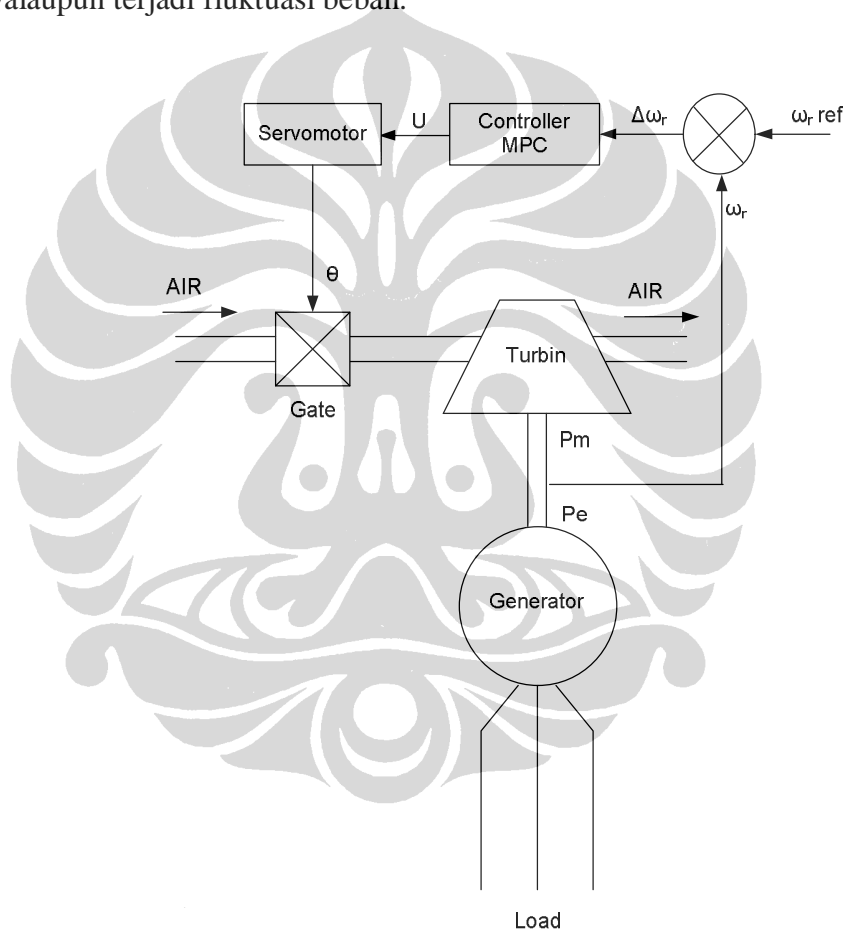


BAB 3 PEMODELAN DAN DISAIN PENGENDALI SISTEM PLTMH

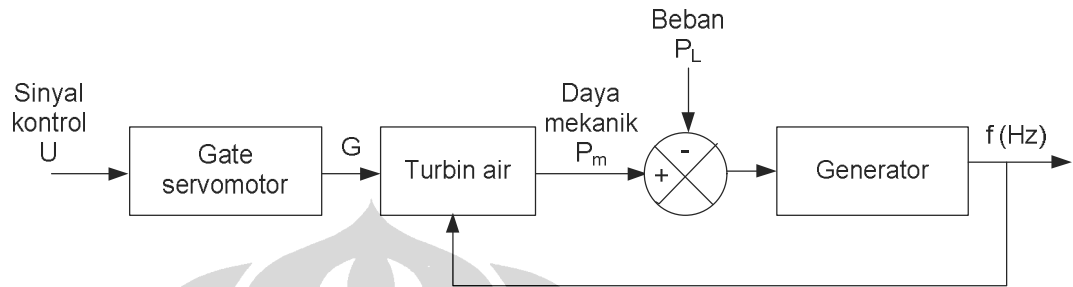
Konsep pengendalian frekuensi (kecepatan) dapat dilihat pada Gambar 3.1. Jika kecepatan (frekuensi) tidak sesuai dengan *set point* maka sinyal *error* akan dikirimkan ke pengendali lalu pengendali akan memberikan sinyal kepada servomotor sebagai penggerak katup (*gate*) untuk membuka atau menutup aliran air. Dengan pengaturan ini maka kecepatan (frekuensi) akan tetap terjaga konstan walaupun terjadi fluktuasi beban.



Gambar 3.1 Diagram blok sistem pengendali frekuensi

3.1 Pemodelan fisik sistem pembangkit listrik tenaga mikrohidro

Pemodelan sistem PLTMH menggunakan metode pemodelan fisik yaitu penurunan model menggunakan persamaan diferensial non linier. Diagram blok pemodelan sistem PLTMH dapat dilihat pada Gambar 3.2



Gambar 3.2 Diagram blok sistem PLTMH

3.1.1 Pemodelan sistem hidrolis [7]

Pemodelan sistem hidrolis terdiri dari pipa pesat dan turbin air. Pemodelan ini memakai asumsi-asumsi sebagai berikut :

1. Tahanan pada hidrolis di abaikan
2. Pipa pesat kaku (tidak elastik) dan air tidak ditekan (*incompressible*)
3. Variasi kecepatan aliran air berhubungan langsung dengan bukaan *gate* dan dengan akar dari *net head*.
4. Daya output turbin sebanding dengan perkalian *head* dan kapasitas air

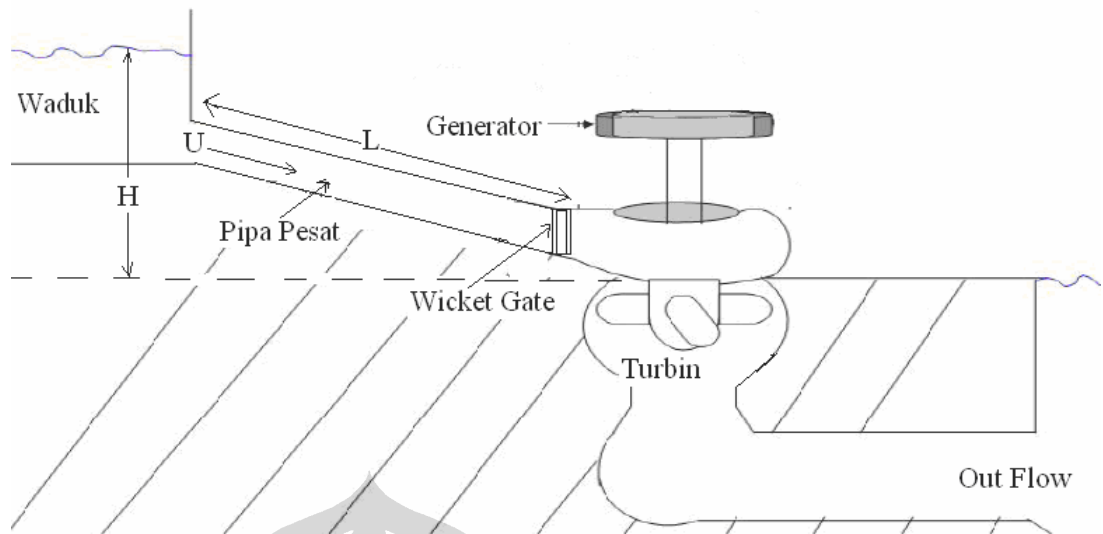
Komponen-komponen sistem hidrolis dapat dilihat pada Gambar 3.3

Karakteristik turbin dan pipa pesat ditentukan oleh tiga dasar persamaan sebagai berikut:

- Kecepatan air dalam pipa pesat
- Daya mekanik turbin
- Percepatan air dalam pipa

Persamaan kecepatan air dalam pipa pesat:

$$U = G\sqrt{H} \quad (3.1)$$



Gambar 3.3 Diagram sistem hidrolik

Persamaan daya turbin:

$$P = HU \quad (3.2)$$

Percepatan air dalam pipa karena perubahan *head* pada turbin, berdasarkan hukum Newton II tentang gerakan, dinyatakan dalam persamaan:

$$(rLA) \frac{d\Delta U}{dt} = -A(r g) \Delta H \quad (3.3)$$

Persamaan diatas diubah dalam bentuk normalisasi dengan membagi kedua sisi dengan $A r a_g H_r U_r$, percepatan dalam bentuk persamaan normalisasi menjadi:

$$\frac{LU_r}{a_g H_r} \frac{d \left(\frac{\Delta U}{U_r} \right)}{dt} = - \frac{\Delta H}{H_r} \quad (3.4)$$

$$T_w = \frac{LU_r}{gH_r} = \frac{LQ_r}{gAH_r}$$

Maka persamaan (3.4) dapat ditulis menjadi :

$$\frac{d\bar{\Delta U}}{dt} = - \frac{1}{T_w} \Delta \bar{H} \quad (3.5)$$

Dalam bentuk Laplace :

$$\bar{U}_{(s)} = - \frac{1}{T_w s} (\bar{H} - \bar{H}_0)_{(s)} \quad (3.6)$$

Dimana *subscript r* menyatakan nilai rata-rata. Penulisan dalam per unit variabelnya menggunakan superbar.

Hubungan antara kapasitas air yang mengalir (debit air) Q dengan kecepatan air dalam pipa :

$$Q = A U \quad (3.7)$$

Daya mekanik yang dihasilkan turbin P_m adalah

$$P_m = P - P_l - P_{damping} \quad (3.8)$$

P_l merepresentasikan rugi-rugi daya pada turbin

$$P_l = U_{NL} H \quad (3.9)$$

Dengan U_{NL} merepresentasikan kecepatan aliran air tanpa beban.

$$P_{damping} = DG\Delta w \quad (3.10)$$

$P_{damping}$ merepresentasikan efek damping karena friksi dan sebanding dengan kecepatan sudut rotor.

Substitusi persamaan (3.9) dan (3.10) ke persamaan (3.8), maka daya mekanik menjadi:

$$P_m = UH - U_{NL} H - DG\Delta w \quad (3.11)$$

Dalam bentuk normalisasi

$$\frac{P_m}{P_r} = \left(\frac{U}{U_r} - \frac{U_{NL}}{U_r} \right) \frac{H}{H_r} - D \frac{G}{G_r} \frac{\Delta w}{\Delta w_r} \quad (3.12)$$

$$\bar{P}_m = (\bar{U} - \bar{U}_{NL}) \bar{H} - D \bar{G} \bar{\Delta w}_r \quad (3.13)$$

Dalam bentuk Laplace

$$\bar{P}_{m(s)} = (\bar{U}_{(s)} - \bar{U}_{NL(s)}) \bar{H}_{(s)} - D \bar{G}_{(s)} \bar{\Delta w}_{r(s)} \quad (3.14)$$

Substitusi persamaan (3.6), (3.1) ke persamaan (3.14)

$$\bar{P}_{m(s)} = \left(-\frac{1}{T_w s} (\bar{H} - \bar{H}_0)_{(s)} - \bar{U}_{NL(s)} \right) \frac{\bar{U}_{(s)}^2}{\bar{G}_{(s)}^2} - D \bar{G}_{(s)} \bar{\Delta w}_{r(s)} \quad (3.15)$$

Tabel 3.1. Keterangan Parameter Sistem Hidrolik

Parameter	Simbol	Satuan
Kecepatan air	U	(m/s)
Posisi <i>gate</i>	G	(Pu)
<i>Head</i> hidrolik sampai <i>gate</i>	H	(m)
Inisial nilai <i>steady state</i> dari H	H_o	(m)
Daya turbin	P_m	Watt
Kapasitas aliran	Q	(m^3 / s)
Luas penampang pipa	A	(m^2)
Kerapatan air	r	(kg / m^3)
Percepatan karena Gravitasi	g	(m / s^2)
Panjang pipa	L	(m)
Massa air dalam pipa	rLA	
Perubahan kenaikan tekanan pada <i>gate</i> turbin	$r g\Delta H$	
Waktu	T	(s)

3.1.2 Pemodelan *Gate*

Gate berfungsi sebagai pintu gerbang untuk mengalirkan air dari *penstock* menuju turbin. Sebagai penggerak buka dan tutup *gate* menggunakan dc servo motor. *Input* dc servo motor adalah tegangan dan *output*-nya adalah posisi sudut yang dihubungkan secara langsung dengan peralatan mekanik (*gate*) melalui roda gigi. Oleh karena itu, karakteristik *gate* dimodelkan sebagai servomotor.[3]

Sistem motor dc servo direpresentasikan dalam persamaan :

$$V_{(s)} = E_{b(s)} + I_{(s)}(R_a + sL_a) \quad (3.16)$$

$$E_{b(s)} = K_b s q_{m(s)} \quad (3.17)$$

Torsi motor dinyatakan dalam persamaan :

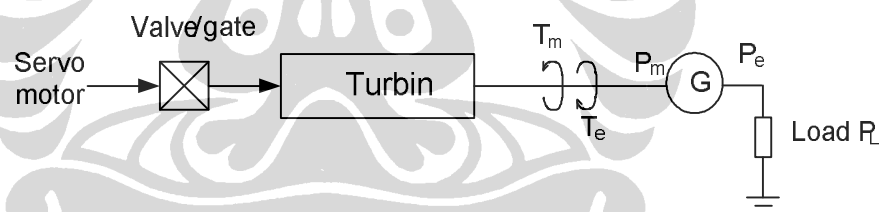
$$J s^2 q_{m(s)} + B s q_{m(s)} = T_{(s)} = I_{a(s)} K_t \quad (3.18)$$

Tabel 3.2 Keterangan parameter *gate* servomotor

Parameter	Simbol	Satuan
Konstanta <i>back emf</i> motor	K_b	(V sec/rad)
Konstanta torsi motor	K_t	(N m/A)
Resistansi lilitan armature	R_a	(Ω)
Induktansi lilitan <i>armature</i> (diabaikan)	L_a	(mH)
Momen inersia motor	J	(kg m ²)
Koefisien gesek motor	B	(N m/rad/sec)

3.1.3 Pemodelan sistem listrik

Pemodelan sistem listrik terdiri dari generator, beban dan pengaturan kecepatan. Konsep dasar dari pengaturan kecepatan dengan mempertimbangkan unit pembangkit terpisah yang menyuplai beban lokal dilustrasikan pada Gambar 3.4.



Gambar 3.4 Generator menyuplai beban lokal

Turbin (menghasilkan torsi mekanik T_m) dihubungkan secara langsung menggunakan *shaft* dengan generator (menghasilkan torsi elektrik T_e) sehingga kecepatan gerak antara keduanya harus sinkron. Hubungan gerakan antara turbin dan generator dimodelkan dalam persamaan gerakan mekanik (*swing equation*).[3] Persamaan ini akan merepresentasikan karakteristik mekanik dari mesin sinkron.

Persamaan gerakan (*swing equation*)

Generator sinkron membangkitkan torsi elektromagnetik T_e dan Turbin menghasilkan torsi mekanik T_m , pada saat kondisi *steady state* dengan rugi-rugi diabaikan keduanya akan bekerja pada kecepatan sinkron w_m , sehingga didapat persamaan:

$$T_m = T_e \quad (3.20)$$

Jika dalam kondisi *Steady state* terjadi perubahan beban maka akan mengakibatkan percepatan ($T_m > T_e$) atau perlambatan ($T_m < T_e$) torsi pada rotor T_a .

$$T_a = T_m - T_e \quad (3.21)$$

$$J \frac{dw_m}{dt} = T_a = T_m - T_e \quad (3.21)$$

Dengan:

T_a = torsi percepatan (N.m)

T_m = Torsi mekanik (N.m)

T_e = Torsi elektrik (N.m)

J = kombinasi inersia generator dan turbin (kg.m^2)

w_m = kecepatan sinkron, (mekanik.rad/s)

Persamaan (3.21) dinormalisasi dalam bentuk inersia konstan (H) per unit, yang didefinisikan sebagai energi kinetik dalam *watt-second* pada kecepatan rata-rata dibagi dengan VA *base*. w_{Om} digunakan untuk menunjukkan kecepatan sinkron rata-rata dalam mekanik radian per second, konstanta inersia adalah sebagai berikut:

$$H = \frac{1}{2} \frac{Jw_{Om}^2}{VA_{base}} \quad (3.22)$$

$$J = \frac{2H}{w_{Om}^2} VA_{base}$$

Substitusi persamaan (3.22) ke persamaan (3.21) menjadi:

$$\frac{2H}{w_{Om}^2} VA_{base} \frac{dw_m}{dt} = T_m - T_e$$

$$2H \frac{d\left(\frac{w_m}{w_{Om}}\right)}{dt} = \frac{T_m - T_e}{VA_{base} / w_{Om}}$$

Catatan bahwa $T_{base} = VA_{base} / w_{Om}$, sehingga persamaan gerakan dalam bentuk per unit adalah :

$$2H \frac{d\bar{w}_r}{dt} = \bar{T}_m - \bar{T}_e \quad (3.23)$$

Atau

$$\frac{d\Delta\bar{w}_r}{dt} = \frac{1}{2H} (\bar{T}_m - \bar{T}_e) \quad (3.24)$$

Untuk selanjutnya persamaan diatas tidak menggunakan superbar (-) untuk indentifikasi per unit, kita asumsikan variable Δw_r , T_m , dan T_e sudah dalam per unit (pu).

Hubungan antara daya P dan torsi T adalah sebagai berikut:

$$P = w_r T \quad (3.25)$$

Dengan mempertimbangkan perubahan kecil (dinyatakan dalam Δ) dari nilai awal (dinyatakan dengan subcrip 0), dapat ditulis:

$$P = P_o + \Delta P$$

$$T = T_o + \Delta T$$

$$w_r = w_o + \Delta w_r$$

Dari persamaan (3.29)

$$P_o + \Delta P = (w_o + \Delta w_r)(T_o + \Delta T)$$

Hubungan antara nilai dengan orde tinggi diabaikan, sehingga :

$$\Delta P = w_o \Delta T + T_o \Delta w_r \quad (3.26)$$

Oleh karena itu,

$$\Delta P_m - \Delta P_e = w_o (\Delta T_m - \Delta T_e) + (T_{mo} - T_{eo}) \Delta w_r \quad (3.27)$$

Karena dalam kondisi *steady state*, torsi elektrik dan mekanikal sama, $T_{mo} = T_{eo}$.

Dengan kecepatan dalam satuan pu. $w_o = 1$, maka

$$\Delta P_m - \Delta P_e = \Delta T_m - \Delta T_e \quad (3.28)$$

Maka persamaan (3.24) dapat ditulis menjadi:

$$\frac{d\Delta w_r}{dt} = \frac{1}{2H} \left(\Delta P_m - \Delta P_e \right) \quad (3.29)$$

3.1.4 Pemodelan beban [7]

Secara umum, beban pada sistem daya listrik terdiri dari bermacam-macam peralatan listrik. Ada beban yang tidak sensitif terhadap perubahan frekuensi, misalnya beban resistif, seperti lampu dan beban pemanas. Ada juga beban yang sangat sensitif terhadap perubahan frekuensi misalnya, beban motor, seperti kipas angin dan pompa.

Daya listrik generator terdiri dari gabungan dua macam beban yaitu:

$$\Delta P_e = \Delta P_L + D\Delta w_r \quad (3.30)$$

Dengan:

ΔP_L = perubahan beban *non-frequency-sensitive* (Watt)

$D\Delta w_r$ = perubahan beban *frequency sensitive*

D = konstanta damping beban (*load damping*) (%)

Damping menyatakan sebagai persen perubahan pada beban dibagi persen perubahan pada frekuensi.

Dengan mensubstitusi persamaan (3.30) ke persamaan (3.29), hubungan generator dan beban menjadi:

$$\frac{d\Delta w_r}{dt} = \frac{1}{2H} \left(\Delta P_m - \Delta P_L - D\Delta w_r \right) \quad (3.31)$$

3.2 Pemodelan Ruang Keadaan

Untuk memudahkan dalam merancang pengendali diperlukan model linier yang merupakan perkiraan model yang mengacu pada sistem *non* linier, dimana hanya *valid* pada daerah sekitar titik operasi atau titik kesetimbangan dari sistem.

Titik operasi pada sistem PLTMH berada di titik $u_0 = 19,2$ V dan $y_0 = 44$ Hz. Linierisasi dilakukan dengan menggunakan fasilitas linier analysis pada simulink matlab. Hasil linierisasi dalam bentuk persamaan ruang keadaan sebagai berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -0.1573 & 0.4606 & 0 & -0.01784 \\ 0 & -2.166 & 0 & 0.1042 \\ 0 & 0 & -9.167 & -5.0000 \\ 0 & 0 & 0.8333 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & -0.1667 \\ 0 & 0 \\ 120 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.32)$$

$$y = [50 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (3.33)$$

Dengan,

Ø Nama *State* :

x_1 = Kecepatan putar generator atau frekuensi

x_2 = Daya mekanik turbin

x_3 = Kecepatan posisi sudut *gate* servomotor

x_4 = Posisi sudut *gate* servomotor

Ø Nama *input* :

u_1 = Tegangan *gate*

u_2 = Perubahan beban

Ø Nama *Output* :

y = kecepatan putar

3.3 Penentuan Waktu Pencuplikan

Model di atas adalah model *continuous-time*. Karena model linier yang digunakan pada algoritma MPC merupakan persamaan *diference* maka persamaan ruang keadaan pada persamaan (3.32) dan (3.33) harus diubah kedalam bentuk diskrit. Sebelum dilakukan perubahan bentuk kontinyu ke diskrit, terlebih dahulu ditentukan waktu pencuplikan h . Pemilihan waktu pencuplikan yang terlalu kecil

dapat menghasilkan nilai masukan proses yang terlalu besar, sedangkan waktu pencuplikan yang terlalu besar dapat mengakibatkan gagalnya rekonstruksi sinyal diskrit terhadap sinyal kontinu.[8] Waktu pencuplikan ditentukan berdasarkan waktu *settling time* yang diperoleh dari uji lup terbuka dengan masukan fungsi *step*. Waktu pencuplikan ditentukan dengan persamaan :

$$\frac{1}{20}T_{95} \leq h \leq \frac{1}{5}T_{95} \quad (3.34)$$

Dengan T_{95} adalah *settling time* dengan kriteria 5 %.

Dari uji lup terbuka dengan nilai masukan pada titik operasinya yaitu $u_0 = 19.2V$ didapatkan interval waktu pencuplikan sebagai berikut: $0.75 \leq h \leq 3$. Sehingga waktu pencuplikan ditentukan sebesar $h = 3$ detik.

Dengan menggunakan waktu pencuplikan $h = 3$ detik dan mengikutsertakan *zero-order-hold* sebagai bagian dari sistem kontinu, maka persamaan ruang keadaan diskrit sistem PLTMH adalah:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 0.6238 & 0.1427 & 0.0003 & 0.0041 \\ 0 & 0.0015 & 0.0014 & 0.0154 \\ 0 & 0 & -0.0139 & -0.1445 \\ 0 & 0 & 0.0240 & 0.2511 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} -0.0159 & -0.3986 \\ 0.7829 & 0 \\ 3.468 & 0 \\ 17.97 & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \quad (3.35)$$

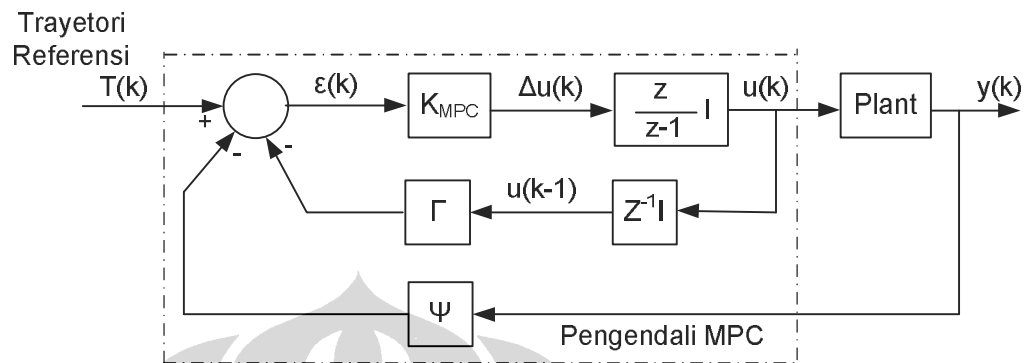
$$y(k) = [50 \ 0 \ 0 \ 0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} \quad (3.36)$$

Masukan sistem pada model terdiri dari tegangan *gate* servomotor dan perubahan beban yang merupakan fungsi gangguan.

3.4 Algoritma Model Predictive Control Tanpa constraint

Struktur pengendali MPC tanpa *constraint* untuk model ruang keadaan terdapat pada Gambar 3.5. Dari blok diagram tersebut, terlihat bahwa prediksi

perubahan sinyal masukan sekarang $\Delta u(k)$ membutuhkan data dari variabel keadaan sekarang $x(k)$ dan masukan satu langkah sebelumnya $u(k-1)$.

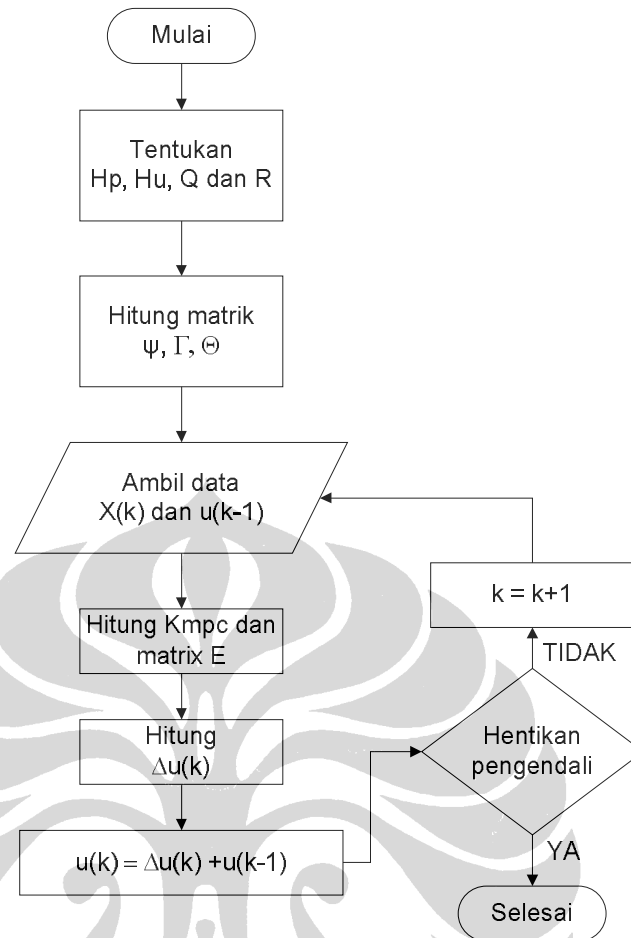


Gambar 3.5 Blok diagram pengendali MPC tanpa *constraint*

Algoritma perhitungan perubahan sinyal kendali pada MPC tanpa *constraint* adalah sebagai berikut:

1. Parameter pengendali yang terlebih dahulu harus ditentukan antara lain horizon prediksi (H_p), horizon kendali (H_u), matriks faktor bobot kesalahan (\underline{Q}), dan matriks faktor bobot perubahan sinyal kendali (\underline{R}).
2. Matriks $\underline{\Psi}$, $\underline{\Gamma}$, dan $\underline{\Theta}$ dihitung dengan menggunakan persamaan (2.16)
3. Ambil data $x(k)$ dan $u(k-1)$
4. Hitung matriks \underline{E} dengan menggunakan persamaan (2.25)
5. Hitung perubahan sinyal kendali $\Delta u(k)$ dengan menggunakan persamaan (2.35)
6. Hitung sinyal kendali $u(k) = \Delta u(k) + u(k-1)$

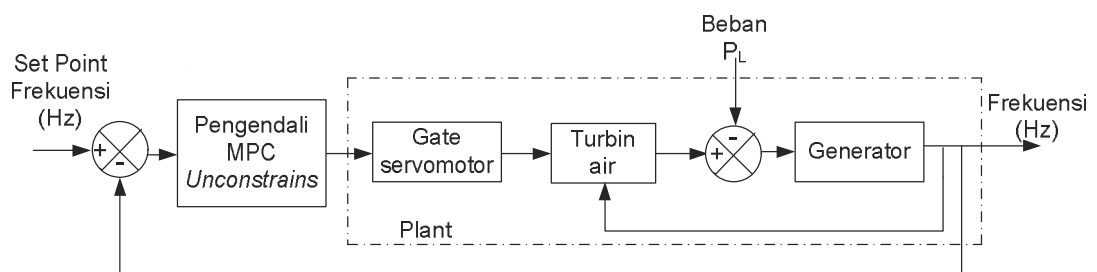
Diagram alir untuk perhitungan sinyal kendali dengan menggunakan MPC tanpa *constraint* adalah seperti pada Gambar 3.6.



Gambar 3.6 Diagram alir algoritma MPC tanpa *constraint*

3.5 Algoritma Pengendali MPC tanpa *constraint* dengan gangguan

Diagram blok pengendali MPC tanpa *constraint* pada model yang mengandung gangguan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Diagram blok pengendali MPC *unconstraint*, *plant* + gangguan

Struktur model *plant* yang dikendalikan menjadi:

$$\underline{x}(k+1) = \underline{A}\underline{x}(k) + \underline{B}\underline{u}(k) + \underline{B}_d \underline{v}(k) \quad (3.37)$$

$$\underline{y}(k) = \underline{C}\underline{x}(k) \quad (3.38)$$

Model linier pada persamaan (3.35) berubah menjadi:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 0.6238 & 0.1427 & 0.0004 & 0.0041 \\ 0 & 0.0015 & 0.0015 & 0.0154 \\ 0 & 0 & -0.0139 & -0.1445 \\ 0 & 0 & 0.0241 & 0.2511 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} -0.0159 \\ 0.7829 \\ 3.4681 \\ 17.9743 \end{bmatrix} [u(k)] + \begin{bmatrix} -0.3987 \\ 0 \\ 0 \\ 0 \end{bmatrix} [v(k)]$$

$$\underline{y}(k) = \begin{bmatrix} 50 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} \quad (3.39)$$

Langkah-langkah perhitungan sinyal kendali sama seperti langkah sub bab (3.4), hanya terjadi perbedaan dalam perhitungan *error* yang terjadi. Persamaan matematis matrik *error* \underline{E} , adalah sebagai berikut:

$$\underline{E}(k) = \underline{T}(k) - \underline{C}_Y \underline{\Psi} \underline{x}(k) - \underline{C}_Y \underline{\Gamma} \underline{u}(k-1) - \underline{C}_Y \underline{\Xi} \underline{v}(k-1) \quad (3.40)$$

$$\text{Dengan } \underline{\Xi} = \underline{C}_Y \begin{bmatrix} B_d & 0 & \mathbf{L} & 0 \\ AB_d & B_d & \mathbf{L} & 0 \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ A^{H_p-1} B_d & A^{H_p-2} B_d & \mathbf{L} & B_d \end{bmatrix}$$

3.6 Perhitungan Penguat Pengendali MPC tanpa *constraint*

Berikut ini adalah contoh langkah-langkah yang dilakukan untuk perhitungan pengendali dengan metode MPC tanpa *constraint*. Parameter pengendali yang digunakan adalah sebagai berikut :

- *Prediction horizon*, $H_p = 6$
- *Control Horizon*, $H_u = 2$, $H_w = 1$
- Faktor Pembobot $\underline{Q} = \underline{I}_{H_p}$ dan $\underline{R} = \underline{I}_{H_u}$

- vektor keadaan (*state*), $n = 4$
- *Input* sistem, $l = 1$
- *Output* sistem, $m = 1$

Matrix variabel keadaan:

$$A = \begin{bmatrix} 0.6238 & 0.1427 & 0.0004 & 0.0041 \\ 0 & 0.0015 & 0.0015 & 0.0154 \\ 0 & 0 & -0.0139 & -0.1445 \\ 0 & 0 & 0.0241 & 0.1511 \end{bmatrix} \quad B = \begin{bmatrix} -0.0159 \\ 0.7829 \\ 3.4681 \\ 17.9743 \end{bmatrix}$$

$$B_d = \begin{bmatrix} -0.3987 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad C = [50 \ 0 \ 0 \ 0]$$

Dimensi matrik dari tiap parameter dapat dilihat pada Tabel 3.3

Tabel 3.3 Dimensi matriks parameter pengendali MPC

Matriks	Dimensi	
Q	$m(H_p - H_w + 1) \times m(H_p - H_w + 1)$	$= 6 \times 6$
R	$lH_u \times lH_u$	$= 2 \times 2$
Ψ	$m(H_p - H_w + 1) \times n$	$= 6 \times 4$
Γ	$m(H_p - H_w + 1) \times l$	$= 6 \times 1$
Θ	$m(H_p - H_w + 1) \times lH_u$	$= 6 \times 2$
H	$lH_u \times lH_u$	$= 2 \times 2$
G	$lH_u \times 1$	$= 2 \times 1$
E	$m(H_p - H_w + 1) \times 1$	$= 6 \times 1$

Untuk mendapatkan sinyal kendali, dilakukan tahapan perhitungan sebagai berikut:

1. Menghitung matriks Ψ , Γ , dan Θ dihitung dengan menggunakan persamaan (2.16)

$$\underline{C}_y \underline{\Psi} = \begin{bmatrix} \underline{C} & \underline{0}_{4 \times 1} & \underline{L} & \underline{0}_{4 \times 1} \\ \underline{0}_{4 \times 1} & \underline{C} & \underline{L} & \underline{0}_{4 \times 1} \\ \underline{M} & \underline{M} & \underline{O} & \underline{M} \\ \underline{0}_{4 \times 1} & \underline{0}_{4 \times 1} & \underline{L} & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{A} \\ \underline{M} \\ \underline{A}^6 \end{bmatrix} = \begin{bmatrix} 31.1908 & 7.1349 & 0.0177 & 0.2063 \\ 19.4573 & 4.4616 & 0.0263 & 0.2879 \\ 12.1378 & 2.7832 & 0.0200 & 0.2175 \\ 7.5717 & 1.7362 & 0.0134 & 0.1447 \\ 4.7234 & 1.0831 & 0.0085 & 0.0924 \\ 2.9465 & 0.6756 & 0.0054 & 0.0581 \end{bmatrix}$$

$$\underline{C}_y \underline{\Gamma} = \begin{bmatrix} \underline{C} & \underline{0}_{4 \times 1} & \underline{L} & \underline{0}_{4 \times 1} \\ \underline{0}_{4 \times 1} & \underline{C} & \underline{L} & \underline{0}_{4 \times 1} \\ \underline{M} & \underline{M} & \underline{O} & \underline{M} \\ \underline{0}_{4 \times 1} & \underline{0}_{4 \times 1} & \underline{L} & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{B} \\ \underline{M} \\ \sum_{i=0}^1 \underline{A}^i \underline{B} \\ \underline{M} \\ \sum_{i=0}^5 \underline{A}^i \underline{B} \end{bmatrix} = \begin{bmatrix} -0.7956 \\ 8.0627 \\ 16.5115 \\ 22.4764 \\ 26.3621 \\ 28.8252 \end{bmatrix}$$

$$\underline{C}_y \underline{\Theta} = \begin{bmatrix} \underline{C} & \underline{0}_{4 \times 1} & \underline{L} & \underline{0}_{4 \times 1} \\ \underline{0}_{4 \times 1} & \underline{C} & \underline{L} & \underline{0}_{4 \times 1} \\ \underline{M} & \underline{M} & \underline{O} & \underline{M} \\ \underline{0}_{4 \times 1} & \underline{0}_{4 \times 1} & \underline{L} & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{B} & \underline{L} & \underline{0}_{4 \times 1} \\ \underline{AB} + \underline{B} & \underline{O} & \underline{M} \\ \underline{M} & \underline{L} & \underline{M} \\ \sum_{i=0}^5 \underline{A}^i \underline{B} & \underline{L} & \sum_{i=0}^4 \underline{A}^i \underline{B} \end{bmatrix} = \begin{bmatrix} -0.7956 & 0 \\ 8.0627 & -0.7956 \\ 16.5115 & 8.0627 \\ 22.4764 & 16.5115 \\ 26.3621 & 22.4764 \\ 28.8252 & 26.3621 \end{bmatrix}$$

2. Perhitungan untuk konstanta K_{MPC} menggunakan rumus :

$$K_{MPC} = (\underline{\Theta}^T \underline{Q} \underline{\Theta} + \underline{R})^{-1} \underline{\Theta}^T \underline{Q}$$

$$= \begin{bmatrix} -0.0054 & 0.0616 & 0.0466 & 0.0180 & -0.0045 & -0.0195 \\ 0.0065 & -0.0745 & -0.0507 & -0.0109 & 0.0200 & 0.0406 \end{bmatrix}$$

3. Matriks \underline{E} dihitung berdasarkan persamaan (2.25). Nilai \underline{E} akan diperbaharui terus menerus seiring dengan perubahan keluaran sistem. Didapatkan matriks \underline{E} sebagai berikut :

$$\underline{E}(1) = \begin{bmatrix} 11.9604 \\ 23.2037 \\ 32.5557 \\ 38.9452 \\ 43.0629 \\ 45.6628 \end{bmatrix}$$

4. Nilai optimal $\underline{\Delta u}(k)$ dapat dihitung dengan menggunakan persamaan (2.33)

$$\underline{\Delta u}(1) = \begin{bmatrix} 2.4960 \\ -1.0131 \end{bmatrix}$$

Setelah nilai matriks $\underline{\Delta u}(k)$ didapatkan, maka 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.

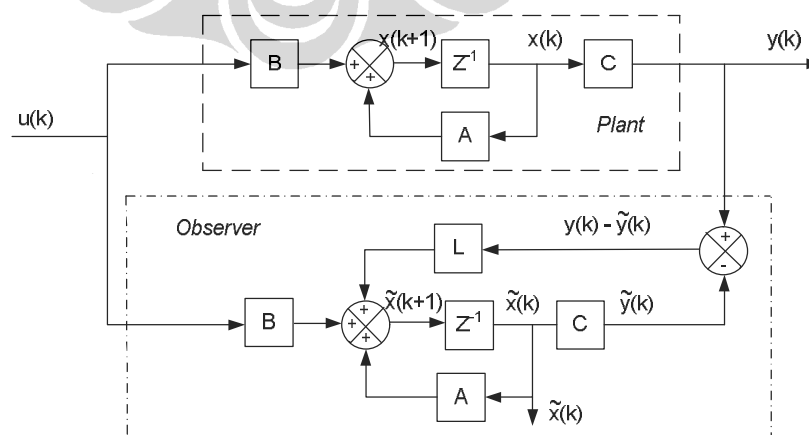
5. Untuk memperbaharui sinyal kendali, nilai $\underline{\Delta u}(k)$ inilah yang akan dijumlahkan dengan nilai sinyal kendali sebelumnya ($u(k-1)$).

$$\begin{aligned} \underline{u}(k) &= \underline{u}(k-1) + \underline{\Delta u}(k) \\ &= 0 + 2.4960 = 2.4960 \end{aligned}$$

Untuk perhitungan sinyal kendali MPC tanpa *constraint* dengan gangguan, langkah perhitungannya sama dengan langkah diatas tetapi perhitungan matriks \underline{E} menggunakan persamaan (3.40).

3.7 Model Observer

Gambar 3.8 adalah diagram blok observer Luenberger atau *full-order observer* yang sederhana. Observer ini merupakan observer identitas, karena $\tilde{x}(k) = I x_e(k)$ atau dituliskan $\tilde{x}(k) = x_e(k)$.



Gambar 3.8 Observer Luenberger dan Sistem

Universitas Indonesia

3.7.1 Uji *Observability*

Langkah pertama yang harus dilakukan untuk merancang sebuah observer yaitu uji *observability* dari sebuah sistem. Uji *observability* sistem dimaksudkan untuk mengetahui apakah sistem tersebut benar-benar dapat diobservasi dan untuk mengetahui apakah *state-state* yang diobservasi tersebut dapat mewakili keadaan sistem yang sebenarnya.

Untuk melakukan pengujian *observability* dari suatu sistem, langkah yang harus dilakukan adalah membentuk matriks *observability* seperti yang ditunjukkan oleh persamaan berikut :

$$\begin{bmatrix} C^T \\ C^T M A^T C^T \\ \dots \\ C^T M^{n-1} A^{T(n-1)} C^T \end{bmatrix} \quad (3.41)$$

Dengan:

- Ø n adalah jumlah *state* yang dimiliki oleh sebuah sistem
- Ø Sistem *observable* jika matriks *observability* memiliki *rank* sebanyak n (jumlah *state*)

Dengan menggunakan matriks C dan A pada persamaan (3.39) ke dalam persamaan (3.41), didapatkan matriks *observability* sebagai berikut:

$$N = obsv = \begin{bmatrix} 50.0000 & 0 & 0 & 0 \\ 31.1908 & 7.1349 & 0.0177 & 0.2063 \\ 19.4573 & 4.4616 & 0.0263 & 0.2879 \\ 12.1378 & 2.7832 & 0.0200 & 0.2175 \end{bmatrix} \quad (3.42)$$

Rank dari matriks *obsv* adalah 4, dengan *rank* matriks A dari model. Hal ini menunjukkan bahwa sistem bersifat *Observable* sempurna atau dengan kata lain semua *state* dari sistem dapat diobservasi.

3.7.2 Pemilihan Nilai *Eigen* model Estimasi

Untuk mengetahui nilai *Eigen* menggunakan persamaan :

$$\det(A - \lambda I) = 0 \quad (3.43)$$

Dengan menggunakan persamaan (3.43), didapatkan nilai *Eigen* dari model sistem adalah:

$$\begin{aligned}
 I_1 &= 0.6238 \\
 I_2 &= 0.0015 \\
 I_3 &= 0.0000 \\
 I_4 &= 0.2374
 \end{aligned}$$

Karena nilai *Eigen* model sistem semua berada pada skala $-1 \leq P \leq 1$ maka sistem adalah stabil.

Dalam perancangan observer ini, nilai *Eigen* digeser kekiri mendekati nol agar model prediksi menjadi lebih stabil. Sehingga nilai *Eigen* yang diinginkan menjadi :

$$\begin{aligned}
 m_1 &= I_1 - 0.1 = 0.5238 \\
 m_2 &= I_2 = 0.0015 \\
 m_3 &= I_3 = 0.0000 \\
 m_4 &= I_4 - 0.1 = 0.1372
 \end{aligned}$$

3.7.3 Pemilihan Gain Observer L

Gain L atau matriks penguat umpan balik observer dapat diperoleh dengan menggunakan persamaan Ackermann pada persamaan (2.43)

Dengan $\Phi(z)$ adalah karakteristik polinomial dengan nilai *Eigen* yang diinginkan untuk *state observer*.

$$\Phi(z) = (z - m_1)(z - m_2)(z - m_3)(z - m_4) = z^4 - 0.6625z^3 + 0.0729z^2 - 0.0001z$$

$$\Phi(A) = A^4 - 0.6625A^3 + 0.0729A^2 - 0.0001A$$

$$= \begin{bmatrix} 0.0189 & 0.0043 & 0 & 0.0004 \\ 0 & 0 & -0 & -0.0001 \\ 0 & 0 & 0.0001 & 0.0010 \\ 0 & 0 & -0.0002 & -0.0017 \end{bmatrix}$$

Diperoleh nilai parameter L,

$$L = \begin{bmatrix} 0.0040 \\ -0.0028 \\ 0.0259 \\ -0.0450 \end{bmatrix}$$

Maka persamaan untuk *full-order state observer* adalah:

$$\hat{\underline{x}}(k+1) = (\underline{A} - \underline{L} \underline{C})\hat{\underline{x}}(k) + \underline{B}u(k) + \underline{L}y(k) \quad (3.44)$$

Atau

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \\ \hat{x}_4(k+1) \end{bmatrix} = \begin{bmatrix} 0.4238 & 0.1427 & 0.0044 & 0.0041 \\ 0.1390 & 0.0015 & 0.0015 & 0.0154 \\ -1.2949 & 0 & -0.0139 & -0.1445 \\ 2.2499 & 0 & 0.0241 & 0.2511 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \\ \hat{x}_4(k) \end{bmatrix} + \begin{bmatrix} -0.0159 \\ 0.7829 \\ 3.4681 \\ 17.9743 \end{bmatrix} [u(k)] + \begin{bmatrix} 0.0040 \\ -0.0028 \\ 0.0259 \\ -0.0450 \end{bmatrix} [y(k)]$$

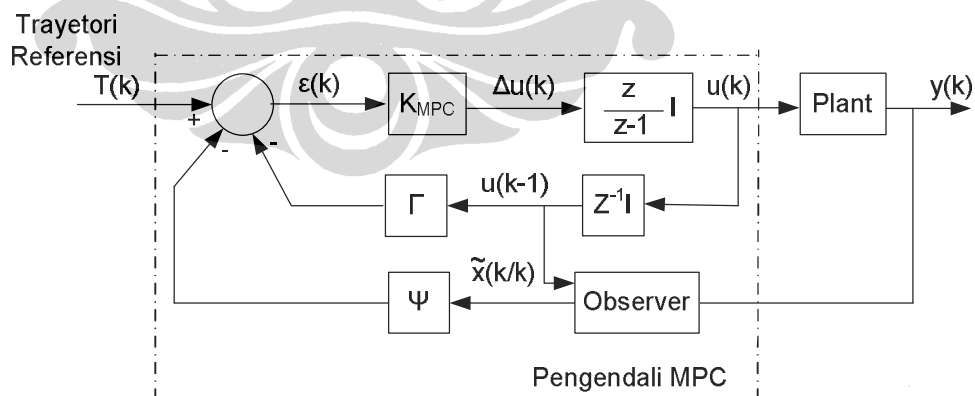
3.8 Penggunaan Observer pada Pengendali *Model Predictive unconstraint*

Observer digunakan untuk mengestimasi variabel *state* yang tidak terukur oleh sensor. Langkah-langkah perhitungan parameter pengendali sama dengan langkah pada sub bab (3.2), perbedaannya hanya pada perhitungan $\underline{Y}(k)$ dan $\underline{E}(k)$. Untuk pengendali MPC tanpa *constraint* dengan observer persamaan (2.4) dan (2.5) menjadi:

$$\underline{Y}(k) = \underline{C}_Y \underline{\Psi} \tilde{\underline{x}}(k/k) + \underline{C}_Y \underline{\Gamma} u(k-1) + \underline{C}_Y \underline{\Theta} \Delta \underline{U}(k) \quad (3.45)$$

$$\underline{E}(k) = \underline{T}(k) - \underline{C}_Y \underline{\Psi} \tilde{\underline{x}}(k/k) - \underline{C}_Y \underline{\Gamma} u(k-1) \quad (3.46)$$

Diagram blok pengendali MPC tanpa *constraint* dengan observer dapat dilihat pada Gambar 3.9



Gambar 3.9 Diagram blok pengendali MPC tanpa *constraint* dan observer