 2.34 maka akan diperoleh :

$$V(k) = \left\| \underline{C}_y \underline{\Theta} \Delta \underline{U}(k) - \underline{\mathrm{E}}(k) \right\|_Q^2 + \left\| \underline{\Delta U}(k) \right\|_R^2$$

$$= \left[ \underline{\Delta U}(k)^T \underline{\Theta}^T \underline{C}_y - \underline{\mathrm{E}}(k)^T \right] \underline{Q} \left[ \underline{C}_y \underline{\Theta} \underline{\Delta U}(k) - \underline{\mathrm{E}}(k) \right] + \underline{\Delta U}^T(k) \underline{R} \underline{\Delta U}(k)$$

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

2.37)

Karena fungsi kriteria digunakan untuk mencari perubahan nilai sinyal masukan optimal $\underline{\Delta U}(k)$, maka bagian $\underline{E}(k)^T \underline{Q}\underline{E}(k)$ dianggap sebagai konstanta yang tidak berpengaruh dalam minimasi fungsi kriteria tersebut, sehingga fungsi kriteria dapat ditulis sebagai berikut :

$$\underline{V}(k) = \text{konst} - \underline{\Delta U}^T(k)\underline{\mathcal{G}} + \underline{\Delta U}^T(k)\underline{\mathcal{H}}\,\underline{\Delta U}(k) \tag{2.38}$$

di mana

$$\underline{\mathcal{G}} = 2\underline{\Theta}^T \underline{C}_y^T \underline{Q}\underline{\mathcal{E}}(k) \tag{2.39}$$

dan

$$\underline{\mathcal{H}} = \underline{\Theta}^T \underline{C}_y^T \underline{Q}\underline{C}_y \underline{\Theta} + \underline{R} \tag{2.40}$$

Nilai optimal $\underline{\Delta U}(k)$ dapat dihitung dengan membuat gradien dari $V(k)$ bernilai nol [Mcjw00]. Gradien dari $V(k)$ pada persamaan (2.28) adalah sebagai berikut

$$\nabla_{\underline{\Delta U}(k)} \underline{V}(k) = -\underline{\mathcal{G}} + 2\underline{\mathcal{H}}\,\underline{\Delta U}(k) \tag{2.41}$$

Dengan membuat nol persamaan (2.41), maka didapatkan nilai optimal dari perubahan sinyal kendali sebagai berikut

$$\underline{\Delta U}(k)_{opt} = \frac{1}{2}\underline{\mathcal{H}}^{-1}\underline{\mathcal{G}} \tag{2.42}$$

Setelah nilai matriks $\underline{\Delta U}(k)$ didapatkan, maka perlu diingat bahwa nilai yang digunakan untuk mengubah sinyal kendali hanya nilai dari baris pertama matriks $\underline{\Delta U}(k)$ sedangkan nilai dari baris yang lain dari matriks $\underline{\Delta U}(k)$ dibuang [Mcjw00].

## 2.4. Strategi Penurunan Sinyal Kendali MPC dengan *constraints*

Pada kasus MPC dengan *constraints*, selama *constraints* yang ada tidak aktif maka penyelesaiannya sama dengan MPC tanpa *constraints*. Namun bila ada *constraints* yang aktiif maka minimisasi fungsi kriteria perlu memperhatikan adanya *constraints* tersebut. Minimisasi fungsi kriteria tidak lagi dilakukan dengan menggunakan gradien seperti pada kasus tanpa *constraints*, tetapi dilakukan supaya nilai perubahan sinyal optimal dari hasil minimisasi juga memenuhi pertidaksamaan *constraints* yang ada. Minimisasi fungsi kriteria :

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

Berdasarkan pada pertidaksamaan *constraints* $\underline{\Omega}\Delta U(k) \leq \underline{\omega}$, merupakan permasalahan optimasi yang dikenal dengan *Quadratic Programming*. Solusi pada *Quadratic Programming* (QP) untuk kondisi normal menghasilkan nilai yang *feasible*, di mana nilai ini akan menghasilkan nilai fungsi kriteria minimum dan memenuhi pertidaksamaan yang ada. Pada optimasi dengan *constraints* terdapat masalah yang sering muncul yaitu solusi yang *infeasible*. Solusi *infeasible* merupakan solusi di mana hasil minimasi dari fungsi kriteria tidak memenuhi pertidaksamaan *constraints* atau sebaliknya. *QP solver* (biasanya dilakukan oleh komputer) akan menghentikan proses perhitungan bila terjadi solusi yang *infeasible*. Hal ini tidak dapat diterima karena sinyal kendali hasil komputasi harus selalu ada sebagai masukan bagi *plant*, sehingga sangatlah penting untuk membuat metode cadangan dalam memperhitungkan sinyal masukan bila MPC diimplementasikan.

## 2.5. Metode Quadratic Programming

Fungsi kriteria pada pengendali MPC dengan *constraints* sama dengan pengendali MPC tanpa *constraints*. Persmasalahan utama proses optimasi ini adalah

$$\text{Meminimalkan } \underline{\Delta U}^T(k)\underline{\mathcal{H}}\,\underline{\Delta U}(k) - \underline{\Delta U}^T(k)\underline{\mathcal{G}} \tag{2.43}$$

Berdasarkan pada pertidaksamaan *constraint* (2.31), atau dapat ditulis menjadi

$$\min_{\underline{\theta}} \frac{1}{2}\underline{\theta}^T \underline{\Phi}\,\underline{\theta} + \underline{\phi}\underline{\theta} \tag{2.44}$$

berdasarkan

$$\underline{\Omega}\underline{\theta} \leq \underline{\omega} \tag{2.45}$$

Bentuk (2.44) dan (2.45) adalah masalah optimasi standar yang disebut sebagai permasalahan *Quadratic Programming* (QP). Bila ada bagian yang aktif pada himpunan *constraints* pada (2.45), maka bagian aktif tersebut akan membuat pertidaksamaan (2.45) menjadi suatu persamaan. Misalkan $\underline{\Omega}_a$ adalah bagian yang aktif maka subyek dari (2.44) menjadi

$$\underline{\Omega}_a\underline{\theta} = \underline{\omega}_a \tag{2.46}$$

Permasalahan optimasi (2.44) dengan subyek terhadap persamaan (2.46) dapat diselesaikan dengan teori pengali *Lagrange*

$$\min_{\theta,\lambda} L(\theta,\lambda) \tag{2.47}$$

dengan

$$L(\theta,\lambda) = \frac{1}{2}\underline{\theta}^T \underline{\Phi}\,\underline{\theta} + \underline{\phi}\underline{\theta} + \underline{\lambda}(\underline{\Omega}_a\underline{\theta} - \underline{\omega}_a) \tag{2.48}$$

Dengan melakukan diferensiasi parsial terhadap $\underline{\theta}$ dan $\underline{\lambda}$ dari persamaan (2.48), maka didapatkan

$$\nabla_\theta L(\theta,\lambda) = \underline{\Phi}\underline{\theta} + \underline{\phi} + \underline{\Omega}_a^T \underline{\lambda} \tag{2.49}$$

$$\nabla_\lambda L(\theta,\lambda) = \underline{\Omega}_a \underline{\theta} - \underline{\omega}_a \tag{2.50}$$

atau dapat ditulis ke dalam bentuk matriks sebagai berikut

$$\nabla L(\theta,\lambda) = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\theta} \\ \underline{\lambda} \end{bmatrix} - \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \tag{2.51}$$

Dengan membuat $\nabla L(\theta,\lambda) = 0$, maka didapatkan solusi optimal sebagai berikut

$$\begin{bmatrix} \underline{\theta} \\ \underline{\lambda} \end{bmatrix}_{opt} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix}^{-1} \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \tag{2.52}$$

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*, di mana nilai yang dihasilkan tidak memenuhi pertidaksamaan *constraints* yang ada. *QP solver* (biasanya komputer) akan menghentikan proses penghitungan 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. Beberapa pendekatan yang dapat dilakukan untuk menghindari terjadinya solusi yang *infeasible* pada MPC antara lain [Mcjw00] :

- Menghindari *constraints* pada keluaran.
- Mengatur *constraints* untuk setiap langkah pencuplikan *k*.
- Mengatur *horizon* untuk setiap langkah pencuplikan *k*.

## 2.6. Diskritisasi Persamaan Continuous-Time State Space

$$e^{\underline{A}t} = 1 + \underline{A}t + \frac{1}{2!}\underline{A}^2 t^2 + \ldots + \frac{1}{k!}\underline{A}^k t^k + \ldots = \sum_{k=0}^{\infty} \frac{\underline{A}^k t^k}{k!} \qquad (2.53)$$

karena konvergensi dari *infinite* $\sum_{k=0}^{\infty} \underline{A}^k t^k$ maka persamaan 2.53 dapat didiferensialkan sebagai berikut :

$$\frac{d}{dt} e^{\underline{A}t} = \underline{A} + \underline{A}^2 t + \frac{\underline{A}^3 t^2}{2!} + \ldots + \frac{\underline{A}^k t^{k-1}}{(k-1)!} + \ldots$$

$$= \underline{A}\left[ I + \underline{A}t + \frac{\underline{A}^2 t^2}{2!} + \ldots + \frac{\underline{A}^{k-1} t^{k-1}}{(k-1)!} + \ldots \right] = \underline{A}e^{\underline{A}t}$$

$$= \left[ \underline{I} + \underline{A}t + \frac{\underline{A}^2 t^2}{2!} + \ldots + \frac{\underline{A}^{k-1} t^{k-1}}{(k-1)!} + \ldots \right]\underline{A} = e^{\underline{A}t}\,\underline{A} \qquad (2.54)$$

Matriks eksponensial memiliki penjabaran sebagai berikut :

$$e^{\underline{A}(t+s)} = e^{\underline{A}t} e^{\underline{A}s} \qquad (2.55)$$

jika $s = -t$, maka

$$e^{\underline{A}t} e^{-\underline{A}t} = e^{-\underline{A}t} e^{\underline{A}t} = e^{\underline{A}(t-t)} = \underline{I} \qquad (2.56)$$

$e^{\underline{A}t}$ nonsingular, saat *inverse* $e^{\underline{A}t}$ yaitu $e^{-\underline{A}t}$ ada nilainya.

$$e^{(\underline{A}+\underline{B})t} = e^{\underline{A}t} e^{\underline{B}t} \qquad \text{jika } \mathbf{AB = BA} \qquad (2.57)$$

$$e^{(\underline{A}+\underline{B})t} \neq e^{\underline{B}t} e^{\underline{A}t} \qquad \text{jika } \mathbf{AB \neq BA} \qquad (2.58)$$

Persamaan *continuous-time state space* yang akan didiskritisasi adalah sebagai berikut :

$$\dot{\underline{x}} = \underline{A}x + \underline{B}u \qquad (2.59)$$

$$y = \underline{C}x + \underline{D}u \qquad (2.60)$$

di mana x merupakan vektor keadaan (*n-vector*), u merupakan vektor input (*r-vector*), $\underline{A}$ adalah matriks konstan berdimensi nxn, dan $\underline{B}$ adalah matriks konstan berdimensi rxr.

Persamaan 2.59 dapat ditulis kembali sebagai berikut :

$$\dot{\underline{x}}(t) - \underline{A}x(t) = \underline{B}u(t) \tag{2.61}$$

dengan mengalikan kedua sisi pada persamaan 2.59 dengan $e^{-At}$ diperoleh :

$$e^{-\underline{A}t}\left[\dot{x}(t) - \underline{A}x(t)\right] = \frac{d}{dt}\left[e^{-\underline{A}t}x(t)\right] = e^{-\underline{A}t}\underline{B}u(t) \tag{2.62}$$

Integralkan kedua sisi persamaan 2.62 di antara 0 dan t diperoleh :

$$e^{-At}x(t) = x(0) + \int_0^t e^{-\underline{A}\tau}\underline{B}u(\tau)d\tau$$

$$x(t) = e^{\underline{A}t}x(0) + \int_0^t e^{\underline{A}(t-\tau)}\underline{B}u(\tau)d\tau \tag{2.63}$$

Pemecahan persamaan keadaan dimulai dengan inisialisasi keadaan $x(t_0)$ yaitu :

$$x(t) = e^{\underline{A}(t-t_0)}x(t_0) + \int_{t_0}^t e^{\underline{A}(t-\tau)}\underline{B}u(\tau)d\tau \tag{2.64}$$

Diasumsikan input vektor *u(t)* berubah hanya pada ruang *sampling* singkat yang sama. Operasi *sampling* disini dapat diubah-ubah dan merupakan permisalan. Selanjutnya hitung persamaan keadaan waktu diskrit di mana $t = kT$ dan k = 0,1,2,… , maka persamaan 2.59 dan 2.60 dapat dituliskan sebagai berikut:

$$x((k+1)T) = G(T)x(kT) + H(T)u(kT) \tag{2.65}$$

Matriks $\underline{G}$ dan $\underline{H}$ bergantung pada periode sampling T. Saat periode sampling T ditetapkan, $\underline{G}$ dan $\underline{H}$ merupakan matriks konstan. Persamaan 2.63 digunakan untuk menentukan $\underline{G}(T)$ dan $\underline{H}(T)$. Asumsikan input *u(t)* di-*sample* dan ditempatkan pada *zero-order*. Sehingga semua komponen *u(t)* konstan melalui interval di antara dua sampling singkat yang berurutan.

$$u(t) = u(kT), \qquad \text{untuk } kT \le t < kT+T \tag{2.66}$$

$$\underline{x}((k+1)T) = e^{\underline{A}(k+1)T} x(0) + e^{\underline{A}(k+1)T} \int_0^{(k+1)T} e^{-\underline{A}\tau} \underline{B}\underline{u}(\tau) d\tau \tag{2.67}$$

dan $\underline{x}(kT) = e^{\underline{A}kT} x(0) + e^{\underline{A}kT} \int_0^{kT} e^{-\underline{A}\tau} \underline{B}\underline{u}(\tau) d\tau$ (2.68)

kalikan persamaan 2.68 dengan $e^{\underline{A}T}$ dan kurangi dari persamaan 2.67, diperoleh :

$$\underline{x}((k+1)T) = e^{\underline{A}T} \underline{x}(kT) + e^{\underline{A}(k+1)T} \int_0^{(k+1)T} e^{-\underline{A}\tau} \underline{B}\underline{u}(\tau) d\tau \tag{2.69}$$

Karena $u(t) = u(kT)$ untuk $kT \le t < kT+T$, maka dapat disubtitusikan $u(\tau) = u(kT) =$ konstan ke dalam persamaan 2.69. *u(t)* mungkin melompati $t = kT+T$ dan juga *u(kT+T)* mungkin berbeda dari *u(kT)*. Saat melangkah di *u(τ)* pada $\tau = kT + T$, batas atas integrasi tidak mempengaruhi nilai integral pada persamaan 2.69, karena *integrand* tidak melibatkan fungsi *impulse*. Maka dapat ditulis sebagai berikut :

$$. \; \underline{x}((k+1)T) = e^{\underline{A}T} \underline{x}(kT) + e^{\underline{A}T} \int_0^{T} e^{-\underline{A}t} \underline{B}\underline{u}(kT) dt$$

$$= e^{\underline{A}T} \underline{x}(kT) + \int_0^{T} e^{\underline{A}\lambda} \underline{B}\underline{u}(kT) d\lambda \tag{2.70}$$

di mana $\lambda = T - t$, jika mendefinisikan :

$$\underline{G}(T) = e^{\underline{A}T} \tag{2.71}$$

$$\underline{H}(T) = \left( \int_0^{T} e^{\underline{A}\lambda} d\lambda \right) \underline{B} \tag{2.72}$$

Maka persamaan 2.70 menjadi seperti yang tertulis pada persamaan 2.65, $\underline{x}((k+1)T) = \underline{G}(T)\underline{x}(kT) + \underline{H}(T)\underline{u}(kT)$. Persamaan outputnya menjadi :

$y(kT) = \underline{C}\underline{x}(kT) + \underline{D}\underline{u}(kT)$

Matriks $\underline{G}(T)$ dan $\underline{H}(T)$ tergantung pada periode sampling T sedangkan matriks $\underline{A}$ dan $\underline{B}$ konstan dan tidak bergantung pada periode sampling T. Jika matriks A nonsingular persamaan 2.72 dapat disederhanakan menjadi :

$$\underline{H}(T) = \left( \int_0^T e^{A\lambda} d\lambda \right) \underline{B} = \underline{A}^{-1}(e^{AT} - I)\underline{B} = (e^{AT} - I)\underline{A}^{-1}\underline{B} \tag{2.73}$$

Hal yang perlu ditekankan :

1. Di dalam pendekatan *state space*, mengasumsikan vector input konstan di antara dua sampling singkat manapun yang berurutan, secara sederhana model waktu diskrit dapat diperoleh dengan mengintegralkan persamaan keadaan waktu kontinu melalui satu periode sampling. Persamaan keadaan waktu diskrit yang diberikan pada persamaan 2.65 disebut *zero-order hold equivalent* dari persamaan keadaan waktu kontinu yang diberikan pada persamaan 2.64.
2. Secara umum, dalam mengbah persamaan sistem waktu kontinu ke dalam persamaan sistem waktu diskrit, beberapa jenis perkiraan dibutuhkan.
3. Untuk $T \ll 1$, $\underline{G}(T) \div \underline{G}(0) = e^{A0} = I$, periode sampling T menjadi sangat kecil, G(T) mendekati matriks identitas.