



UNIVERSITAS INDONESIA

**VALIDASI MODEL DINAMIKA UAV-200 LAPAN
MENGUNAKAN ALGORITMA STATE SPACE MPC**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

YUANANDA

0706268114

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JANUARI 2012**



UNIVERSITAS INDONESIA

**VALIDASI MODEL DINAMIKA UAV-200 LAPAN
MENGUNAKAN ALGORITMA STATE SPACE MPC**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar sarjana

Disusun Oleh :

YUANANDA

0706268114

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS INDONESIA
JANUARI 2012**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Yuananda
NPM : 0706268114
Tanda Tangan : 
Tanggal : 18 Januari 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Yuananda
NPM : 0706268114
Program Studi : Teknik Elektro
Judul Seminar : Validasi Model dinamika UAV-200 LAPAN
menggunakan algoritma State Space MPC

Telah berhasil diajukan kepada Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Ir. Wahidin Wahab M.Sc., Ph.D.
NIP : 195311251979021001

Penguji : Prof. Dr.Eng. Drs. Benyamin Kusumoputro M.Eng.
NIP : 195711171987031001

Penguji : Dr. Abdul Muis S.T., M.Eng
NIP : 197509011999031003

Ditetapkan di : Depok

Tanggal : 18 Januari 2012

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan YME karena atas berkat rahmat dan karunia-Nya lah saya dapat menyelesaikan skripsi ini. Penulisan skripsi dilakukan sebagai salah satu syarat untuk menjadi Sarjana Teknik di Departemen Teknik Elektro FTUI. Saya menyadari bahwa tanpa bantuan banyak pihak, skripsi ini tidak mungkin terselesaikan. Oleh sebab itu, pada kesempatan ini saya ingin mengucapkan terima kasih kepada pihak-pihak yang membantu saya dalam segala hal mengenai penyusunan skripsi baik secara langsung maupun tidak langsung. Hal ini saya tujukan kepada:

1. Ir. Wahidin Wahab M.Sc., Ph.D selaku dosen Pembimbing Akademik dan Pembimbing skripsi saya, yang telah meluangkan waktunya serta masukan-masukan selama bimbingan.
2. Herma Yudhi Irwanto, M. Eng yang telah meluangkan waktu dan juga masukan-masukannya dan juga staff LAPAN yang telah memberikan model matematikanya

Skripsi ini tentunya masih jauh dari kesempurnaan. Oleh karena itu, saya mohon maaf apabila terdapat kekurangan dalam penyusunan skripsi ini. Dengan segala kerendahan hati, saya sangat mengharapkan kritik dan saran untuk memperbaiki skripsi ini pada khususnya dan kemampuan penulis pada umumnya. Semoga tugas akhir ini dapat berguna dan memberikan manfaat bagi kita semua.

Depok, 18 Januari 2012

Yuananda

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Nama : Yuananda
NPM : 0706268114
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

Demi 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 :

Validasi Model dinamika UAV-200 LAPAN menggunakan algoritma State Space MPC

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencatumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta.
Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 18 Januari 2012

Yang menyatakan



(Yuananda)

(Yuananda)

ABSTRAKSI

Nama : Yuananda
Program Studi : Teknik Elektro
Judul : Validasi Model dinamika UAV-200 LAPAN menggunakan algoritma State Space MPC

Pada skripsi ini dilakukan pembahasan secara umum mengenai MPC dan jenis MPC yang digunakan. Model yang digunakan adalah model dinamika berupa matriks state space sistem. Simulasi akan dilakukan dengan menggunakan metode State Space MPC dengan menggunakan perangkat lunak computer MATLAB yang bertujuan untuk melihat penerapan sistem dengan MPC pada Guided Flight Vehicle dan juga melihat sampai sejauh mana State Space MPC dapat berperan mengatasi disturbance yang mengganggu model system tersebut dan seberapa jauh UAV-200 LAPAN dapat diikuti oleh MPC dengan berbagai prediction horizon dan moving horizon.

Kata Kunci: State Space MPC, Guided Flight Vehicle

ABSTRACT

Name : Yuananda
Study Program : Electrical Engineering
Title : Dynamic Model Validation for UAV-200 LAPAN using State Space Algorithm

In this final project, a general knowledge about MPC will be discussed. The type of MPC that will be discussed is the State Space MPC method. The model that will be used is a dynamic model of the system in state space matrices form. Simulation will be done by using State Space MPC method using computer software, in this case is MATLAB. This simulation aims to see how the system of a Guided Flight Vehicle that applied with MPC strategy will react and also to see how far MPC can handle the disturbance and how far UAV-200 LAPAN can be followed by MPC with various prediction horizon and moving horizon as well.

Keywords: State Space MPC, Guided Flight Vehicle

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR.....	iv
HALAMAN PERSETUJUAN PUBLIKASI SKRIPSI.....	ivi
ABSTRAKSI.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	x
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	2
1.3 Pembatasan Masalah.....	2
1.4 Metode Penelitian.....	2
1.5 Sistematika Penulisan.....	2
BAB 2 LANDASAN TEORI.....	4
2.1 Pengertian MPC.....	4
2.2 Perancangan dan Implementasi Strategi Kontrol MPC.....	8
2.3 Pengetesan Proses.....	11
2.4 State Space MPC.....	12
BAB 3 DESKRIPSI SISTEM.....	16
3.1 Dinamika Guided Missile.....	16
3.2 UAV-200.....	17
BAB 4 SIMULASI DAN ANALISIS.....	23
4.1 Simulasi Matra Longitudinal.....	23
4.2 Simulasi Matra Lateral Directional.....	29
BAB 5 KESIMPULAN.....	34
DAFTAR ACUAN.....	35

DAFTAR GAMBAR

Gambar 2.1 Struktur Dasar MPC	4
Gambar 2.2 Konsep MPC	6
Gambar 2.3 Skema MPC	7
Gambar 2.4 Algoritma MPC	8
Gambar 2.5 Perancangan MPC	9
Gambar 3.1 Missile Dynamics	16
Gambar 3.2 Blok Diagram Model Dinamika dengan MPC	17
Gambar 4.1 Hasil Simulasi Matra Longitudinal dengan $p=10$ dan $m=2$	23
Gambar 4.2 Hasil Simulasi Matra Longitudinal dengan $p=15$ dan $m=3$	24
Gambar 4.3 Hasil Simulasi Matra Longitudinal dengan $p=18$ dan $m=3$	24
Gambar 4.4 Hasil Simulasi Matra Longitudinal dengan Disturbance 0,1%	25
Gambar 4.5 Hasil Simulasi Matra Longitudinal dengan Disturbance 1%	26
Gambar 4.6 Hasil Simulasi Matra Longitudinal dengan Disturbance 2%	27
Gambar 4.7 Hasil Simulasi Matra Longitudinal dengan Disturbance 5%	27
Gambar 4.8 Hasil Simulasi Matra Longitudinal dengan Disturbance 10%	28
Gambar 4.9 Hasil Simulasi Matra Lateral Direct dengan $p=10$ dan $m=2$	29
Gambar 4.10 Hasil Simulasi Matra Lateral Direct dengan $p=15$ dan $m=3$	29
Gambar 4.11 Hasil Simulasi Matra Lateral Direct dengan $p=20$ dan $m=5$	30
Gambar 4.12 Hasil Simulasi Matra Lateral Direct dengan $p=25$ dan $m=6$	30
Gambar 4.13 Hasil Simulasi Matra Lateral Direct dengan $p=30$ dan $m=7$	31
Gambar 4.14 Hasil Simulasi Matra Lateral Direct dengan Disturbance 0,1%	31
Gambar 4.15 Hasil Simulasi Matra Lateral Direct dengan Disturbance 1%	32
Gambar 4.16 Hasil Simulasi Matra Lateral Direct dengan Disturbance 2%	33
Gambar 4.17 Hasil Simulasi Matra Lateral Direct dengan Disturbance 5%	33
Gambar 4.18 Hasil Simulasi Matra Lateral Direct dengan Disturbance 10%	34

DAFTAR TABEL

Tabel 3.2.a. Matriks A Matra Longitudinal	20
Tabel 3.2.b. Matriks B Matra Longitudinal	20
Tabel 3.2.c. Matriks C Matra Longitudinal.....	20
Tabel 3.2.d. Matriks D Matra Longitudinal	21
Tabel 3.2.e. Matriks K Matra Longitudinal	21
Tabel 3.2.f. Matriks $x(0)$ Matra Longitudinal	21
Tabel 3.2.g. Matriks A Matra Lateral Directional.....	22
Tabel 3.2.h. Matriks B Matra Lateral Directional.....	22
Tabel 3.2.i. Matriks C Matra Lateral Directional.....	22
Tabel 3.2.j. Matriks D Matra Lateral Directional	22
Tabel 3.2.k. Matriks K Matra Lateral Directional.....	23
Tabel 3.2.l. Matriks $x(0)$ Matra Lateral Directional.....	23

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penerapan sistem pengendalian sudah menjadi hal yang asing lagi dalam kehidupan modern. Dengan berkembang pesatnya teknologi pengendalian dalam bidang industri, proses produksi dapat berlangsung tanpa henti sehingga memperoleh keuntungan yang maksimal. Namun pengendalian tersebut juga mulai diterapkan pada bidang kemiliteran. Akhir-akhir ini, sering timbul konflik antar negara seperti masalah wilayah teritorial, dapat mengakibatkan terjadinya peperangan yang dapat menimbulkan korban tak bersalah. Maraknya tragedi bom di beberapa negara yang disebabkan oleh aksi teroris membuat tingkat kewaspadaan masyarakat meninggi. Terjadinya konflik tersebut disebabkan oleh adanya kesalahpahaman ataupun pihak lain yang mencoba mencari keuntungan. Dengan adanya konflik tersebut, memicu suatu Negara untuk memperkuat pertahanannya demi mempertahankan kedaulatan dan keutuhan wilayah negara tersebut, dengan memanfaatkan teknologi yang lebih maju.

Peralatan militer seperti roket kendali merupakan sesuatu yang wajib dimiliki dalam bidang pertahanan militer Negara. Bagi negara kecil maupun menengah yang teknologinya kurang mendukung, maka harus membeli dari negara produsen. Roket kendali yang digunakan, dan teknologi yang diadopsi jaman dulu sangatlah berbeda dengan komponen elektronik saat ini. Bahkan masih banyak mengandalkan gas generator untuk membangkitkan energy listriknya, dan komponen sensornya maupun electronic guided controlnya dirasa sudah tidak diproduksi lagi saat ini. Walaupun semua itu dapat difungsikan lagi jika diupgrade ke negara produsen, namun hal tersebut dapat menimbulkan ketergantungan terhadap negara produsennya. Tidak jarang saat dilakukan flight test secara langsung, trajektorinya masih melenceng jauh dari yang ditetapkan. Sehingga masalah autopilot roket kendali tersebut menjadi masalah utama dalam bidang ini.

Dalam skripsi ini, akan dilakukan suatu validasi model dinamik dari roket kendali yaitu UAV-200 LAPAN dengan menggunakan algoritma State Space MPC. Model UAV-200 LAPAN yang digunakan berupa model matematikanya dalam bentuk matriks state space. Cara kerja dan konsep dari State Space MPC akan dibahas, kemudian dilihat pengaruhnya terhadap kestabilan sistem, serta sampai sejauh mana State Space MPC dapat berperan mengatasi disturbance yang mengganggu model system tersebut dan seberapa jauh UAV-200 LAPAN dapat diikuti oleh MPC dengan berbagai prediction horizon dan moving horizon.

1.2 Tujuan

Adapun tujuan dari Skripsi adalah sebagai berikut:

- Mencoba mensimulasikan sistem kendali dengan State Space MPC pada model UAV-200 dari LAPAN.

1.3 Pembatasan Masalah

Pada laporan skripsi ini, lingkup permasalahan yang akan dibahas adalah:

1. MPC dan jenis MPC yang digunakan yaitu State Space MPC
2. Penerapan sistem pengendalian UAV-200 LAPAN dengan State Space MPC terhadap disturbance.

1.4 Metodologi Penulisan

Metode penulisan yang dilakukan pada laporan skripsi ini adalah:

- Data-data studi lapangan, berupa pengetahuan baik dari pembimbing maupun percobaan yang dilakukan.
- Data-data studi kepustakaan, berupa literatur dan sumber tertulis lainnya baik dari dalam perusahaan, buku-buku perpustakaan maupun dari media internet yang terkait dengan topik penulisan laporan skripsi ini.

1.5 Sistematika Penulisan

Laporan ini dibagi menjadi beberapa bab. Bab pertama menjelaskan tentang Latar Belakang, Perumusan Masalah, Tujuan, Pembatasan Masalah,

Metodologi Penulisan, dan Sistematika Penulisan. Bab kedua menjelaskan tentang teori yang berhubungan dengan pembahasan skripsi yaitu state space MPC. Bab ketiga menjelaskan tentang sistem yang akan divalidasi yaitu UAV-200. Bab keempat menjelaskan mengenai simulasi sistem dengan menggunakan perangkat lunak MATLAB. Bab kelima menjelaskan tentang kesimpulan dari pembahasan dan simulasi yang telah dilakukan dalam sistem pengendaliannya.

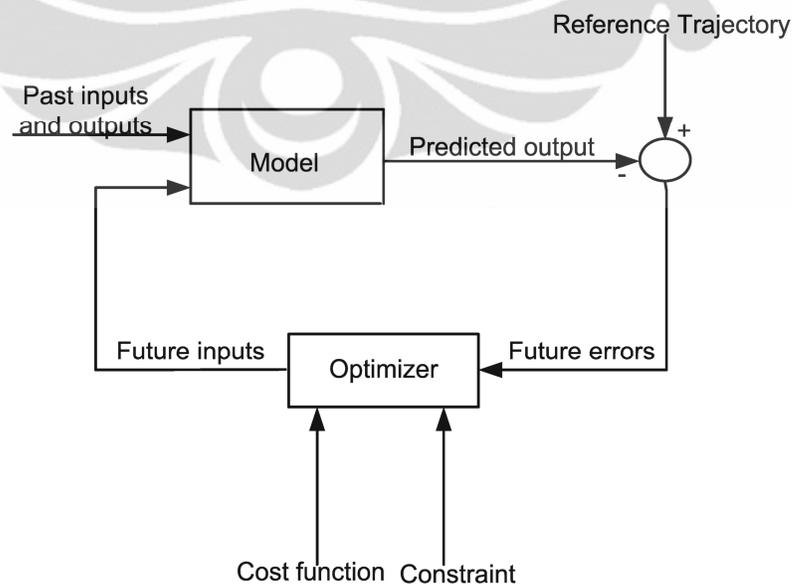


BAB 2

LANDASAN TEORI

2.1 Model Predictive Control (MPC)

Model Predictive Control (MPC) merupakan suatu metode pengendalian yang berbasis model dengan memprediksikan keluaran proses dengan memberikan dan menyesuaikan sinyal kendali ke proses. MPC mengandalkan dinamika proses dari sistem yang dimodelkan, yang paling sering adalah model empirical linear yang didapatkan dari identifikasi sistem. Model yang digunakan ditujukan untuk menggambarkan perilaku dari dinamika sistem yang kompleks. Model akan digunakan untuk memprediksi perilaku dari variable terikat (output) dari sistem dinamik yang dimodelkan sehubungan dengan perubahan variable bebas (input) dalam proses. MPC menggunakan model dan hasil pengukuran plant saat sekarang untuk menghitung nilai selanjutnya dari variable bebas. MPC kemudian mengirimkan sinyal kendali yang berisi kumpulan nilai variable bebas pada langkah-langkah selanjutnya untuk diterapkan dalam proses[7].

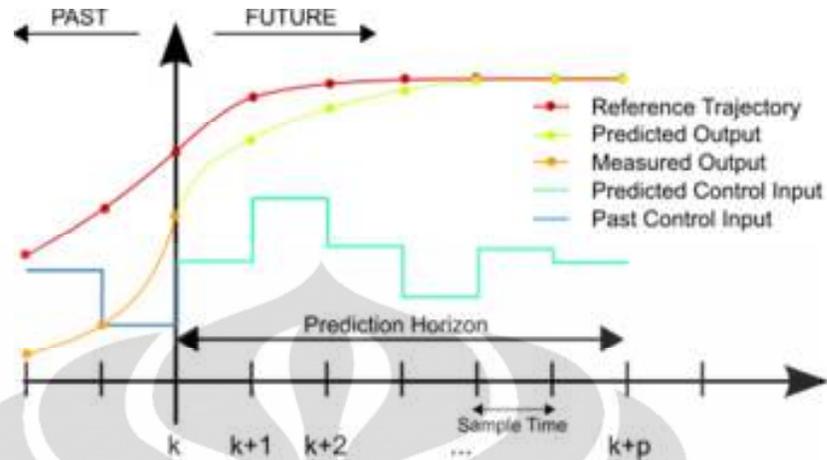


Gambar 2.1 Struktur Dasar MPC[6]

MPC menggunakan fungsi kriteria yang sama dengan LQR (Linear Quadratic Regulator), yakni fungsi kriteria quadratic. Berdasarkan fungsi kriteria ini, MPC menghasilkan control input yang optimal untuk beberapa waktu ke depan (hasil prediksi), tetapi hanya control input saat ini saja yang diterapkan ke plant. Pada waktu berikutnya, perhitungan berdasarkan cost function itu diulangi kembali dan hanya kontrol input saat itu saja yang diterapkan, dan begitu selanjutnya[8].

MPC didasarkan pada iterasi dan optimasi dari beberapa langkah waktu ke depan yang terbatas dari sebuah model plant. Pada waktu t , kondisi dari plant yang sekarang ini disampling dan sebuah strategi pengendalian yang dapat meminimalisasikan biaya akan dilakukan melalui sebuah algoritma minimalisasi algoritma numeric, dalam horizon waktu yang relative singkat di masa depan $[t, t+T]$. Secara khusus sebuah perhitungan online digunakan untuk mengeksplorasi trajektori state dari state yang sekarang dan mencari sebuah strategi pengendalian yang meminimalisasikan biaya sampai waktu $t+T$. Hanya langkah pertama dari strategi pengendalian yang akan diimplementasikan, kemudian kondisi dari plant disampling lagi dan perhitungan akan diulangi mulai dari kondisi plant yang sekarang, menghasilkan sebuah sinyal kendali baru dan jalur kondisi baru yang telah diprediksi. Horizon yang telah diprediksi terus bergeser ke depan dan untuk alasan ini MPC juga disebut pengendalian horizon yang semakin sempit[7].

Pengendalian prediktif merupakan jenis pengendali yang didisain berdasarkan model suatu proses. Model tersebut digunakan untuk menghitung suatu kumpulan data prediksi keluaran proses. Berdasarkan data prediksi tersebut, sinyal kendali yang akan diberikan ke proses dihitung dengan melakukan minimalisasi suatu fungsi kriteria sehingga selisih antara data prediksi keluaran proses tersebut dengan data masukan referensi yang bersesuaian menjadi minimal[5].



Gambar 2.2 Konsep MPC[7]

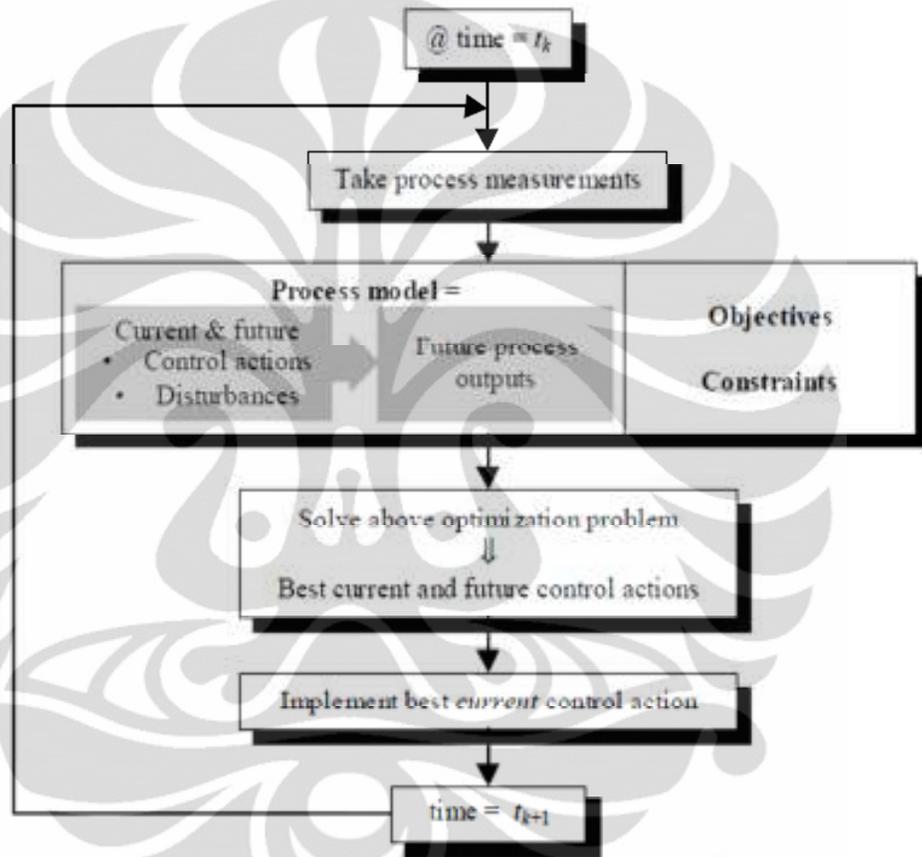
Gambar di atas menunjukkan bahwa model dari proses dihitung berdasarkan prediksi dari kurva lintasan variable yang dikontrol. Setelah mengoreksi ketidaksesuaian antara nilai yang diprediksi dan nilai yang sebenarnya dari variable yang dikontrol, selanjutnya prediksi variable yang dikontrol akan dikurangi dengan lintasan prediksi set poin untuk menghasilkan vector eror yang merupakan input dari algoritma pengendalian dalam mengeluarkan sinyal kendali (variable yang dimanipulasi). Selanjutnya, sinyal kendali inilah yang digunakan untuk mengatur aksi pada actuator agar nilai actual proses dapat mencapai set poin yang diinginkan[5].

MPC merupakan algoritma pengendalian multivariable yang menggunakan[6]

- Model dinamika internal proses
- Tindakan pengendalian pada langkah-langkah sebelumnya
- Optimasi fungsi kriteria J pada horizon terprediksi yang semakin sempit untuk menghitung langkah pengendalian yang optimum

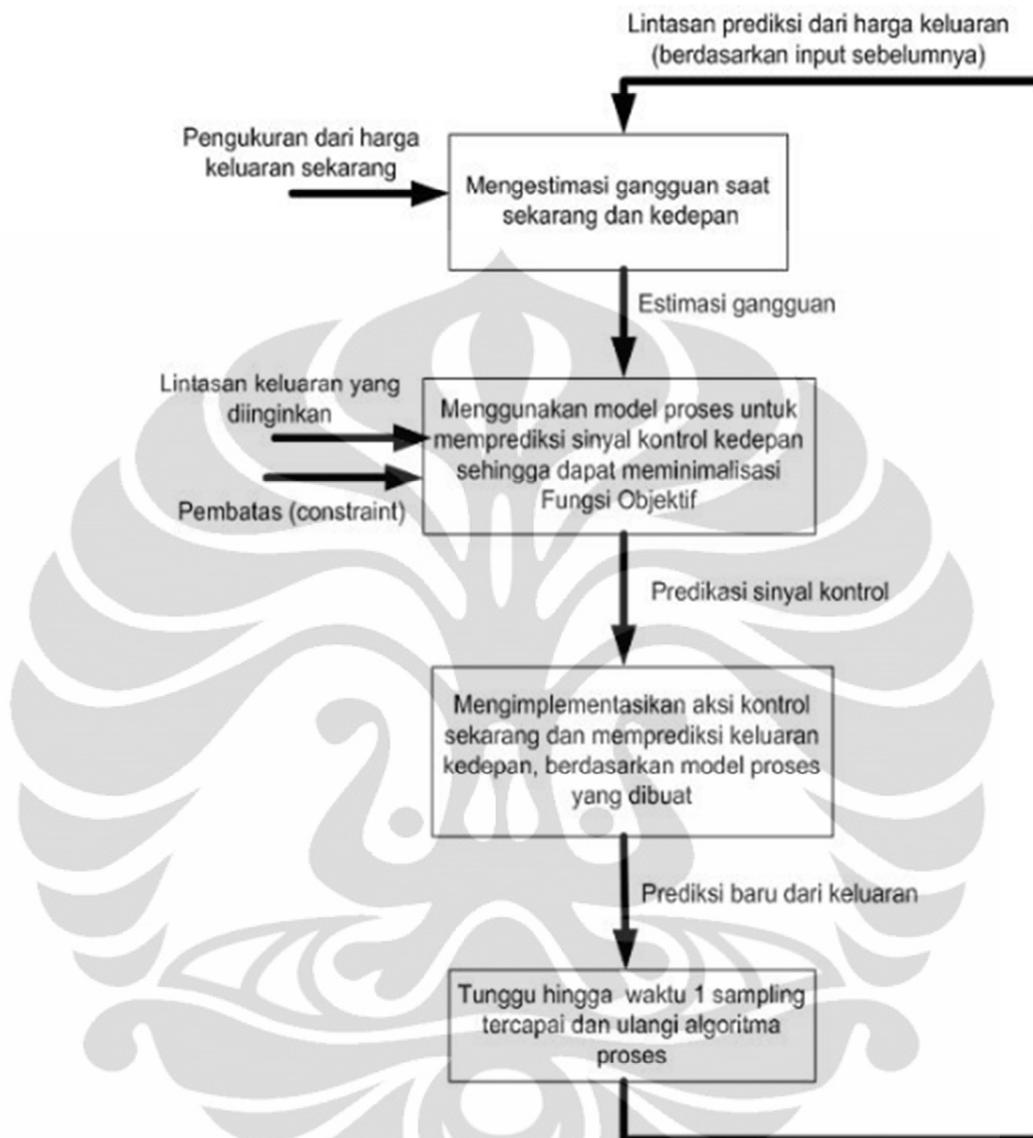
Skema MPC dimulai pada saat t sekarang (misal dinyatakan $t(k)$) dengan melakukan pengukuran pada proses dari sistem. Optimalisasi dari pengontrol MPC sangat bergantung pada kualitas model proses yang didapat. Sementara dalam beberapa kasus, model proses yang didapat terkadang tidak sesuai dengan kondisi proses sesungguhnya, sehingga terkadang perlu dilakukan identifikasi proses secara berulang untuk dapat memastikan akurasi model

proses yang dihasilkan. Walaupun demikian, akurasi model proses tidak akan pernah mencapai kesempurnaan dikarenakan teknik linierisasi suatu elemen proses yang tak linier. Untuk lebih menyempurnakan model proses yang dihasilkan, perbaiki model proses identifikasi dengan pengetahuan tentang proses sebenarnya. Skemanya dapat dilihat sebagai berikut:



Gambar 2.3 Skema Sistem MPC[3]

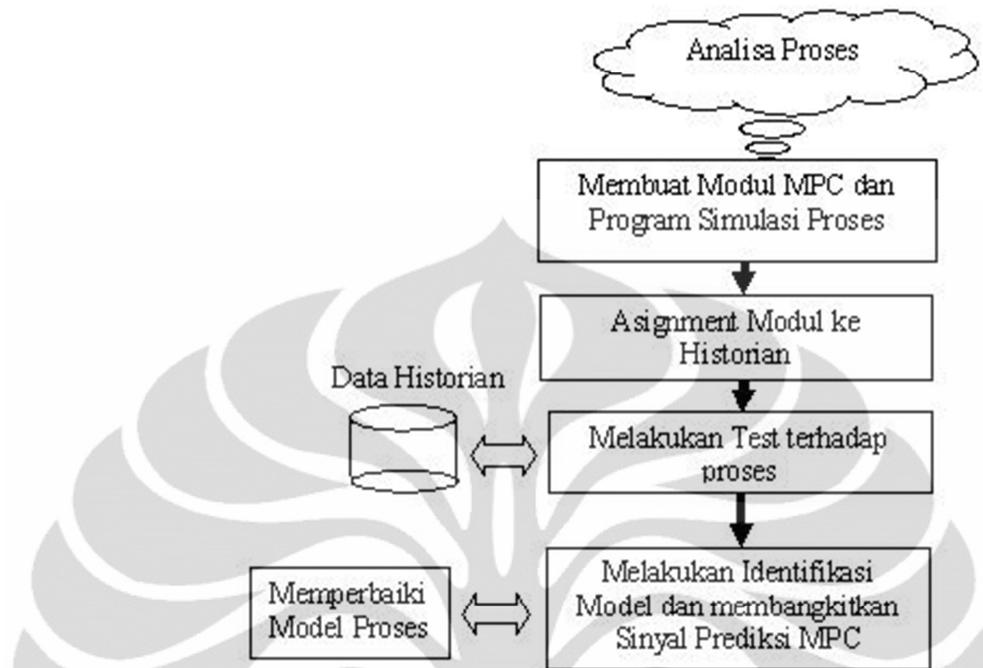
Pengontrol MPC akan bekerja berdasarkan model proses yang dihasilkan. Kemampuan pengontrol MPC dalam mengestimasi adanya gangguan untuk keadaan sekarang dan ke depan termasuk poin yang penting, sehingga performansi dari sistem kendali dapat terjaga pada kondisi kestabilan yang maksimal[5]. Algoritma MPC berupa langkah-langkah secara berurut yang secara umum dapat digambarkan sebagai berikut :



Gambar 2.4 Algoritma MPC[5]

2.2 Perancangan dan Implementasi Strategi Kontrol MPC

Perancangan dan tahap-tahap implementasi strategi kendali MPC dapat dilihat pada diagram blok di bawah ini[5]:



Gambar 2.5 Perancangan MPC[5]

1. Analisa Proses

Tujuan melakukan kajian ilmiah mengenai objek plant yang akan dikendalikan adalah mendapatkan persamaan dinamika dari proses. Analisa proses akan memberikan formulasi yang jelas dari objektif sistem kendali dan keterbatasan (limitations) yang sesuai dengan pengertian MPC sebenarnya. Atau dengan kata lain, analisa proses akan memberikan konfigurasi kendali proses dalam terminologi MPC. Input dan Output proses digolongkan dalam 4 kategori yang berbeda. Pengolongan ini berdasarkan penggunaannya dalam pengontrolan proses:

- *Manipulated Variable (MV)* – input proses yang diatur untuk menjaga output proses tetap pada setpoint.
- *Controlled Variable (CV)* -- output proses yang ingin dijaga pada harga setpoint
- *Disturbance Variable (DV)* – input proses yang memberi efek terhadap proses dan output variabel tertentu

- *Constraint (AV)* – output proses yang harus dijaga dalam suatu definisi batas jangkauan operasi yang berlawanan dengan setpoint. Proses input dan output dapat juga didefinisikan sebagai *teroptimisasi* jika pada proses input atau output tersebut terdapat sebuah pertimbangan ekonomis atau kinerja untuk menekan proses variabel dalam arah yang diinginkan hingga beberapa constraint (batasan) proses menjadi aktif. Pada langkah ini juga akan didefinisikan variabel *Manipulated (MV)*, *Disturbance (DV)*, *Controlled (CV)* dan *Constraint (AV)* yang berkaitan dengan proses yang ingin dikontrol.
2. Membuat modul program MPC. Modul dapat berisikan informasi algoritma, kondisi, *alarm*, *display*, *historical information*, dan karakteristik lainnya yang menyatakan perlengkapan proses. Algoritma adalah langkah-langkah logika yang menyatakan bagaimana modul berfungsi.
 3. Setelah modul program dirancang dalam suatu AREA project, langkah selanjutnya adalah *assignment program* ke *data historian* yang terdapat dalam *database* program DeltaV.
 4. Melakukan tes terhadap proses dan data yang didapat akan disimpan secara otomatis dalam *data historian*. Berdasarkan data-data yang tersimpan dalam *data historian*, selanjutnya akan dibangkitkan model dari proses dan sinyal kendali prediksi.
 5. Setelah identifikasi model proses dan membangkitkan sinyal kendali prediksi, dilakukan perbaikan model sehingga lebih mendekati model proses yang sebenarnya.
 6. Melakukan tes simulasi untuk memastikan algoritma program sudah berjalan dengan benar. Jika unjuk kerja respon pengendalian kurang baik, lakukan tuning parameter pengendali sehingga didapatkan respon terbaik.

2.3 Pengetesan Proses

Pengetesan proses dilakukan dengan tujuan untuk mendapatkan solusi terbaik dari kinerja struktur pengetesan dalam mendapatkan data analisa yang optimal. Ada banyak teknik yang digunakan dalam pengetesan proses. Prosedur pengetesan yang umum dijumpai adalah[5]:

- Step Test

Pengetesan proses dengan cara memberikan perubahan pada input proses secara step sehingga memberi pengaruh pada output proses. Satu kerugian dari penggunaan metode ini adalah bahwa pada kondisi awal dan akhir pengetesan, proses harus dalam keadaan tunak (steadystate). Selain itu untuk mendapatkan hasil yang representatif, perubahan input yang besar sering dibutuhkan untuk mendapatkan hasil jelas yang berada diluar bandwidth noise data proses.

- Pseudo Random Binary Sequence (PRBS)

Prinsip pengetesan proses dengan sinyal PRBS adalah membuat perubahan input kecil secara acak untuk membangkitkan gangguan (perturbation) yang kontinyu pada variabel output. Salah satu keuntungan penggunaan dari pendekatan ini adalah amplitude perubahan input yang dibutuhkan dapat lebih kecil jika dibandingkan dengan perubahan step pada *step testing*. Selain itu, proses pengetesan dapat dilakukan tanpa harus menunggu proses dalam keadaan tunak (steadystate). Jika pengetesan dengan sinyal PRBS dilakukan, sinyal input secara teoritis disebut white (uncorelated) dan akan menghasilkan parameter model estimasi yang lebih baik. Frekuensi dari PRBS dapat dipilih untuk putaran (flips) cepat (*fast*) atau lambat (*slow*).Pemilihan frekuensi ini dapat menentukan jenis informasi terbaik yang akan didapat, misalnya untuk *fast* akan memberikan informasi yang akurat mengenai deadtime, *slow* akan memberikan informasi *steadystate gain* yang tepat sedangkan medium memberikan informasi *time constant* lebih baik. Dalam disain pengetesan proses dengan metoda PRBS, terdapat beberapa parameter penting yang akan memberikan pengaruh terhadap kualitas data yang didapatkan:

- i. Fluktuasi Amplitudo (Move Amplitude)

Pilih amplitudo yang cukup besar sehingga dapat dilihat efek dari pergerakan data, tetapi jangan terlalu besar sehingga dapat menyebabkan proses *upset* (di atas setpoint) dan operator harus mengkompensasi variabel yang lain. Jika dibutuhkan, amplitudo dapat diatur selama proses pengetesan. Jika proses mengalami *noise* yang tinggi dan pergerakan input terlalu kecil sehingga berada dalam daerah *bandwith noise*, maka perioda pengetesan harus lebih lama untuk mendapatkan kualitas model yang lebih baik.

ii. Frekuensi (Flip Time)

Flip time seharusnya ditentukan sebagai fungsi dari proses, normalnya antara 1/5 hingga 1 kali konstanta waktu proses (*time constant*) Untuk *noise* yang besar, frekuensi PRBS yang rendah (flop time lama) akan dapat menentukan nilai *gain* yang sesuai.

iii. Lama Pengetesan (*Tes Length*)

Lama pengetesan akan menentukan kualitas data yang didapatkan. Untuk *noise* yang besar, membutuhkan waktu pengetesan yang lebih lama. Lama pengetesan direduksi oleh peningkatan pergerakan amplituda dan peningkatan *signal to noise ratio* (SNR).

2.4 State Space MPC

State Space MPC merupakan salah satu metode MPC dalam melakukan pengendalian terhadap suatu sistem. Model proses yang digunakan pada teknik kendali ini adalah model state space dari sistem :

$$\begin{aligned}x(k + 1) &= Ax(k) + B_u u(k) + B_d d(k) + w(k) \\z(k) &= Cx(k) \\y(k) &= Cx(k) + v(k)\end{aligned}\tag{1}$$

dimana $y(k)$ dan $z(k)$ adalah output yang terukur dan output dengan bebas noise, $u(k)$ adalah control input, $w(k)$ dan $v(k)$ adalah noise diasumsikan sebagai white noise, $d(k)$ adalah disturbance yang terukur.

Integrator dihasilkan dengan menggunakan extended model state space yang menggunakan state vector yang didifferensialkan $\Delta x(k) = x(k) - x(k - 1)$ dan output $z(k)$ sebesar 1, sehingga menghasilkan :

$$\begin{aligned} \begin{bmatrix} \Delta x(k+1) \\ z(k+1) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B_u \\ CB_u \end{bmatrix} \Delta u(k) + \begin{bmatrix} B_d \\ CB_d \end{bmatrix} \Delta d(k) + \begin{bmatrix} I \\ C \end{bmatrix} \Delta w(k) \\ z(k) &= [0 \quad I] \begin{bmatrix} \Delta x(k) \\ z(k) \end{bmatrix} \\ y(k) &= z(k) + v(k) \end{aligned} \quad (2)$$

Atau dapat dituliskan sebagai :

$$\begin{aligned} \bar{x}(k+1) &= \bar{A}\bar{x}(k) + \bar{B}u\Delta u(k) + \bar{B}d\Delta d(k) + \Delta\bar{w}(k) \\ z(k) &= \bar{C}\bar{x}(k) \\ y(k) &= z(k) + v(k) \end{aligned} \quad (3)$$

State vector diestimasi dengan menggunakan state observer berdasarkan persamaan (3), diberikan dalam persamaan berikut :

$$\begin{aligned} \varepsilon(k) &= y(k) - \bar{C}\hat{x}(k|k-1) \\ \hat{x}(k+1|k) &= \bar{A}\hat{x}(k|k-1) + \bar{B}_u\Delta u(k) + \bar{B}_d\Delta d(k) + K\varepsilon(k) \end{aligned} \quad (4)$$

Observer akan menghasilkan prediksi satu langkah ke depan dari extended state vector. Prediksi selanjutnya didapatkan dengan mengulang persamaan (4) dengan asumsi $\Delta u(k) = 0, k > m$ $\Delta d(k) = 0, k > 1$, dan $\varepsilon(k) = 0, k > 1$. Perkalian dengan \bar{C} menghasilkan prediksi besar z, berdasarkan state yang diperkirakan, hasil ukuran yang sebenarnya dan langkah output yang dimanipulasi selanjutnya. Output diprediksi sebesar p langkah ke depan dan sinyal kendali prediksi sebesar m langkah ke depan, $m \leq p$.

$$U(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+m-1) \end{bmatrix}, \quad Z(k) = \begin{bmatrix} z(k) \\ \vdots \\ z(k+p-1) \end{bmatrix} \quad (5)$$

Kemudian proses variable yang diprediksi :

$$Z(k+1|k) = \begin{bmatrix} \bar{C}\bar{A} \\ \bar{C}\bar{A}^2 \\ \vdots \\ \bar{C}\bar{A}^p \end{bmatrix} \hat{x}(k|k-1) + \begin{bmatrix} \bar{C}\bar{B}_u & 0 & \dots & 0 \\ \bar{C}\bar{A}\bar{B}_u & \bar{C}\bar{B}_u & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}\bar{A}^{(p-1)}\bar{B}_u & \dots & \dots & \bar{C}\bar{A}^{(p-m)}\bar{B}_u \end{bmatrix} \Delta U(k) + \begin{bmatrix} \bar{C}\bar{B}_d \\ \bar{C}\bar{A}\bar{B}_d \\ \vdots \\ \bar{C}\bar{A}^{(p-1)}\bar{B}_d \end{bmatrix} \Delta d(k) + \begin{bmatrix} \bar{C}K \\ \bar{C}\bar{A}K \\ \vdots \\ \bar{C}\bar{A}^{(p-1)}K \end{bmatrix} \varepsilon(k) \quad (6)$$

Atau dapat dituliskan seperti persamaan berikut :

$$Z(k+1|k) = S^X \hat{x}(k|k-1) + S^U \Delta U(k) + S^d \Delta d(k) + S^e \varepsilon(k) \quad (9)$$

$\varepsilon(k)$ menunjukkan feedback berdasarkan besar $y(k)$ yang paling baru.

Control error berdasarkan prediction horizon adalah perbedaan antara prediksi dan trajectory dari set poin yang selanjutnya

$$E(k+1) = Z(k+1|k) - R(k+1) \quad (8)$$

Nilai perubahan sinyal kendali prediksi didapatkan dengan meminimasi fungsi kriteria yang ditentukan yang umumnya merupakan selisih antara keluaran prediksi dengan trajektori acuan, dengan persamaan sebagai berikut :

$$V(k) = \sum_{i=1}^p Q_i \|\hat{y}(k+i|k) - r(k+i|k)\|^2 + \sum_{i=0}^{m-1} R_i \|\Delta \hat{u}(k+i|k)\|^2 \quad (9)$$

dimana :

$\hat{y}(k+i|k)$: estimasi output prediksi i langkah ke depan saat k

$r(k+i|k)$: trajektori acuan i langkah ke depan saat k

$\Delta \hat{u}(k+i|k)$: estimasi perubahan sinyal kendali prediksi i langkah ke depan

Q_i dan R_i : matriks faktor bobot

Matriks bobot Q dan R merupakan matriks diagonal dengan persamaan $Q = c_1 I_{M \times p}$ dan $R = c_2 I_{N \times m}$ dimana c_1 dan c_2 merupakan nilai scalar tertentu yang besarnya dipilih berdasarkan kecocokan sistem. Nilai c_1 dan c_2 ditentukan dengan melihat nilai matriks bobot yang menghasilkan keluaran sistem terbaik mengikuti sinyal referensi.

Untuk meminimasi fungsi kriteria tersebut, dibutuhkan matriks E(k), yaitu selisih antara nilai trajektori acuan yang akan datang dengan respon bebas sistem. Respon bebas adalah nilai keluaran di sepanjang horizon saat tidak ada perubahan sinyal kendali ($\Delta U(k) = 0$). Persamaan matematis dari E(k) sebagai berikut :

$$E(k) = R(k) - S^X x(k) - S^U u(k-1) \quad (10)$$

Persamaan tersebut kemudian diubah menjadi suatu persamaan yang mengandung E(k) dan $\Delta U(k)$ seperti berikut :

$$\begin{aligned} V(k) &= Q \|S^X \Delta U(k) - E(k)\|^2 + R \|\Delta U(k)\|^2 \\ &= \left(\Delta U^T(k) S^{X^T} - E^T(k) \right) Q (S^X \Delta U(k) - E(k)) + \Delta U^T(k) R \Delta U(k) \\ &= E^T(k) Q E(k) - \Delta U^T(k) 2 S^{X^T} Q E(k) + \Delta U^T(k) (S^{X^T} Q S^X + R) \Delta U(k) \\ V(k) &= c + \Delta U^T(k) G + \Delta U^T(k) H + \Delta U(k) \end{aligned} \quad (11)$$

Nilai $E^T(k) Q E(k)$ merupakan nilai konstan karena tidak mengandung $\Delta U(k)$. Nilai minimum bisa didapat dengan mencari turunan pertama atau gradien $\nabla_{\Delta U(k)} V(k) = 0$. Persamaan gradient tersebut menjadi :

$$\nabla_{\Delta U(k)} V(k) = -G + 2H \Delta U(k) \quad (12)$$

Sehingga dengan $\nabla_{\Delta U(k)} V(k) = 0$ didapat $\Delta U(k)$ optimal :

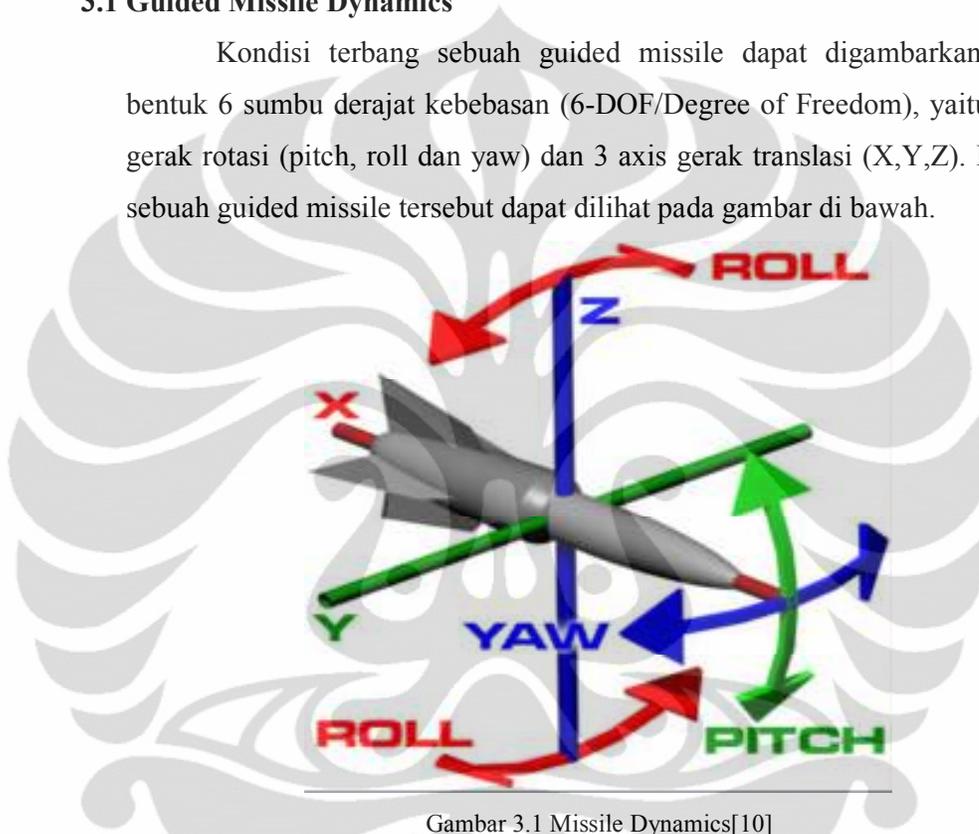
$$\Delta U(k)_{opt} = \frac{1}{2} H^{-1} G \quad (13)$$

BAB 3

DESKRIPSI SISTEM

3.1 Guided Missile Dynamics

Kondisi terbang sebuah guided missile dapat digambarkan dalam bentuk 6 sumbu derajat kebebasan (6-DOF/Degree of Freedom), yaitu 3 axis gerak rotasi (pitch, roll dan yaw) dan 3 axis gerak translasi (X,Y,Z). Ilustrasi sebuah guided missile tersebut dapat dilihat pada gambar di bawah.



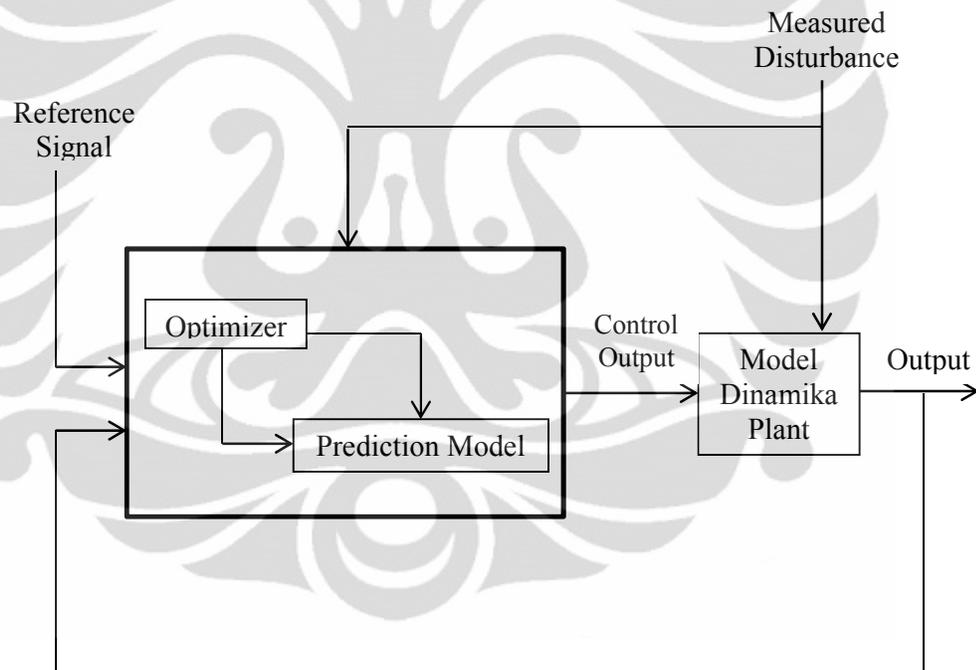
Gambar 3.1 Missile Dynamics[10]

Gerakan roll merupakan gerakan memutar yang berporos pada sumbu x. Sedangkan gerakan pitch adalah gerakan memutar yang berporos pada sumbu y. Adapun gerakan yaw adalah gerakan memutar yang berporos pada sumbu z. Dengan kondisi terbang yang seperti itu, sebuah missile akan cenderung untuk tumbling atau wobbling jika sistem stability controlnya tidak bekerja secara akurat. Disinilah letak fungsi sistem autopilot, dengan MPC diharapkan kondisi tersebut dapat diatasi dengan cepat dan tegas. MPC tersebut akan mendapat masukan 6-DOF sensor yang dipasang pada Centre of Gravity, C_G missile.

3.2 UAV-200

Jenis Guided Flight Vehicle yang akan digunakan adalah guided missile (roket kendali berdiameter 200 mm) yang selanjutnya disebut UAV-200. UAV-200 ini merupakan hasil rancangan LAPAN. UAV-200 ini dalam design awalnya dirancang sebagai berikut[1] :

- Diameter 200 mm
- Speed 0.6 Mach (740 km/jam)
- Cruise altitude 500-700 m
- Tail control model, no aileron
- Booster – Sustained scheme



Gambar 3.2 Blok Diagram Model Dinamika dengan MPC[1]

Dalam design guided missile tersebut, autopilot sistem mempunyai 2 fungsi yang sangat vital yaitu mengontrol stabilitas terbang wahana dan mengantarkan wahana sampai mendekati koordinat target pada range tertentu,

dan proses detail selanjutnya akan lebih banyak dihandle oleh seeker sistem. Tahapan proses tersebut adalah sebagai berikut :

1. Sebelumnya wahana akan diluncurkan dengan roket pendorong (Booster) berbahan bakar padat.
2. Setelah booster ini lepas, gaya dorong wahana akan digantikan oleh sustainer rocket / turbo jet. Dari tahap inilah fungsi autopilot dimulai.
3. Sistem autopilot akan menggerakkan flap tail control untuk menjaga stabilitas, dan terbang melalui way points menuju sasaran.

Guided missile ini dalam desainnya dirancang tidak mempunyai aileron sebagaimana layaknya pesawat terbang pada umumnya. Wing di bagian tengah hanya untuk menjaga keseimbangan dan membuat wahana dapat terbang datar dengan angle of attack tertentu. Seluruh kendali akan dibebankan pada tail flap yang dikontrol penuh oleh sistem autopilot[1].

Untuk mendeteksi seluruh attitude guided missile tersebut digunakan sensor IMU 6-DOF (terdiri atas 3 axis gyro, 3 axis accelerometer dan 3 axis magnetometer), sensor navigasi GPS dan sensor moving flap encoder, termasuk pula sistem actuator dan given module untuk missile airframe.

Plant dikendalikan dengan melakukan penyederhanaan (linearisasi) menjadi gerakan longitudinal dan lateral. Microcontroller didisain untuk continuous time maupun discrete time agar dapat diimplementasikan pada kendali roll, pitch dan yaw, serta memastikan dinamika close loop yang diinginkan pada saat ketidakpastian, gangguan, dan derau yang muncul.

Roket ini didisain dengan tail control model dan motor berbahan bakar propelan padat. Actuator tail mempunyai 4 control surface (fin) yang bergerak secara mandiri pada porosnya. Controller akan memberikan sinyal masukan kepada 4 servo aktuator untuk menggerakkan fin. Pada kondisi akselerasi normal, data roll / pitch / yaw diukur oleh accelerometer dan rate gyro yang berasal dari 6-DOF sensor, yang nantinya akan memberikan sinyal feedback kepada controller[1].

Penurunan state space untuk missile dilakukan secara terpisah untuk gerakan / matra longitudinal (bidang X-Z) dan lateral directional (bidang X-Y

dan Y-Z). State (keadaan) yang berada di dalam matra longitudinal tersebut adalah[1] :

- U : kecepatan pada arah sumbu X (m/s)
- α : sudut serang (rad)
- θ : sudut pitch (rad)
- q : pitch rate (rad/sec)

Sedangkan state yang berada di dalam matra lateral directional adalah :

- β : sudut slip samping (rad)
- ϕ : sudut roll (rad)
- p : roll rate (rad/sec)
- r : yaw rate (rad/sec)

Persamaan state space matra longitudinal mempunyai matriks[1] :

$$\begin{Bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{Bmatrix} = \begin{bmatrix} x_u & x_\alpha & x_\theta & x_q \\ z_u & z_\alpha & z_\theta & z_q \\ j_1 & j_2 & j_3 & \frac{v_o}{c} \\ m_M & m_M & m_M & m_M \end{bmatrix} \cdot \begin{Bmatrix} u \\ \alpha \\ \theta \\ q \end{Bmatrix} + \begin{bmatrix} x_{\delta_e} \\ z_{\delta_e} \\ f \\ m_{\delta_e} \end{bmatrix} \{\delta_e\} \quad (14)$$

$$\begin{Bmatrix} y1 \\ y2 \\ y3 \\ y4 \end{Bmatrix} = \begin{bmatrix} 1 & a_1 & a_2 & a_3 \\ b_1 & \frac{180}{\pi} & b_2 & b_3 \\ c_1 & c_2 & \frac{180}{\pi} & c_3 \\ d_1 & d_2 & d_3 & \frac{180}{\pi} \cdot \frac{v_o}{b} \end{bmatrix} \cdot \begin{Bmatrix} u \\ \alpha \\ \theta \\ q \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \{\delta_e\} \quad (15)$$

Sedangkan matra lateral directional mempunyai matriks[1] :

$$\begin{Bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{Bmatrix} = \begin{bmatrix} y_\beta & y_\phi & y_p & y_r \\ k_1 & k_2 & 2 \frac{v_o}{b} & k_3 \\ l_\beta & l_1 & l_p & l_r \\ n_\beta & n_1 & n_p & n_r \end{bmatrix} \cdot \begin{Bmatrix} \beta \\ \phi \\ p \\ r \end{Bmatrix} + \begin{bmatrix} y_{\delta_a} & y_{\delta_r} \\ 0 & 0 \\ l_{\delta_a} & l_{\delta_r} \\ n_{\delta_a} & n_{\delta_r} \end{bmatrix} \begin{Bmatrix} \delta_a \\ \delta_r \end{Bmatrix} \quad (16)$$

$$\begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} = \begin{bmatrix} \frac{180}{\pi} & a_1 & a_2 & a_3 \\ b_1 & \frac{180}{\pi} & b_1 & b_1 \\ c_1 & c_2 & \frac{180}{\pi} \cdot \frac{V_o}{b} & c_3 \\ d_1 & d_2 & d_3 & \frac{180}{\pi} \cdot \frac{V_o}{b} \end{bmatrix} \cdot \begin{Bmatrix} \beta \\ \phi \\ p \\ r \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta_a \\ \delta_r \end{Bmatrix} \quad (17)$$

dengan y_1 , y_2 , y_3 , dan y_4 merupakan keluaran sistem yang ingin dipantau, sehingga didapatkan[1] :

$$\begin{aligned} \frac{dx}{dt} &= Ax(t) + Bu(t) + Ke(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{aligned} \quad (18)$$

A=

$$\begin{bmatrix} -0.0033023 & -0.0047349 & -0.027142 & -0.053565 \\ 0.18869 & -0.072498 & -0.28092 & -0.17847 \\ 0.083629 & -1.9967 & -2.5143 & 2.1588 \\ 0.10748 & -1.2199 & -36.108 & 0.0045206 \end{bmatrix}$$

B=

$$\begin{bmatrix} 0.072309 \\ 0.44139 \\ 0.16886 \\ 26.084 \end{bmatrix}$$

C=

$$\begin{bmatrix} 6756.1 & -997.61 & 85.131 & -1.1602 \\ 1.4025 & -0.1358 & 1.8293 & 0.03978 \\ 1.6214 & 12.041 & 0.9647 & 0.015358 \\ 1.7041 & -2.2829 & -0.71929 & 2.2621 \end{bmatrix}$$

D=

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

K=

$$\begin{bmatrix} 0.020431 & -0.043918 & 0.01899 & 0.11641 \\ -0.078755 & 0.22033 & -0.57745 & 0.71378 \\ -2.0676 & 5.7993 & -6.0704 & -0.95548 \\ -2.3214 & 36.652 & -86.097 & 30.188 \end{bmatrix}$$

x(0)=

$$\begin{bmatrix} -0.12653 \\ -1.142 \\ -1.5856 \\ 5.9206 \end{bmatrix}$$

Sedangkan untuk matriks lateral directionalnya :

A=

$$\begin{bmatrix} 0.31214 & 0.00017775 & 0.18241 & 1.2228 \\ 0.16623 & -0.70382 & 9.2507 & -1.4765 \\ -0.0628 & -9.1178 & -0.20375 & -0.64504 \\ -1.7858 & 1.1911 & -1.0161 & -5.1241 \end{bmatrix}$$

B=

$$\begin{bmatrix} 0.24306 & 0.0065519 \\ -0.11684 & -0.50048 \\ -2.4968 & 4.6704 \\ -17.685 & -1.1581 \end{bmatrix}$$

C=

$$\begin{bmatrix} -0.016539 & 1.5101 & 0.22822 & -0.044744 \\ -22.332 & 0.16362 & -0.038845 & -0.40372 \\ -5.4017 & 0.15725 & -3.7315 & -22.128 \\ -0.3145 & -1.9117 & 11.952 & -1.3882 \end{bmatrix}$$

D=

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

K=

$$\begin{bmatrix} 0.025025 & -0.35723 & -0.028558 & -0.024558 \\ 3.9196 & 0.46907 & 0.25254 & -2.5861 \\ 1.6535 & -0.14015 & -0.12577 & 2.1522 \\ 1.1606 & -17.576 & -0.5217 & 0.2 \end{bmatrix}$$

x(0)=

$$\begin{bmatrix} -7.7948e - 005 \\ -0.004556 \\ -0.0023018 \\ 0.00013901 \end{bmatrix}$$

Persamaan model dinamik tersebut mengacu kepada paper Bapak Herma Yudhi Irwanto selaku staff LAPAN yang sedang menjalankan pendidikan doctor di Universitas Indonesia. Data-data dari matriks matra longitudinal dan lateral directional ini yang akan disimulasikan dengan menggunakan MATLAB yang akan dibahas dalam bab selanjutnya.

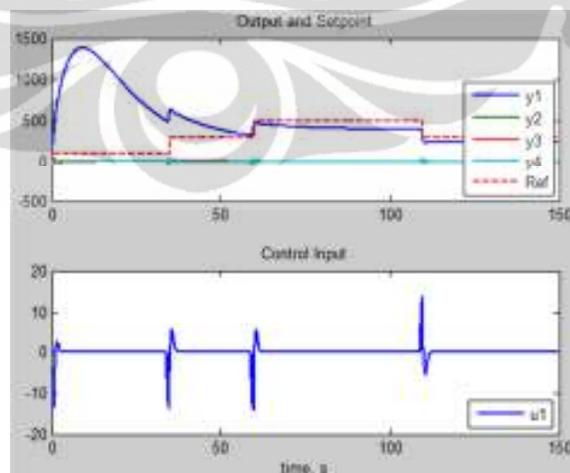
BAB 4

SIMULASI DAN ANALISIS

Simulasi dari sistem UAV-200 LAPAN ini akan dilakukan dengan menggunakan perangkat lunak computer, yaitu MATLAB dengan versi MATLAB 7.6.0 (R2008a). Simulasi yang akan dilakukan ini akan melihat seberapa besar prediction horizon dan moving horizon yang harus digunakan agar mendapatkan hasil pengendalian yang baik dan juga melihat sejauh mana sistem dapat stabil mengikuti referensi yang diberikan dengan adanya disturbance. Semakin besar disturbancenya maka sistem akan cenderung tidak stabil sampai batas dimana sistem tidak dapat lagi mengikuti referensi.

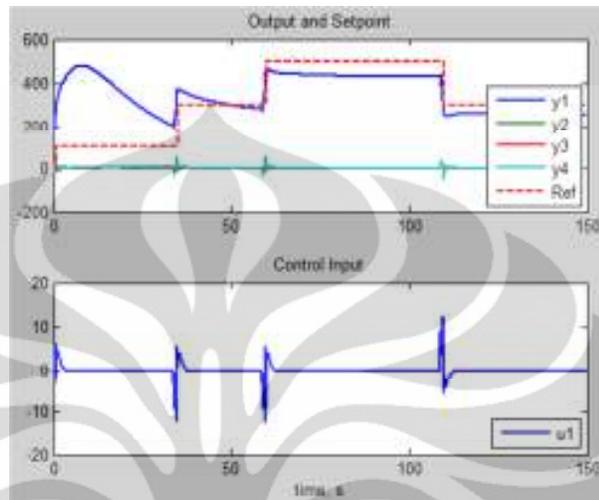
4.1 Simulasi Matra Longitudinal

Simulasi pada matra longitudinal ini, diawali dengan mencari besar prediction horizon dan moving horizon yang digunakan agar hasil pengendalian yang didapat baik. Pertama, akan coba disimulasikan dengan menggunakan prediction horizon (p) sebesar 10 dan moving horizon (m) sebesar 2 sebagai berikut

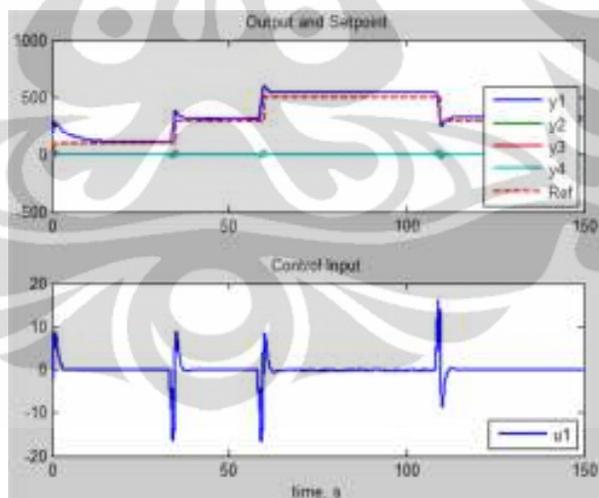


Gambar 4.1 Hasil Simulasi Matra Longitudinal dengan $p=10$ dan $m=2$

Dapat dilihat pada gambar di atas, p dan m yang digunakan masih belum optimal. Berhubung semakin besarnya nilai p dan m maka akan semakin baik keluarannya, maka akan dicoba dengan beberapa p dan m yang lebih besar.



Gambar 4.2 Hasil Simulasi Matra Longitudinal dengan $p=15$ dan $m=3$

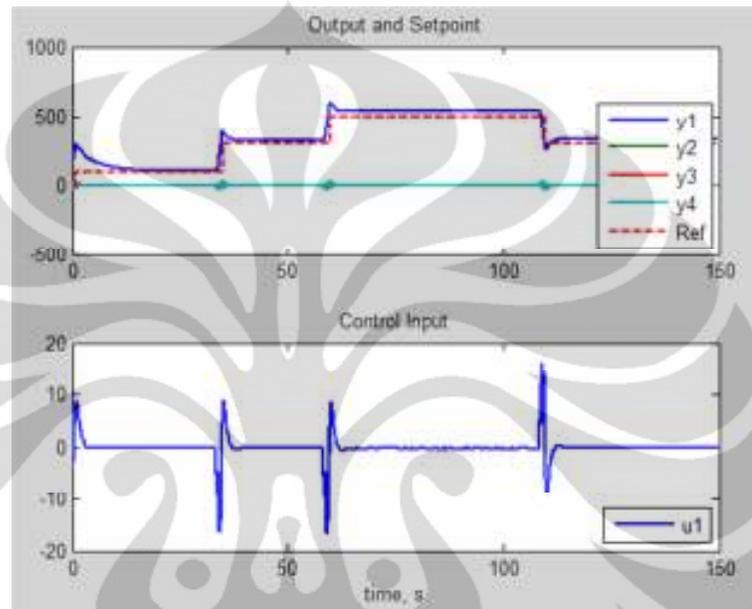


Gambar 4.3 Hasil Simulasi Matra Longitudinal dengan $p=18$ dan $m=3$

Dari simulasi yang dilakukan dengan mencoba nilai prediction horizon dan moving horizon yang tepat, seperti yang terlihat pada gambar di atas, didapat bahwa untuk matra longitudinal, prediction horizon dan moving horizon yang baik untuk digunakan adalah masing-masing sebesar 18 dan 3.

Kemudian akan dilihat seberapa besar disturbance yang dapat diatasi pada matra longitudinal. Hasil simulasi yang didapatkan adalah

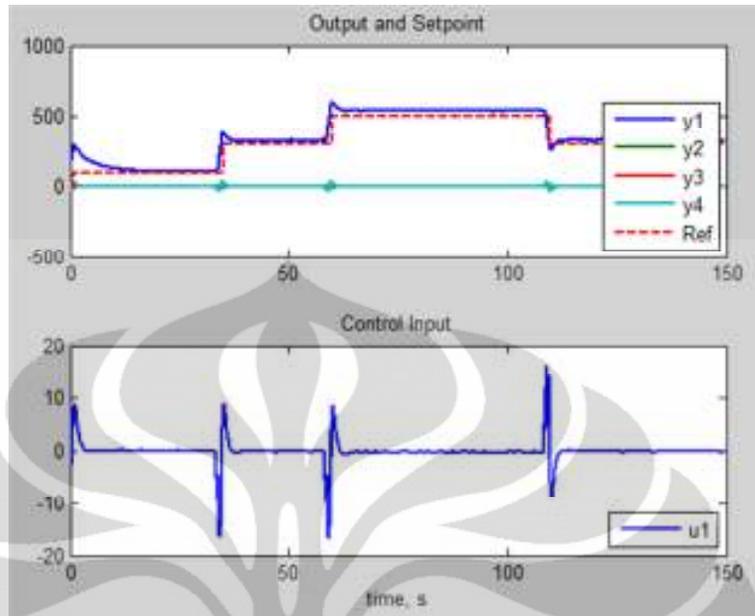
1. Disturbance = 0.1%



Gambar 4.4 Hasil Simulasi Matra Longitudinal dengan Disturbance 0.1%

Simulasi dilakukan dalam waktu 150 detik, menggunakan prediction horizon (p) dan moving horizon (m) sebesar 18 dan 3, seperti yang telah disimulasikan sebelumnya, dengan adanya disturbance sebesar 0,1%. Hasil simulasi dengan besar disturbance 0,1% pada gambar di atas, dapat dilihat bahwa y1 menunjukkan hasil yang mendekati referensi dibanding yang output yang lain. Walaupun pada referensi awal terdapat overshoot dan juga waktu transien yang cukup signifikan, namun pada referensi berikutnya sistem sudah dapat menyesuaikan, sehingga hasil outputnya mulai bisa mengikuti referensi yang ditentukan.

2. Disturbance = 1%

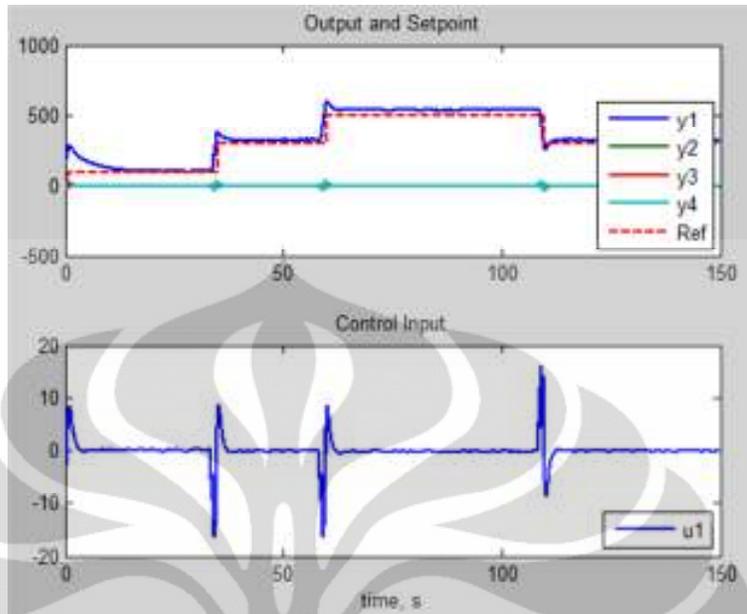


Gambar 4.5 Hasil Simulasi Matra Longitudinal dengan Disturbance 1%

Pada simulasi ini juga digunakan p dan m yang sama dengan sebelumnya, namun bertambah disturbance menjadi 1%. Dapat dilihat bahwa hasil yang didapat masih baik mengikuti referensi hampir sama dengan hasil disturbance 0,1%. Namun di sini dapat dilihat bahwa pada saat sistem mempertahankan nilai pada referensi tertentu, terjadi munculnya beberapa ripple yang cukup kecil.

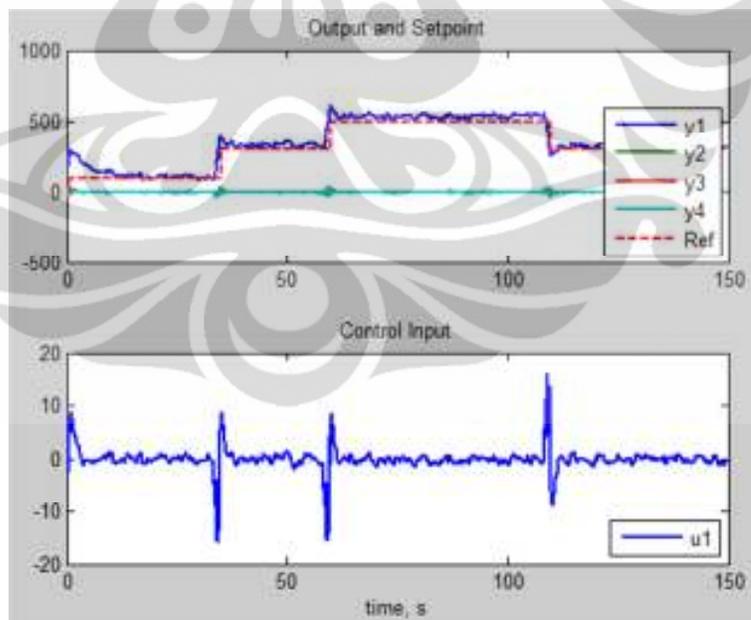
3. Disturbance = 2%.

Simulasi dengan disturbance sebesar 2% ini menggunakan p dan m yang sama. Seperti yang terlihat pada gambar di bawah ini, hasil simulasi dengan menggunakan disturbance yang bertambah menjadi sebesar 2%, walaupun hasil yang didapat masih cukup baik, namun ripple yang muncul bertambah banyak dan juga lebih besar dari sebelumnya.



Gambar 4.6 Hasil Simulasi Matra Longitudinal dengan Disturbance 2%

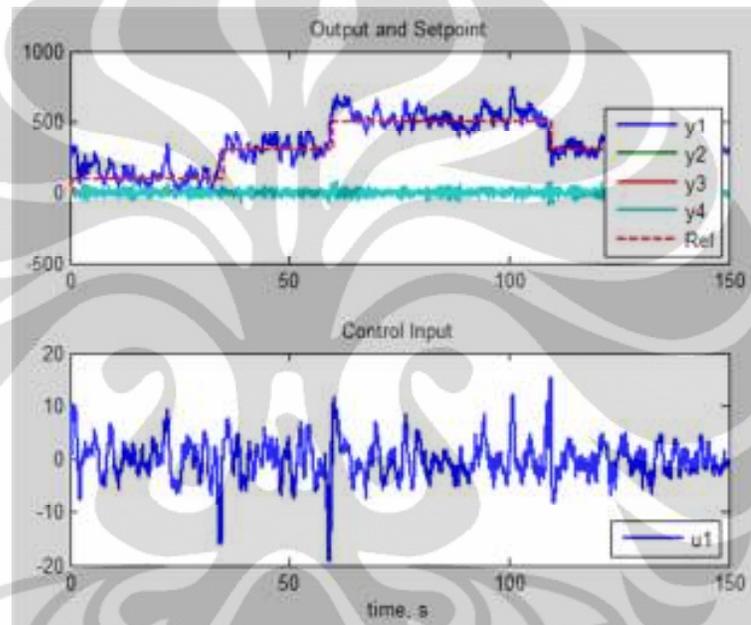
4. Disturbance = 5%



Gambar 4.7 Hasil Simulasi Matra Longitudinal dengan Disturbance 5%

Simulasi sistem tetap menggunakan p dan m yang sama, namun dengan bertambahnya disturbance sebesar 5%, ripple yang muncul menjadi bertambah banyak dari disturbance 2%. Pada gambar di atas dapat dilihat pula bahwa

control input yang diberikan pun mulai menjadi tidak teratur yang tentunya mengakibatkan hasil keluaran sistem agak tidak sesuai dengan referensinya.. Hal ini menunjukkan bahwa sistem sudah mendekati batas untuk dapat mengikuti referensi yang ditentukan. Maka dapat dilihat pada gambar di bawah ini jika disturbance bertambah lebih besar lagi, sistem akan menjadi tidak stabil.

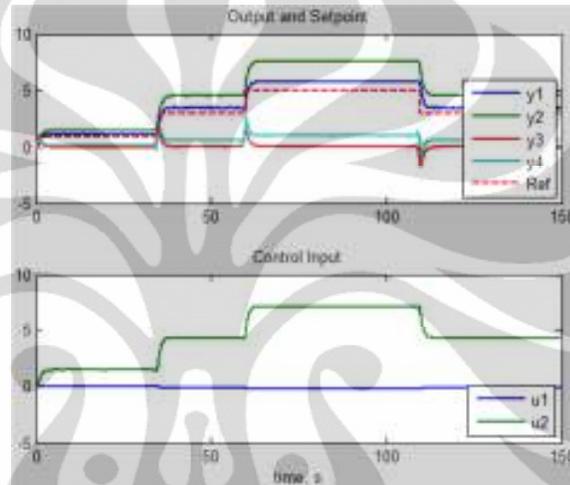


Gambar 4.8 Hasil Simulasi Matra Longitudinal dengan Disturbance 10%

Hasil simulasi menggunakan disturbance sebesar 10% ini menunjukkan bahwa sistem sudah tidak dapat lagi mengikuti referensi yang diberikan, sehingga batas disturbance yang dapat digunakan pada UAV-200 dengan menggunakan state space MPC pada matra longitudinal untuk sistem pengendaliannya adalah sebesar 5%.

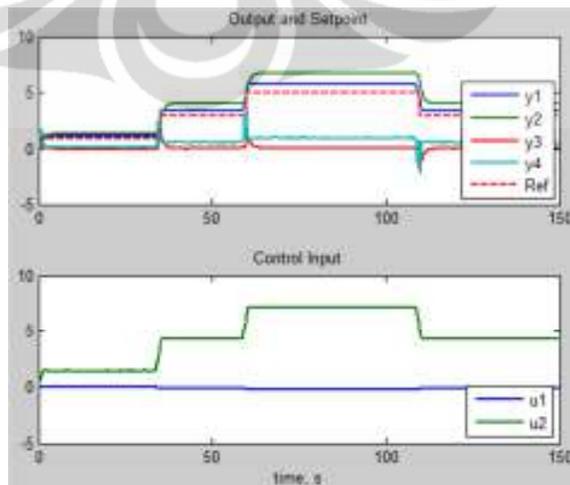
4.2 Simulasi Matra Lateral Directional

Seperti yang telah dilakukan pada matra longitudinal, maka simulasi pada matra lateral directional ini juga diawali dengan mencari besar prediction horizon dan moving horizon yang digunakan agar hasil pengendalian yang didapat baik. Pertama, akan coba disimulasikan mulai dengan menggunakan prediction horizon (p) sebesar 10 dan moving horizon (m) sebesar 2 sebagai berikut.

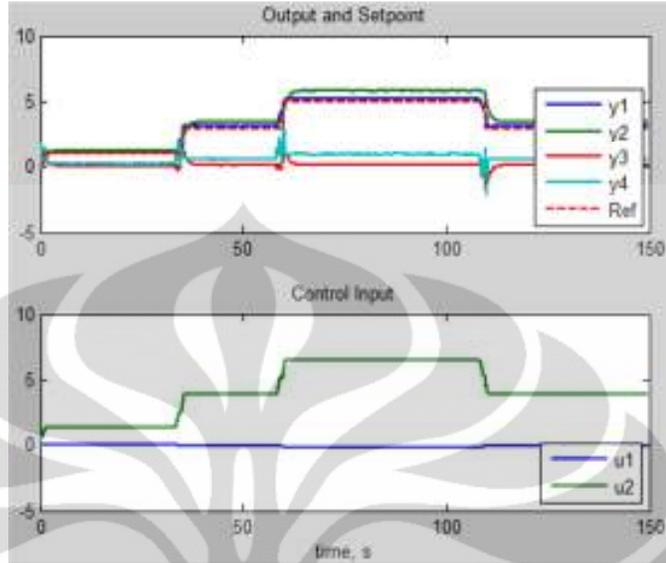


Gambar 4.9 Hasil Simulasi Matra Lateral Directional dengan $p=10$ dan $m=2$

Dapat dilihat pada gambar di atas, p dan m yang digunakan masih belum optimal. Berhubung semakin besarnya nilai p dan m maka akan semakin baik keluarannya, maka akan dicoba dengan beberapa p dan m yang lebih besar.

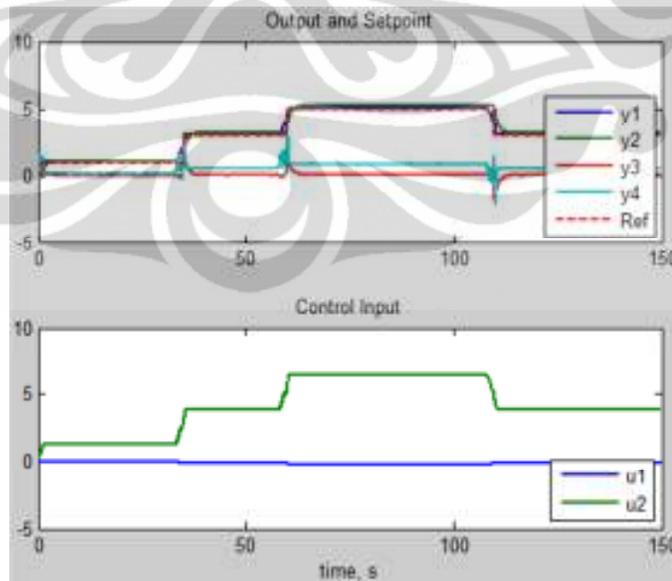


Gambar 4.10 Hasil Simulasi Matra Lateral Directional dengan $p=15$ dan $m=3$

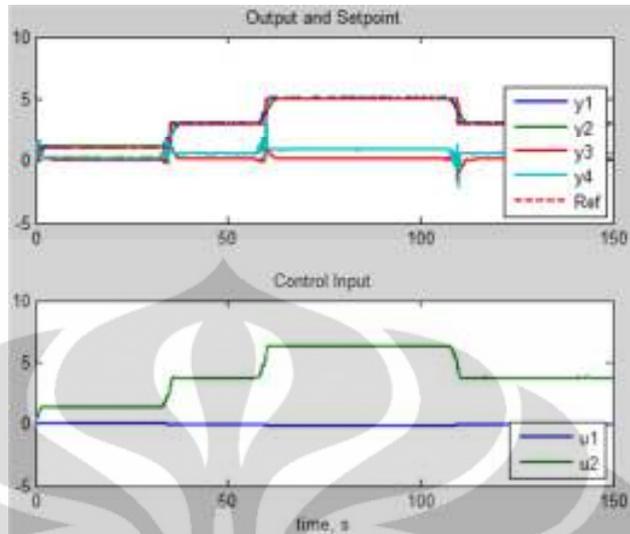


Gambar 4.11 Hasil Simulasi Matra Lateral Directional dengan $p=20$ dan $m=5$

Pada gambar di atas dapat dilihat hasil simulasi dengan $p=15$ & $m=3$ dan juga $p=20$ & $m=5$ yang didapat masih kurang baik, maka nilai p dan m akan coba dibesarkan lagi.



Gambar 4.12 Hasil Simulasi Matra Lateral Directional dengan $p=25$ dan $m=6$

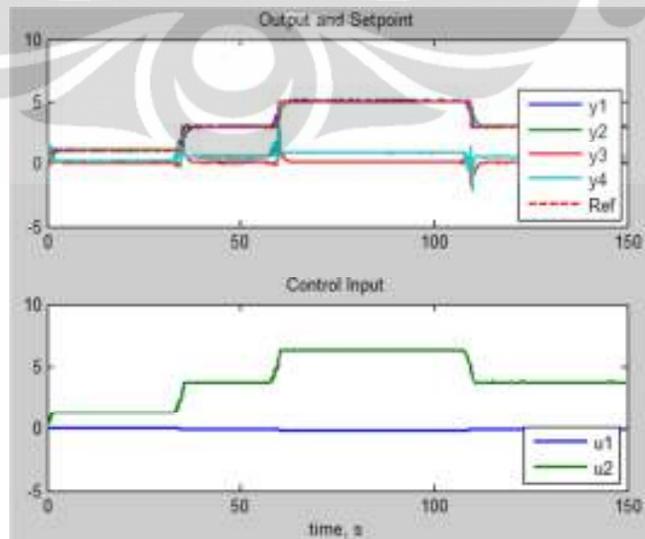


Gambar 4.13 Hasil Simulasi Matra Lateral Directional dengan $p=30$ dan $m=7$

Dari simulasi yang dilakukan dengan mencoba nilai prediction horizon dan moving horizon yang tepat, seperti yang terlihat pada gambar di atas, didapat bahwa untuk matra lateral directional, prediction horizon dan moving horizon yang baik untuk digunakan adalah masing-masing sebesar 18 dan 3.

Kemudian akan dilihat seberapa besar disturbance yang dapat diatasi pada matra lateral directional. Hasil simulasi yang didapatkan adalah

1. Disturbance = 0.1%

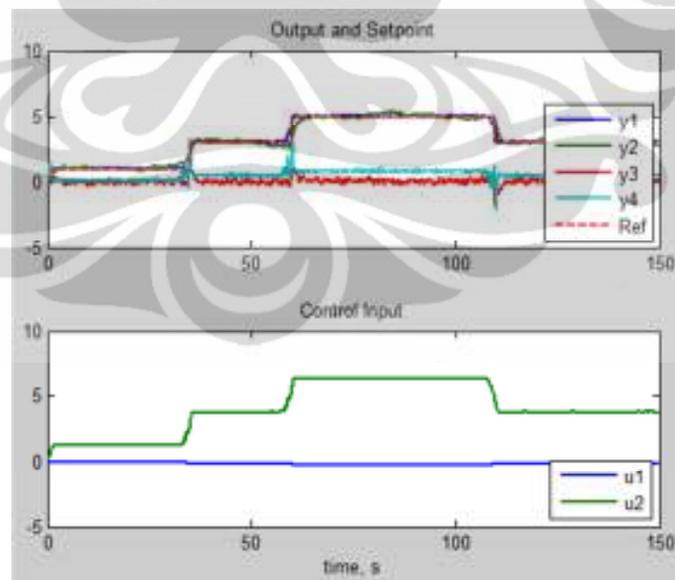


Gambar 4.14 Hasil Simulasi Matra Lateral Directional dengan Disturbance 0,1%

Simulasi dilakukan dalam waktu 150 detik, menggunakan prediction horizon (p) dan moving horizon (m) sebesar 30 dan 7, seperti yang telah disimulasikan sebelumnya, dengan adanya disturbance sebesar 0,1%. Hasil simulasi dengan besar disturbance 0,1% pada gambar di atas, dapat dilihat bahwa sistem dapat mengikuti referensi dengan baik, y_1 dan y_2 yang masing-masing merupakan sudut yaw dan sudut roll.

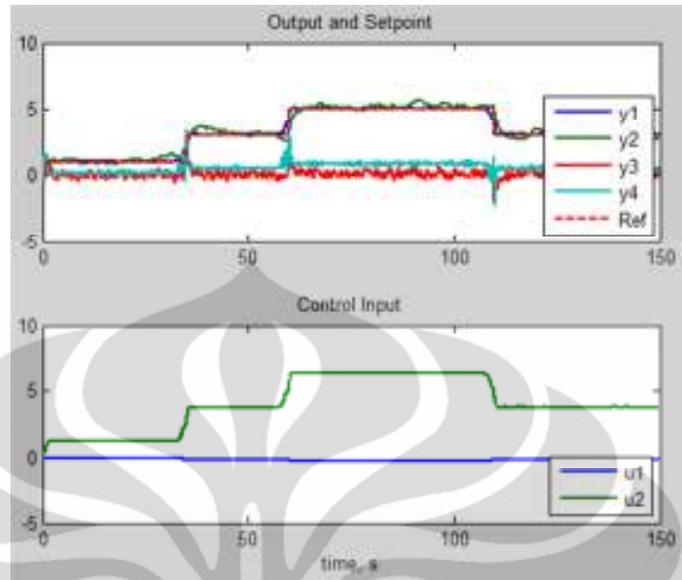
2. Disturbance = 1%

Pada simulasi ini juga digunakan p dan m yang sama dengan sebelumnya, namun bertambah disturbance menjadi 1%. Pada simulasi dengan disturbance 1% seperti terlihat pada gambar di bawah dapat dilihat bahwa hasil yang didapat masih baik mengikuti referensi hampir sama dengan hasil disturbance 0,1%. Namun mirip dengan hasil simulasi matra longitudinal bahwa pada saat sistem mempertahankan nilai pada referensi tertentu, terjadi munculnya beberapa ripple yang cukup kecil.



Gambar 4.15 Hasil Simulasi Matra Lateral Directional dengan Disturbance 1%

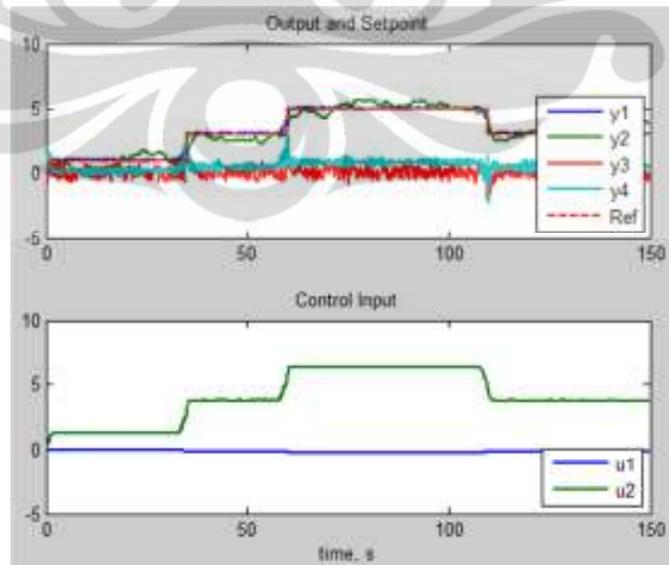
3. Disturbance = 2%



Gambar 4.16 Hasil Simulasi Matra Lateral Directional dengan Disturbance 2%

Simulasi dengan disturbance sebesar 2% ini menggunakan p dan m yang sama. Seperti yang terlihat pada gambar di bawah ini, hasil simulasi dengan menggunakan disturbance yang bertambah menjadi sebesar 2% ini, walaupun hasil yang didapat masih cukup baik, namun ripple yang muncul bertambah banyak dan juga besar dari sebelumnya.

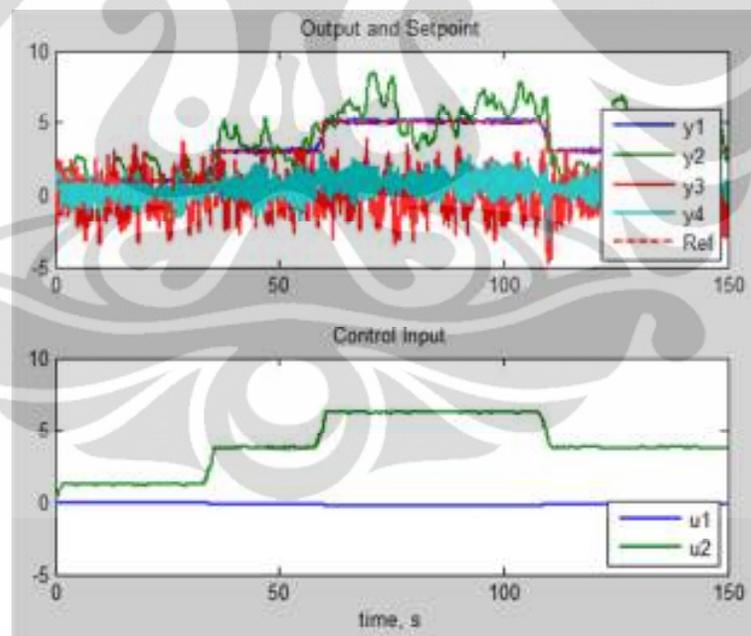
4. Disturbance = 5%



Gambar 4.17 Hasil Simulasi Matra Lateral Directional dengan Disturbance 5%

Simulasi sistem tetap menggunakan p dan m yang sama, namun dengan bertambahnya disturbance sebesar 5%. Seperti yang terlihat pada gambar di atas, pada hasil simulasi sistem dengan disturbance sebesar 5%, ripple yang muncul menjadi bertambah banyak dari disturbance 2%. Hal ini menunjukkan bahwa sistem sudah mendekati batas untuk dapat mengikuti referensi yang ditentukan. Dengan menambah besar disturbancenya lagi, sistem akan menjadi tidak stabil seperti yang dapat dilihat pada gambar di bawah ini yang menunjukkan hasil simulasi dengan disturbance sebesar 10%.

Hasil simulasi menggunakan disturbance sebesar 10% ini menunjukkan bahwa sistem sudah tidak dapat lagi mengikuti referensi yang diberikan, sehingga batas disturbance yang dapat digunakan pada UAV-200 dengan menggunakan state space MPC pada matra lateral directional untuk sistem pengendaliannya adalah sebesar 5%.



Gambar 4.18 Hasil Simulasi Matra Lateral Directional dengan Disturbance 10%

BAB 5

KESIMPULAN

5.1 Kesimpulan

Skripsi ini membahas mengenai validasi model dinamika dengan algoritma state space MPC untuk sistem pengendalian pada guided flight vehicle yang berupa UAV-200 milik LAPAN. Dengan mencoba memvariasikan besar disturbance yang ada, maka dapat dilihat sampai sejauh mana state space MPC ini dapat berperan dalam pengendalian sistem pengendaliannya. Dari hasil simulasi dan analisa maka dapat ditarik kesimpulan sebagai berikut

1. Model dinamika UAV-200 LAPAN ini dapat divalidasi dengan cukup baik dengan algoritma state space MPC.
2. Prediction horizon dan moving horizon yang baik digunakan pada matra longitudinal masing-masing sebesar 18 dan 3. Sedangkan untuk matra lateral directional, prediction horizon dan moving horizonnya masing-masing sebesar 30 dan 7.
3. Algoritma state space MPC dapat mengikuti perubahan model dinamika ini sampai dengan batas nilai disturbance sebesar 5% untuk matra longitudinal dan juga 5% untuk matra lateral directionalnya.

DAFTAR ACUAN

- [1] Irwanto, Herma Yudhi. "Pra Penelitian II : *Autopilot Sistem Berbasis Model Predictive Control untuk Guided Flight Vehicle*," Universitas Indonesia, 2011.
- [2] Lundh, Michael and Molander, Mats. "State Space Models in Model Predictive Control," ABB Automation Products AB.
- [3] Patra, Debadatta, Jena, Debasish, and Mohanty, Sunil Kumar, "*Model Predictive Control*," National Institute of Technology Rourkela, 2007.
- [4] Surayuda, Rise Hapshari. "*Perancangan Pengendali Model Predictive Control dengan Constraint pada Sistem Tata Udara Presisi*," Depok : Universitas Indonesia, 2010.
- [5] Amir, Mohamad Fauzan. "Kontrol Proses Berbasis Model," Institut Teknologi Bandung.
- [6] Garcia, C.E, Prett, D.M. and Morari, M. "*Model Predictive Control : Theory and Practice – a Survey*," Automatica, 1989.
- [7] <http://en.wikipedia.org/wiki/MPC>.
- [8] <http://akirajunto.wordpress.com/2010/08/01/mpc-model-predictive-control>
- [9] The MathWorks, Inc. (2008). *MatLab* (Versi 7.6.0.324) [Perangkat lunak komputer]. U.S.A: The MathWorks, Inc. (MatLab Help)
- [10] <http://www.sv.vt.edu/classes/ESM4714/methods/EEG.html>.