

**PERANCANGAN DAN IMPLEMENTASI PENGENDALI
MODEL PREDICTIVE CONTROL DENGAN CONSTRAINT UNTUK
PENGATURAN TEMPERATUR PADA
*LEVEL/FLOW AND TEMPERATURE PROCESS RIG 38-003***

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**Nama Hermanto Ang
NPM 0404030458**



**UNIVERSITAS INDONESIA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2008**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber yang baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar**

Nama : Hermanto Ang

NPM : 04 04 03 045 8

Tanda Tangan :

Tanggal : Juni 2008

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :Hermanto Ang
Nama :Hermanto Ang
NPM :0404030458
Program Studi :Teknik Elektro
Judul Skripsi :Perancangan dan Implementasi Pengendali *Model Predictive Control* dengan *Constraint* untuk Pengaturan Temperatur pada *Level/Flow and Temperature Process Rig 38-003*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana pada Program Teknik Elektro Fakultas Teknik Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Ir. Aries Subiantoro, M.Sc. ()
Penguji : Dr. Ir. Feri Yusivar, M.Eng. ()
Penguji : Dr. Ir. Ridwan Gunawan, M.T. ()

Depok, Juni 2008

UCAPAN TERIMAKASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmatNya, penulis dapat menyelesaikan skripsi ini. Penyusunan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, baik dari masa perkuliahan sampai pada penyusunan skripsi ini sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Aries Subianto, M.Sc., selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran didalam mengarahkan penulis dalam penyusunan skripsi ini.
2. Orangtua dan keluarga saya yang telah memberikan bantuan dukungan material maupun moril.
3. Sahabat yang telah banyak membantu penulis dalam menyelesaikan skripsi ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan saudara-saudara semua. Dan semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, Juni 2008

Penulis

Publikasi Karya Ilmiah untuk Kepentingan Akademis
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Hermanto Ang

NPM/NIP : 0404030458

Program Studi : Teknik Elektro

Fakultas : Teknik

Jenis karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Non- Eksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul : Perancangan dan Implementasi Pengendali *Model Predictive Control* dengan *Constraint* untuk Pengaturan Temperatur pada *Level/Flow and Temperature Proses Rig 38-003*

beserta perangkat yang ada (bila diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelolanya dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juni 2008

Yang menyatakan

(Hermanto Ang)

ABSTRAK

Nama : Hermanto Ang
Program studi : Teknik Elektro
Judul : Perancangan dan Implementasi Pengendali *Model Predictive Control* dengan *Constraint* untuk Pengaturan Temperatur pada *Level/Flow and Temperature Process Rig 38-003*

Pada sistem kendali konvensional, batasan-batasan seperti amplitudo dan *slew rate* sinyal kendali tidak diperhitungkan pada proses pengendalian. Hal ini tentu dapat menyebabkan hasil kendali menjadi kurang baik, terutama jika terjadi pemotongan paksa terhadap sinyal kendali sebelum masuk ke *plant*. Untuk mengatasi hal tersebut dirancanglah suatu pengendali *Model Predictive Control* (MPC). Dengan MPC, keluaran proses yang akan datang dapat diprediksi dan batasan-batasan yang ada tidak diabaikan sehingga keluaran sistem menjadi bagus. Selain keluaran sistem menjadi bagus, adanya batasan juga dapat membuat kinerja alat menjadi optimal.

Skripsi ini bertujuan untuk merancang jenis pengendali *Model Predictive Control* (MPC) yang akan diterapkan pada sebuah sistem nyata *Level/Flow and Temperature Process Rig 38-003* dengan metode *Quadratic Programming*. Dalam merancang pengendali MPC untuk *Level/Flow and Temperature Process Rig 38-003* ini, penulis menggunakan model yang berbentuk ruang keadaan yang didapat dengan menggunakan metode Kuadrat Terkecil berdasarkan pada data masukan dan data variabel keadaan alat. Masukan sistem adalah tegangan untuk mengatur kondisi *servo valve* dan keluran yang akan dikendalikan adalah temperatur air hasil keluaran *Heat Exchanger* sebelum masuk ke sistem *Radiator Cooler*.

Dari uji eksperimen terbukti bahwa metode pengendali MPC dengan *constraints* memberikan hasil yang lebih baik dibandingkan dengan metode pengendali Ruang Keadaan. Hal tersebut dapat dilihat dari tanggapan sistem hasil pengendalian MPC dengan *constraints* yang lebih halus dibandingkan dengan tanggapan sistem hasil pengendalian dengan metode pengendali Ruang Keadaan.

Perubahan sinyal kendali pengendali MPC dengan *constraints* juga jauh lebih halus dibandingkan dengan perubahan sinyal kendali pengendali Ruang Keadaan. Kondisi ini akan meningkatkan ketahanan fisik sistem selama uji eksperimen.

Kata Kunci :

batasan, pengendali, model, prediksi

ABSTRACT

Name : Hermanto Ang
Study Program : Electrical Engineering
Title : The Designing and Implementing of Model Predictive Controller with Constraint for Level/Flow and Temperature Process Rig 38-003 Temperature Control

In conventional control system, some constraints such as amplitude and control signal's slew rate are not included in the controlling process. So, the result of the control process is not good enough especially if the control signal is forcibly cut before entering the plant. In order to overcome this problem, a Model Predictive Controller is designed. In this MPC control scheme, the few next steps of process output are going to be predicted and some constraints will be ignored so the system output will become precise. In other hand, the occurrence of constraints will improve system's performance into an optimum condition.

The final purpose of this thesis is to design a Model Predictive Controller (MPC) using Quadratic Programming method which will be applied on a real time system of Level/Flow and Temperature Process Rig 38-003. In designing MPC controller for Level/Flow and Temperature Process Rig 38-003, the writer uses system's model on state space form which is obtained by using Least Square method in the basis of input and state variables data of the plant. Input for the plant is voltage which will be used to control the position of servo valve whereas the controlled output is water temperature on the pipe that connects Heat Exchanger's output line and Radiator Cooler's input line.

Experiments conducted prove that MPC with constraints controlling scheme will give a better results than State Controller controlling scheme. Generally, it can be seen that system response to MPC controller is much smoother than system response to State Controller. MPC controller also has smoother control signal variance compared to State Controller control signal

variance. This condition will actually raise the system's physical reliability during the experiment.

Keywords :

constraint, controller, model, prediction



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
LEMBAR PENGESAHAN.....	iii
UCAPAN TERIMA KASIH	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	v
ABSTRAK.....	vi
ABSTRACT.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
1. PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	2
1.3. Pembatasan Masalah.....	2
1.4. Sistematika Penulisan.....	3
2. DASAR TEORI.....	4
2.1. Identifikasi Sistem	4
2.2. Model Predictive Control.....	6
2.2.1. Konsep Dasar Model Predictive Control.....	6
2.2.2. Fungsi Kriteria pada Model Predictive Control.....	9
2.2.3. Model Proses.....	10
2.2.4. Prediksi.....	10
2.2.5. Strategi Pengendali Model Predictive Control tanpa Constraint.....	14
2.2.6. Strategi Pengendali Model Predictive Control dengan Constraint.....	16
2.2.6.1. Pembentukan Constraints.....	16
2.2.6.2. Metode Quadratic Programming.....	18
2.3. Reduced-Order State Observer.....	20
2.3.1. Pembentukan Persamaan State dan Persamaan Keluaran.....	21

2.3.2. Pembentukan Persamaan Dinamik Reduced Order Observer	23
3. PERANCANGAN SISTEM.....	25
3.1. Deskripsi Proses.....	25
3.1.1. Level/Flow and Temperature Process Rig 38-003.....	25
3.1.2. Kalibrasi Komponen.....	27
3.2. Interkoneksi Alat.....	28
3.3. Pembentukan Model Level/Flow and Temperature Process Rig 38-003.....	30
3.3.1. Penentuan Daerah Kerja.....	30
3.3.2. Identifikasi Model Sistem.....	31
3.3.3. Perancangan Reduced-Order State Observer.....	34
3.3.3.1. Pengetesan Observability Sistem.....	34
3.3.3.2. Pembentukan Persamaan Karakteristik Observer...	35
3.3.4. Identifikasi Model Sistem dengan Vektor Kompensasi Nilai Masukan.....	36
3.4. Algoritma Model Predictive Control dengan Constraints.....	40
3.5. Perhitungan Sinyal Kendali.....	45
4. UJI COBA DAN ANALISA.....	54
4.1. Uji Eksperimen Pengendali MPC Tanpa Nilai Trayektori Acuan yang Akan Datang.....	54
4.1.1. Uji Eksperimen untuk Nilai <i>Prediction Horizon</i> yang Bervariasi.....	54
4.1.2. Pengaruh Nilai Faktor Bobot Perubahan Sinyal Kendali (R) pada Hasil Pengendalian MPC.....	59
4.2. Uji Eksperimen Pengendali MPC dengan Nilai Trayektori Acuan yang Akan Datang	62
4.3. Perbandingan Kinerja Pengendali Metode <i>MPC with Constraints</i> dengan Metode Pengendali Ruang Keadaan.....	64
4.3.1. Landasan Teori Pengendali Ruang Keadaan.....	64
4.3.2. Uji Eksperimen dan Analisa.....	67
4.3.2.1. Pengetesan <i>Controllability</i> dan Perancangan	

Pengendali.....	67
4.3.2.2. Uji Eksperimen Pengendali Ruang Keadaan.....	68
5. KESIMPULAN.....	73
DAFTAR REFERENSI.....	74
LAMPIRAN.....	75



DAFTAR GAMBAR

Gambar 2.1.	Struktur pengendali MPC.....	8
Gambar 2.2.	Kalkulasi keluaran proses dan pengendali terprediksi.....	8
Gambar 2.3.	Skematik <i>observed-state feedback control system</i>	21
Gambar 2.4.	Skematik <i>reduced-order observer</i> dengan <i>state feedback control system</i>	24
Gambar 3.1.	Sketsa <i>Temperature Process Rig 38-600</i>	25
Gambar 3.2.	Sketsa <i>Process Interface 38-200</i>	26
Gambar 3.3.	Sketsa <i>Process Controller 38-300</i>	27
Gambar 3.4.	Sketsa <i>Level/Flow and Temperature Process Rig 38-003</i> ...	27
Gambar 3.5.	Ilustrasi arah aliran sinyal pada sistem.....	29
Gambar 3.6.	Grafik karakteristik <i>Level/Flow and Temperature Process Rig 38-003</i> : Tanggapan Masukan vs Keluaran.....	30
Gambar 3.7.	Blok simulink untuk proses identifikasi.....	31
Gambar 3.8.	(a) Grafik sinyal masukan. (b) Grafik dari <i>Thermistor T4</i>	32
Gambar 3.9.	Blok diagram <i>reduced-order observer</i>	36
Gambar 3.10.	Blok simulink proses identifikasi model dengan vektor kesalahan masukan.....	36
Gambar 3.11.	(a) Grafik sinyal masukan (b). Grafik dari <i>Thermistor</i> (c). Grafik dari <i>reduced-order observer</i>	38
Gambar 3.12.	Grafik keluaran proses vs keluaran model.....	40
Gambar 3.13.	Selisih keluaran model dengan keluaran proses.....	40
Gambar 3.14.	Blok diagram pengendali MPC dengan <i>constraints</i>	41
Gambar 3.15.	Diagram alir algoritma MPC dengan <i>constraints</i>	42
Gambar 3.16.	Diagram alir metode <i>Active Set</i> untuk menyelesaikan <i>Quadratic Programming</i>	45
Gambar 4.1.	Keluaran sistem hasil uji eksperimen dengan <i>prediction horizon</i> yang berbeda.....	55
Gambar 4.2.	Sinyal kendali hasil uji eksperimen dengan <i>prediction horizon</i> yang berbeda.....	56

Gambar 4.3.	Hasil estimasi <i>reduced order observer</i> untuk uji eksperimen dengan <i>prediction horizon</i> yang berbeda.....	57
Gambar 4.4.	Keluaran sistem hasil uji eksperimen dengan nilai R yang berbeda.....	59
Gambar 4.5.	Sinyal kendali hasil uji eksperimen dengan nilai R yang berbeda.....	60
Gambar 4.6.	Hasil estimasi <i>reduced order observer</i> untuk uji eksperimen dengan nilai R yang berbeda.....	61
Gambar 4.7.	Keluaran sistem hasil uji eksperimen dengan nilai trayektori acuan yang akan datang diketahui.....	63
Gambar 4.8.	Sinyal kendali uji eksperimen dengan nilai trayektori acuan yang akan datang diketahui.....	63
Gambar 4.9.	Hasil estimasi <i>reduced order observer</i> untuk uji eksperimen dengan nilai trayektori acuan yang akan datang diketahui.....	64
Gambar 4.10.	Pengendali ruang keadaan lingkaran tertutup.....	65
Gambar 4.11.	Pengendali ruang keadaan lingkaran tertutup dengan penguat <i>precompensator</i>	66
Gambar 4.12.	Hasil keluaran uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint.....	68
Gambar 4.13.	Sinyal kendali uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint.....	69
Gambar 4.14.	Hasil estimasi <i>reduced-order observer</i> untuk uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint.....	69
Gambar 4.15.	Hasil keluaran uji eksperimen pengendali ruang keadaan dengan perubahan setpoint.....	70
Gambar 4.16.	Sinyal kendali uji eksperimen pengendali ruang keadaan dengan perubahan setpoint.....	70
Gambar 4.17.	Hasil estimasi <i>reduced-order observer</i> untuk uji eksperimen pengendali ruang keadaan dengan perubahan setpoint.....	71

1. PENDAHULUAN

1.1. Latar Belakang

Dengan perkembangan yang sangat pesat dalam dunia industri belakangan ini, sistem *Heat Exchanger* sangat banyak dipakai dalam membantu serangkaian proses dalam industri. Sistem *Heat Exchanger* merupakan sebuah sistem yang tersusun dari dua buah sistem dasar yaitu sistem *reservoir* dan sistem *heater/cooler*.

Dalam merancang pengendali untuk sebuah sistem *Heat Exchanger*, dibutuhkan beberapa pengendali yang ditempatkan pada masing-masing sistem dasar pembentuk *Heat Exchanger* yaitu pengendali ketinggian air *reservoir* (*water level controller*) dan pengendali temperatur *heater/cooler* (*heater/cooler temperature controller*).

Model Predictive Control merupakan suatu metodologi pengendalian yang saat ini memiliki pengaruh besar dalam dunia industri dibandingkan dengan pengendali konvensional seperti *Two-Degree of Freedom* ataupun Aturan Kendali Kenaikan.

Pada sistem kendali konvensional, batasan-batasan (*constraints*) seperti amplitudo dan *slew rate* sinyal kendali tidak diperhitungkan pada proses pengendalian. Hal ini tentu dapat menyebabkan hasil kendali menjadi kurang baik, terutama jika terjadi pemotongan paksa terhadap sinyal kendali sebelum masuk ke *plant*. Pemotongan sinyal kendali biasanya terjadi ketika nilai trayektori acuan berubah secara mendadak. Hal tersebut tentu tidak akan terjadi pada MPC karena pengendali dapat memprediksi keluaran proses yang akan datang serta tidak mengabaikan batasan-batasan yang ada. Selain agar keluaran sistem menjadi bagus, adanya batasan pada proses pengendali dapat membuat kinerja alat menjadi optimal sehingga alat tidak cepat rusak dan dapat beroperasi dalam jangka waktu yang lama.

Banyaknya faktor yang harus diperhitungkan pada pengendali MPC membuat algoritma MPC menjadi sangat panjang dan rumit. Akan

tetapi dengan kecepatan komputasi perangkat keras saat ini, tidak lagi menjadi masalah utama. Masalah utama metode MPC adalah keperluan akan model proses. Model proses pada MPC berguna untuk memprediksi keluaran sistem sehingga pengendali MPC dapat memberikan sinyal masukan yang sesuai. Oleh sebab itu, algoritma MPC membutuhkan model proses yang baik.

1.2. Tujuan

Tujuan dari skripsi ini adalah untuk merancang sebuah pengendali *Model Predictive Control* (MPC) dengan batasan dan mengimplementasikan pengendali ini ke sistem *Level/Flow and Temperature Process Rig 38-003* pada sistem *Heat Exchanger* yang digunakan untuk mengendalikan temperatur air pada saluran sebelum memasuki *heat exchanger*. Metode perancangan pengendali MPC yang dipakai adalah metode *Quadratic Programming*, dirancang dengan menggunakan fasilitas program *s-function* pada *MatLab 7.0*

1.3. Pembatasan Masalah

Skripsi ini membahas perancangan MPC dengan batasan menggunakan metode *Quadratic Programming* dalam menghitung besar perubahan sinyal kendali. Batasan (*Constraints*) yang digunakan adalah amplitudo dan *slew rate* sinyal kendali.

Model yang digunakan pada skripsi ini adalah model ruang keadaan linier dengan tegangan untuk *servo valve* pada *Temperature Process Rig 38-600* sebagai masukan dan temperatur T4 (terletak pada saluran yang mengalirkan fluida yang menuju ke *cooling radiator*) sebagai keluaran sistem.

1.4. Sistematika Penulisan

Penulisan skripsi ini akan dibagi ke dalam lima bab yang akan menjelaskan secara bertahap mengenai keseluruhan isi skripsi ini.

Bab satu merupakan pendahuluan yang berisi latar belakang, tujuan, pembatasan masalah, dan sistematika penulisan. Bab dua membahas dasar teori yaitu tentang konsep dasar perancangan pengendali *Model Predictive Control* (MPC) dengan batasan. Bab tiga membahas mengenai perangkat keras dan perangkat lunak yang digunakan. Bab empat berisi hasil uji coba dan analisa mengenai hasil percobaan yang dilakukan. Bab lima merupakan kesimpulan dari keseluruhan pembahasan dalam laporan skripsi ini.

2. DASAR TEORI

2.1. Identifikasi Sistem

Pada skripsi ini, model proses ditentukan berdasarkan data masukan dan keluaran dengan menggunakan metode Kuadrat Terkecil. Inti dari metode Kuadrat Terkecil adalah bahwa kecocokan antara model dengan sistem yang akan diidentifikasi diperoleh dengan meminimumkan selisih kuadrat antara keluaran model dengan keluaran sistem yang diidentifikasi untuk semua N data pengamatan [1]. Selisih kuadrat antara keluaran model dan keluaran sistem dapat dinyatakan dalam fungsi kriteria berikut

$$J_{LS} = \sum_{i=1}^N \varepsilon_i^2 = \sum_{i=1}^N (y(i) - \hat{y}(i))^2 \quad (2.1)$$

dengan :

- J_{LS} = fungsi kriteria
- ε_i = kesalahan prediksi data ke- i
- $y(i)$ = data keluaran ke- i
- $\hat{y}(i)$ = prediksi keluaran ke- i

Fungsi kriteria pada persamaan (2.1) disebut juga sebagai *loss function*.

Keluaran model untuk satu langkah prediksi kedepan dari model dinamik orde- n adalah sebagai berikut

$$\hat{y}(k) = -a_1 y(k-1) - \dots - a_n y(k-n) + b_1 u(k-1) + \dots + b_n u(k-n) \quad (2.2)$$

Persamaan (2.2) kemudian dapat ditulis ke dalam bentuk vektor matriks sebagai berikut

$$\hat{y}(k) = \underbrace{\begin{bmatrix} -y(k-1) & \cdots & -y(k-n) & u(k-1) & \cdots & u(k-n) \end{bmatrix}}_{\underline{\rho}^T} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\underline{\hat{\theta}}} \quad (2.3)$$

Dengan mensubstitusikan persamaan (2.3) ke persamaan (2.1), maka persamaan *loss function* J_{LS} menjadi

$$J_{LS} = \sum_{i=1}^N (y(i) - \underline{\rho}^T(i) \underline{\hat{\theta}})^2 \quad (2.4)$$

Untuk sejumlah N data, persamaan (2.3) dapat ditulis kembali dalam bentuk matriks menjadi

$$\underbrace{\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(N) \end{bmatrix}}_{\underline{\hat{y}}} = \underbrace{\begin{bmatrix} -y(0) & \cdots & -y(-n) & u(0) & \cdots & u(-n) \\ -y(1) & \cdots & -y(1-n) & u(1) & \cdots & u(1-n) \\ \vdots & & \vdots & \vdots & & \vdots \\ -y(N-1) & \cdots & -y(N-n) & u(N-1) & \cdots & u(N-n) \end{bmatrix}}_{\underline{P}} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\underline{\hat{\theta}}} \quad (2.5)$$

atau

$$\underline{\hat{y}} = \underline{P} \underline{\hat{\theta}} \quad (2.6)$$

Supaya persamaan (2.4) dapat diminimasi, maka persamaan (2.4) harus dinyatakan dalam bentuk

$$\begin{aligned} J_{LS} &= (\underline{y} - \underline{P} \underline{\hat{\theta}})^T (\underline{y} - \underline{P} \underline{\hat{\theta}}) \\ &= \underline{y}^T \underline{y} - \underline{\hat{\theta}}^T \underline{P}^T \underline{y} - \underline{y}^T \underline{P} \underline{\hat{\theta}} + \underline{\hat{\theta}}^T \underline{P}^T \underline{P} \underline{\hat{\theta}} \end{aligned} \quad (2.7)$$

Selanjutnya dengan membuat turunan pertama $J_{LS}(\underline{\theta})$ terhadap $\underline{\hat{\theta}}$ menjadi nol :

$$\left. \frac{\partial J_{LS}(\underline{\theta})}{\partial \underline{\theta}} \right|_{\underline{\theta}=\underline{\hat{\theta}}} = -2\underline{P}^T \underline{y} + 2\underline{P}^T \underline{P} \underline{\hat{\theta}} = 0$$

maka didapatkan rumus untuk menghitung parameter estimasi $\underline{\hat{\theta}}$ sebagai berikut

$$\underline{\hat{\theta}} = (\underline{P}^T \underline{P})^{-1} \underline{P}^T \underline{y} \quad (2.8)$$

2.2. Model Predictive Control (MPC)

2.2.1. Konsep Dasar Model Predictive Control

Model Predictive 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 [2] :

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 di atas diulang dengan menggunakan informasi terakhir.

Metode MPC memiliki beberapa keunggulan dibandingkan dengan metode pengendali lainnya, di antaranya 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 *multivariable*.
4. Mempunyai kompensasi terhadap waktu tunda.
5. Mempunyai kemampuan dari pengendali *feed forward* untuk mengkompensasi gangguan yang terukur.
6. Mudah untuk mengimplementasikan pengendali yang diperoleh.
7. Dapat memperhitungkan batasan atau *constraint* dalam merancang pengendali.
8. Sangat berguna jika sinyal acuan untuk masa yang akan datang diketahui.

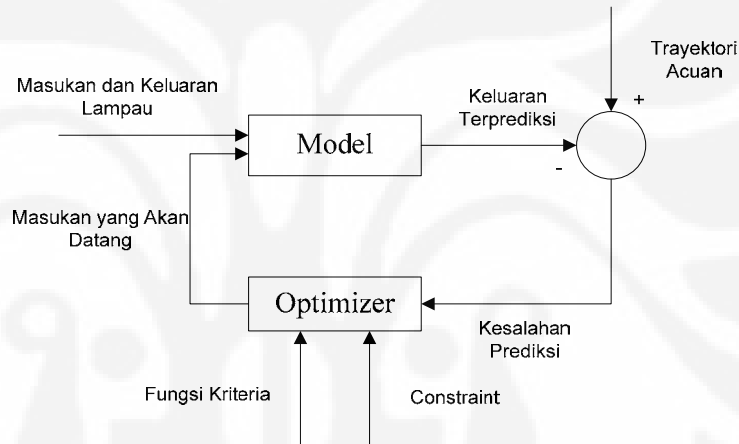
Selain beragam keuntungan yang dimiliki, metode MPC juga mempunyai kelemahan, yaitu masalah penurunan aturan sinyal kendali yang cukup kompleks dan keperluan akan model proses yang baik.

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 [1] :

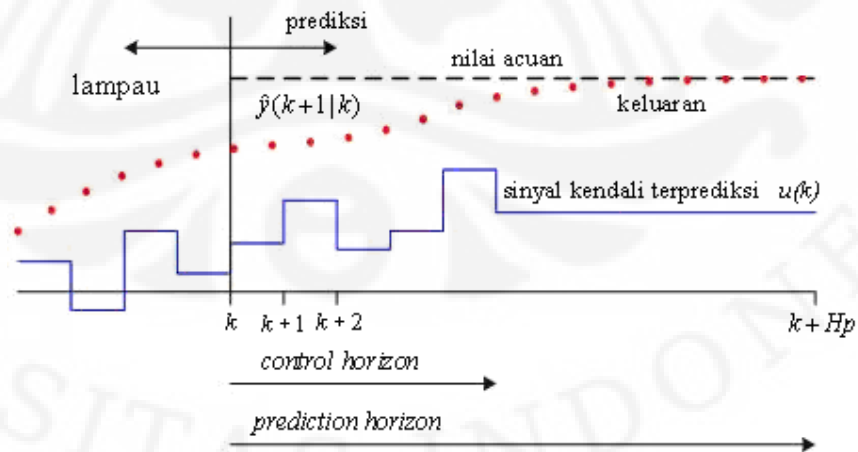
1. Keluaran proses yang akan datang untuk rentang *horizon Hp* yang ditentukan yang dinamakan sebagai *prediction horizon*, diprediksi pada setiap waktu pencuplikan dengan menggunakan model proses. Keluaran proses terprediksi ini $y(k+i/k)$ untuk $i = 1 \dots Hp$ bergantung pada nilai masukan dan keluaran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



Gambar2.1. Struktur pengendali MPC



Gambar2.2. Kalkulasi keluaran proses dan pengendali terprediksi

2.1.2. 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

$$V(k) = \sum_{i=1}^{H_p} \|\underline{\hat{y}}(k+i|k) - \underline{r}(k+i|k)\|_{\underline{Q}(i)}^2 + \sum_{i=0}^{H_u-1} \|\underline{\Delta\hat{u}}(k+i|k)\|_{\underline{R}(i)}^2 \quad (2.9)$$

dengan :

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

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

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

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

H_p = *prediction horizon*

H_u = *control horizon*

Dari persamaan fungsi kriteria tersebut, selalu dibuat asumsi bahwa nilai $H_u < H_p$ dan $\underline{\Delta\hat{u}}(k+i|k) = 0$ untuk $i \geq H_u$, sehingga nilai masukan terprediksi $\underline{\hat{u}}(k+i|k) = \underline{\hat{u}}(k+H_u-i|k)$ untuk semua $i \geq H_u$ seperti yang terlihat pada gambar 2.2.

Bentuk dari fungsi kriteria pada persamaan (2.9) menyatakan bahwa vektor kesalahan $\underline{\hat{y}}(k+i|k) - \underline{r}(k+i|k)$ 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 $\underline{Q}(i)$ bernilai nol pada langkah yang diinginkan. Selain vektor kesalahan, fungsi kriteria pada persamaan (2.9) juga memperhitungkan perubahan vektor masukan dalam rentang *control horizon*. Pemilihan penggunaan $\underline{\Delta\hat{u}}(k+i|k)$ yang pada fungsi kriteria bertujuan untuk meminimumkan perubahan sinyal kendali yang masuk ke *plant*.

2.1.3. Model Proses

Pada pembahasan skripsi ini, model proses yang digunakan berupa model ruang keadaan diskrit linier seperti berikut :

$$\underline{x}(k+1) = \underline{A}\underline{x}(k) + \underline{B}\underline{u}(k) \quad (2.10)$$

$$\underline{y}(k) = \underline{C}\underline{x}(k) \quad (2.11)$$

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 $n \times n$

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

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

Model ruang keadaan pada persamaan (2.10) dan (2.11) adalah model ruang keadaan untuk proses yang bersifat linier. Pada skripsi ini, vektor masukan $\underline{u}(k)$ dan keluaran $\underline{y}(k)$ masing-masing berdimensi satu.

2.1.4. Prediksi

Dalam menyelesaikan masalah pengendali prediktif, nilai keluaran terprediksi $\hat{y}(k+i|k)$ 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 (2.10) dan (2.11). Perhitungan prediksi variabel keadaan adalah sebagai berikut

$$\hat{\underline{x}}(k+1|k) = \underline{A}\underline{x}(k) + \underline{B}\hat{\underline{u}}(k|k) \quad (2.12)$$

$$\hat{\underline{x}}(k+2|k) = \underline{A}\hat{\underline{x}}(k+1|k) + \underline{B}\hat{\underline{u}}(k+1|k)$$

$$= \underline{A}^2 \underline{x}(k) + \underline{A}\underline{B}\hat{u}(k|k) + \underline{B}\hat{u}(k+1|k) \quad (2.13)$$

$$\vdots$$

$$\begin{aligned} \hat{x}(k+Hp|k) &= \underline{A}\hat{x}(k+Hp-1|k) + \underline{B}\hat{u}(k+Hp-1|k) \\ &= \underline{A}^{Hp} \underline{x}(k) + \underline{A}^{Hp-1} \underline{B}\hat{u}(k|k) + \dots + \underline{B}\hat{u}(k+Hp-1|k) \end{aligned} \quad (2.14)$$

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

Sekarang, diasumsikan bahwa nilai masukan hanya berubah pada waktu $k, k+1, \dots, k+Hu-1$, dan setelah itu menjadi konstan, sehingga didapatkan bahwa $\hat{u}(k+i|k) = \hat{u}(k+Hu-1|k)$ untuk $Hu \leq i \leq Hp-1$. Selanjutnya, perhitungan prediksi diubah sehingga mengandung $\underline{\Delta}\hat{u}(k+i|k)$ daripada $\hat{u}(k+i|k)$, dengan

$$\underline{\Delta}\hat{u}(k+i|k) = \hat{u}(k+i|k) - \hat{u}(k+i-1|k) \quad (2.15)$$

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

$$\hat{u}(k|k) = \underline{\Delta}\hat{u}(k|k) + \underline{u}(k-1|k) \quad (2.16)$$

$$\hat{u}(k+1|k) = \underline{\Delta}\hat{u}(k+1|k) + \underline{\Delta}\hat{u}(k|k) + \underline{u}(k-1|k) \quad (2.17)$$

$$\vdots$$

$$\hat{u}(k+Hu-1|k) = \underline{\Delta}\hat{u}(k+Hu-1|k) + \dots + \underline{\Delta}\hat{u}(k|k) + \underline{u}(k-1|k) \quad (2.18)$$

Dengan mensubstitusikan persamaan (2.16) – (2.18) ke persamaan (2.12) – (2.14), diperoleh persamaan

$$\hat{x}(k+1|k) = \underline{A}\underline{x}(k) + \underline{B}[\underline{\Delta}\hat{u}(k|k) + \underline{u}(k-1)] \quad (2.19)$$

$$\begin{aligned} \hat{x}(k+2|k) &= \underline{A}^2 \underline{x}(k) + \underline{A}\underline{B}[\underline{\Delta}\hat{u}(k|k) + \underline{u}(k-1)] \\ &\quad + \underline{B}[\underline{\Delta}\hat{u}(k+1|k) + \underline{\Delta}\hat{u}(k|k) + \underline{u}(k-1)] \\ &\quad \underbrace{\hspace{10em}}_{\hat{u}(k+1|k)} \end{aligned}$$

$$= \underline{A}^2 \underline{x}(k) + (\underline{A} + \underline{I}) \underline{B} \underline{\Delta} \hat{u}(k | k) + \underline{B} \underline{\Delta} \hat{u}(k + 1 | k) + (\underline{A} + \underline{I}) \underline{B} u(k - 1) \quad (2.20)$$

$$\vdots$$

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

$$+ \underline{B} \underline{\Delta} \hat{u}(k + Hu - 1 | k) + (\underline{A}^{Hu-1} + \dots + \underline{A} + \underline{I}) \underline{B} u(k - 1) \quad (2.21)$$

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

$$\hat{\underline{x}}(k + Hu + 1 | k) = \underline{A}^{Hu+1} \underline{x}(k) + (\underline{A}^{Hu} + \dots + \underline{A} + \underline{I}) \underline{B} \underline{\Delta} \hat{u}(k | k) + \dots$$

$$+ (\underline{A} + \underline{I}) \underline{B} \underline{\Delta} \hat{u}(k + Hu - 1 | k) + (\underline{A}^{Hu} + \dots + \underline{A} + \underline{I}) \underline{B} u(k - 1) \quad (2.22)$$

$$\vdots$$

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

$$+ (\underline{A}^{Hp-Hu} + \dots + \underline{A} + \underline{I}) \underline{B} \underline{\Delta} \hat{u}(k + Hu - 1 | k)$$

$$+ (\underline{A}^{Hp-1} + \dots + \underline{A} + \underline{I}) \underline{B} u(k - 1) \quad (2.23)$$

Akhirnya, persamaan (2.19) – (2.23) dapat disusun ke dalam bentuk vektor matriks sebagai berikut

$$\begin{aligned}
 \begin{bmatrix} \hat{\underline{x}}(k+1|k) \\ \vdots \\ \hat{\underline{x}}(k+Hu|k) \\ \hat{\underline{x}}(k+Hu+1|k) \\ \vdots \\ \hat{\underline{x}}(k+Hp|k) \end{bmatrix} &= \underbrace{\begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^{Hu} \\ \underline{A}^{Hu+1} \\ \vdots \\ \underline{A}^{Hp} \end{bmatrix}}_{\Psi} \underline{x}(k) + \underbrace{\begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} \end{bmatrix}}_{\Gamma} \underline{u}(k-1) \\
 &+ \underbrace{\begin{bmatrix} \underline{B} & \cdots & \underline{0}_{n \times l} \\ \underline{AB} + \underline{B} & \cdots & \underline{0}_{n \times l} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} & \cdots & \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} & \cdots & \underline{AB} + \underline{B} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^{Hp-Hu} \underline{A}^i \underline{B} \end{bmatrix}}_{\Theta} \begin{bmatrix} \Delta \hat{\underline{u}}(k) \\ \vdots \\ \Delta \hat{\underline{u}}(k+Hu-1) \end{bmatrix} \quad (2.24)
 \end{aligned}$$

Lampau
Prediksi

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

$$\hat{\underline{y}}(k+1|k) = \underline{C} \hat{\underline{x}}(k+1|k) \quad (2.25)$$

$$\hat{\underline{y}}(k+2|k) = \underline{C} \hat{\underline{x}}(k+2|k) \quad (2.26)$$

⋮

$$\hat{\underline{y}}(k+Hp|k) = \underline{C} \hat{\underline{x}}(k+Hp|k) \quad (2.27)$$

Persamaan (2.25) – (2.27) kemudian dapat ditulis kedalam vektor matriks sebagai berikut

$$\begin{bmatrix} \hat{\underline{y}}(k+1|k) \\ \vdots \\ \hat{\underline{y}}(k+Hp|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{C} & \underline{0}_{m \times n} & \cdots & \underline{0}_{m \times n} \\ \underline{0}_{m \times n} & \underline{C} & \cdots & \underline{0}_{m \times n} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{m \times n} & \underline{0}_{m \times n} & \cdots & \underline{C} \end{bmatrix}}_{\underline{C}_y} \begin{bmatrix} \hat{\underline{x}}(k+1|k) \\ \vdots \\ \hat{\underline{x}}(k+Hp|k) \end{bmatrix} \quad (2.28)$$

2.1.5. Strategi Pengendali *Model Predictive Control* tanpa *Constraints*

Fungsi kriteria yang akan diminimumkan sama seperti pada persamaan (2.9) dan dapat ditulis sebagai berikut

$$V(k) = \|\underline{Y}(k) - \underline{T}(k)\|_{\underline{Q}}^2 + \|\underline{\Delta U}(k)\|_{\underline{R}}^2 \quad (2.29)$$

dimana

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

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

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

$$\underline{Q} = \begin{bmatrix} Q(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q(Hp) \end{bmatrix} \quad (2.30)$$

$$\underline{R} = \begin{bmatrix} R(0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R(Hu-1) \end{bmatrix} \quad (2.31)$$

Berdasarkan pada persamaan ruang keadaan (2.24) dan (2.28), maka matriks $\underline{Y}(k)$ dapat ditulis dalam bentuk

$$\underline{Y}(k) = \underline{C}_Y \underline{\Psi} x(k) + \underline{C}_Y \underline{\Gamma} u(k-1) + \underline{C}_Y \underline{\Theta} \underline{\Delta U}(k) \quad (2.32)$$

Selain matriks-matriks di atas, didefinisikan juga suatu matriks penjejukan 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 ($\Delta U(k) = 0$) [3]. Persamaan matematis dari matriks $\underline{E}(k)$ adalah sebagai berikut

$$\underline{E}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi}_x(k) - \underline{C}_y \underline{\Gamma} u(k-1) \quad (2.33)$$

Persamaan (2.29) kemudian dapat ditulis kembali dalam bentuk yang mengandung matriks $\underline{E}(k)$ dan $\Delta U(k)$ sebagai berikut

$$V(k) = \|\underline{C}_y \underline{\Theta} \Delta U(k) - \underline{E}(k)\|_{\underline{Q}}^2 + \|\Delta U(k)\|_{\underline{R}}^2 \quad (2.34)$$

$$= [\Delta U^T(k) \underline{\Theta}^T \underline{C}_y^T - \underline{E}^T(k)] \underline{Q} [\underline{C}_y \underline{\Theta} \Delta U(k) - \underline{E}(k)] + \Delta U^T(k) \underline{R} \Delta U(k) \quad (2.35)$$

$$= \underbrace{\underline{E}^T(k) \underline{Q} \underline{E}(k)}_{\underline{c}_1} - \underbrace{\Delta U^T(k) 2 \underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{E}(k)}_{\underline{G}} + \underbrace{\Delta U^T(k) [\underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{C}_y \underline{\Theta} + \underline{R}]}_{\underline{H}} \Delta U(k) \quad (2.36)$$

Pada persamaan (2.36), bagian $\underline{E}^T(k) \underline{Q} \underline{E}(k)$ tidak mengandung unsur $\Delta U(k)$ sehingga bagian tersebut bisa dianggap konstan sehingga bagian tersebut tidak diikutsertakan dalam proses optimasi untuk menghitung nilai $\Delta U(k)$. Persamaan (2.36) kemudian dapat ditulis kembali menjadi

$$V(k) = \underline{c}_1 - \Delta U^T(k) \underline{G} + \Delta U^T(k) \underline{H} \Delta U(k) \quad (2.37)$$

dimana

$$\underline{G} = 2 \underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{E}(k) \quad (2.38)$$

dan

$$\underline{H} = \underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{C}_y \underline{\Theta} + \underline{R} \quad (2.39)$$

Nilai optimal $\Delta U(k)$ dapat dihitung dengan membuat gradien dari $V(k)$ bernilai nol [3]. Gradien $V(k)$ dari persamaan (2.37) adalah

$$\nabla_{\underline{\Delta U}(k)} \underline{V}(k) = -\underline{G} + 2\underline{\mathcal{H}} \underline{\Delta U}(k) \quad (2.40)$$

Dengan membuat nol nilai $\nabla_{\underline{\Delta U}(k)} \underline{V}(k)$ pada persamaan (2.40), maka didapatkan nilai optimal dari perubahan sinyal kendali sebagai berikut

$$\underline{\Delta U}(k)_{opt} = \frac{1}{2} \underline{\mathcal{H}}^{-1} \underline{G} \quad (2.41)$$

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 [3].

2.1.6. Strategi Pengendali *Model Predictive Control* dengan *Constraints*

2.1.6.1. Pembentukan *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

$$\underline{FU}(k) \leq \underline{f} \quad (2.42)$$

$$\underline{E\Delta U}(k) \leq \underline{e} \quad (2.43)$$

Pada algoritma MPC, yang akan dihitung adalah nilai optimal perubahan sinyal kendali $\underline{\Delta U}(k)$ sehingga sangat perlu untuk mengubah bentuk *constraints* yang belum mengandung $\underline{\Delta U}(k)$ menjadi bentuk *constraints* yang mengandung $\underline{\Delta U}(k)$. Sebagai contoh adalah pertidaksamaan (2.42), karena pada pertidaksamaan (2.42) belum mengandung $\underline{\Delta U}(k)$ maka bentuk pertidaksamaan (2.42) harus diubah terlebih dahulu menjadi bentuk yang mengandung $\underline{\Delta U}(k)$.

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

$$\underline{u}_{\min} \leq \underline{u}(k) \leq \underline{u}_{\max} \quad (2.44)$$

Pertidaksamaan (2.44) dapat ditulis menjadi dua bentuk yang terpisah seperti berikut ini

$$-\underline{u}(k) \leq -\underline{u}_{\min} \quad (2.45)$$

$$\underline{u}(k) \leq \underline{u}_{\max} \quad (2.46)$$

Pertidaksamaan (2.45) dan (2.46) masing-masing dapat ditulis dalam bentuk yang mengandung $\underline{\Delta U}(k)$ menjadi

$$-\underline{F}'\underline{\Delta U}(k) \leq -\underline{u}_{\min} + \underline{F}_1 u(k-1) \quad (2.47)$$

$$\underline{F}'\underline{\Delta U}(k) \leq \underline{u}_{\max} - \underline{F}_1 u(k-1) \quad (2.48)$$

dimana

$$\underline{F}' = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{Hu \times Hu} \quad (2.49)$$

dan

$$\underline{F}_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{Hu \times 1} \quad (2.50)$$

Untuk pertidaksamaan (2.43), bentuknya tidak perlu diubah lagi karena pada pertidaksamaan tersebut sudah mengandung unsur $\underline{\Delta U}(k)$.

Pertidaksamaan (2.43), (2.47), dan (2.48) kemudian dapat disusun menjadi sebuah vektor matriks sebagai berikut

$$\underbrace{\begin{bmatrix} -\underline{F}' \\ \underline{F}' \\ \underline{E} \end{bmatrix}}_{\underline{\Omega}} \underbrace{\Delta U(k)}_{\underline{\delta}} \leq \underbrace{\begin{bmatrix} -\underline{u}_{\min} + \underline{F}_1 u(k-1) \\ \underline{u}_{\max} - \underline{F}_1 u(k-1) \\ \underline{e} \end{bmatrix}}_{\underline{\omega}} \quad (2.51)$$

Vektor matriks pada pertidaksamaan (2.51) digunakan pada perhitungan nilai optimal perubahan sinyal kendali $\underline{\Delta U}(k)_{opt}$.

2.1.6.2. Metode *Quadratic Programming*

Fungsi kriteria pada pengendali MPC dengan *constraints* sama dengan fungsi kriteria pada pengendali MPC tanpa *constraints* (persamaan (2.37)). Permasalahan utama proses optimasi ini adalah meminimalkan fungsi kriteria

$$\underline{\Delta U}^T(k) \underline{H} \underline{\Delta U}(k) - \underline{\Delta U}^T(k) \underline{G} \quad (2.52)$$

berdasarkan pada pertidaksamaan *constraint* (2.51) atau

$$\min_{\underline{\delta}} \frac{1}{2} \underline{\delta}^T \underline{\Phi} \underline{\delta} + \underline{\phi} \underline{\delta} \quad (2.53)$$

berdasarkan pada *constraints*

$$\underline{\Omega} \underline{\delta} \leq \underline{\omega} \quad (2.54)$$

Bentuk (2.53) dan (2.54) adalah masalah optimasi standar yang disebut sebagai permasalahan *Quadratic Programming* (QP). Bila ada bagian yang aktif di dalam himpunan *constraints* pada persamaan (2.54), maka bagian aktif tersebut akan membuat pertidaksamaan (2.54) menjadi suatu persamaan

$$\underline{\Omega}_a \underline{\delta} = \underline{\omega}_a \quad (2.55)$$

dengan matriks $\underline{\Omega}_a$ adalah bagian yang aktif dari matriks pertidaksamaan (2.54). Persamaan (2.55) kemudian dijadikan sebagai *constraints* dari fungsi kriteria pada persamaan (2.53).

Permasalahan optimasi persamaan (2.53) dengan subyek terhadap persamaan (2.55) dapat diselesaikan dengan teori pengali *Lagrange*

$$\min_{\underline{\delta}, \underline{\lambda}} L(\underline{\delta}, \underline{\lambda}) \quad (2.56)$$

dengan

$$L(\underline{\delta}, \underline{\lambda}) = \frac{1}{2} \underline{\delta}^T \underline{\Phi} \underline{\delta} + \underline{\phi} \underline{\delta} + \underline{\lambda} (\underline{\Omega}_a \underline{\delta} - \underline{\omega}_a) \quad (2.57)$$

Selanjutnya dengan melakukan diferensiasi parsial terhadap $\underline{\delta}$ dan $\underline{\lambda}$ dari persamaan (2.57), maka didapatkan kondisi *Karush-Kuhn-Tucker* sebagai berikut

$$\nabla_{\underline{\delta}} L(\underline{\delta}, \underline{\lambda}) = \underline{\Phi} \underline{\delta} + \underline{\phi} + \underline{\Omega}_a^T \underline{\lambda} \quad (2.58)$$

$$\nabla_{\underline{\lambda}} L(\underline{\delta}, \underline{\lambda}) = \underline{\Omega}_a \underline{\delta} - \underline{\omega}_a \quad (2.59)$$

atau

$$\nabla L(\underline{\delta}, \underline{\lambda}) = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix} - \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \quad (2.60)$$

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

$$\begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix}_{opt} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix}^{-1} \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \quad (2.61)$$

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

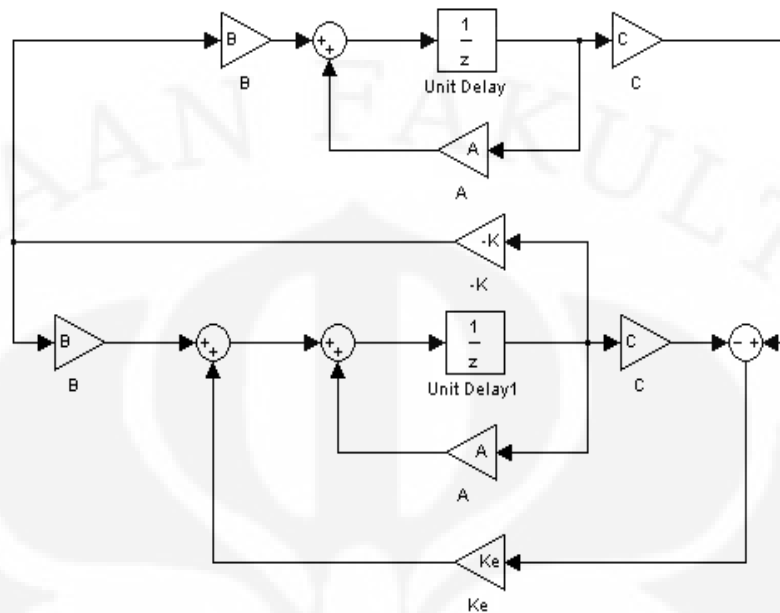
infeasible. Hal ini tentu tidak dapat diterima karena sinyal kendali hasil komputasi harus selalu ada untuk digunakan sebagai masukan bagi *plant*, sehingga sangat penting untuk membuat metode cadangan dalam menghitung sinyal masukan ketika algoritma MPC diterapkan. Beberapa pendekatan yang dapat dilakukan untuk menghindari terjadinya solusi yang *infeasible* pada MPC antara lain [3] :

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

2.3. **REDUCED-ORDER STATE OBSERVER [4]**

Dalam suatu proses kontrol industri, sensor merupakan hal yang memegang peranan yang sangat penting. Semua data atau yang sering disebut dengan *state* yang akan di kalkulasi oleh pengendali adalah data yang direkam oleh sensor-sensor yang ada dalam sistem tersebut. Dalam banyak kasus praktikal, hanya sedikit variabel *state* yang terukur dan sisanya adalah *state* yang tidak dapat terukur oleh sensor. Oleh karena itu, *state variable* yang tidak terukur dapat di estimasi dan hal ini sering disebut dengan proses observasi. Sistem nyata membutuhkan observasi atau estimasi *state variable* yang tidak terukur dari data-data keluaran dan variabel kendali. Untuk melakukan observasi sebagian *state variable* yang tidak terukur maka dapat dipakai algoritma observasi yang disebut *Reduced-Order State Observation*.

Berikut ini adalah skematik sederhana dari sebuah *Observed-State Feedback Control System* :



Gambar 2.3. Skematik *Observed-State Feedback Control System*

Untuk mendisain sebuah *reduced-order state observer*, asumsikan bahwa *state vector* $x(k)$ adalah sebuah n -vektor dan *output vector* $y(k)$ adalah m -vektor yang dapat diukur. Sehingga kita harus melakukan estimasi untuk sejumlah $n-m$ vektor.

2.3.1. Pembentukan Persamaan *State* dan Persamaan Keluaran

Reduced-order observer dapat didisain dengan melakukan proses partisi *state vector* $x(k)$ kedalam dua bagian yaitu :

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

dimana $x_1(k)$ adalah bagian dari *state vector* yang dapat terukur (sehingga $x_1(k)$ adalah sebuah m -vektor) sedangkan $x_2(k)$ adalah bagian dari *state vector* yang tidak dapat terukur (sehingga $x_2(k)$ adalah sebuah $(n-m)$ vektor). Partisi persamaan keadaan sistem menjadi seperti berikut :

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k) \quad (2.62)$$

$$y(k) = [I \mid 0] \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (2.63)$$

dengan menuliskan ulang persamaan (2.62), maka persamaan untuk bagian *state vector* yang dapat diukur menjadi :

$$x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_1u(k)$$

atau

$$x_1(k+1) - A_{11}x_1(k) - B_1u(k) = A_{12}x_2(k) \quad (2.64)$$

dimana persamaan pada ruas kiri adalah *state* yang dapat diukur.

Persamaan (2.64) ini sering disebut juga dengan persamaan keluaran.

Dari persamaan (2.62) juga dapat dibentuk sebuah persamaan *state* yang tidak dapat diukur yaitu :

$$x_2(k+1) = A_{21}x_1(k) + A_{22}x_2(k) + B_2u(k) \quad (2.65)$$

Persamaan (2.65) ini sering disebut dengan persamaan *state*.

Persamaan-persamaan diatas dapat dianalogikan dengan persamaan keluaran dan persamaan *state* dari sebuah *full-order observer* yaitu :

$$y(k) = Cx(k)$$

dengan

$$x_1(k+1) - A_{11}x_1(k) - B_1u(k) = A_{12}x_2(k)$$

dan

$$x(k+1) = Ax(k) + Bu(k)$$

dengan

$$x_2(k+1) = A_{21}x_1(k) + [A_{22}x_2(k) + B_2u(k)]$$

Untuk mendisain *reduced-order observer*, kita dapat membuat persamaan *observer* sebagai berikut :

$$\tilde{x}_2(k+1) = (A_{22} - K_e A_{12})\tilde{x}_2(k) + A_{21}x_1(k) + B_2u(k) + K_e [x_1(k+1) - A_{11}x_1(k) - B_1u(k)] \quad (2.66)$$

Persamaan (2.66) dapat dianalogikan dengan persamaan *observer* pada *full-order observer* yaitu sebagai berikut :

$$\tilde{x}(k+1) = (A - K_e C)\tilde{x}(k) + Bu(k) + K_e y(k) \quad (2.67)$$

2.3.2. Pembentukan Persamaan Dinamik *Reduced-Order Observer*

Berdasarkan persamaan (2.63), terdapat sebuah hubungan bahwa $y(k) = x_1(k)$, jika hubungan ini dimasukkan ke dalam persamaan (2.66) maka akan didapat :

$$\tilde{x}_2(k+1) = (A_{22} - K_e A_{12})\tilde{x}_2(k) + K_e y(k+1) + (A_{21} - K_e A_{11})y(k) + (B_2 - K_e B_1)u(k) \quad (2.68)$$

Persamaan (2.68) diatas masih memiliki nilai $y(k+1)$ sehingga kita harus mengukur nilai ini dan hal ini merupakan sesuatu yang menyulitkan sehingga persamaan (2.68) diatas dapat dimodifikasi menjadi berikut :

$$\begin{aligned} & \tilde{x}_2(k+1) - K_e y(k+1) \\ &= (A_{22} - K_e A_{12})\tilde{x}_2(k) + (A_{21} - K_e A_{11})y(k) + (B_2 - K_e B_1)u(k) \\ &= (A_{22} - K_e A_{12})[\tilde{x}_2(k) - K_e y(k)] + [(A_{22} - K_e A_{12})K_e + A_{21} - K_e A_{11}]y(k) + (B_2 - K_e B_1)u(k) \end{aligned} \quad (2.69)$$

definisikan bahwa

$$x_2(k) - K_e y(k) = \tilde{x}_2 - K_e x_1(k) = \eta(k) \quad (2.70)$$

dan

$$\tilde{x}_2(k) - K_e y(k) = \tilde{x}_2 - K_e x_1(k) = \tilde{\eta}(k) \quad (2.71)$$

Sehingga persamaan (2.69) dapat ditulis sebagai berikut :

$$\tilde{\eta}(k+1) = (A_{22} - K_e A_{12})\tilde{\eta}(k) + [(A_{22} - K_e A_{12})K_e + A_{21} - K_e A_{11}]y(k) + (B_2 - K_e B_1)u(k) \quad (2.72)$$

Persamaan (2.71) dan (2.72) menunjukkan sebuah persamaan dinamik *reduced-order observer* sehingga kita tidak perlu lagi harus mengukur $y(k+1)$.

Persamaan kesalahan *observer* adalah sebagai berikut :

$$e(k) = \eta(k) - \tilde{\eta}(k) = x_2(k) - \tilde{x}_2(k) \quad (2.73)$$

atau

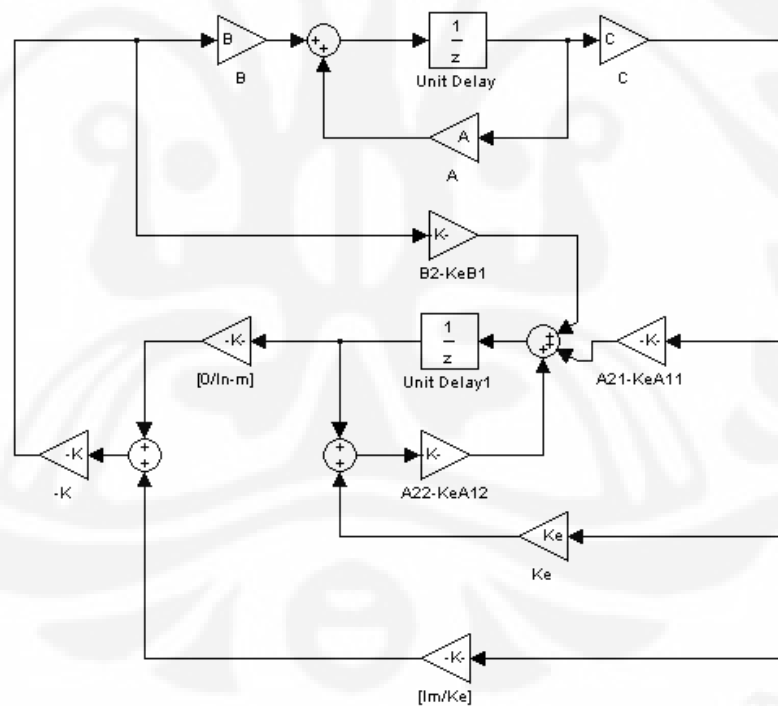
$$e(k+1) = (A_{22} - K_e A_{12})e(k) \quad (2.74)$$

Dengan menggunakan persamaan (2.74), maka didapat persamaan karakteristik dari *reduced-order observer* adalah sebagai berikut :

$$|zI - A_{22} + K_e A_{12}| = 0 \quad (2.75)$$

dengan K_e adalah matriks penguat umpan balik *observer* yang dapat dihitung dengan memilih lokasi kutub-kutub *observer* lingkaran tertutup.

Berikut ini skematik *reduced-order observer* dengan *state feedback control system* :



Gambar 2.4. Skematik *reduced-order observer* dengan *state feedback control system*

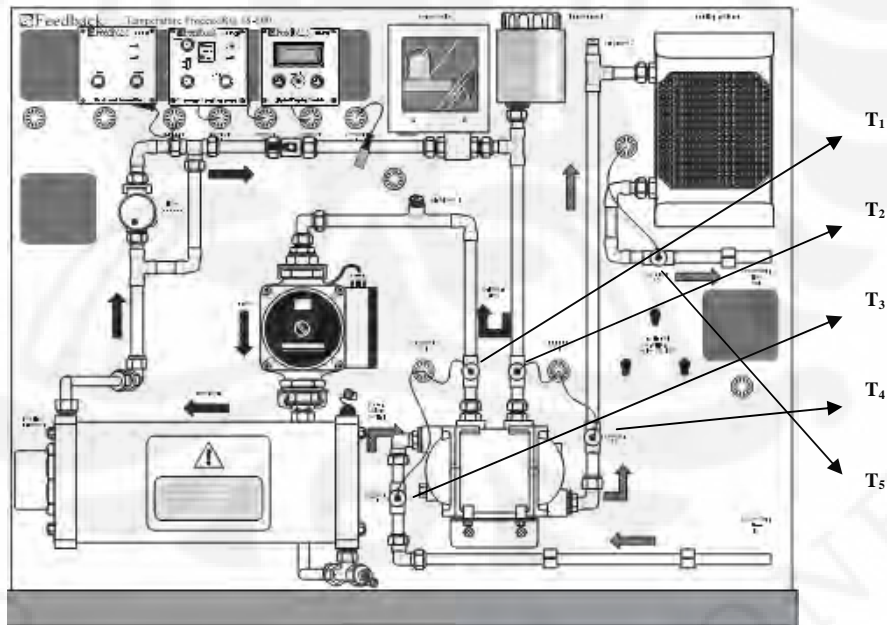
3. PERANCANGAN SISTEM

3.1. Deskripsi Proses

3.1.1. Level/Flow and Temperature Process Rig 38-003 [5]

Sistem nyata yang akan dipakai dalam skripsi ini adalah sistem *Level/Flow and Temperature Process Rig 38-003*. Sistem ini merupakan gabungan dari dua buah sistem yang dapat berdiri sendiri yaitu *Basic Process Rig 38-100* dan *Temperature Process Rig 38-600*. *Basic Process Rig 38-100* akan dipakai sebagai penyalur fluida primer ke *Temperature Process Rig 38-600*. *Temperature Process Rig 38-600* tersusun dari beberapa komponen dasar, antara lain : *Closed Primary Hot Water Circuit*, *Electrical Heater*, *Heat Exchanger*, *2 Motorized Flow Valves*, *Pulse Flow Sensor*, *5 Thermistor Temperature Sensor*, *Fan-Assisted Cooling Radiator*, *Signal Conditioning Units*.

Berikuti ini diberikan sketsa dari *Temperatur Process Rig 38-600* :



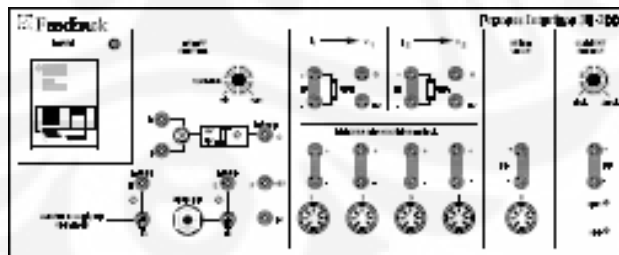
Gambar 3.1. Sketsa *Temperature Process Rig 38-600*

Berikut ini hubungan-hubungan dari kelima *Thermistor* diatas :

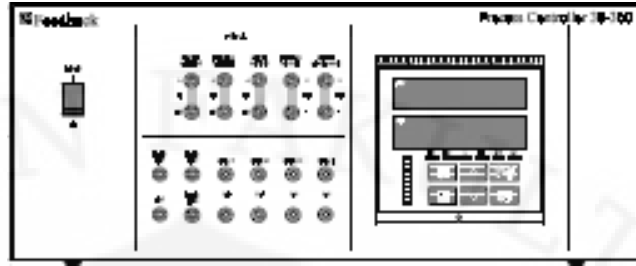
- T2 seharusnya lebih rendah dari T1, panas ditransfer dari bagian primer ke bagian sekunder.
- T3 seharusnya lebih rendah dari T4, air dingin dipanaskan didalam *Heat Exchanger*.
- T5 seharusnya lebih rendah dari T4, air yang dipanaskan sebelumnya akan didinginkan sebelum dikembalikan ke *sump*.

Selain komponen-komponen diatas, terdapat sebuah sistem yang disebut *Process Interface* yang dihubungkan ke sistem *Temperatur Process Rig 38-600*. *Process Interface* ini bertugas menyediakan semua outlet daya yang dibutuhkan oleh *Basic Process Rig 38-100*, sensor dan *Process Controller*. *Process Interface* juga memiliki input 4 – 20 mA, sebuah sumber arus 4 – 20 mA, konverter arus ke tegangan, komparator tegangan dengan *variable hysteresis*. Sistem proteksi nya disediakan oleh *residual current circuit breaker*.

Process Controller merupakan suatu sistem pengendali *ABB Kent-Taylor Commander 300* dengan standar industri. *Process Controller* juga menyediakan semua fasilitas masukan dan keluaran yang dibutuhkan untuk pengendalian *Temperature Process Rig 38-600*. *Process Controller* ini sangat compatible dengan *Process Interface* dan kedua alat ini akan memberikan suatu media konfigurasi sistem yang mudah dilakukan.



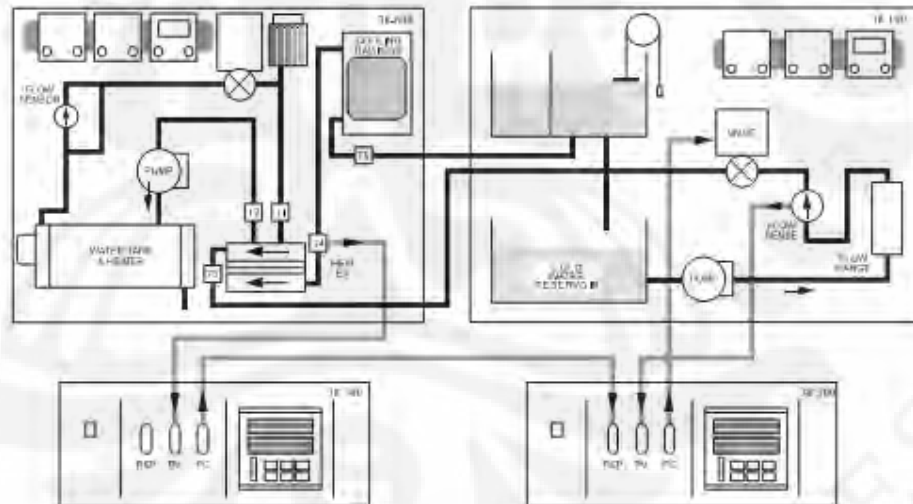
Gambar 3.2. Sketsa *Process Interface 38-200*



Gambar 3.3 Sketsa *Process Controller 38-300*

Pada skripsi ini, sumber fluida utama untuk *Temperatur Process Rig 38-600* akan disediakan secara langsung oleh *Basic Process Rig 38-100*. Oleh karena itu, *outlet Basic Process Rig 38-100* akan dihubungkan ke *inlet Temperatur Process Rig 38-600* sedangkan *outlet Temperatur Process Rig 38-600* akan dihubungkan ke *sump Basic Process Rig 38-100*.

Berikut ini adalah sketsa dari gabungan *Basic Process Rig 38-100* dan *Temperatur Process Rig 38-600* yang selanjutnya akan dinamakan dengan *Level/Flow and Temperatur Process Rig 38-003* :



Gambar 3.4. Sketsa *Level/Flow and Temperatur Process Rig 38-003*

3.1.2. Kalibrasi Komponen

Kalibrasi merupakan langkah pertama yang mutlak dilakukan pada sebuah sistem sebelum sistem tersebut dipakai. Berikut ini beberapa

kalibrasi yang harus dilakukan pada sistem *Level/Flow and Temperature Process Rig 38-003*.

Kalibrasi Servo Valve

Untuk melakukan kalibrasi *servo valve*, buka MV2 pada kondisi terbuka penuh dan MV3 pada kondisi setengah terbuka. Nyalakan *Process Interface* dan pompa kemudian naikkan keluaran sumber arus pada *Process Interface* dari nilai minimal ke maksimal secara perlahan.

Kalibrasi Temperature Sensor Pack 38-440

Untuk melakukan kalibrasi *Temperature Sensor Pack 38-440*, nyalakan *Digital Display Module 38-490* dan ubah dalam tampilan %. Ubah *Thermistor Temperature Transmitter 38-441* ke sensor A. Tekan tombol kalibrasi 25°C dan sesuaikan baut nol (*zero screw*) untuk mengubah tampilan 38-490 menjadi 25. Tekan tombol kalibrasi 80°C dan sesuaikan baut *span (span screw)* untuk menampilkan 80 pada 38-441. Tampilan tersebut kini menunjukkan suhu dalam satuan derajat *celcius*.

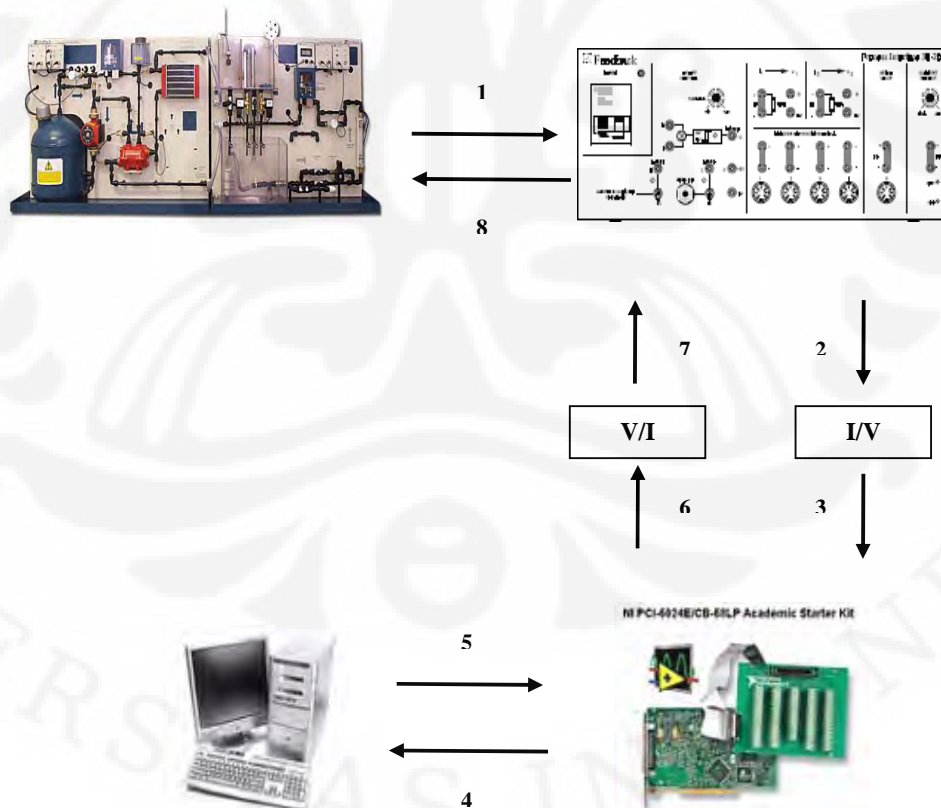
3.2. Interkoneksi Alat

Pengambilan data masukan dan data keluaran dilakukan dengan menggunakan blok *Simulink* yang tersedia dalam *Matlab*. Oleh karena itu, diperlukan adanya interkoneksi antara *Level/Flow and Temperature Process Rig 38-003* dengan komputer yang digunakan. Interkoneksi antara *Level/Flow and Temperature Process Rig 38-003* dengan komputer ini akan dijelaskan berdasarkan urutan dari keluarnya sinyal keluaran proses sampai dengan pengendalian *servo valve* oleh sinyal kendali yang dibangkitkan oleh komputer dan juga urutan sebaliknya yaitu hasil pengendalian *servo valve* untuk dibaca di komputer, seperti yang terlihat pada gambar 3.5.

Sinyal yang diterima dan dihasilkan oleh *Level/Flow and Temperature Process Rig 38-003* adalah sinyal arus listrik *analog*. Oleh

sebab itu, untuk menghubungkan sistem ini dengan komputer yang hanya dapat menerima dan menghasilkan sinyal tegangan *digital*, maka diperlukan adanya rangkaian *V/I* dan *I/V converter* serta ADC dan DAC agar aliran data dapat terjadi.

Keluaran yang dihasilkan sistem (1) dalam bentuk sinyal arus listrik *analog* diterima oleh *Process Interface*, lalu sinyal ini diubah menjadi sinyal tegangan (2 dan 3) oleh *I/V converter* yang ada didalam *Process Interface*. Sinyal keluaran sistem ini lalu diubah oleh ADC menjadi sinyal *digital* sehingga dapat diolah oleh pengendali yang telah dirancang pada komputer. Proses pengendalian ini kemudian menghasilkan sinyal kendali, yang kemudian diubah menjadi sinyal arus listrik *analog* oleh DAC dan *V/I converter* (6 dan 7) sebelum masuk ke *Process Interface*. Sinyal kendali dalam bentuk arus listrik *analog* ini kemudian diteruskan oleh *Proses Interface* untuk menjadi masukan bagi *servo valve*.



Gambar 3.5. Ilustrasi arah aliran sinyal pada sistem

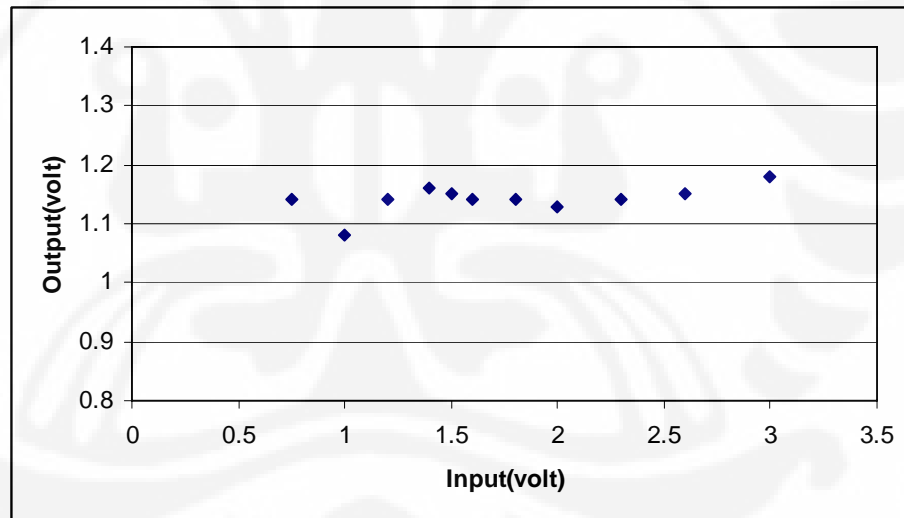
3.3. Pembuatan Model *Level/Flow and Temperature Process Rig 38-003*

3.3.1. Penentuan Daerah Kerja

Setiap sistem dalam dunia nyata merupakan sistem yang memiliki sifat *nonlinear* seperti halnya *Level/Flow and Temperature Process Rig 38-003*. Oleh karena keperluan ini, harus dilakukan penentuan daerah kerja dari *Level/Flow and Temperature Process Rig 38-003* yaitu daerah dimana tanggapan suatu sistem dianggap linier terhadap masukannya. Dengan adanya daerah kerja ini, diasumsikan pula bahwa sistem hanya beroperasi di sekitar daerah kerja ini saja.

Dalam mendapatkan daerah kerja linier dari *Level/Flow and Temperature Process Rig 38-003*, maka dilakukan serangkaian uji coba yaitu dengan melakukan uji coba sistem *open loop* dengan memberikan masukan *step* yang bervariasi dan mencatat hasil keluaran sistem yaitu pada sensor T4.

Berikut ini diberikan grafik hasil percobaan *open loop* yang dilakukan :



Gambar 3.6. Grafik karakteristik *Level/Flow and Temperature Process Rig 38-003* : Tanggapan Masukan vs Keluaran

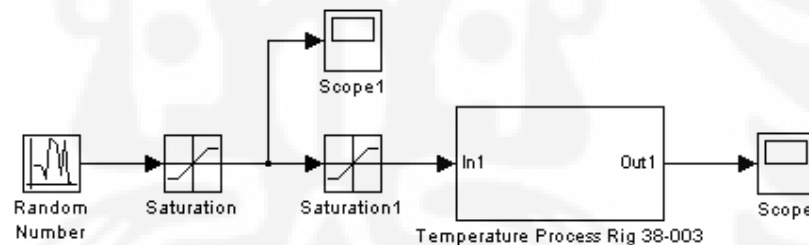
Dari grafik ini dapat dilihat bahwa temperatur pada sensor T4 bersifat tidak linier untuk masukan-masukan dengan nilai 0.75 volt sampai 2 volt. Sedangkan untuk nilai masukan 2 volt sampai 3 volt, sistem menunjukkan sifat linier. Sehingga dapat dikatakan bahwa sistem akan

bersifat linier untuk nilai masukan 2 volt sampai 3 volt. Kecenderungan ini sesuai dengan logika bahwa semakin banyak air panas yang masuk ke *Heat Exchanger* maka temperatur T4 akan semakin meningkat. Daerah kerja *Level/Flow and Temperature Process Rig 38-100* dapat dilihat berada pada daerah masukan 2.3 volt.

3.3.2. Identifikasi Model Sistem

Untuk mencari model proses, digunakan metode Identifikasi dengan Kuadrat Terkecil. Model proses ini akan digunakan sebagai dasar dalam perancangan *reduced-order observer*. *Reduced-order Observer* ini nantinya akan disertakan dalam proses identifikasi model sistem dengan vektor kesalahan masukan. Model sistem dengan vektor kompensasi nilai masukan inilah yang akan dipakai sebagai basis dalam algoritma pengendali MPC.

Berikut ini adalah blok simulink proses identifikasi model proses :



Gambar 3.7. Blok simulink untuk proses identifikasi

Pada proses identifikasi dengan metode kuadrat terkecil, data masukan dan data keluaran akan direkam untuk menentukan parameter-parameter model sistem yang dirumuskan sebagai berikut :

$$(1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4})y(t) = (b_1 z^{-1} + b_2 z^{-2})u(t) + e(t) \quad (3.1)$$

$$\text{atau} \quad \frac{y(t)}{u(t)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}} \quad (3.2)$$

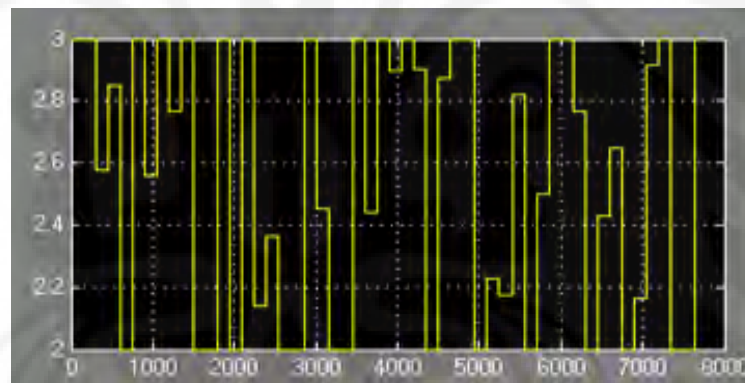
Persamaan (3.1) juga dapat dimodifikasi menjadi persamaan dibawah ini dengan nilai awal pada pencuplikan ke 4 :

$$\underbrace{\begin{bmatrix} \hat{y}(4) \\ \hat{y}(5) \\ \vdots \\ \hat{y}(k) \end{bmatrix}}_{\hat{\underline{y}}} = \underbrace{\begin{bmatrix} -y(3) & -y(2) & -y(1) & -y(0) & u(3) & u(2) \\ -y(4) & -y(3) & -y(2) & -y(1) & u(4) & u(3) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(k-1) & -y(k-2) & -y(k-3) & -y(k-4) & u(k-1) & u(k-2) \end{bmatrix}}_{\underline{x}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ b_1 \\ b_2 \end{bmatrix}}_{\hat{\underline{\theta}}} \quad (3.3)$$

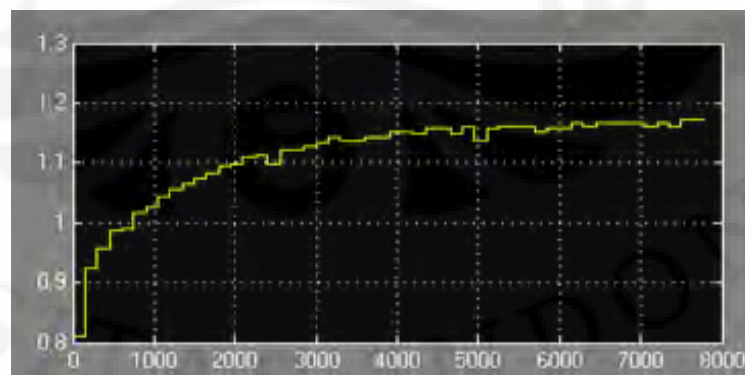
Untuk mencari parameter $\hat{\theta}$, dapat digunakan rumusan sebagai berikut :

$$\hat{\theta} = (\underline{x}^T \underline{x})^{-1} (\underline{x}^T \hat{\underline{y}}) \quad (3.4)$$

Pada percobaan, masukan yang digunakan adalah berupa *Random Number* dengan nilai rata-rata \bar{v} 2,5 dan variansi σ_v^2 bernilai 1, sedangkan nilai *sampling time* h yang digunakan adalah 150 detik. Data masukan dan keluaran terlihat pada gambar 3.8.



(a)



(b)

Gambar 3.8. (a) Grafik sinyal masukan. (b) Grafik dari *Thermistor* T4

Berdasarkan data masukan dan keluaran yang direkam, maka dengan menggunakan persamaan (3.3) dan (3.4) didapat parameter model sebagai berikut :

$$\hat{\theta} = \begin{bmatrix} -0.6644 \\ -0.5595 \\ 0.0915 \\ 0.1358 \\ 0.0008 \\ 0.0019 \end{bmatrix}$$

Setelah parameter $\hat{\theta}$ didapat, maka dengan menggunakan persamaan (3.1) didapat model fungsi alih *Level/Flow and Temperature Process Rig 38-003* seperti berikut :

$$\frac{Y(z)}{U(z)} = \frac{0.0008z^{-1} + 0.0019z^{-2}}{1 - 0.6644z^{-1} - 0.5595z^{-2} + 0.0915z^{-3} + 0.1358z^{-4}} \quad (3.5)$$

Bentuk model fungsi alih pada persamaan (3.5) dapat diubah kedalam persamaan ruang keadaan yaitu sebagai berikut :

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 0.6644 & 1 & 0 & 0 \\ 0.5595 & 0 & 1 & 0 \\ -0.0915 & 0 & 0 & 1 \\ -0.1358 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0.0008 \\ 0.0019 \\ 0 \\ 0 \end{bmatrix} u(k) \quad (3.6)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}$$

Untuk melakukan pengetesan kedekatan antara model dengan proses, maka dibuat blok simulink model dan diberikan masukan yang sama dengan proses mencari daerah kerja sistem dan hasilnya cukup mendekati kondisi yang sebenarnya.

3.3.3. Perancangan *Reduced-Order State Observer*

3.3.3.1. Pengetesan *Observability* Sistem

Ada satu langkah penting yang harus dilakukan terlebih dahulu sebelum merancang sebuah *observer* yaitu pengetesan kondisi *observability* dari sebuah sistem. Pengecekan *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.

Asumsikan bahwa model ruang keadaan sistem *Level/Flow and Temperature Process Rig 38-003* yang ditunjukkan pada persamaan (3.6) dapat diwakili oleh persamaan berikut :

$$x((k+1)T) = Ax(kT) + Bu(kT) \quad (3.7)$$

$$y(kT) = Cx(kT) \quad (3.8)$$

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

$$\begin{bmatrix} C^T : A^T C^T : \dots : (A^T)^{n-1} C^T \end{bmatrix} \quad (3.9)$$

dimana :

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

Berdasarkan persamaan (3.9) maka matriks *observability* dari *Level/Flow and Temperature Process Rig 38-003* adalah sebagai berikut :

$$\begin{bmatrix} 1 & 0.6644 & 1 & 0.9452 \\ 0 & 1 & 0.6644 & 1.0009 \\ 0 & 0 & 1 & 0.6644 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rank dari matriks *observability* diatas adalah 4 sehingga sistem *Level/Flow and Temperature Process Rig 38-003* dikatakan *fully observable* atau dengan kata lain semua *state* dari sistem dapat diobservasi.

3.3.3.2. Pembentukan Persamaan Karakteristik *Observer*

Setelah identifikasi model sistem, maka didapat model seperti pada persamaan (3.6). Untuk merancang *reduced-order observer*, maka persamaan (3.6) harus dipartisi sebagaimana yang dilakukan dalam persamaan (2.62) dan (2.63) dimana *state vector* yang dapat diukur adalah $x_{11}(k)$ dan *state vector* yang tidak dapat diukur adalah $x_{21}(k), x_{22}(k),$ dan $x_{23}(k)$.

Sehingga diperoleh :

$$x_1(k) = x_{11}(k) \text{ dan } x_2(k) = \begin{bmatrix} x_{21}(k) \\ x_{22}(k) \\ x_{23}(k) \end{bmatrix}$$

$$A_{11} = [0.6644], A_{12} = [1 \ 0 \ 0], A_{21} = \begin{bmatrix} 0.5595 \\ -0.0915 \\ -0.1358 \end{bmatrix}, A_{22} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_1 = [0.0008], B_2 = \begin{bmatrix} 0.0019 \\ 0 \\ 0 \end{bmatrix}$$

Parameter K_e (matriks penguat umpan balik *observer*) dapat diperoleh dengan menggunakan persamaan (2.75) dengan letak kutub-kutub *observer* lingkaran tertutup yang diinginkan berada pada titik :

$$\text{desired } eig_1 = 0.4$$

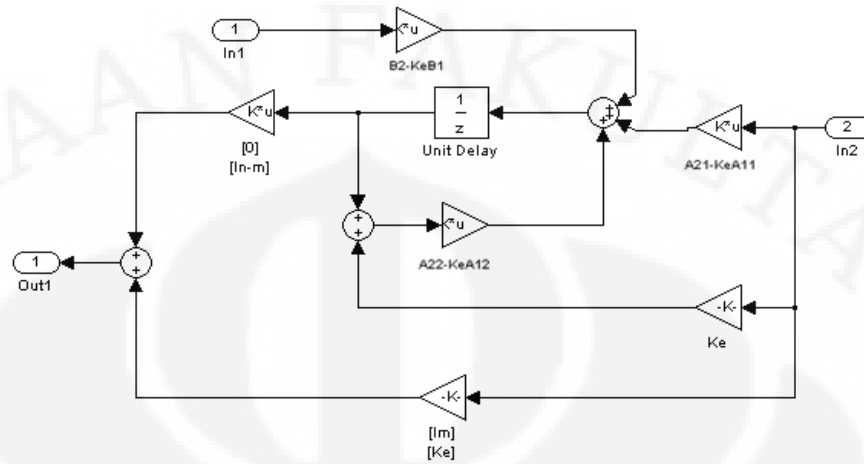
$$\text{desired } eig_{2,3} = -0.3 \pm 0.2175i$$

Berikut ini adalah persamaan karakteristik *observer* yang akan dirancang :

$$|zI - A_{22} + K_e A_{12}| = (z - 0.4)(z + 0.3 + 0.2175i)(z + 0.3 - 0.2175i)$$

$$\text{didapat nilai } K_e = \begin{bmatrix} 0.2 \\ -0.1027 \\ -0.0549 \end{bmatrix}$$

Berikut ini adalah blok diagram *reduced-order observer* :

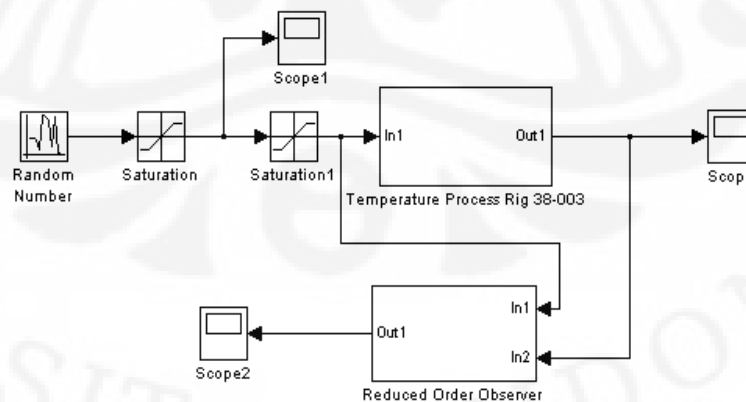


Gambar 3.9. Blok diagram *reduced-order observer*

3.3.4. Identifikasi Model Sistem dengan Vektor Kompensasi Nilai Masukan

Model yang didapat pada bagian 3.2.2. kurang baik bila dipakai dalam algoritma pengendali MPC. Hal ini dikarenakan tidak adanya faktor kompensasi kesalahan masukan yang sangat diperlukan bila kita menghadapi sistem yang tidak linier. Oleh karena itu, proses identifikasi model sistem akan kembali dilakukan. Proses identifikasi model sistem ini akan melibatkan *reduced-order observer* karena algoritma identifikasi ini memerlukan semua data *state vector* yang dimiliki sistem.

Berikut ini adalah blok simulink untuk proses identifikasi :



Gambar 3.10. Blok simulink proses identifikasi model dengan vektor kesalahan masukan

Pada saat pengambilan data, terdapat satu buah masukan dan dua buah variabel keadaan yang dicatat sebagai informasi untuk menentukan nilai parameter-parameter estimasi model ruang keadaan dari alat tersebut.

Model ruang keadaan sistem adalah sebagai berikut.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} u(k) + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix} \quad (3.10)$$

atau

$$\underbrace{\begin{bmatrix} x_1(1) & x_2(1) & x_3(1) & x_4(1) \\ x_1(2) & x_2(2) & x_3(2) & x_4(2) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(k+1) & x_2(k+1) & x_3(k+1) & x_4(k+1) \end{bmatrix}}_{\underline{X}} = \underbrace{\begin{bmatrix} x_1(0) & x_2(0) & x_3(0) & x_4(0) & u(0) & 1 \\ x_1(1) & x_2(1) & x_3(1) & x_4(1) & u(1) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1(k) & x_2(k) & x_3(k) & x_4(k) & u(k) & 1 \end{bmatrix}}_{\underline{P}} \underbrace{\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \\ b_1 & b_2 & b_3 & b_4 \\ k_1 & k_2 & k_3 & k_4 \end{bmatrix}}_{\underline{\hat{\theta}}} \quad (3.11)$$

Dari persamaan (3.11), maka dapat diturunkan rumus untuk menghitung nilai parameter-parameter estimasi $\hat{\underline{\theta}}$. Langkah-langkah untuk menghitung nilai parameter estimasi $\hat{\underline{\theta}}$ adalah sebagai berikut

1. Memodifikasi fungsi kriteria menjadi

$$\begin{aligned} J_{LS} &= (\underline{X} - \underline{P}\hat{\underline{\theta}})^T (\underline{X} - \underline{P}\hat{\underline{\theta}}) \\ &= \underline{X}^T \underline{X} - \hat{\underline{\theta}}^T \underline{P}^T \underline{X} - \underline{X}^T \underline{P} \hat{\underline{\theta}} + \hat{\underline{\theta}}^T \underline{P}^T \underline{P} \hat{\underline{\theta}} \end{aligned} \quad (3.12)$$

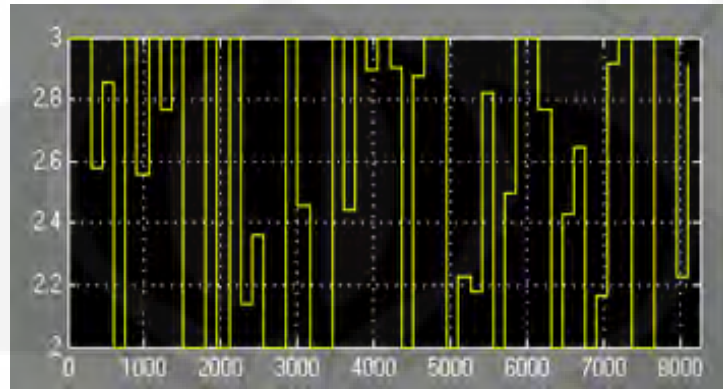
2. Dengan membuat turunan pertama dari J_{LS} terhadap $\underline{\theta}$ bernilai nol, maka didapatkan persamaan

$$\left. \frac{\partial J_{LS}(\underline{\theta})}{\partial \underline{\theta}} \right|_{\underline{\theta}=\hat{\underline{\theta}}} = -2\underline{P}^T \underline{X} + 2\underline{P}^T \underline{P} \hat{\underline{\theta}} = 0 \quad (3.13)$$

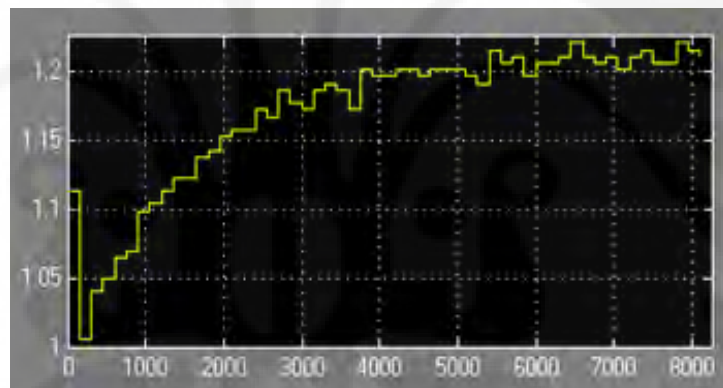
3. Dari persamaan (3.13), maka didapat rumus untuk menghitung nilai parameter estimasi

$$\hat{\underline{\theta}} = (\underline{P}^T \underline{P})^{-1} \underline{P}^T \underline{X} \quad (3.14)$$

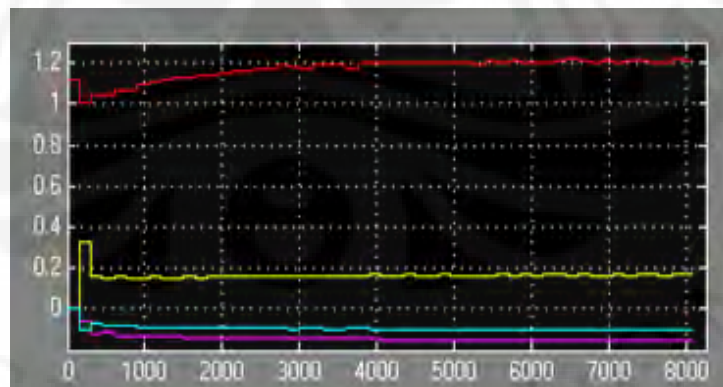
Pada percobaan, masukan yang digunakan adalah berupa *Random Number* dengan nilai rata-rata \bar{v} 2,5 dan variansi σ_v^2 bernilai 1, sedangkan nilai *sampling time* h yang digunakan adalah 150 detik. Data masukan dan keluaran terlihat pada gambar 3.11.



(a)



(b)



(c)

Gambar 3.11. (a) Grafik sinyal masukan (b). Grafik dari *Thermistor* T4
(c). Grafik dari *reduced-order observer*

Berdasarkan data masukan dan keluaran yang direkam, maka dengan menggunakan persamaan (3.14) didapat parameter model sebagai berikut :

$$\hat{\theta} = \begin{bmatrix} 0.7549 & 0.2839 & -0.0576 & -0.0883 \\ 0.0107 & -0.2 & 0.1027 & 0.0549 \\ -0.2171 & 1 & 2.558e-13 & 1.8474e-13 \\ -0.9994 & 1.59e-12 & 1 & -9.0949e-13 \\ 0.0003 & 0.0017 & 0.0001 & 0 \\ 0.1633 & -2.84e-14 & 238422e-14 & 0 \end{bmatrix}$$

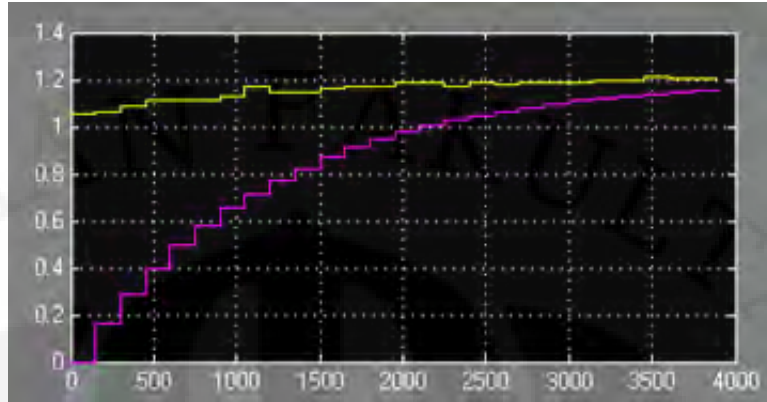
Setelah parameter $\hat{\theta}$ diketahui, maka didapatkan persamaan model ruang keadaan linier dari *Level/Flow and Temperature Process Rig 38-003* sebagai berikut :

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 0.7549 & 0.0107 & -0.2171 & -0.9994 \\ 0.2839 & -0.2 & 1 & 0 \\ -0.0576 & 0.1027 & 0 & 1 \\ -0.0883 & 0.0549 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0.0003 \\ 0.0017 \\ 0.0001 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0.1633 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

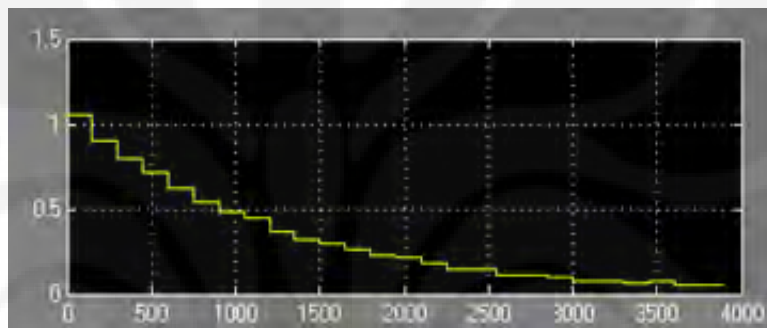
dengan vektor K adalah vektor kompensasi nilai masukan.

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}$$

Grafik keluaran proses dan keluaran model dapat dilihat pada gambar 3.12 sedangkan grafik selisih keluaran proses dan keluaran model pada tangki kedua dapat dilihat pada gambar 3.13.



Gambar 3.12. Grafik keluaran proses vs keluaran model



Gambar 3.13. Selisih keluaran model dengan keluaran proses

Dari gambar 3.12 dan gambar 3.13 terlihat bahwa model yang digunakan sudah cukup baik karena selisih antara keluaran proses dan keluaran model relatif kecil. Berdasarkan hasil estimasi, didapatkan nilai *lost function* (J_{LS})

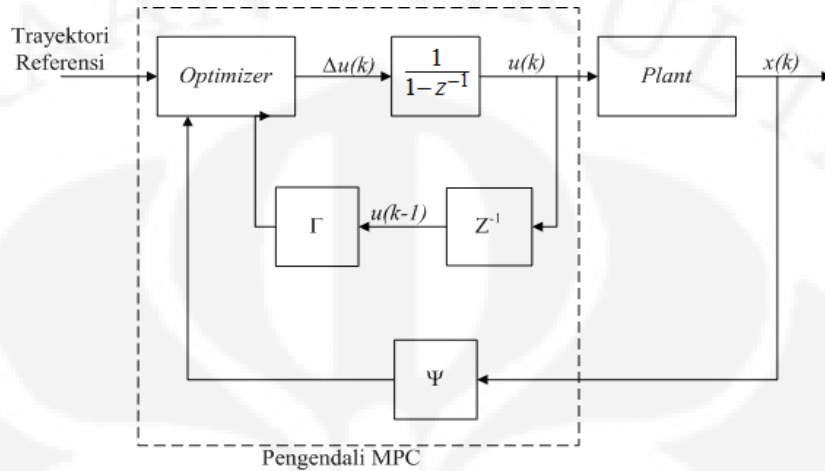
$$J_{LS} = \frac{1}{N} \sum_{i=1}^N (y_2 - \hat{y}_2)^2 = 0,177597$$

Dari perhitungan ternyata didapatkan nilai J_{LS} yang cukup kecil, hal ini membuktikan bahwa model yang akan digunakan sudah cukup baik.

3.4. Algoritma *Model Predictive Control* dengan *Constraints*

Struktur pengendali MPC dengan *constraint* untuk model ruang keadaan terdapat pada gambar 3.14. Dari blok diagram tersebut, terlihat bahwa prediksi perubahan sinyal masukan sekarang ($\Delta u(k)$) membutuhkan

data dari variabel keadaan sekarang $\underline{x}(k)$ dan masukan satu langkah sebelumnya $u(k-1)$.



Gambar 3.14. Blok diagram pengendali MPC dengan *constraints*.

Algoritma perhitungan perubahan sinyal kendali pada MPC dengan *constraints* adalah sebagai berikut :

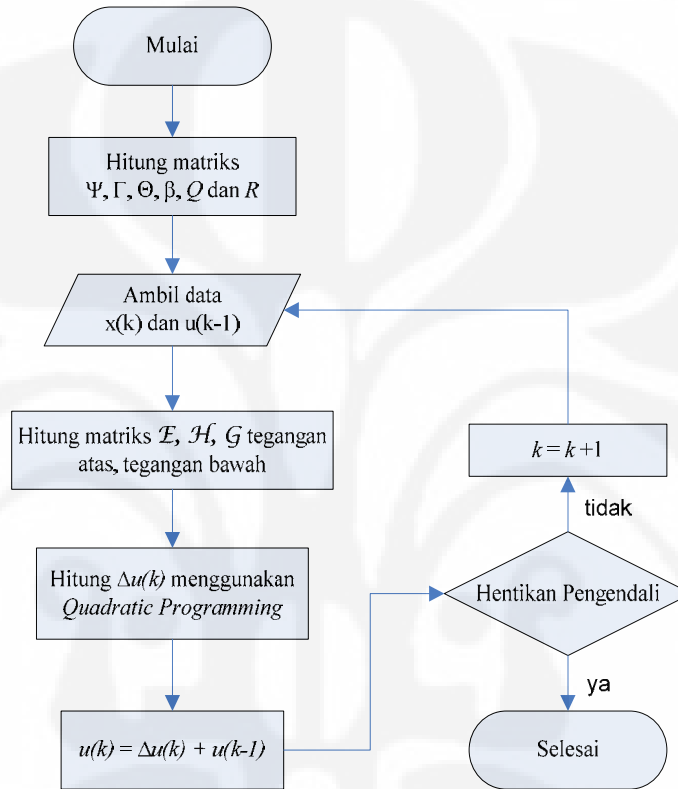
1. Parameter pengendali yang terlebih dahulu harus ditentukan antara lain horizon prediksi (H_p), horizon kendali (H_u), matriks faktor bobot kesalahan (Q), dan matriks faktor bobot perubahan sinyal kendali (R).
2. Matriks \underline{F} dihitung dengan menggunakan persamaan (2.33), serta matriks \underline{H} dan \underline{G} yang terdapat pada fungsi kriteria persamaan (2.37) dihitung masing-masing dengan menggunakan persamaan (2.39) dan (2.38).
3. Parameter batasan (*constraints*) fisik sistem diubah ke dalam bentuk pertidaksamaan yang memiliki hubungan dengan perubahan sinyal kendali (ΔU).

$$\underline{\Omega} \Delta U(k) \leq \underline{\omega} \quad (3.15)$$

4. Menghitung perubahan sinyal kendali optimal Δu_{opt} dengan menggunakan metode *Quadratic Programming*.
5. Menghitung sinyal kendali $u(k)$ dimana

$$u(k) = \Delta u(k) + u(k-1) \quad (3.16)$$

Diagram alir untuk perhitungan sinyal kendali dengan menggunakan MPC dengan *constraints* adalah seperti pada gambar 3.15.



Gambar3.15. Diagram alir algoritma MPC dengan *constraints*.

Metode yang digunakan pada *Quadratic Programming* dalam menghitung nilai ΔU adalah *Active Set* dengan alur operasi seperti dijelaskan berikut ini [2].

1. Fungsi kriteria pada persamaan (2.37), diubah menjadi seperti berikut

$$V(\underline{\Delta U}(k)) = \frac{1}{2} \underline{\Delta U}^T(k) 2\underline{\mathcal{H}} \underline{\Delta U}(k) - \underline{\Delta U}^T(k) \underline{\mathcal{G}} \quad (3.17)$$

berdasarkan *constraints*

$$\underline{\Omega} \underline{\Delta U}(k) \leq \underline{\omega} \quad (3.18)$$

2. Nilai $\underline{\Delta U}_r$ dipilih sedemikian sehingga pertidaksamaan *constraints* (3.18) menjadi sebuah persamaan seperti berikut

$$\underline{\Omega}_r \underline{\Delta U}_r = \underline{\omega}_r \quad (3.19)$$

Elemen yang membuat pertidaksamaan menjadi persamaan disebut elemen aktif .

3. Menghitung nilai \underline{d} yang merupakan pergerakan $\underline{\Delta U}_r$ dalam meminimasi fungsi kriteria sehingga fungsi kriteria pada persamaan (3.17) berubah menjadi

$$\begin{aligned} V(\underline{\Delta U}_r + \underline{d}) &= \frac{1}{2} (\underline{\Delta U}_r + \underline{d})^T 2\underline{\mathcal{H}}(\underline{\Delta U}_r + \underline{d}) - (\underline{\Delta U}_r + \underline{d})^T \underline{\mathcal{G}} \\ &= \frac{1}{2} \underline{d}^T \underbrace{2\underline{\mathcal{H}}}_{\underline{\Phi}} \underline{d} + \underline{d}^T \underbrace{(2\underline{\mathcal{H}} \underline{\Delta U}_r - \underline{\mathcal{G}})}_{\underline{\phi}_r} + V(\underline{\Delta U}_r) \end{aligned} \quad (3.20)$$

Nilai \underline{d} tidak boleh mempengaruhi pertidaksamaan *constraints* (3.18), sehingga persamaan *constraints* untuk persamaan (3.20) adalah

$$\underline{\Omega}_r \underline{d} = \underline{0} \quad (3.21)$$

4. Dari persamaan (3.20) dan (3.21), nilai optimal \underline{d} sepanjang *constraints* yang aktif dapat dihitung dengan menyelesaikan fungsi kuadrat berikut

$$\min \frac{1}{2} \underline{d}^T \underline{\Phi} \underline{d} + \underline{d}^T \underline{\phi}_r \quad (3.22)$$

dengan *constraints*

$$\underline{\Omega}_r \underline{d} = \underline{0} \quad (3.23)$$

Nilai pengali *Lagrange* λ_r untuk persamaan (3.22) dan (3.23) dihitung berdasarkan kondisi *Karush-Kuhn-Tucker* (KKT) seperti berikut

$$\begin{bmatrix} \underline{\Phi} & \underline{\Omega}^T \\ \underline{\Omega} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{d} \\ \underline{\lambda}_r \end{bmatrix} = \begin{bmatrix} -\underline{\phi}_r \\ \underline{0} \end{bmatrix} \quad (3.24)$$

dimana nilai $\underline{\Delta U}_r$ yang terdapat pada matriks $\underline{\phi}_r$ ditentukan pada langkah (2). Hasil perhitungan \underline{d} dan $\underline{\lambda}_r$ akan mempengaruhi tahapan berikutnya, yaitu :

- a. Jika semua $\lambda_r > 0$ dan $\underline{d} = 0$, maka proses komputasi selesai dan nilai $\underline{\Delta U}_r$ merupakan nilai optimal untuk $\underline{\Delta U}(k)$.
 - b. Jika semua $\lambda_r > 0$ dan ada nilai $d \neq 0$, maka lanjut ke langkah (5).
 - c. Jika ada nilai $\lambda_r < 0$, maka *constraint* yang memiliki nilai λ_r paling negatif dibuang, kemudian lanjut ke langkah (5).
5. Nilai faktor koreksi pergerakan nilai optimal α_r dihitung dengan menggunakan rumus

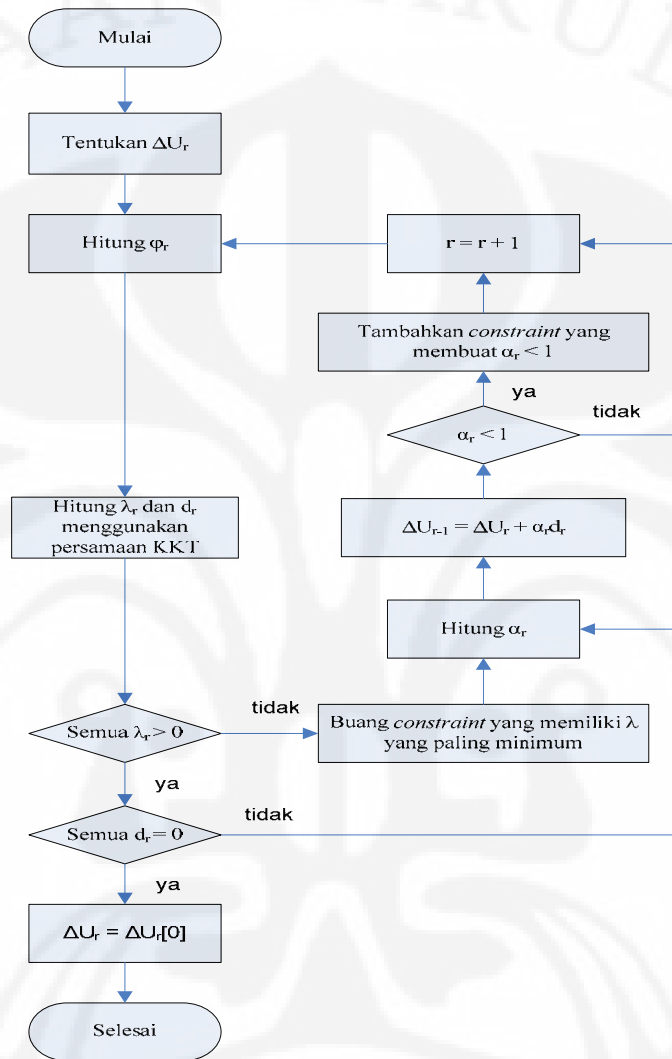
$$\alpha_r = \min \left(1, \min_{\substack{i \in \underline{\Omega}_r \\ a_i d > 0}} \frac{b_i - a_i \underline{\Delta U}_r}{a_i d} \right) \quad (3.25)$$

dengan a_i adalah baris dari pertidaksamaan batasan yang tidak aktif dan b_i adalah batasannya. Selanjutnya, nilai $\underline{\Delta U}_r$ dalam arah \underline{d} dihitung sebagai berikut

$$\underline{\Delta U}_{r+1} = \underline{\Delta U}_r + \alpha_r \underline{d} \quad (3.26)$$

6. Jika nilai $\alpha_r < 1$, maka *constraint* yang membuat nilai $\alpha_r < 1$ ditambahkan ke $\underline{\Omega}_r$.
7. Tetapkan $r = r + 1$ dan kembali ke langkah (3) untuk proses iterasi berikutnya.

Diagram alir metode *Active Set* untuk menyelesaikan *Quadratic Programming* seperti yang terdapat pada gambar 3.16.



Gambar 3.16. Diagram alir metode *Active Set* untuk menyelesaikan *Quadratic Programming*.

3.5. Perhitungan Sinyal Kendali

Berikut ini adalah contoh perhitungan sinyal kendali dengan metode MPC dengan *constraints*. Spesifikasi pengendali yang digunakan pada pengendali MPC berikut ini adalah sebagai berikut :

- Nilai *control horizon* $H_u = 2$

- Nilai *prediction horizon* $H_p = 10$
- Faktor bobot kesalahan $\underline{Q} = \underline{I}_{H_p}$
- Faktor bobot perubahan sinyal kendali $\underline{R} = 0,5\underline{I}_{H_u}$
- Trayektori acuan $r(k) = 1.0$
- Matriks variabel keadaan

$$\underline{A} = \begin{bmatrix} 0.7549 & 0.0107 & -0.2171 & -0.9994 \\ 0.2839 & -0.2 & 1 & 0 \\ -0.0576 & 0.1027 & 0 & 1 \\ -0.0883 & 0.0549 & 0 & 0 \end{bmatrix}, \underline{B} = \begin{bmatrix} 0.0003 \\ 0.0017 \\ 0.0001 \\ 0 \end{bmatrix}, \text{ dan } \underline{K} = \begin{bmatrix} 0.1633 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Karena pada persamaan ruang keadaan terdapat faktor kompensasi untuk nilai masukan, yakni vektor \underline{K} , maka perhitungan nilai prediksi variabel keadaan pada persamaan (2.16) berubah menjadi

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \vdots \\ \hat{x}(k+H_u|k) \\ \vdots \\ \hat{x}(k+H_p|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^{H_u} \\ \vdots \\ \underline{A}^{H_p} \end{bmatrix}}_{\underline{\Psi}} \underline{x}(k) + \underbrace{\begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^{H_u-1} \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^{H_p-1} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\Gamma}} \underline{u}(k-1) +$$

$$\underbrace{\begin{bmatrix} \underline{B} & \cdots & \underline{0}_{n \times 1} \\ \underline{A}\underline{B} + \underline{B} & \ddots & \vdots \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{H_p-1} \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^{H_p-H_u} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\Theta}} \begin{bmatrix} \Delta \hat{u}(k) \\ \vdots \\ \Delta \hat{u}(k+H_u-1) \end{bmatrix} + \underbrace{\begin{bmatrix} \underline{K} \\ \vdots \\ \sum_{i=0}^{H_u-1} \underline{A}^i \underline{K} \\ \vdots \\ \sum_{i=0}^{H_p-1} \underline{A}^i \underline{K} \end{bmatrix}}_{\underline{\beta}}$$

(3.27)

Contoh dari setiap tahap untuk menghitung sinyal kendali dengan algoritma *Model Predictive Control* dengan *constraints* adalah seperti berikut

1. Matriks $\underline{C}_y \underline{\Psi}$, $\underline{C}_y \underline{\Gamma}$, $\underline{C}_y \underline{\Theta}$, dan $\underline{C}_y \underline{\beta}$ dihitung dengan menggunakan persamaan (3.27)

$$\underline{C}_y \underline{\Psi} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^2 \\ \vdots \\ \underline{A}^{10} \end{bmatrix} = \begin{bmatrix} 0.7549 & 0.0107 & -0.2171 & -0.9994 \\ 0.6737 & -0.0712 & -0.1532 & -0.9715 \\ 0.5829 & -0.0476 & -0.2175 & -0.8264 \\ 0.5120 & -0.0519 & -0.1742 & -0.8001 \\ 0.4525 & -0.0459 & -0.1631 & -0.6859 \\ 0.3985 & -0.0404 & -0.1442 & -0.6153 \\ 0.3520 & -0.0362 & -0.1269 & -0.5424 \\ 0.3106 & -0.0318 & -0.1127 & -0.4787 \\ 0.2742 & -0.0282 & -0.0992 & -0.4231 \\ 0.2421 & -0.0249 & -0.0877 & -0.3733 \end{bmatrix}$$

$$\underline{C}_y \underline{\Gamma} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^1 \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^9 \underline{A}^i \underline{B} \end{bmatrix} = \begin{bmatrix} 0.0003 \\ 0.0002229 \\ 0.0000657 \\ 0.0000722 \\ 0.0000479 \\ 0.0000413 \\ 0.0000365 \\ 0.0000313 \\ 0.0000279 \\ 0.0000245 \end{bmatrix}$$

$$\underline{C}_y \underline{\Theta} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{B} & \cdots & \underline{0}_{2 \times 1} \\ \underline{AB+B} & \ddots & \vdots \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^9 \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^8 \underline{A}^i \underline{B} \end{bmatrix} = \begin{bmatrix} 0.0003 & 0 \\ 0.0002229 & 0.0003 \\ 0.0000657 & 0.0002229 \\ 0.0000722 & 0.0000657 \\ 0.0000479 & 0.0000722 \\ 0.0000413 & 0.0000479 \\ 0.0000365 & 0.0000413 \\ 0.0000313 & 0.0000365 \\ 0.0000279 & 0.0000313 \\ 0.0000245 & 0.0000279 \end{bmatrix}$$

$$\underline{C}_y \underline{\beta} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{K} \\ \vdots \\ \sum_{i=0}^1 \underline{A}^i \underline{K} \\ \vdots \\ \sum_{i=0}^9 \underline{A}^i \underline{K} \end{bmatrix} = \begin{bmatrix} 0.1633 \\ 0.1233 \\ 0.1100 \\ 0.0952 \\ 0.0836 \\ 0.0739 \\ 0.0651 \\ 0.0575 \\ 0.0507 \\ 0.0448 \end{bmatrix}$$

2. Batasan sinyal kendali dengan nilai *control horizon* sama dengan dua dapat dinyatakan dalam bentuk pertidaksamaan seperti berikut

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} 1.0 \leq \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} 3.0 \quad (3.28)$$

Pertidaksamaan (3.28) harus diubah kedalam bentuk pertidaksamaan yang mengandung $\Delta u(k)$, dimana

$$u(k) = \Delta u(k) + u(k-1) \text{ dan}$$

$$u(k+1) = \Delta u(k+1) + \Delta u(k) + u(k-1)$$

Sehingga didapatkan hasil transformasi pertidaksamaan (3.28) sebagai berikut

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} 1.0 \leq \begin{bmatrix} \Delta u(k) \\ \Delta u(k) + \Delta u(k+1) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k-1) \quad (3.29)$$

$$\begin{bmatrix} \Delta u(k) \\ \Delta u(k) + \Delta u(k+1) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k-1) \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} 3.0 \quad (3.30)$$

Dengan menggeser semua elemen yang mengandung Δu ke sebelah kiri dan yang tidak mengandung Δu ke sebelah kanan tanda pertidaksamaan (\leq), maka persamaan (3.29) dan (3.30) masing-masing dapat ditulis kembali menjadi

$$\begin{bmatrix} -\Delta u(k) \\ -\Delta u(k) - \Delta u(k+1) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} [-1.0 + u(k-1)] \quad (3.31)$$

$$\begin{bmatrix} \Delta u(k) \\ \Delta u(k) + \Delta u(k+1) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} [3.0 - u(k-1)] \quad (3.32)$$

atau

$$\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ -1 & 0 \\ -1 & -1 \end{bmatrix}}_{\underline{\Omega}} \underbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix}}_{\underline{\Delta U}} \leq \underbrace{\begin{bmatrix} 3.0 - u(k-1) \\ 3.0 - u(k-1) \\ -1.0 + u(k-1) \\ -1.0 + u(k-1) \end{bmatrix}}_{\underline{\omega}} \quad (3.33)$$

3. Matriks \underline{G} dan \underline{H} masing-masing dihitung dengan menggunakan persamaan (2.38), dan (2.39). Dengan membuat nilai matriks \underline{Q} sama dengan \underline{I}_{10} dan matriks \underline{R} bernilai $0,5\underline{I}_2$, maka matriks \underline{H} dapat dihitung sebagai berikut

$$\underline{H} = (\underline{C}_y \ \underline{\omega})^T \underline{Q} \underline{C}_y \ \underline{\omega} + \underline{R}$$

$$\underline{H} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

sedangkan nilai matriks \underline{G} untuk $k = 1$ adalah

$$\underline{G} = 2(\underline{\omega} \underline{C}_y)^T \underline{Q} \underline{\mathcal{E}}(1)$$

$$\underline{G} = \begin{bmatrix} 0.0015 \\ 0.0015 \end{bmatrix}$$

Perhitungan untuk matriks $\underline{\mathcal{E}}$ agak berbeda dengan persamaan (2.33) dimana

$$\underline{\mathcal{E}}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi} \underline{x}(k) - \underline{C}_y \underline{\Gamma} u(k-1) - \underline{C}_y \underline{\beta} \quad (3.34)$$

Dengan memisalkan pada $k = 1$ $\underline{x}(1) = [0 \ 0]^T$, $u(0) = 0$ dan

$$\underline{T}(1) = \begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{bmatrix}$$

maka didapatkan

$$\underline{E}(1) = \begin{bmatrix} 0.8367 \\ 0.8767 \\ 0.8900 \\ 0.9048 \\ 0.9164 \\ 0.9261 \\ 0.9349 \\ 0.9425 \\ 0.9493 \\ 0.9552 \end{bmatrix}$$

4. Menghitung nilai \underline{d} dan $\underline{\lambda}_r$ sebagai berikut :

- a. Nilai matriks $\underline{\Omega}_1$ dan $\underline{\Delta U}_1$ dipilih sedemikian rupa sehingga isi matriks $\underline{\Delta U}_1$ membuat pertidaksamaan $\underline{\Omega}_1 \underline{\Delta U} \leq \underline{\omega}_1$ menjadi aktif dan memenuhi persamaan

$$\underline{\Omega}_1 \underline{\Delta U}_1 = \underline{\omega}_1 \quad (3.35)$$

Matriks $\underline{\Omega}_1$ yang dipilih adalah

$$\underline{\Omega}_1 = [1 \quad 0],$$

Untuk menentukan nilai matriks $\underline{\omega}_1$, ada beberapa hal yang harus diperhatikan, yaitu :

- i. Jika selisih antara batas tegangan maksimum dan nilai sinyal masukan sebelumnya ($u(k-1)$) lebih besar dari *slew rate* maksimumnya (Δu_{maks}), maka batas tegangan maksimum u_{maks} harus diubah menjadi

$$u_{maks} = u(k-1) + \Delta u_{maks} \quad (3.36)$$

- i. Hal yang sama juga berlaku untuk batas tegangan minimum dimana ketika selisih antara nilai sinyal masukan sebelumnya ($u(k-1)$) dengan nilai batas tegangan minimum lebih besar daripada *slew rate* maksimumnya (Δu_{maks}), maka batas tegangan minimum u_{min} harus diubah menjadi

$$u_{min} = u(k-1) - \Delta u_{maks} \quad (3.37)$$

Besarnya *slew rate* maksimum pada percobaan ini adalah satu ($\Delta u_{maks} = 1$), sehingga isi matriks $\underline{\omega}_1$ yang memenuhi kedua syarat di atas dan bersesuaian dengan matriks $\underline{\Omega}_1$ adalah

$$\underline{\omega}_1 = [1 - u(0)] = [1]$$

Supaya persamaan (3.30) terpenuhi, maka isi matriks $\underline{\Delta U}_1$ yang harus digunakan adalah

$$\underline{\Delta U}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- b. Nilai \underline{d} dan $\underline{\lambda}_r$ dihitung dengan menggunakan persamaan

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_1 \end{bmatrix} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_1^T \\ \underline{\Omega}_1 & \underline{0} \end{bmatrix}^{-1} \begin{bmatrix} -\underline{\phi}_1 \\ \underline{0} \end{bmatrix} \quad (3.38)$$

dengan $\underline{\Phi}$ dan $\underline{\phi}$ merupakan bagian dari persamaan (3.20), maka

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_1 \end{bmatrix} = \begin{bmatrix} 1.0000 & 0 & 1 \\ 0 & 1.0000 & 0 \\ 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0.9985 \\ -0.0015 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_1 \end{bmatrix} = \begin{bmatrix} 0.0000 \\ -0.0015 \\ 0.9985 \end{bmatrix}$$

Karena semua $\underline{\lambda}_1$ bernilai positif dan ada bagian dari \underline{d} yang tidak bernilai nol, maka perlu dilakukan langkah berikutnya tanpa harus membuang *constraints* yang ada.

- c. Nilai α_1 dihitung dengan menggunakan persamaan (3.25) dan didapatkan nilai α_1 sama dengan nol. Karena nilai α_1 kurang dari satu ($\alpha_1 < 1$) maka ada *constraint* yang membuat nilai α_1 menjadi nol ditambahkan ke matriks $\underline{\Omega}_1$. *Constraint* yang ditambahkan ke matriks $\underline{\Omega}_1$ adalah *constraint* yang terdapat pada baris kedua dari matriks $\underline{\Omega}$ pada persamaan (3.33). Selanjutnya, matriks $\underline{\Omega}_1$ berubah menjadi matriks $\underline{\Omega}_2$, dimana

$$\underline{\Omega}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

sedangkan isi matriks $\underline{\Delta U}_2$ adalah seperti berikut

$$\underline{\Delta U}_2 = \underline{\Delta U}_1 + \alpha_1 \underline{d} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- d. Dengan mengulang langkah (4.b), maka proses perhitungan untuk mendapatkan nilai \underline{d} dan $\underline{\lambda}_2$ yang baru adalah

$$\begin{bmatrix} \underline{d} \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1.0000 & 0 & 1 & 1 \\ 0 & 1.0000 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0.9985 \\ -0.0015 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \underline{d} \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 1.0000 \\ 0.0015 \end{bmatrix}$$

Karena semua nilai \underline{d} sama dengan nol dan semua isi matriks λ_2 lebih besar dari nol, maka proses perhitungan selesai dan nilai $\underline{\Delta U}_2$ adalah nilai optimal yang membuat fungsi kriteria pada persamaan (3.17) menjadi minimum.

2. Nilai $\Delta u(k)$ yang digunakan untuk memperbaiki sinyal kendali hanya nilai pada baris pertama matriks $\underline{\Delta U}$ sedangkan isi baris yang lainnya dibuang karena pada proses pencuplikan berikutnya sudah didapatkan nilai $\Delta u(k)$ yang baru. Dari contoh perhitungan pada langkah (4), maka nilai $u(k)$ yang harus diberikan ke *plant* adalah sebagai berikut :

$$u(1) = \Delta u(1) + u(0)$$

dengan

$$\underline{\Delta U}_{opt} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ dan } \Delta u(k) = \underline{\Delta U}_{opt} [1,1] = 1$$

maka

$$u(1) = 1 + 0 = 1 \text{ Volt}$$

Untuk menghitung besar sinyal kendali pada proses pencuplikan berikutnya dapat dilakukan dengan mengulang langkah-langkah di atas tetapi dimulai hanya dari langkah (3).

4. UJI EKSPERIMEN DAN ANALISA

Bab ini membahas analisa dari uji eksperimen pengendalian *Level/Flow and Temperature Process Rig 38-003* dengan menggunakan metode *Model Predictive Control with Constraint* dengan memakai beberapa parameter penalaan yang berbeda-beda. Uji eksperimen dilakukan dengan memakai bantuan salah satu *toolbox* yang terdapat pada *MATLAB 7.0* yaitu *toolbox SIMULINK*.

Tujuan dari uji eksperimen yang dilakukan adalah untuk mengetahui kinerja dari pengendali *MPC with Constraint* dengan parameter penalaan yang berbeda-beda. Selain membahas kinerja pengendalian *Level/Flow and Temperature Process Rig 38-003* dengan pengendali *MPC with Constraint*, pada bab ini juga akan dibahas mengenai perbandingan antara hasil pengendalian dengan pengendali MPC dan dengan menggunakan metode pengendali ruang keadaan.

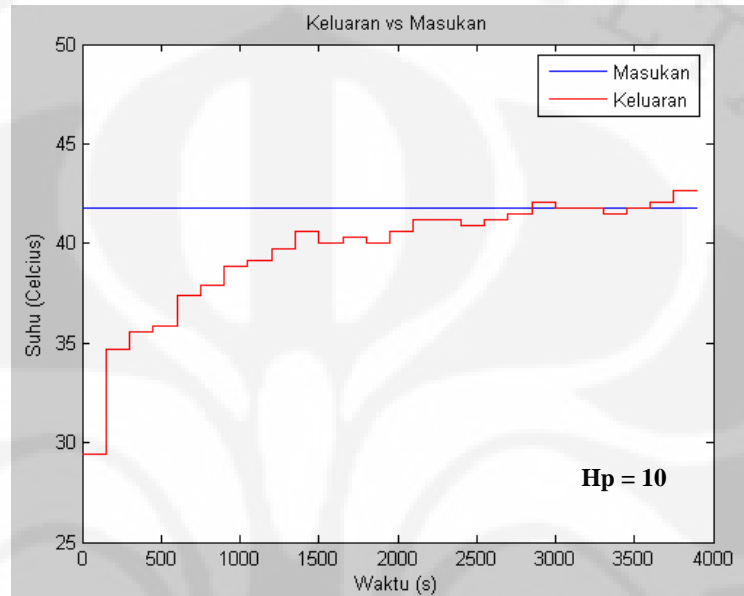
4.1. Uji Eksperimen Pengendali MPC Tanpa Nilai Trayektori Acuan yang Akan Datang

4.1.1. Uji Eksperimen untuk Nilai *Prediction Horizon* yang Bervariasi

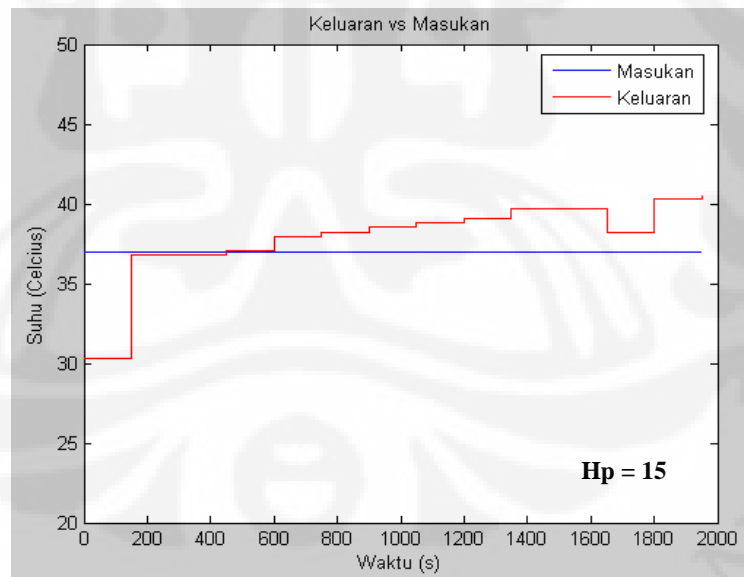
Uji eksperimen dilakukan dengan menggunakan bantuan *toolbox SIMULINK* yang terdapat pada *MATLAB 7.0*. Blok *simulink* yang dipakai untuk melakukan uji eksperimen untuk nilai *prediction horizon* yang bervariasi dilampirkan pada lampiran L.1.

Untuk melihat pengaruh variasi nilai *prediction horizon* terhadap hasil pengendalian *MPC with constraint*, maka nilai *control horizon* (H_u) dibuat tetap yaitu sebesar 2, nilai faktor bobot perubahan sinyal kendali (R) sebesar $0.2I_{H_u}$ dan nilai faktor bobot kesalahan sebesar I_{H_p} sedangkan nilai *prediction horizon* (H_p) dibuat bervariasi pada nilai $H_p = 10$ dan $H_p = 15$.

Berikut ini diberikan hasil uji eksperimen pengendalian *Level/Flow and Temperature Process Rig 38-003* dengan pengendali *MPC with constraint* dengan nilai *prediction horizon* yang bervariasi :



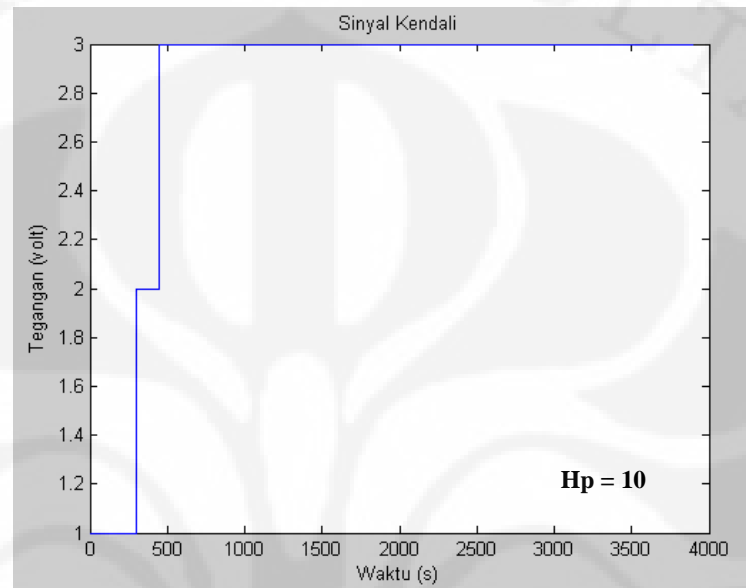
(a)



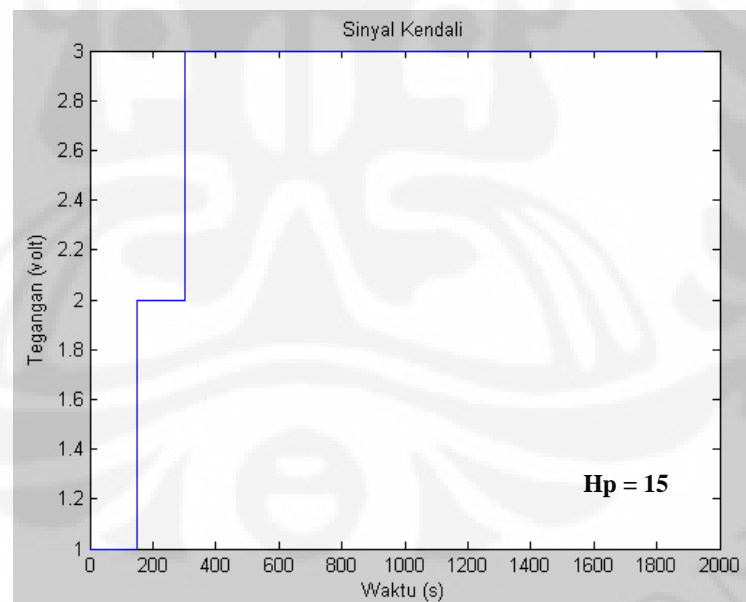
(b)

Gambar 4.1. Keluaran sistem hasil uji eksperimen dengan *prediction horizon* yang berbeda

Berikut ini diberikan grafik sinyal kendali yang diberikan ke *Level/Flow and Temperature Process Rig 38-003* dari pengendali MPC *with constraint* dengan nilai *prediction horizon* yang bervariasi :



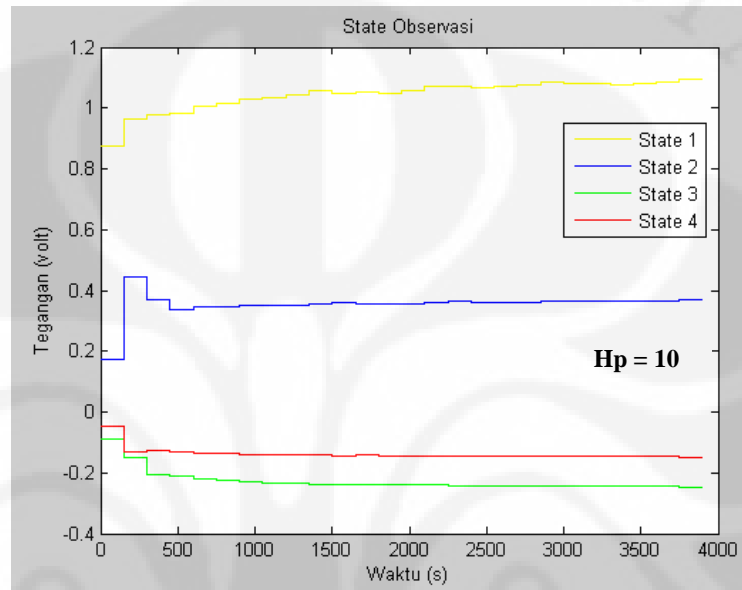
(a)



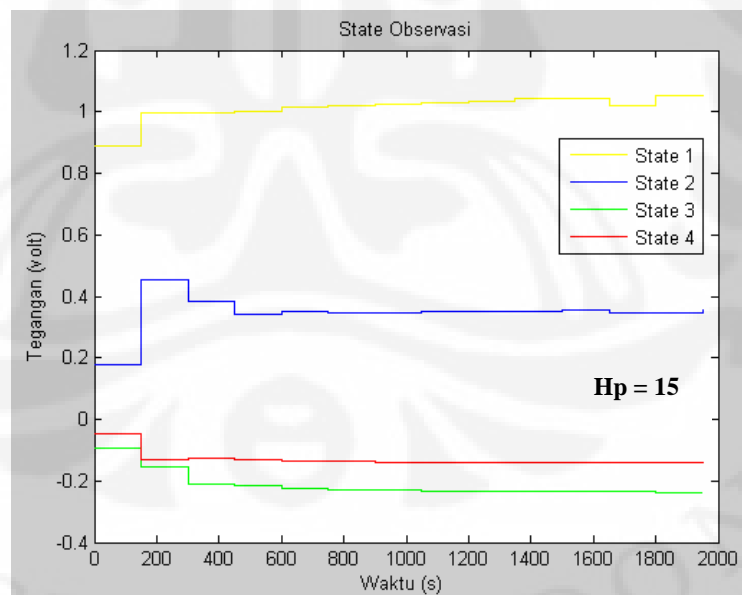
(b)

Gambar 4.2. Sinyal kendali hasil uji eksperimen dengan *prediction horizon* yang berbeda

Berikut ini diberikan grafik hasil observasi dari *reduced-order state observer* yang digunakan untuk mengestimasi keadaan-keadaan sistem untuk pengendali *MPC with constraint* dengan nilai *prediction horizon* yang bervariasi :



(a)



(b)

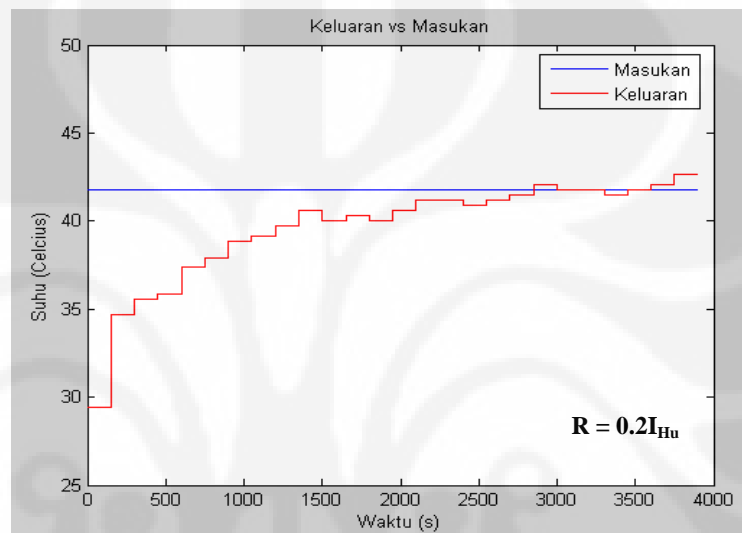
Gambar 4.3. Hasil estimasi *reduced order observer* untuk uji eksperimen dengan *prediction horizon* yang berbeda.

Berdasarkan gambar 4.1. dapat dilihat bahwa hasil uji eksperimen pengendalian sistem dengan *prediction horizon* sebesar 10 lebih baik bila dibandingkan dengan hasil uji eksperimen pengendalian sistem dengan *prediction horizon* sebesar 15. Hal ini menandakan bahwa hasil pengendalian akan menjadi lebih baik bila nilai *prediction horizon* yang diterapkan ke sistem mendekati nilai *control horizon*. Jika selisih antara *prediction horizon* dan *control horizon* terlalu besar maka kemampuan pengendali MPC untuk memprediksi keluaran sistem dan sinyal kendali yang akan dikirimkan ke sistem akan menurun karena sinkronisasi antara *prediction horizon* dan *control horizon* akan semakin sulit. Semakin dekat nilai *control horizon* dengan nilai *prediction horizon*, variansi perubahan sinyal kendali akan semakin kecil. Hal tersebut dapat terlihat pada saat akan terjadi perubahan trayektori acuan. Dengan nilai *control horizon* yang hampir sama dengan nilai *prediction horizon*, prediksi perubahan sinyal kendali menyesuaikan dengan nilai prediksi keluaran sehingga variansi perubahan sinyal kendali tidak terlalu besar

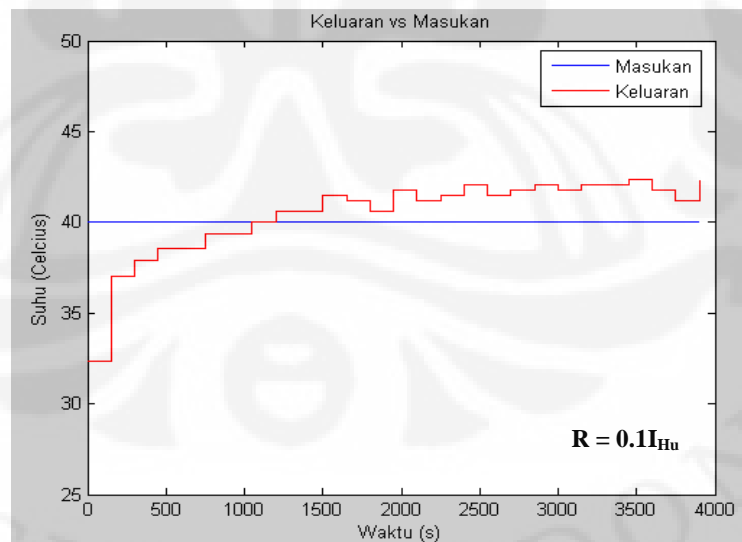
Dari gambar 4.1. juga dapat dilihat bahwa untuk mencapai daerah *setpoint* yang ditentukan, sistem membutuhkan waktu yang cukup lama walaupun memakai algoritma pengendali MPC. Selain itu, waktu respon sistem juga sangat tergantung dari kondisi awal sistem ketika akan dijalankan. Karena dalam skripsi ini yang diatur adalah temperatur, maka kondisi awal temperatur *Thermistor T4* merupakan salah satu faktor yang sangat berpengaruh terhadap hasil pengendalian. Hal ini disebabkan karena saluran penyalur fluida pada *Level/Flow and Temperature Process Rig 38-003* terbuat dari besi sehingga panas yang dibawa oleh fluida dapat terakumulasi dan terjebak didalam pipa besi sedangkan sistem pendingin yang disebut *fan-assisted cooling radiator* tidak dapat menjamin temperatur fluida yang masuk ke *inlet heat exchanger* tetap berada pada level 25 – 30°C.

4.1.2. Pengaruh Nilai Faktor Bobot Perubahan Sinyal Kendali (R) pada Hasil Pengendalian MPC

Untuk melihat pengaruh nilai faktor bobot perubahan sinyal kendali (R) pada hasil pengendalian MPC, maka dilakukan uji eksperimen pada sistem dengan membuat nilai diagonal matriks R berbeda-beda yaitu 0.1 dan 0.2, sedangkan nilai parameter pengendali lainnya dibuat tetap yaitu $H_p = 10$, $H_u = 2$ dan $Q = I_{H_p}$.



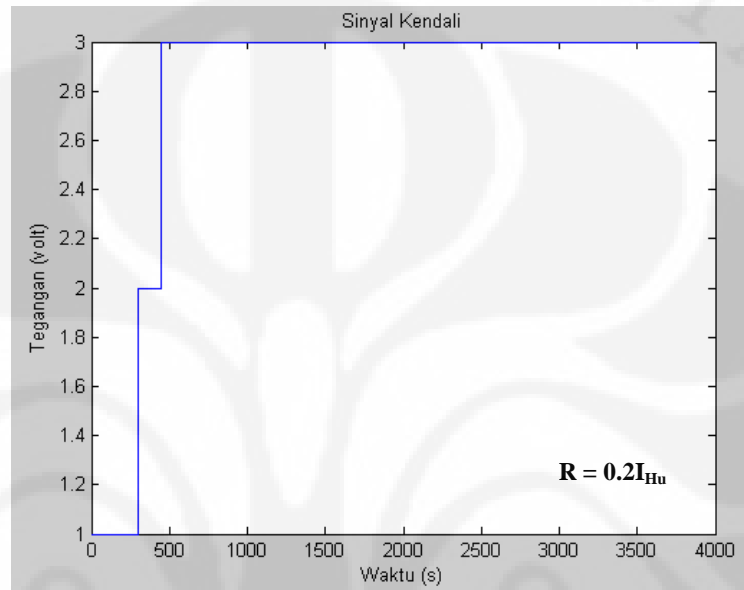
(a)



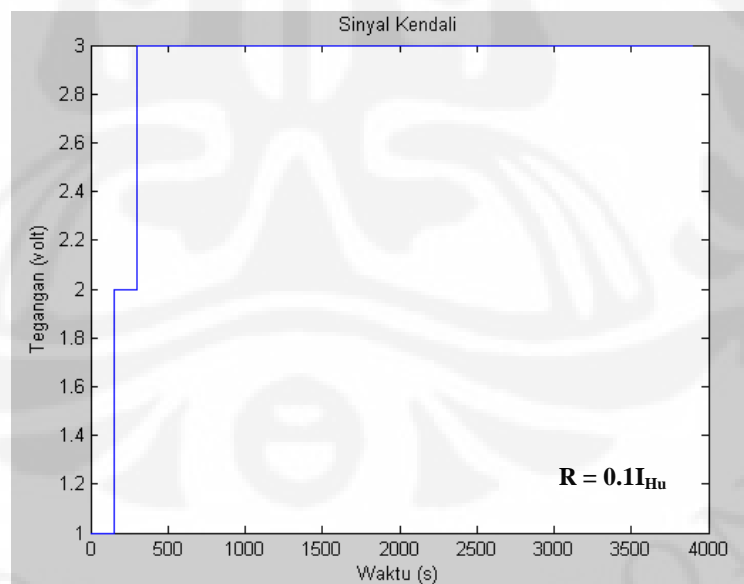
(b)

Gambar 4.4. Keluaran sistem hasil uji eksperimen dengan nilai R yang berbeda

Berikut ini diberikan grafik sinyal kendali yang diberikan ke *Level/Flow and Temperature Process Rig 38-003* dari pengendali *MPC with constraint* dengan nilai faktor bobot perubahan sinyal kendali yang bervariasi :



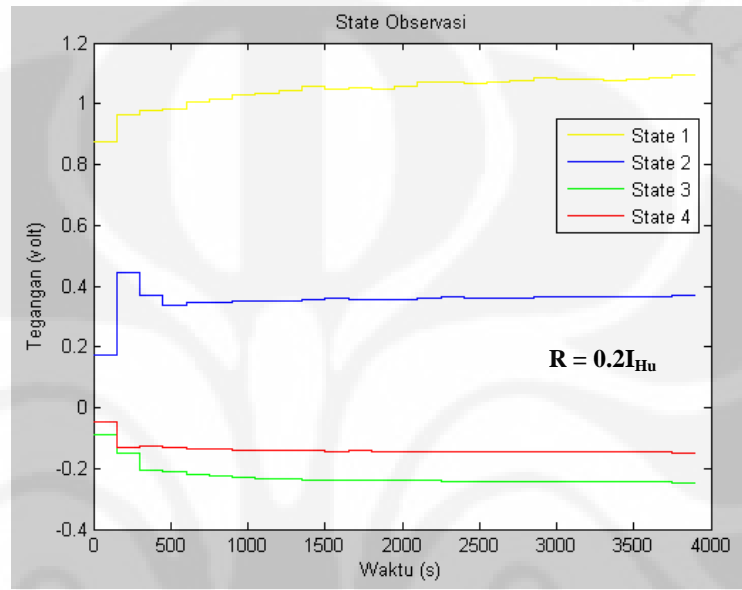
(a)



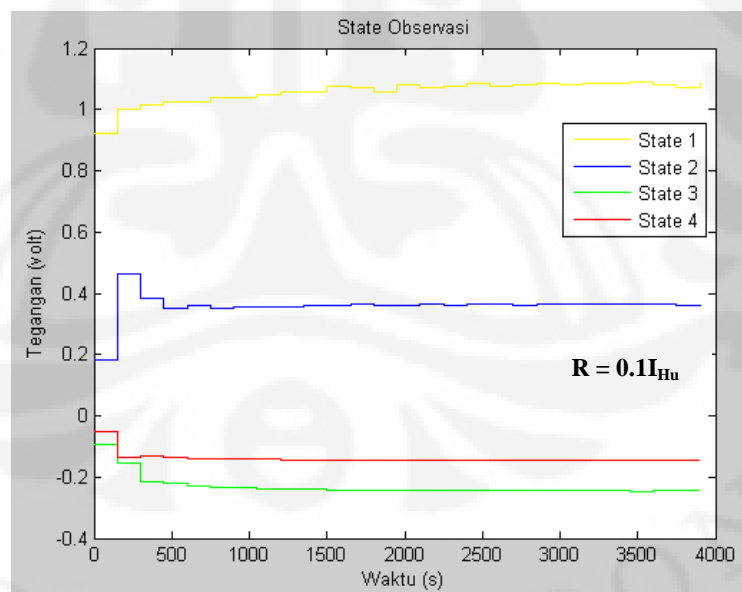
(b)

Gambar 4.5. Sinyal kendali hasil uji eksperimen dengan nilai R yang berbeda

Berikut ini diberikan grafik hasil observasi dari *reduced-order state observer* yang digunakan untuk mengestimasi keadaan-keadaan sistem untuk pengendali *MPC with constraint* dengan nilai faktor bobot perubahan sinyal kendali yang bervariasi :



(a)



(b)

Gambar 4.6. Hasil estimasi *reduced order observer* untuk uji eksperimen dengan nilai R yang berbeda.

Dari gambar 4.5. dapat dilihat pengaruh perbedaan besar nilai faktor bobot perubahan sinyal kendali yang diterapkan pada algoritma pengendali MPC. Semakin besar nilai R maka sinyal kendali yang diberikan ke sistem akan semakin ditekan. Dengan nilai R yang makin kecil, maka sinyal kendali semakin dilepas sehingga sinyal kendali akan diberikan lebih cepat dibandingkan dengan nilai R yang lebih besar. Hal ini dapat terjadi karena untuk proses identifikasi sistem *Level/Flow and Temperature Process Rig 38-003* hanya memakai sekitar 55 data dan perubahan nilai keluaran dengan basis perubahan acak sinyal kendali menjadi sedikit *bias* karena beberapa faktor seperti panas yang terjebak di pipa besi dan faktor *cooling radiator*. Beberapa faktor ini menyebabkan nilai matriks masukan pada persamaan ruang keadaan menjadi sangat kecil. Kondisi ini akan menyebabkan perhitungan perubahan sinyal kendali (Δu) menjadi sangat kecil sehingga sinyal kendali sulit berubah untuk mengikuti keadaan keluaran sistem.

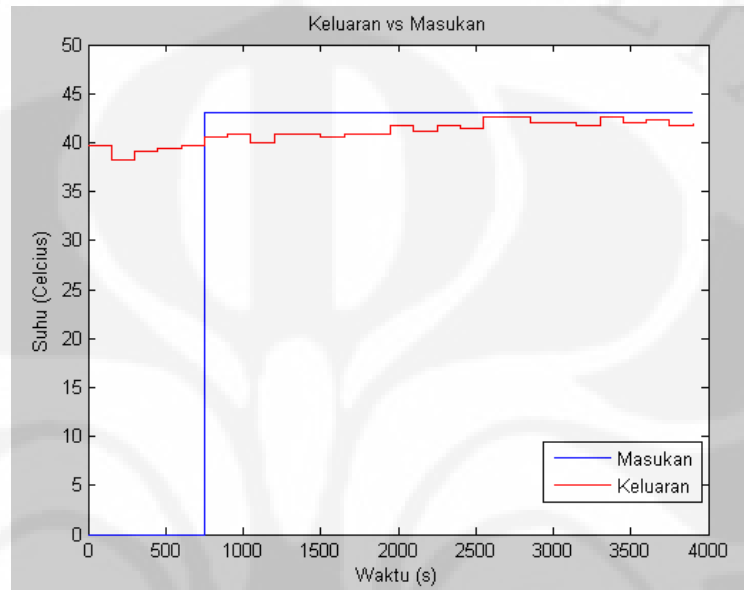
4.2. Uji Eksperimen Pengendali MPC dengan Nilai Trayektori Acuan yang Akan Datang

Pada proses uji eksperimen yang sebelumnya, nilai trayektori acuan yang akan datang belum diketahui dan hanya nilai trayektori acuan saat sekarang yang diketahui. Untuk mengatasi ketidaktahuan sistem pada trayektori acuannya, nilai trayektori acuan yang akan datang dianggap sama dengan nilai trayektori acuan sekarang.

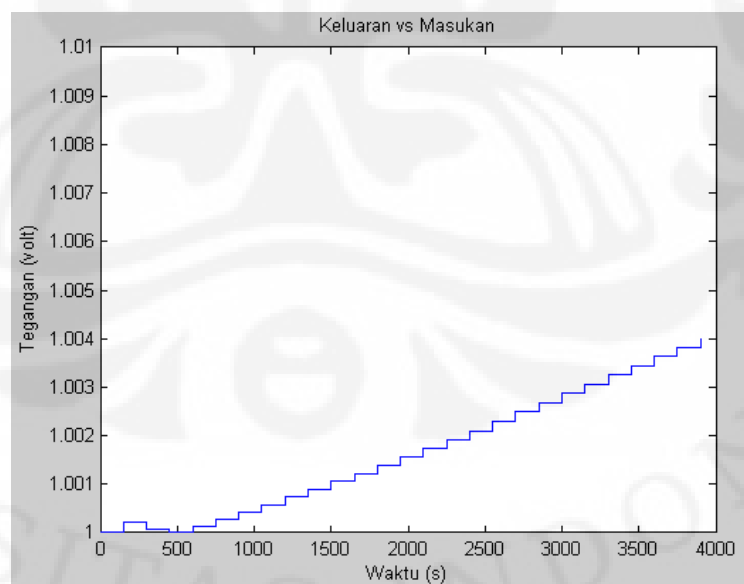
Pada uji eksperimen bagian ini, nilai trayektori acuan yang akan datang diberikan ke pengendali MPC. Langkah pertama yang dilakukan sistem adalah mendeteksi trayektori acuan pada beberapa proses pencuplikan, tergantung pada besarnya nilai *prediction horizon* yang diterapkan ke pengendali MPC. Dengan diketahuinya trayektori acuan untuk masa yang akan datang, keluaran sistem dapat berubah terlebih dahulu sebelum terjadi perubahan trayektori acuan sehingga waktu yang dibutuhkan oleh keluaran sistem untuk mencapai trayektori acuan yang diinginkan menjadi cepat.

Pada uji eksperimen ini, beberapa parameter pengendali MPC yang diterapkan antara lain : $H_p = 5$, $H_u = 2$, $R = 0.5I_{H_u}$ dan $Q = I_{H_p}$.

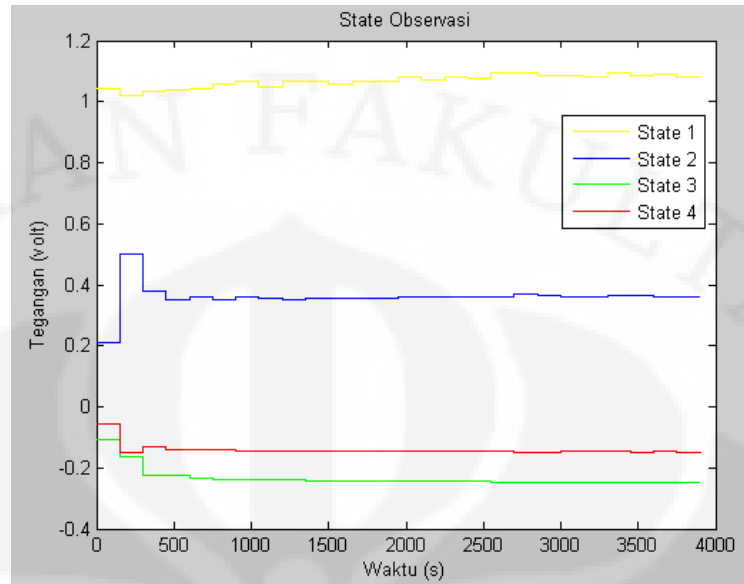
Berikut ini grafik keluaran hasil uji eksperimen dengan trayektori acuan yang akan datang diketahui :



Gambar 4.7. Keluran sistem hasil uji eksperimen dengan nilai trayektori acuan yang akan datang diketahui



Gambar 4.8. Sinyal kendali uji eksperimen dengan nilai trayektori acuan yang akan datang diketahui



Gambar 4.9. Hasil estimasi *reduced order observer* untuk uji eksperimen dengan nilai trayektori acuan yang akan datang diketahui.

Gambar 4.8. menunjukkan grafik sinyal kendali yang diberikan pengendali MPC ke sistem. Pada kondisi awal, nilai sinyal kendali yang diberikan pengendali adalah sebesar 1 volt karena tegangan minimum yang diberikan pada algoritma MPC adalah sebesar 1 volt. Saat sistem mendeteksi akan adanya perubahan *setpoint*, maka sinyal kendali mulai dinaikkan oleh pengendali MPC. Perubahan sinyal kendali yang diberikan pengendali MPC tidak begitu signifikan karena selisih antara *setpoint* dan keadaan keluaran sistem memang tidak begitu signifikan jika dalam satuan tegangan.

4.3. Perbandingan Kinerja Pengendali Metode MPC with Constraints dengan Metode Pengendali Ruang Keadaan

4.3.1. Landasan Teori Pengendali Ruang Keadaan

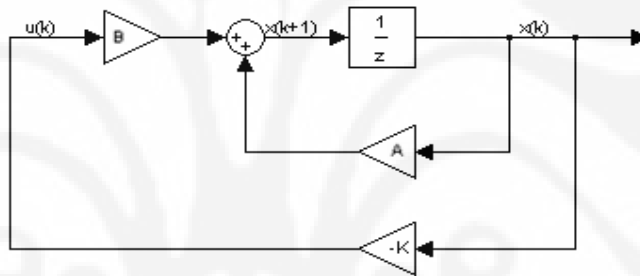
Persamaan ruang keadaan suatu sistem secara umum dapat dinyatakan dalam persamaan sebagai berikut :

$$x(k+1) = Ax(k) + Bu(k) \quad (4.1)$$

$$y(k) = Cx(k) + Du(k) \quad (4.2)$$

Pada dasarnya, metode pengendali ruang keadaan merupakan sebuah metode pengendali penempatan kutub dimana metode pengendaliannya dimulai dengan penentuan kutub-kutub sistem lingkaran tertutup yang didasarkan pada kebutuhan *transient response* dan/atau *frequency response* sistem seperti kecepatan, koefisien redaman atau *bandwidth*. Oleh karena itu, letak kutub-kutub lingkaran tertutup sistem harus ditentukan terlebih dahulu yaitu berada pada posisi $z = \mu_1, z = \mu_2, \dots, z = \mu_n$.

Berikut ini adalah blok diagram pengendali ruang keadaan lingkaran tertutup dengan *necessary and sufficient condition* :



Gambar 4.10. Pengendali ruang keadaan lingkaran tertutup

Sinyal kendali yang diberikan ke sistem sebesar $u(k) = -Kx(k)$ dengan K adalah matriks penguat umpan balik keadaan dan persamaan keadaan sistem menjadi :

$$x(k+1) = (A - BK)x(k) \quad (4.3)$$

Matriks K harus dipilih sehingga membuat nilai *eigen* dari $A - BK$ menjadi kutub-kutub lingkaran tertutup yang diinginkan, $\mu_1, \mu_2, \dots, \mu_n$.

Untuk menghitung besarnya matriks K , dapat digunakan formula *Ackermann* yaitu :

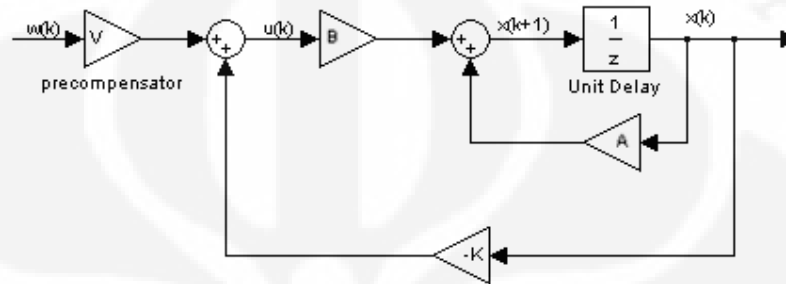
$$\begin{aligned} |zI - A + BK| &= (z - \mu_1)(z - \mu_2) \cdots (z - \mu_n) \\ &= z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \cdots + \alpha_{n-1} z + \alpha_n = 0 \end{aligned} \quad (4.4)$$

$$K = [0 \ 0 \ \cdots \ 0 \ 1] [B \ AB \ \cdots \ A^{n-1}B]^{-1} \phi(G) \quad (4.5)$$

dimana

$$\phi(A) = A^n + \alpha_1 A^{n-1} + \dots + \alpha_{n-1} A + \alpha_n I$$

Berikut ini adalah blok diagram pengendali ruang keadaan lingkaran tertutup dengan penguat *precompensator* :



Gambar 4.11. Pengendali ruang keadaan lingkaran tertutup dengan penguat *precompensator*

Sinyal kendali yang diberikan ke sistem sebesar :

$$u(k) = Vw(k) - Kx(k)$$

Persamaan ruang keadaan sistem menjadi :

$$x(k+1) = Ax(k) + B[Vw(k) - Kx(k)]$$

$$x(k+1) = [A - BK]x(k) + BVw(k)$$

dengan $x(k+1) = x(k)$ saat kondisi *steady state*, maka

$$x(k)[I - A + BK] = BVw(k)$$

$$x(k) = [I - A + BK]^{-1} BVw(k) \quad (4.6)$$

Persamaan keluaran sistem adalah sebagai berikut :

$$y(k) = C[I - A + BK]^{-1} BVw(k) \quad (4.7)$$

dengan $y(k) = w(k)$ saat kondisi *steady state*, maka

$$w(k) = C[I - A + BK]^{-1} BVw(k)$$

Sehingga persamaan penguat *precompensator* didapat sebagai berikut :

$$V = [C[I - A + BK]^{-1} B]^{-1} \quad (4.8)$$

4.3.2. Uji Eksperimen dan Analisa

4.3.2.1. Pengetesan *Controllability* dan Perancangan Pengendali

Persamaan ruang keadaan sistem yang dipakai adalah persamaan 3.6. Sebelum menerapkan pengendali ruang keadaan, pengetesan *controllability* sistem harus dilakukan dengan membentuk matriks *controllability* sebagai berikut :

$$\begin{bmatrix} B & AB & \dots & (A)^{n-1} B \end{bmatrix} \quad (4.9)$$

dimana :

- n adalah jumlah *state* yang dimiliki oleh sebuah sistem.
- sistem *controllable* jika matriks *controllability* memiliki *rank* sebanyak n (jumlah *state*)

Berdasarkan persamaan (4.9) maka matriks *controllability* dari *Level/Flow and Temperature Process Rig 38-003* adalah sebagai berikut :

$$\begin{bmatrix} 0.0008 & 0.0024 & 0.0021 & 0.0027 \\ 0.0019 & 0.0004 & 0.0013 & 0.0008 \\ 0 & -0.0001 & -0.0003 & -0.0005 \\ 0 & -0.0001 & -0.0003 & -0.0003 \end{bmatrix}$$

Rank dari matriks *controllability* diatas adalah 4 sehingga sistem *Level/Flow and Temperature Process Rig 38-003* dikatakan *fully controllable*.

Nilai eigen dari persamaan ruang keadaan sistem adalah :

$$eig = \begin{bmatrix} 1.001 \\ 0.5579 \\ -0.4412 + 0.2175i \\ -0.4412 - 0.2175i \end{bmatrix}$$

Letak kutub-kutub lingkaran tertutup sistem yang diinginkan adalah :

$$desired\ poles = \begin{bmatrix} 0.8 \\ 0.4 \\ -0.3 + 0.2175i \\ -0.3 - 0.2175i \end{bmatrix}$$

Matriks penguat umpan balik keadaan diperoleh dengan menggunakan persamaan 4.5 atau dengan menggunakan perintah *Acker* pada MATLAB 7.0. dan diperoleh nilai sebagai berikut :

$$K = [149.5873 \quad -29.0894 \quad 368.2748 \quad -145.8690]$$

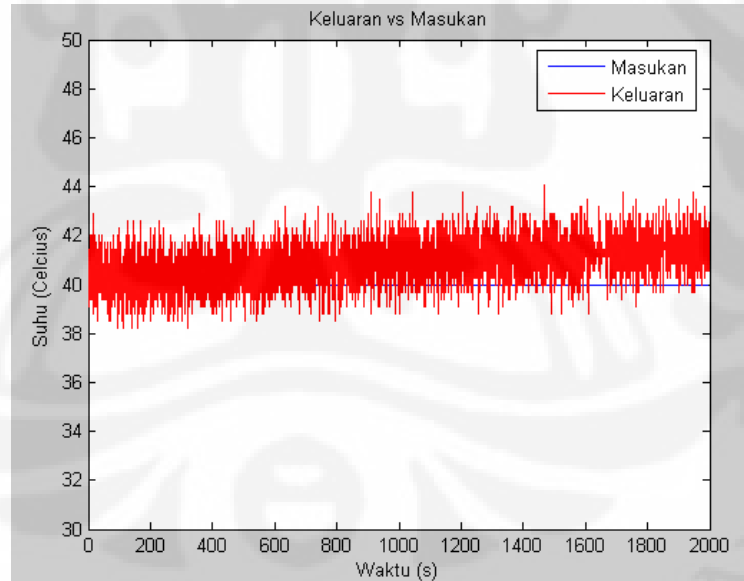
Sedangkan penguat *precompensator* diperoleh dengan menggunakan persamaan 4.8. dan diperoleh nilai sebagai berikut :

$$V = 77.2136$$

4.3.2.2. Uji Eksperimen Pengendali Ruang Keadaan

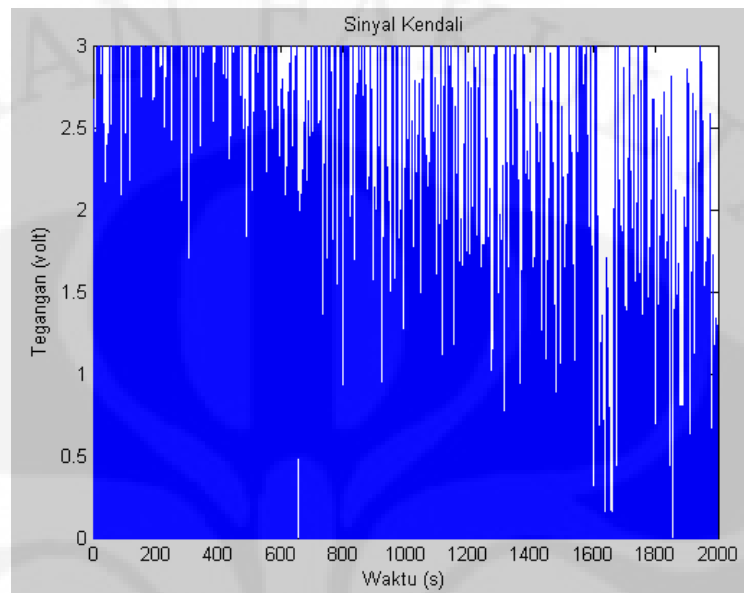
Parameter-parameter pengendali ruang keadaan yang telah didapat akan diterapkan pada sistem *Level/Flow and Temperature Process Rig 38-003*. Blok SIMULINK yang dipakai untuk uji eksperimen pengendali ruang keadaan dilampirkan pada lampiran L.3.

Berikut ini hasil keluaran uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint :



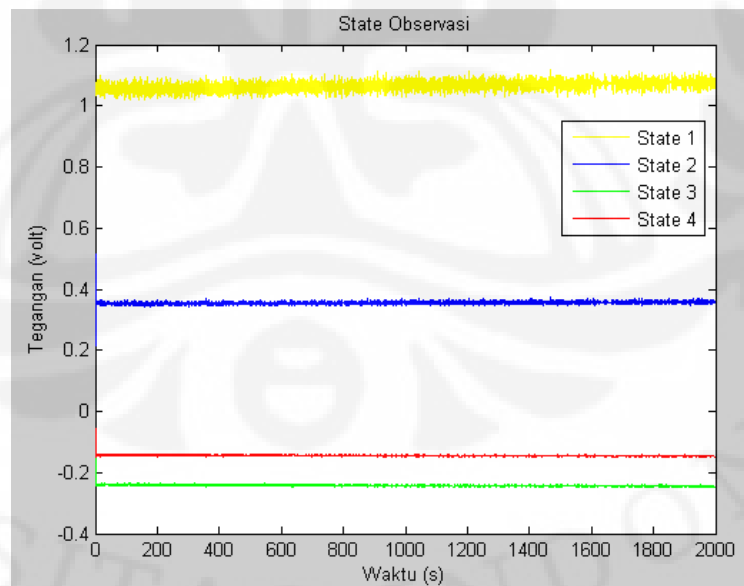
Gambar 4.12. Hasil keluaran uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint

Berikut ini sinyal kendali uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint :



Gambar 4.13. Sinyal kendali uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint

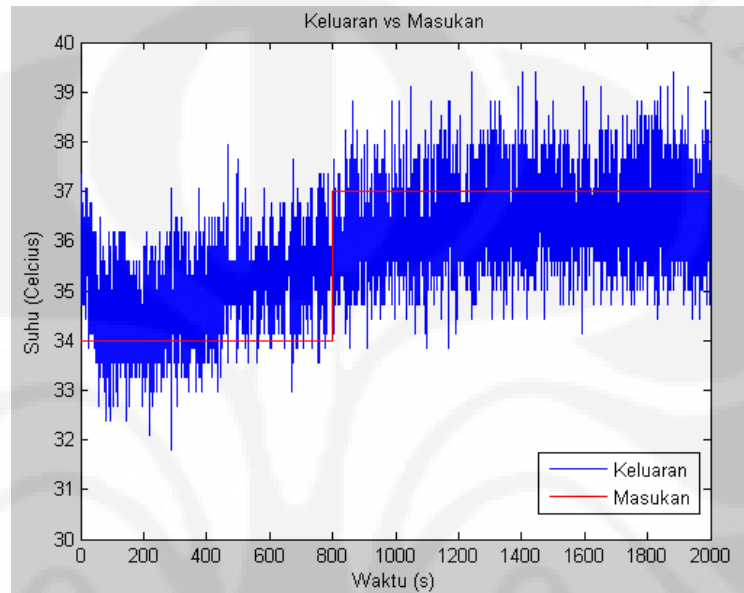
Berikut ini hasil estimasi reduced-order observer untuk uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint :



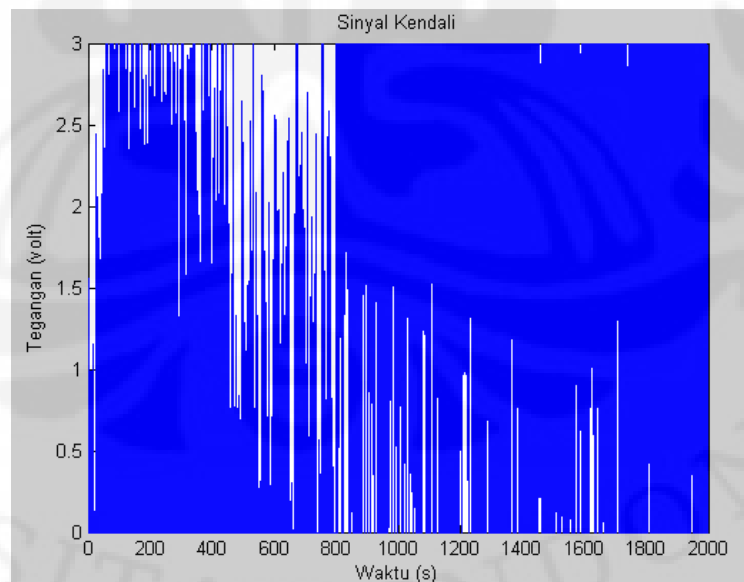
Gambar 4.14. Hasil estimasi reduced-order observer untuk uji eksperimen pengendali ruang keadaan tanpa perubahan setpoint

Uji eksperimen pengendali ruang keadaan juga dilakukan dengan mengubah nilai setpoint yang diberikan untuk melihat kinerja pengendali.

Berikut ini hasil keluaran uji eksperimen pengendali ruang keadaan dengan perubahan setpoint :

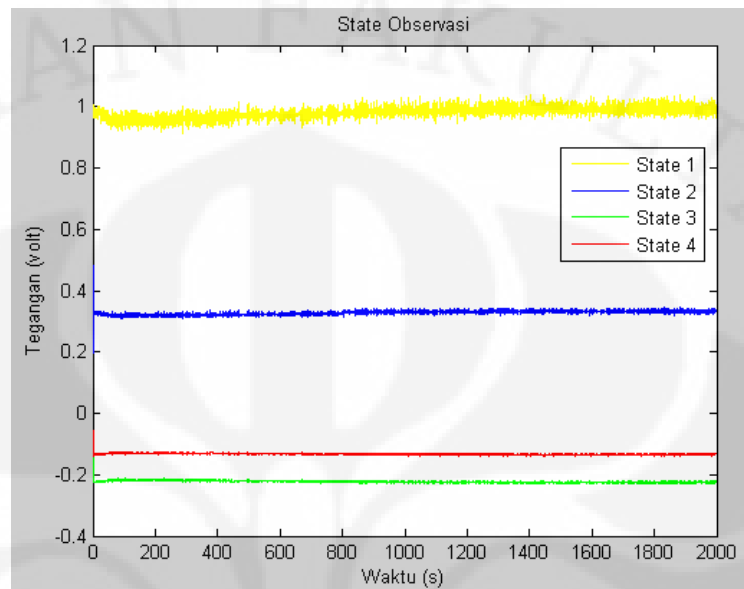


Gambar 4.15. Hasil keluaran uji eksperimen pengendali ruang keadaan dengan perubahan setpoint



Gambar 4.16. Sinyal kendali uji eksperimen pengendali ruang keadaan dengan perubahan setpoint

Berikut ini hasil estimasi reduced-order observer untuk uji eksperimen pengendali ruang keadaan dengan perubahan setpoint :



Gambar 4.17. Hasil estimasi reduced-order observer untuk uji eksperimen pengendali ruang keadaan dengan perubahan setpoint

Setelah melakukan dua uji eksperimen pengendali ruang keadaan yaitu uji eksperimen tanpa perubahan setpoint dan uji eksperimen dengan perubahan setpoint, kinerja pengendali ruang keadaan dapat dikatakan cukup baik karena pengendali dapat membuat sistem berusaha mengikuti nilai masukan yang diberikan. Hanya saja di beberapa tempat masih terdapat selisih antara masukan dan keluaran. Hal ini disebabkan beberapa faktor seperti panas yang terjebak di pipa besi dan faktor *cooling radiator*. Selisih ini sebenarnya sudah diatasi oleh perlakuan sinyal kendali yang diberikan pengendali ruang keadaan dimana pengendali memberikan reaksi seperti penurunan sinyal kendali ketika keluaran telah melebihi masukan dan juga sebaliknya.

Walaupun keluaran sistem cukup baik, metode pengendali ruang keadaan masih memiliki kekurangan yaitu tidak dapat diperhitungkannya besar perubahan sinyal kendali dan batasan sinyal kendali pada proses pengendalian seperti pada MPC. Akibatnya, variasi perubahan sinyal kendali menjadi cukup besar (gambar (4.13) dan gambar (4.16)) dan besar

sinyal kendali bisa menjadi tidak terbatas. Besarnya sinyal kendali dapat dibatasi dengan menggunakan blok saturasi sehingga sinyal kendali yang masuk ke *plant* akan dipotong jika melebihi tegangan maksimum atau tegangan minimum yang diperbolehkan. Jika sinyal kendali yang masuk ke *plant* terus-menerus dipotong, maka akan membuat hasil kendali menjadi tidak bagus. Karena pada uji eksperimen ini sinyal kendali yang terpotong hampir tidak ada, maka keluaran sistem hasil pengendalian menjadi cukup baik.

5. KESIMPULAN

Dari keseluruhan pembahasan dalam skripsi ini dapat disimpulkan beberapa hal, yaitu :

1. Pemakaian *reduced-order observer* untuk mengestimasi variabel keadaan sistem dapat menggantikan keterbatasan *hardware* sistem tetapi hasil estimasi yang dikumpulkan tidak sebaik bila menggunakan data yang dikumpulkan secara langsung oleh sensor.
2. Penggeseran nilai eigen *reduced-order observer* tidak boleh terlalu jauh dari nilai eigen sistem agar tidak menghilangkan karakteristik alami sistem. Nilai eigen sistem berada pada nilai $eig_1 = 0.8$ dan $eig_{2,3} = -0.3 \pm 0.2175i$, kemudian digeser ke nilai *desired* $eig_1 = 0.4$ dan *desired* $eig_{2,3} = -0.3 \pm 0.2175i$.
3. Keluaran sistem hasil pengendalian *MPC with constraint* akan semakin baik bila nilai *prediction horizon* mendekati nilai *control horizon*.
4. Semakin besar nilai faktor bobot perubahan sinyal kendali R , maka perubahan sinyal kendali dapat semakin ditekan.
5. Kombinasi terbaik nilai *prediction horizon*, *control horizon* dan faktor bobot perubahan sinyal kendali R untuk sistem Level/Flow and Temperature Process Rig 38-003 adalah $H_p = 10$, $H_u = 2$ dan $R = 0.2I_{Hu}$.
6. Metode *MPC with constraints* dapat menghasilkan keluaran yang lebih baik dibandingkan dengan metode Pengendali Ruang Keadaan karena pada *MPC with constraints* tidak akan terjadi perubahan yang drastis pada sinyal kendali dan pemotongan paksa pada sinyal kendali.

DAFTAR REFERENSI

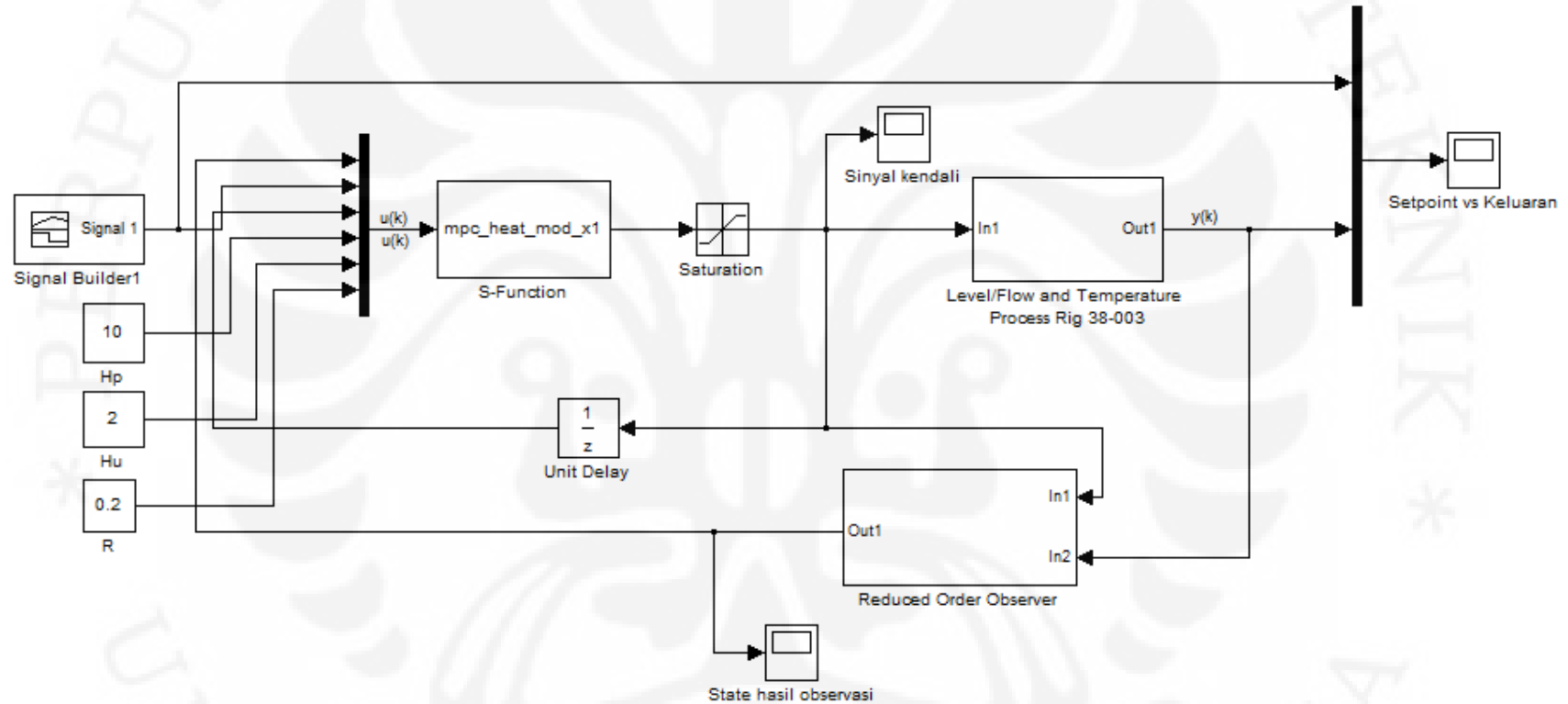
- Subiantoro, Aries, *Diktat Kuliah Sistem Kendali Adaptif* (Depok : Control System Research Group Jurusan Elektro FTUI, 2002)
- E.F. Camacho, C. Bordons, *Model Predictive Control* (Springer-Verlag, 1999)
- J. M. Maciejowski, *Predictive Control with Constraints* (Prentice Hall, 2002)
- Ogata, Katsuhiko, *Discrete-Time Control Systems* (Prentice Hall, 1995)
- PROCON Process Control Trainer, *Temperature – Workbook 38-002*. Feedback.1996.
- Mellon, Carnegie, *Control Tutorials for Matlab*. The University of Michigan.
- Kristiawan, Antonius Yuda, *Aplikasi Model Predictive Control dengan Constraints Sinyal Kendali Berbasis Algoritma Active Set pada Pengendalian Coupled-Tank Control Apparatus PP-100* (Depok : Skripsi Program Sarjana Teknik Elektro FTUI, 2007)

LAMPIRAN



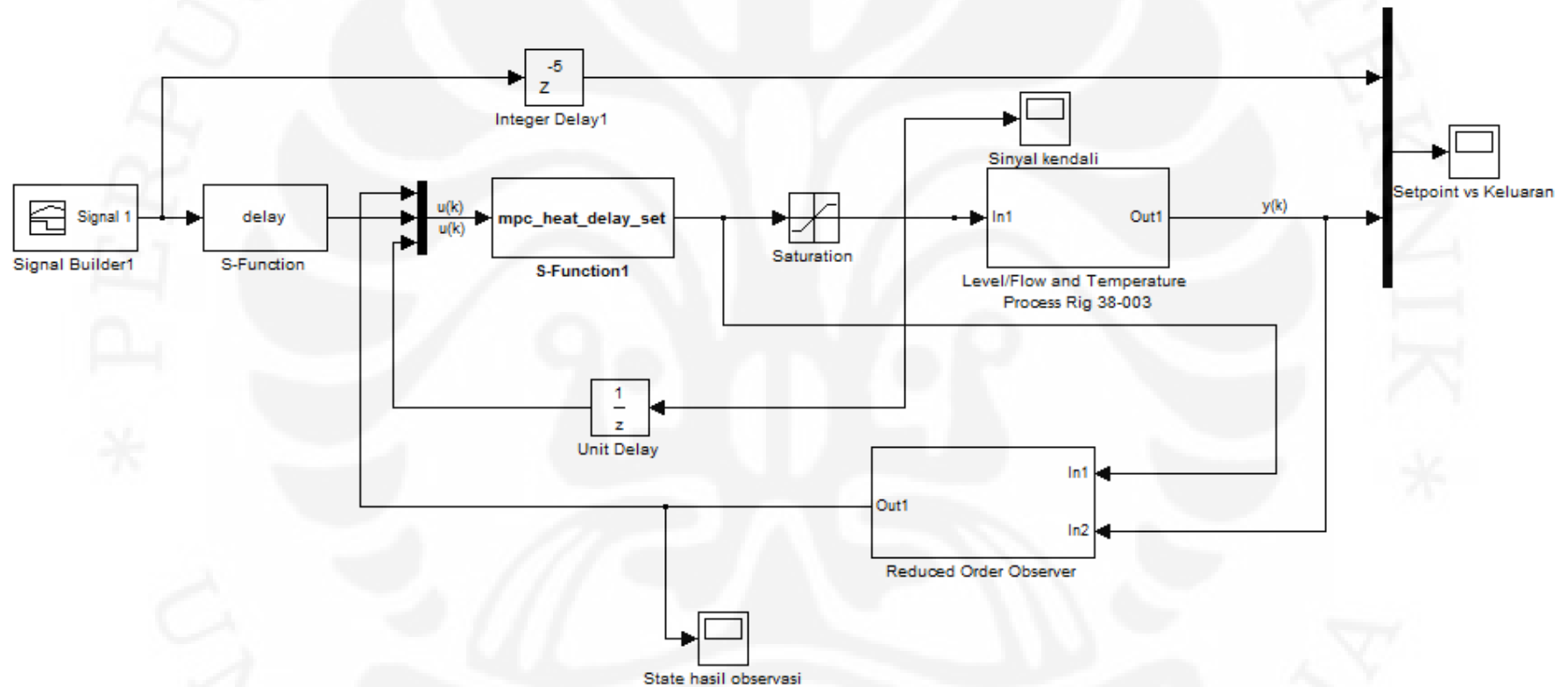
Lampiran 1. Blok SIMULINK

1. Blok SIMULINK pada Uji Eksperimen Pengendalian Menggunakan Metode MPC *with Constraints* tanpa Nilai Trayektori Acuan yang Akan Datang



Gambar L.1. Blok SIMULINK uji eksperimen pengendalian menggunakan metode MPC dengan *constraints* tanpa Nilai Trayektori Acuan yang Akan Datang

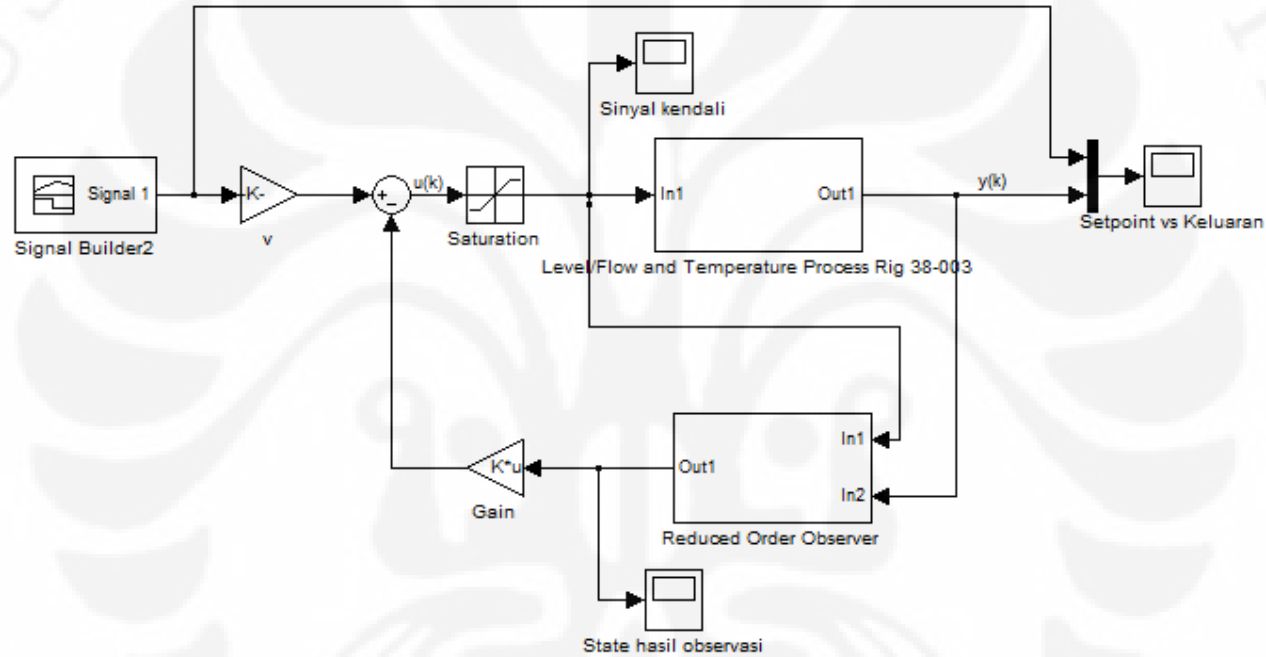
2. Blok SIMULINK pada Uji Eksperimen Pengendalian Menggunakan Metode MPC *with Constraints* dengan Nilai Trayektori Acuan yang Akan Datang



Gambar L.2. Blok SIMULINK uji eksperimen pengendalian menggunakan metode MPC dengan *constraints* dengan Nilai Trayektori Acuan yang Akan Datang

Universitas Indonesia

3. Blok SIMULINK pada Uji eksperimen Pengendalian Menggunakan Metode Aturan Kendali Ruang Keadaan



Gambar L.3. Blok SIMULINK uji eksperimen pengendalian menggunakan metode Aturan Kendali Ruang Keadaan.