TESIS TE- 092099

MODEL PREDICTIVE CONTROL (MPC) DENGAN *CONSTRAIN* UNTUK PENGATURAN *TRACKING* PADA KAPAL LAUT PENGANGKUT BARANG *(CARGO)*

AJI BRAHMA NUGROHO
2211202002

DOSEN PEMBIMBING
Dr.Ir.Mochamad Rameli
Ir. Rusdhianto Effendi, M.T.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015

THESIS

MODEL PREDICTIVE CONTROL (MPC) WITH CONSTRAIN FOR TRACKING CONTROL OF SEA TRANSPORTATION TO SHIP GOODS (CARGO)

AJI BRAHMA NUGROHO
2211202002

SUPERVISOR
Dr.Ir.Mochamad Rameli
Ir. Rusdhianto Effendi, M.T.

MASTER PROGRAM
CONTROL SYSTEM ENGINEERING
DEPARTMENT of INDUSTRIAL TECHNOLOGY
FACULTY of INDUSTRY TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

Oleh:
Aji Brahma Nugroho
NRP. 2211202002

Tanggal Ujian : 26 Januari 2015
Periode Wisuda : 13 Maret 2015

Disetujui oleh :

1. Dr.Ir. Mochammad Rameli (Pembimbing I)
   NIP. 1954 12 27 1981 03 1002

2. Ir. Rusdhianto Effendi, M.T. (Pembimbing II)
   NIP. 1957 04 24 1985 02 1001

3. Prof. Ir. Abdullah Alkaff, M.Sc, Ph.D (Penguji I)
   NIP. 1955 01 23 1980 03 1002

4. Dr. Trihastuti Agustinah, S.T.,M.T. (Penguji II)
   NIP. 1968 0812 1994 0320 01

5. Ir. Ali Fatoni, M.T (Penguji III)
   NIP. 1962 06 03 1989 03 1002

6. Ir. Josaphat Pramudjianto, M.Eng. (Penguji IV)
   NIP. 1962 1005 1990 0310 03

Direktur Program Pascasarjana

Prof. Dr. Adi Soeprijanto, M.T
NIP. 1964 0405 1990 0210 01

MODEL PREDICTIVE CONTROL (MPC) WITH CONSTRAINT FOR TRACKING CONTROL ON SHIP GOODS CARRIER (CARGO)

By : Aji Brahma Nugroho
Student Identity Number : 2211202002
Supervisor : 1. Dr.Ir. Mochamad Rameli
: 2. Ir. Rusdhianto Effendi Ak, MT

ABSTRACT

Auto pilot-based PID has been widely used as a tracking settings on sea transport. Limitations that occur in this sistem that is the parameter settings manually as compensation of disturbance that occurred. This causes considerable interference in the operation of the ship and take more time for delivery process. Based on the tracking control sistem built on a cargo ship based Model Predictive Control with the constraint as an alternative.

The results showed after tested on some form of trajectories. Model predictive control with constraint gives a small error rate is less than 10% of the total mileage of the ship.

Keywords : *Model Predictive Control (MPC) with contrained, maximum angle rate.*

Halaman ini sengaja dikosongkan

MODEL PREDICTIVE CONTROL (MPC) WITH CONSTRAINT FOR TRACKING CONTROL ON SHIP GOODS CARRIER (CARGO)

By	: Aji Brahma Nugroho
Student Identity Number	: 2211202002
Supervisor	: 1. Dr.Ir. Mochamad Rameli
	: 2. Ir. Rusdhianto Effendi Ak, MT

ABSTRACT

Auto pilot-based PID has been widely used as a tracking settings on sea transport. Limitations that occur in this sistem that is the parameter settings manually as compensation of disturbance that occurred. This causes considerable interference in the operation of the ship and take more time for delivery process. Based on the tracking control sistem built on a cargo ship based Model Predictive Control with the constraint as an alternative.

The results showed after tested on some form of trajectories. Model predictive control with constraint gives a small error rate is less than 10% of the total mileage of the ship.

Keywords : *Model Predictive Control (MPC) with contrained, maximum angle rate.*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT yang senantiasa selalu memberikan ridho dan rahmat-Nya. Tak lupa shalawat serta salam penulis haturkan kepada Nabi besar junjungan kita Rasulullah Muhammad SAW sehingga penulis dapat menyelesaikan Tesis ini dengan judul:

***Model Predictive Control* (MPC) dengan *Constrain* untuk Pengaturan *Tracking* pada Kapal Laut Pengangkut Barang (*Cargo*)**

Dalam menyelesaikan Tesis ini, penulis berpegang pada teori yang pernah diperoleh dan bimbingan dari para dosen pembimbing Tesis. Selain itu, dengan adanya bantuan dan dukungan pihak-pihak lain sehingga Tesis ini dapat terselesaikan. Oleh karena itu, pada kesempatan ini penulis hendak menyampaikan ucapan terima kasih dan teriring doa semoga Allah SWT membalas setiap kebaikan dengan kebaikan yang berlipat ganda kepada semua keluarga yang telah memberikan dorongan moral, doa serta semangat untukku dalam mengerjakan Tesis ini. Secara khusus penulis mengucapkan termakasih sebesar-besarnya kepada

1. Bapak Dr.Ir. Mochamad Rameli selaku dosen pembimbing 1 yang penuh kesabaran dalam memberikan pengarahan
2. Bapak Ir. Rusdhianto Effendi Ak, MT sebagai dosen pembimbing 2 yang dengan sabar memberikan pengarahan dan bimbingan.
3. Semua pihak yang tidak dapat disebutkan satu per satu yang telah memberi bantuan baik secara langsung maupun tidak langsung.

Penulis berharap, buku Tesis ini mampu menambah pengetahuan dan pemahaman pada bidang ilmu kontrol, terutama rekan-rekan mahasiswa Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Surabaya, 26 Januari 2015

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PENGESAHAN ii
LEMBAR PERNYATAAN iiv
ABSTRAK vi
ABSTRACT viii
KATA PENGANTAR x
DAFTAR ISI xi
DAFTAR GAMBAR xiv
DAFTAR TABEL .... **Error! Bookmark not defined.**

BAB 1 PENDAHULUAN

1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Batasan Masalah 3
1.5 Kontribusi 3
1.6 Metodologi Penelitian 4

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

2.1 *Model Predictive Control* (MPC) dengan *Constrain* 5
2.2 Dinamika Sistem Kapal Kargo (*Container Ship*) 9
2.3 Model *Plant* Kapal 11
2.3.1 Model Davidson dan Schiff 11
2.3.2 Model Nomoto 13
2.4 Aktuator Kendali pada Kapal Kargo 14
2.5 *Disturbance* pada Sistem 15
2.5.1 Arus Laut (*Ocean Current*) 16
2.5.2 Angin Laut (*Ocean Wind*) 16

2.5.3 Gelombang Laut (*Ocean Wave*) 18

2.6 *Track* (jalur) Sistem 18

BAB 3 PERANCANGAN SISTEM

3.1 Perancangan *Plant* 22

3.1.1 Blok Diagram Model Kapal 23

3.1.2 Blok Diagram Aktuator 23

3.1.3 Blok Diagram Kinematika Kapal 24

3.2 Memasukkan Nilai *Parameter Plant* 25

3.3 Perancangan Kontroler 25

3.4 Perancangan *Trajectory* 28

3.4.1 Tahap Menentukan Koordinat Tujuan 28

3.4.2 Tahap *Trajectory Tracking* 28

BAB 4 HASIL DAN PEMBAHASAN

4.1 Hasil Simulasi Sistem 30

4.2 Hasil Pengujian Simulasi Sistem 32

4.3 Perubahan *Parameter* Kontroler 35

BAB 5 PENUTUP

5.1 Kesimpulan 74

5.2 Saran 75

DAFTAR PUSTAKA 76

LAMPIRAN 78

RIWAYAT PENULIS 80

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Dinamika Kapal 10
Tabel 3.1 *Parameter* Simulasi *Plant* Kapal Kargo 25
Tabel 4.1 *Parameter* Uji Simulasi Lintasan Garis Lurus 31
Tabel 4.2 Analisa *Error Tracking* pada Uji Simulasi Lintasan Lurus........ 33
Tabel 4.3 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 300 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Lurus 38
Tabel 4.4 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 400 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Lurus 40
Tabel 4.5 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 500 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Lurus 43
Tabel 4.6 *Parameter* Uji Simulasi Lintasan Persegi Panjang........ 43
Tabel 4.7 Analisa *Error Tracking* pada Uji Simulasi Lintasan Persegi Panjang 47
Tabel 4.8 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 300 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Persegi Panjang........ 50
Tabel 4.9 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 400 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Persegi Panjang........ 52
Tabel 4.10 Analisa *Response Time* controller MPC dengan *Horizon Prediction* N_p = 500 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Persegi Panjang........ 55
Tabel 4.11 *Parameter* Uji Simulasi Lintasan Bebas 55
Tabel 4.12 Analisa *Error Tracking* pada Uji Simulasi Lintasan Bebas 59
Tabel 4.13 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 300 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Bebas 62
Tabel 4.14 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 400 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Bebas 64
Tabel 4.15 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 500 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Bebas 67
Tabel 4.16 *Parameter* Uji Simulasi Lintasan Lingkaran........ 67
Tabel 4.17 Respon Sudut *Rudder* dari Sistem 70
Tabel 4.18 Analisa Kesalahan dari Simulasi *Tracking* terhadap Target 70
Tabel 4.19 Hasil Variasi Perubahan *Parameter* Kontroler pada Lintasan Lurus71
Tabel 4.20 Hasil Variasi Perubahan *Parameter* Kontroler pada Lintasan Persegi Panjang 71
Tabel 4.21 Hasil Variasi Perubahan *Parameter* Kontroler pada Lintasan Bebas 72

DAFTAR GAMBAR

Gambar 2.1 Konsep Dasar *Model Predictive Control* 5
Gambar 2.2 Blok Simulink *Model Predictive Control with Constrain* 9
Gambar 2.3 Gerak pada Kapal Kargo 9
Gambar 2.4 Vektor A Dalam Bidang xy 11
Gambar 2.5 Diagram Hidrolik Aktuator Kapal Kargo 15
Gambar 2.6 Blok Diagram *Rudder Control loop* 15
Gambar 2.7 Blok Simulink *Uniform Random Number* 16
Gambar 2.8 Koefisien C_{NW} sebagai Fungsi relatif γ_R 17
Gambar 2.9 Koefisien C_{NW} sebagai Fungsi relatif γ_R 17
Gambar 2.10 *Track* Garis Lurus 19
Gambar 2.12 *Track* Lintasan Bebas (*Free Track*) 20
Gambar 3.1 Diagram Blok Penelitian 22
Gambar 3.2 Blok Lengkap Model *Plant* Kapal Kargo 23
Gambar 3.3 Blok Diagram Model Kapal Kargo 23
Gambar 3.4 Blok Diagram Aktuator Model Kapal Kargo 23
Gambar 3.5 Kinematika Model *Plant* Kapal Kargo 24
Gambar 3.6 Blok MPC Kontroler pada MATLAB 25
Gambar 3.7 Menu Utama MPC Kontroler MATLAB 26
Gambar 3.8 *Control and Estimation Tool Manager* MPC 26
Gambar 3.9 *Parameter* MPC 27
Gambar 3.10 *Mode Scenarios* MPC 27
Gambar 3.11 MPC Kontroler *Output* 28
Gambar 4.1 Bentuk Keseluruhan Sistem 30
Gambar 4.2 Lintasan Target Garis Lurus 31
Gambar 4.3 Hasil Simulasi *Tracking* Lintasan Garis Lurus 32
Gambar 4.4 Perbandingan Hasil Simulasi target dan *Tracking* pada Lintasan Garis Lurus 32
Gambar 4.5 Hasil Simulasi Respon Sudut *Yaw* untuk Lintasan Lurus 34
Gambar 4.6 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Lurus 34
Gambar 4.7 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk Lintasan Lurus 35
Gambar 4.8 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 3 pada Lintasan Lurus 36
Gambar 4.9 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 5 pada Lintasan Lurus 36
Gambar 4.10 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 7 pada Lintasan Lurus 37
Gambar 4.11 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 9 pada Lintasan Lurus 37
Gambar 4.12 Perbandingan Respon *Plant Model* terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 3 pada Lintasan Lurus 38
Gambar 4.13 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 5 pada Lintasan Lurus 39

Gambar 4.14 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 7 pada Lintasan Lurus.................. 39
Gambar 4.15 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 9 pada Lintasan Lurus.................. 40
Gambar 4.16 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 3 pada Lintasan lurus.................... 41
Gambar 4.17 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 5 pada Lintasan Lurus.................. 41
Gambar 4.18 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 7 pada lintasan lurus..................... 42
Gambar 4.19 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 9 pada Lintasan Lurus.................. 42
Gambar 4.20 Bentuk Lintasan Target pada Lintasan Persegi Panjang 44
Gambar 4.21 Hasil Simulasi *Tracking* Target pada Lintasan Persegi Panjang..... 44
Gambar 4.22 Perbandingan Hasil Simulasi Target dan Hasil *Tracking* pada Lintasan Persegi Panjang.. 45
Gambar 4.23 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Persegi Panjang 45
Gambar 4.24 Hasil simulasi Respon Sudut *Rudder* untuk Lintasan Persegi Panjang 46
Gambar 4.25 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC pada Lintasan Persegi Panjang 46
Gambar 4.26 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 3 pada Lintasan Persegi Panjang... 48
Gambar 4.27 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 5 pada Lintasan Persegi Panjang... 48
Gambar 4.28 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 7 pada Lintasan Persegi Panjang... 49
Gambar 4.29 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 9 pada Lintasan Persegi Panjang... 49
Gambar 4.30 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 3 pada Lintasan Persegi Panjang... 50
Gambar 4.31 Perbandingan Respon *Plant* model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 5 pada Lintasan Persegi Panjang... 51
Gambar 4.32 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 7 pada Lintasan Persegi Panjang... 51
Gambar 4.33 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 9 pada Lintasan Persegi Panjang... 52
Gambar 4.34 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 3 pada Lintasan Persegi Panjang... 53
Gambar 4.35 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 5 pada Lintasan Persegi Panjang... 53
Gambar 4.36 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 7 pada Lintasan Persegi Panjang... 54
Gambar 4.37 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 9 pada Lintasan Persegi Panjang... 54
Gambar 4.38 Bentuk Lintasan Target untuk Lintasan Bebas 56

Gambar 4.39 Hasil Simulasi *Tracking* pada Lintasan Bebas 56
Gambar 4.40 Perbandingan Hasil Simulasi Target dan Hasil *Tracking* pada Lintasan Bebas 57
Gambar 4.41 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Bebas 57
Gambar 4.42 Hasil Simulasi Respon Sudut *Yaw* pada Lintasan Bebas 58
Gambar 4.43 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC pada Lintasan Bebas 58
Gambar 4.44 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 3 pada Lintasan Bebas 60
Gambar 4.45 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 5 pada Lintasan Bebas 60
Gambar 4.46 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 7 pada Lintasan Bebas 61
Gambar 4.47 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 9 pada Lintasan Bebas 61
Gambar 4.48 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 3 pada Lintasan Bebas 62
Gambar 4.49 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 5 pada Lintasan Bebas 63
Gambar 4.50 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 7 pada Lintasan Bebas 63
Gambar 4.51 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 9 pada Lintasan Bebas 64
Gambar 4.52 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 3 pada Lintasan Bebas 65
Gambar 4.53 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 5 pada Lintasan Bebas 65
Gambar 4.54 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 7 pada Lintasan Bebas 66
Gambar 4.55 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 9 pada Lintasan Bebas 66
Gambar 4.56 Hasil Simulasi *Tracking* Lintasan Lingkaran 68
Gambar 4.57 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Lingkaran... 68
Gambar 4.58 Hasil Simulasi Respon Sudut *Yaw* pada *Tracking* Lintasan Lingkaran.. 69
Gambar 4.59 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC pada Lintasan Lingkaran 69

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Semakin tingginya kebutuhan pada sarana transportasi laut, terutama untuk perdagangan dan jasa pengiriman barang menyebabkan kebutuhan kapal laut pengangkut barang *(cargo)* yang efisien dan cepat semakin meningkat.

Untuk mempermudah navigasi dan operasional pada kapal kargo, sistem autopilot telah dikembangkan dan diimplementasikan untuk mengkontrol arah laju kapal kargo sesuai dengan jalur yang telah ditentukan (*tracking*).

Salah satu komponen yang sangat vital untuk mengatur arah laju dari kapal kargo adalah *rudder*. Sistem kontrol konvensional seperti PID telah digunakan sebagai autopilot kontroler untuk pengaturan *tracking* pada kapal kargo. Permasalahan yang kemudian muncul adalah diperlukannya penambahan kontrol manual *parameter* dari kontroler PID sebagai kompensasi untuk gangguan (*disturbance*) yang terjadi pada kapal, diantaranya adalah angin, dan arus laut.

Karena pengaturan *parameter* kontroler dilakukan secara manual maka kontroler umumnya bekerja dengan sangat baik pada kondisi operasional dengan variasi terbatas. Untuk kondisi operasional dengan variasi yang lebih beragam, dimana *parameter* autopilot harus diubah secara terus-menerus mengikuti perubahan dinamika lingkungan seperti perubahan angin, gelombang, arus dan kedalaman laut maka pengaturan secara manual seringkali cukup merepotkan para kru kapal kargo.

Kesalahan pengaturan akibat *human error* berpotensi sangat besar dan hal ini tentu saja sangat mengganggu dalam proses pelayaran, selain itu *Tracking control* yang buruk berandil besar terhadap pemborosan bahan bakar dan keterlambatan proses pengiriman barang.

Berdasarkan hal diatas, maka penulis Tesis ini tertarik untuk meneliti metode kontrol yang mampu mengatur *rudder* untuk *tracking control* pada kapal kargo yang mampu beradaptasi terhadap perubahan dinamika lingkungan secara

otomatis sehingga diharapkan lebih efisien dalam konsumsi bahan bakar dan waktu.

Metode pengaturan untuk *tracking control* yang telah digunakan dan diteliti sebelumnya diantaranya dilakukan oleh Steiner Ardal, R.Ferreiro Garcia, Yanseng Yang dan Zhen Li

Dalam papernya, Steiner Ardal menggunakan metode *robust control* yang diaplikasikan untuk *automatic steering* pada kapal *cargo* dengan *disturbance* kecepatan angin, gelombang dan arus laut. Kelemahan pada penelitian ini yaitu pada frekuensi *disturbance* yang rendah sistem tetap stabil tetapi tidak dapat mempertahankan arah laju (*tracking*) pada kapal hal ini ditunjukan dengan osilasi yang terlalu besar pada sudut *heading*.

Kelemahan pada penelitian Steiner Ardal kemudian diperbaiki oleh penelitian yang dilakukan oleh R.Ferreiro Garcia, dengan mengajukan metode *fuzzy cascade* yang digabung dengan PID. Strategi yang digunakan adalah dengan membangun *fuzzy rule base* untuk semua variabel *disturbance*. Kelemahan pada penelitian R.F Garcia adalah sistem hanya efisien pada kecepatan rendah dan bekerja pada *disturbance* yang terbatas.

Penelitian berikutnya dilakukan oleh Yanseng Yang menggunakan metode *robust adaptive fuzzy control*. Peneliti ini mengubah sudut pandang dalam menyelesaikan masalah dalam *tracking control* yaitu dengan mengasumsikan bahwa dengan pengaturan *roll* pada kapal kargo maka perfoma *tracking control* akan meningkat. Kelemahan pada penelitian Yanseng Yang adalah timbulnya problema baru yaitu mekanisme *rudder* dipengaruhi oleh *maximum angle* dan *angle rate* dimana metode Yanseng Yang mengabaikan kedua *parameter* ini.

Untuk mengatasi kelemahan tersebut pada tahun 2012 Zhen Li mengajukan metode *Disturbance Compensating* yang digabungkan dengan *Model Predictive Control (MPC)*, pada penelitian ini model yang digunakan telah memperhitungkan satu derajat kebebasan (1 DOF) dinamika pada kapal yaitu *yaw rate* serta telah memperhitungkan batasan mekanik sistem. Kelemahan pada penelitian Zhen Li adalah pada real sistem model dinamika kapal harus memenuhi 2 derajat kebebasan (DOF) yaitu gerak *surge* dan *yaw*.

Berdasarkan hal diatas dan beberapa pustaka yang telah dikaji diusulkan sebuah Tesis yang akan meneliti metode *Model Predictive Control* (MPC) dengan memperhitungkan *constrain* untuk digunakan dalam pengaturan *tracking* pada kapal *cargo* menggunakan model dinamika dengan 2 derajat kebebasan (DOF) yaitu *surge* dan *yaw*.

Alasan pemilihan metode *Model Predictive Control* (MPC) dengan *constrain* dilandasi oleh mekanisme *rudder* dipengaruhi oleh *maximum angle* dan *angle rate.* Pada sistem kendali konvensional batasan-batasan seperti *maximum angle* dan *angle rate* tidak diperhitungkan pada proses pengendalian, hal ini tentu dapat menyebabkan hasil kendali yang kurang baik terutama pada *sistem* berukuran massif seperti kapal kargo. Untuk mengatasi hal itu maka metode *Model Predictive Control* (MPC) dengan *constrain* yang paling sesuai.

1.2 Rumusan Masalah

Merancang kontroler yang mampu mengatur *rudder* untuk mempertahankan arah laju (*tracking*) kapal *cargo* sesuai dengan jalur yang telah ditentukan dan mampu beradaptasi terhadap perubahan kondisi lingkungan secara automatis.

1.3 Tujuan

Merancang kontroler untuk pengaturan *tracking* pada kapal *cargo* berbasis *Model predictive Control (MPC)* dengan *constrain*

1.4 Batasan Masalah

Kapal yang di control berada pada kondisi operasi di laut lepas

Constrain yang digunakan adalah perubahan kecepatan sudut *rudder*

1.5 Kontribusi

Pengembangan model dinamika kapal dengan 2 derajat kebebasan (DOF)

1.6 Metodologi Penelitian

Agar dapat mencapai tujuan yang diharapkan, penelitian ini akan dilakukan melalui beberapa tahapan berikut ini :

1. Kajian Pustaka

   Melakukan telaah beberapa pustaka terkait baik dari artikel penelitian yang telah dipublikasikan maupun buku yang diterbitkan.

2. Analisis beberapa metode yang terkait

   Analisis metode-metode yang terkait dengan persoalan pada sistem kontrol yang akan dirancang dengan tujuan agar memperoleh formulasi yang sesuai untuk sistem tersebut.

3. Merancang sistem kontrol MPC dengan *constrain*.
4. Membuat program simulasi dan pengujian sistem

   Melakukan serangkaian eksperimen untuk mengetahui kinerja dan ketahanan sistem tersebut setelah dilakukan pengembangan dan perbaikan.

5. Analisis hasil pengujian

   Melakukan analisis hasil pengujian dan mengklarifikasi hasil tersebut terhadap tujuan yang telah ditetapkan. Apabila telah memenuhi tujuan berarti penelitian telah berhasil, dan apabila belum memenuhi maka perlu dikaji lebih lanjut solusi atau gagasan *alternative* agar tujuan yang telah ditetapkan dapat dicapai.

6. Penarikan kesimpulan

   Jika hasil evaluasi menunjukkan bahwa tujuan penelitian telah tercapai maka akan ditarik kesimpulan untuk menegaskan bahwa gagasan yang diusulkan berhasil menyelesaikan permasalahan dan memenuh tujuan penelitian.

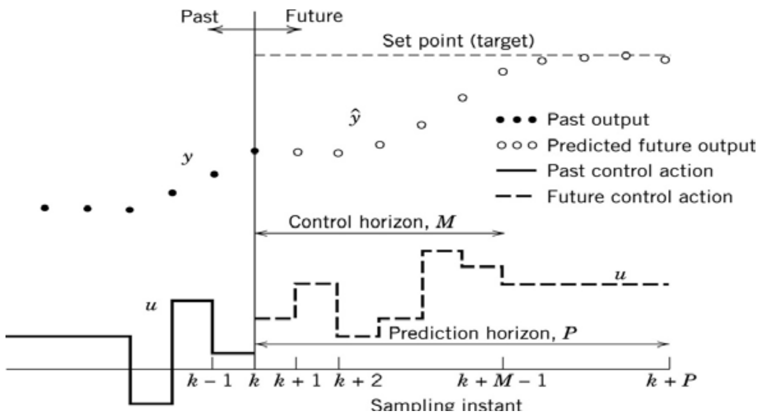
BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Dalam bab ini, telah ditelaah beberapa pustaka khususnya penelitian - penelitian yang terkait dengan topik yang akan diselesaikan. Pustaka tersebut juga telah dipublikasikan dalam jurnal maupun prosiding internasional. Uraian dari tiap-tiap pustaka yang disampaikan akan dikerucutkan untuk mendapatkan beberapa konsep yang digunakan dalam penelitian. Pustaka-pustaka yang mendasari penelitian ini meliputi *Model Predictive Control* (MPC) *with constrain,* dan *Guidance and Control of Ocean Vehicle*.

2.1 *Model predictive Control (MPC)* dengan *Constrain*

Liuping wang dalam bukunya menyatakan bahwa nilai yang akan datang dari variabel keluaran yang diprediksi menggunakan model dinamik dari suatu proses dan pengukuran saat ini merupakan konsep dasar dari algoritma MPC. Tidak seperti metode kompensasi waktu tunda, prediksi tersebut dibuat untuk lebih dari satu waktu tunda kedepan. Perhitungan aksi kontrol didasarkan pada prediksi yang akan datang dan pengukuran saat ini. Variabel yang dimanipulasi $u(k)$ pada *sampling* ke- k dihitung sedemikian rupa sehingga mereka meminimalkan fungsi objektif J seperti yang ditunjukkan pada Gambar 2.1.



Gambar 2.1 Konsep Dasar *Model Predictive Control* (MPC)

Secara sederhana, *Plant* diasumsikan sebagai *single input single output* yang dideskripsikan oleh

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \tag{2.1}$$

$$y(k) = C_m x_m(k) \tag{2.2}$$

Di mana $u(k)$ adalah variabel yang dimanipulasi atau variabel *input*, $y(k)$ adalah *output* proses, dan $x_m(k)$ adalah vektor variabel *state*. Catatan bahwa sistem tersebut memiliki $u(k)$ sebagai *input*-nya.

Dengan mengambil operasi delta pada Persamaan (2.1), maka didapatkan bahwa

$$x_m(k+1) - x_m(k) = A_m(x_m(k) - x_m(k-1)) + B_m(u(k) - u(k-1))$$

Dengan menotasikan delta variabel maka

$$\Delta x_m(k+1) = x_m(k+1) - x_m(k); \Delta x_m(k) = x_m(k) - x_m(k-1)$$

Dan delta variabel kontrol

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

Sehingga persamaan (2.1) *state space*-nya adalah

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \tag{2.3}$$

Catatan bahwa *input* model *state space* adalah $\Delta u(k)$. Langkah selanjutnya adalah menghubungkan $\Delta x_m(k)$ ke *output* $y(k)$

$$x(k) = [\Delta x_m(k)^T \; y(k)]^T$$

$$y(k+1) - y(k) = C_m(x_m(k+1) - x_m(k)) = C_m \Delta x_m(k+1)$$

$$= C_m A_m \Delta x_m + C_m B_m \Delta u(k) \tag{2.4}$$

Dengan mengambil persamaan (2.3) dan (2.4) maka model *state space* adalah

$$\overbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} = \overbrace{\begin{bmatrix} A_m & O_m^T \\ C_m A_m & 1 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \Delta x_m \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^{B}$$

$$y(k) = \overbrace{[O_m \quad 1]}^{C} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \tag{2.5}$$

di mana $O_m = \overbrace{[0 \quad 0 \quad \ldots \quad 0]}^{n_1}$. A, B, C dinamakan *augmented* model yang akan digunakan di dalam perancangan *predictive control*.

Tahapan selanjutnya dalam perancangan sistem *predictive control* adalah menghitung keluaran *plant* yang diprediksi dengan sinyal kontrol yang akan datang sebagai variabel yang dapat disetel. Prediksi tersebut dideskripsikan dalam *optimization window*. Dengan mengamsusikan bahwa *sampling instant* $k_i, k_i > 0$, vektor variabel *state* $x(k_i)$ tersedia melalui pengukuran, *state* $x(k_i)$ menyediakan informasi *Plant* saat ini. Lintasan kontrol yang akan datang dinotasikan oleh $\Delta u(k_i), \Delta u(k_i+1), \ldots, \Delta u(k_i+N_c+1)$, di mana N_c dinamakan *control horizon*. Dengan informasi yang diberikan $x(k_i)$, variabel *state* yang akan datang diprediksi untuk N_p, di mana N_p dinamakan *prediction horizon* atau panjang *window* optimisasi. Variabel *state* yang akan datang sebagai $x(k_i+1|k_i), x(k_i+2|k_i), \ldots., x(k_i+m|k_i), \ldots. x(k_i+N_p|k_i)$ di mana $x(k_i+m|k_i)$ adalah variabel *state* yang diprediksi saat k_i+m dengan informasi *Plant* yang diberikan saat ini $x(k_i)$.

Berdasarkan model *augmented state space* A, B, C, variabel *state* yang akan datang dihitung secara berurutan menggunakan himpunan *parameter* kontrol yang akan datang

$$x(k_i+1|k_i) = Ax(k_i) + B\Delta u(k_i)$$

$$x(k_i+2|k_i) = Ax(k_i+1|k_i) + B\Delta u(k_i+1)$$

$$= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i+1)$$

$$\vdots$$

$$x(k_i+N_p|k_i) = A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + A^{N_p-2}B\Delta u(k_i+1) + \ldots + A^{N_p-N_c}B\Delta u(k_i+N_c-1)$$

Dari variabel *state* yang terprediksi, variabel *output* terprediksi adalah

$$y(k_i+1|k_i) = CAx(k_i) + CB\Delta u(k_i)$$

$$y(k_i+2|k_i) = CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i+1)$$

$$\vdots$$

$$y(k_i+N_p|k_i) = CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) + CA^{N_p-2}B\Delta u(k_i+1) + \ldots + CA^{N_p-N_c}B\Delta u(k_i+N_c-1) \quad (2.6)$$

Dengan mendefinisikan vektor

$$Y = \left[y(k_i + 1|k_i) \quad y(k_i + 2|k_i) \quad \dots \quad y\left(k_i + N_p \middle| k_i\right) \right]^T$$

$$\Delta U = [\Delta u(k_i) \quad \Delta u(k_i + 1) \quad \dots \quad \Delta u(k_i + N_c + 1)]$$

Dengan mengubah persamaan (2.1.6) kedalam bentuk vektor diatas maka

$$Y = F\, x(k_i) + \Phi \Delta U \tag{2.7}$$

di mana

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \; \Phi = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

Untuk *set point* sinyal yang diberikan $r(k_i)$ saat waktu *sampling* k_i, dalam sebuah *prediction horizon*, objektif dari sistem *control horizon* adalah membawa *output* yang terprediksi sedekat mungkit ke *set point* sinyal. Dengan mengasumsikan bahwa data vektor yang berisikan informasi *set point* adalah

$$R_s^T = \overbrace{[1 \quad 1 \quad \dots \quad 1]}^{N_p} \; r(k_i),$$

Dan mendefinisikan *cost function* J yang mencerminkan objektif kontrol sebagai

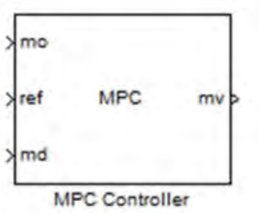
$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \tag{2.8}$$

Dengan mensubtitusikan persamaan (2.7) ke persamaan (2.8) dan diturunkan terhadap delta U sama dengan nol maka akan didapatkan persamaan solusi optimal kontrol seperti pada persamaan (2.1.9) dengan *set point* $r(k_i)$

$$\Delta U = (\Phi^T \Phi + \; R)^{-1} \Phi^T (\bar{R}_s\, r(k_i) - F\, x(k_i)) \tag{2.9}$$

Walaupun optimal *parameter* vektor ΔU berisikan kontrol $\Delta u(k_i), \Delta u(k_i + 1), \dots, \Delta u(k_i + N_c + 1)$, dengan prinsip *receding horizon* kontrol, hanya diterapkan dalam samping pertama pada setiap urutan sehingga nilai dari $\Delta u(k_i)$ adalah $\Delta u(k_i) = \overbrace{[1 \quad 0 \quad \dots \quad 0]}^{N_c} (\Phi^T \Phi + \; R)^{-1} \Phi^T (\bar{R}_s\, r(k_i) - F\, x(k_i))$ (2.10)

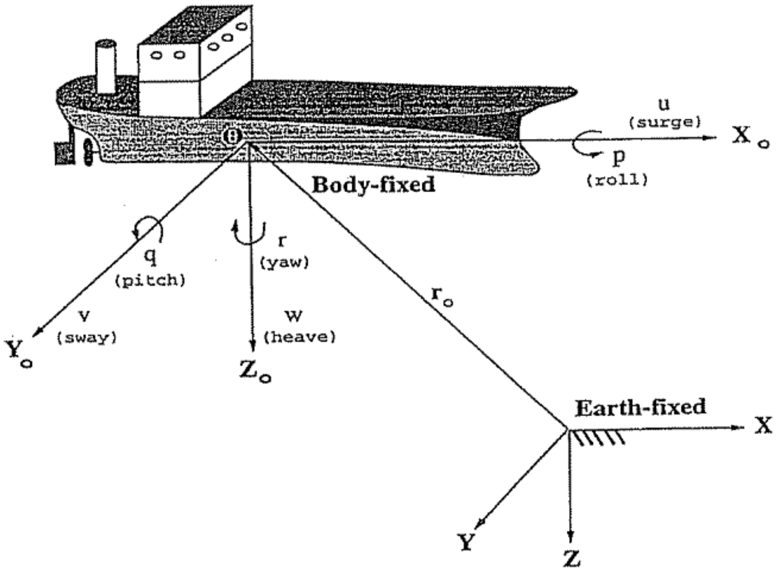
Fungsi operasi *Model Predictive Control* (MPC) *with constrain* telah tersedia dalam bentuk simulink Blok pada *software* MATLAB 14.0 seperti pada Gambar 2.2



Gambar 2.2 Blok Simulink *Model Predictive Control* (MPC) *with Constrain*

2.2 Dinamika sistem kapal cargo (Container Ship)

Pada prinsipnya gerak kapal dibagi dalam enam derajat kebebasan (DOF) yaitu : *surge, sway, yaw, heave, roll,* dan *pitch* seperti pada Gambar 2.3



Gambar 2.3 Gerak pada Kapal Kargo

Pada penelitian ini dinamika kapal yang kami ambil yaitu gerak translasi pada sumbu – x (surge) dan gerak rotasi pada sumbu – z (*yaw*) hal ini didasarkan pada gerak real kapal cargo saat beroperasi pada permukaan laut dengan asumsi :

1. massa kapal terdistribusi homogen
2. bidang pada sumbu - x dan sumbu – z simetri
3. gerak heave, roll dan pitch konstan

Secara umum gerakan yang dialami sebuah kapal ketika melaju dilautan ada dua macam yaitu gerakan translasi dan rotasi dimana ditampilkan pada Tabel 2.1

Tabel 2.1. Dinamika Kapal Kargo

DOF	Gerak Translasi	gaya dan momen	Kecepatan Linear dan angular	Posisi dan sudut Euler
1	Gerak pada arah sumbu - x (surge)	X	u	x
2	Gerak pada arah sumbu - y (sway)	Y	v	y
3	Gerak pada arah sumbu -z (heave)	Z	w	z
DOF	**Gerak Rotasi**	**gaya dan momen**	**Kecepatan Linear dan angular**	**Posisi dan sudut Euler**
4	Rotasi pada sumbu – x (roll)	K	p	ϕ
5	Rotasi pada sumbu – y (pitch)	M	q	θ
6	Rotasi pada sumbu – z (*yaw*)	N	r	ψ

Representasi kinematika gerak pada kapal kargo menggunakan analisa vector yaitu sebagai berikut :

a. Vector dalam bidang *xy*

Dalam bidang sumbu Z tidak ada maka vector $\overline{A}$ dinyatakan dengan :

$$\overline{A} = \overline{A}_x + \overline{A}_y = A_x\hat{i} + A_y\hat{j} \qquad (2.11)$$

dan besarnya :

$$A = \sqrt{A_x^2 + A_y^2} \qquad (2.12)$$

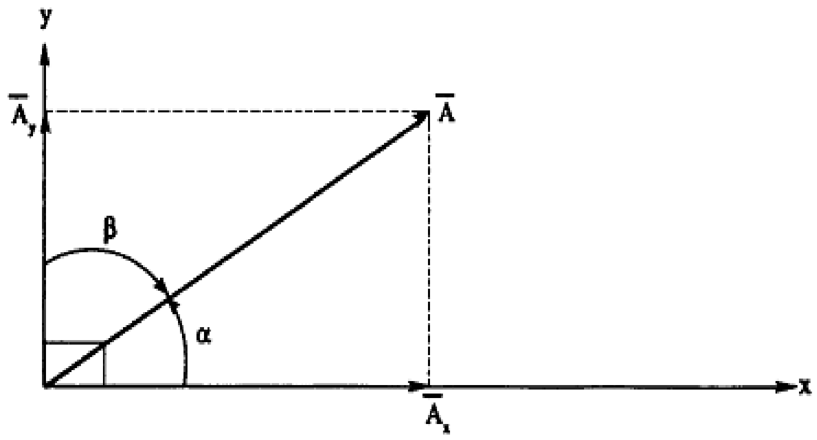
$\overline{A}_x, \overline{A}_y$ dan $\hat{i}, \hat{j}$ masing-masing adalah komponen vector dan vector satuan pada sumbu x dan y .

Disini $\overline{A}_x = A_x\hat{i}$, $\overline{A}_y = A_y\hat{j}$ dan besarnya $A_x = A\cos\alpha$, $A_y = A\cos\beta = A\sin\alpha$

Arah vector $\overline{A}$ terhadap sumbu x, y adalah :

$$Cos\alpha = \frac{A_x}{A} \qquad Cos\beta = Sin\alpha = \frac{A_y}{A} \qquad (2.13)$$

Gambar 2.4 berikut ini menunjukkan vector $\overline{A}$ dalam bidang xy



Gambar 2.4. Vektor $\overline{A}$ Dalam Bidang xy

2.3 Model *Plant* Kapal

2.3.1 Model Davidson dan Schiff

Davidson Sciff memodelkan gaya hidrodinamik dan momen pada kapal untuk gerak pada sumbu – y *(sway)* dan gerak rotasi pada sumbu - z (*yaw*) sebagai berikut :

$$Y = Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_v v + Y_r r + Y_\delta \delta_R \tag{2.14}$$

$$N = N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_v v + N_r r + N_\delta \delta_R \tag{2.15}$$

sehingga persamaan gerak kapal dapat dituliskan sebagai berikut :

$$M\dot{v} + (u_0)v = b\delta_R \tag{2.16}$$

dimana $v = [u, r]^T$ adalah vector state, δ_R adalah sudut *rudder* dan

$$M = \begin{bmatrix} m - Y_{\dot{v}} & mx_G - Y_{\dot{r}} \\ mx_G - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} N = \begin{bmatrix} -Y_v & mu_0 - Y_r \\ -N_v & mx_G u_0 - N_r \end{bmatrix} b = \begin{bmatrix} Y_\delta \\ N_\delta \end{bmatrix} \tag{2.17}$$

dalam matrik M dan N diatas mengandung *parameter* hidrodinamik dari kapal kargo yaitu sebagai berikut :

m = mass of the ship

$Y_{\dot{v}}$ = *derivative of sway force against* $\dot{v}$

Y_v = *derivative of sway force direction of* v

$Y_{\dot{r}}$ = *yaw direction of the force derivative* $\dot{r}$

N_v = *derivative of sway moment of* v

$N_{\dot{v}}$ = *derivative of sway moment on* $\dot{v}$

$N_{\dot{r}}$ = *derivative of yaw moment of* $\dot{r}$

N_r = *derivative of yaw moment on* r

x_G = *center of mass*

Catat bahwa matrik $N(u_0)$ diperoleh dengan penjumlahan damping D dan fungsi *coriolis sentripetal* $C(u_0)$ (fungsi tambahan mu_0 dan $mx_G u_0$) sehingga diperoleh :

$$N(u_0) = C(u_0) + D \tag{2.18}$$

selain itu kami telah memilih matrik inersia sehingga $M \neq M^T$. Hubungan model state-space diperoleh dengan meletakan $x = [v, r]^T$ sebagai state vector dan $u = \delta_R$ maka diperoleh :

$$\dot{x} = Ax + b_1 u \tag{2.19}$$

dengan

$$A = -M^{-1}N = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad b_1 = M^{-1}b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{2.20}$$

dengan koefisien sebagai berikut :

$$\begin{aligned}
a_{11} &= \frac{(I_z - N_{\dot{r}})Y_v - (mx_G - Y_{\dot{r}})N_v}{\det(M)} \\
a_{12} &= \frac{(I_z - N_{\dot{r}})(Y_r - mu_0) - (mx_G - Y_{\dot{r}})(N_{\dot{r}} - mx_G u_0)}{\det(M)} \\
a_{21} &= \frac{(m - Y_{\dot{v}})N_v - (mx_G - N_{\dot{v}})Y_v}{\det(M)} \\
a_{22} &= \frac{(m - Y_{\dot{v}})(N_r - mx_G u_0) - (mx_G - N_{\dot{v}})(Y_r - mu_0)}{\det(M)}
\end{aligned} \tag{2.21}$$

$$\begin{aligned}
b_1 &= \frac{(I_z - N_{\dot{r}})Y_\delta - (mx_G - Y_{\dot{r}})N_\delta}{\det(M)} \\
b_2 &= \frac{(m - Y_{\dot{v}})N_\delta - (mx_G - N_{\dot{v}})Y_\delta}{\det(M)}
\end{aligned} \tag{2.22}$$

2.3.2 Model Nomoto

Dua alternative representasi dari Model Davidson dan Schiff yang kemudian dikembangkan oleh Nomoto, Honda dan Hirano, dimana dalam domain waktu model Nomoto diklasifikasikan berdasarkan orde nya yaitu :

a. Model Nomoto 1st Orde

Model ini diperoleh dengan meletakkan *propeller thrust effective time constant* sebanding dengan :

$$T = T_1 + T_2 - T_3 \ . \tag{2.23}$$

Dalam domain waktu

$$T\dot{\psi} + \dot{\psi} = K\delta \tag{2.24}$$

Dengan fungsi transfer diperoleh sbb :

$$T\frac{d\psi}{dt} + \frac{d\psi}{dt} = K\delta$$

$$T\psi(s) + \psi(s) = K\delta$$

$$\psi(s)(T+1) = K\delta$$

$$\frac{\psi}{\delta}(s) = \frac{K}{s(1+Ts)} \tag{2.25}$$

Model Nomoto 1st orde sebaiknya digunakan untuk model kapal dengan frekuensi kecepatan rendah dan memiliki *single screw propeller.* Untuk kapal yang memiliki jumlah *propeller* lebih dari satu dapat menggunakan Model Nomoto 2nd orde.

b. Model Nomoto 2nd Orde

Model Nomoto 2nd orde menghubungkan sudut *yaw* (ψ) dan sudut *rudder*(δ) menurut domain waktu :

$$T_1T_2\dddot{\psi} + (T_1 + T_2)\ddot{\psi} + \dot{\psi} = K(\delta + T_3\dot{\delta}) \tag{2.26}$$

dengan fungsi transfer diperoleh sbb:

$$T_1T_2\frac{d\psi^3}{dt^2} + (T_1 + T_2)\frac{d\psi^2}{dt} + \frac{d\psi}{dt} = K(\delta + T_3\frac{d\delta}{dt})$$

$$T_1T_2\psi(s)^3 + (T_1 + T_2)\psi(s)^2 + \psi(s) = K(\delta + T_3\delta(s))$$

$$\psi(s)(T_1T_2(s)^2 + (T_1 + T_2)(s) + 1) = K\delta(1 + T_3(s))$$

$$\frac{\psi}{\delta}(s) = \frac{K(1+T_3 s)}{s(1+T_1 s)(1+T_2 s)} \tag{2.27}$$

Keterangan :

ψ = sudut *yaw*

δ = sudut *rudder*

K = Gain

$T_{1,2,3}$ = propeller Thrust time constant

Karena $r(s) = s\,\psi(s)$ maka fungsi transfer diatas dapat juga ditulis sebagai berikut:

$$\frac{r}{\delta}(s) = \frac{K(1+T_3 s)}{(1+T_1 s)(1+T_2 s)} \tag{2.28}$$

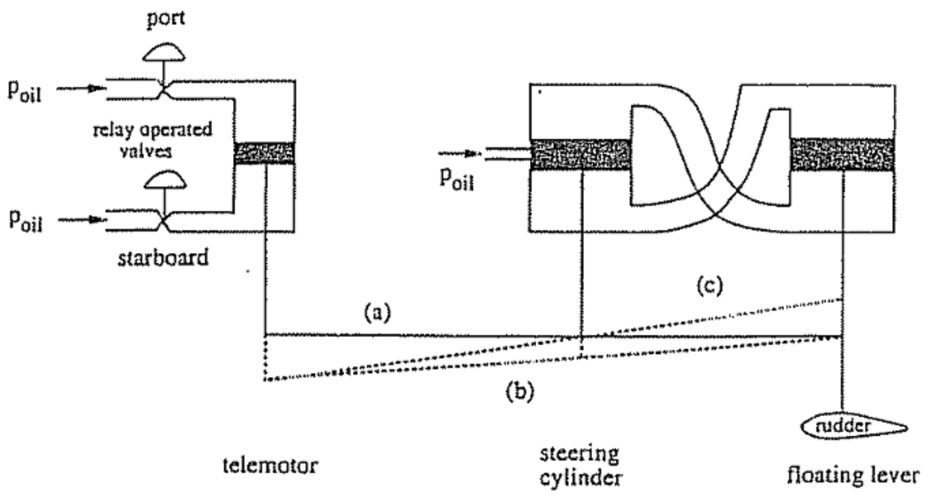
parameter fungsi transfer yang terkait dengan turunan hidrodinamika adalah sebagai berikut :

$$\begin{aligned}
T_1 T_2 &= \frac{\det(M)}{\det(N)} \\
T_1 + T_2 &= \frac{n_{11}m_{22} + n_{22}m_{11} - n_{12}m_{21} - n_{21}m_{12}}{\det(N)} \\
K &= \frac{n_{21}b_1 - n_{11}b_2}{\det(N)} \\
KT_3 &= \frac{m_{21}b_1 - m_{11}b_2}{\det(N)}
\end{aligned} \tag{2.29}$$

dimana elemen m_{ij}, n_{ij} dan b_i (i = 1,2 dan j = 1,2) didefinisikan pada persamaan (2.17).

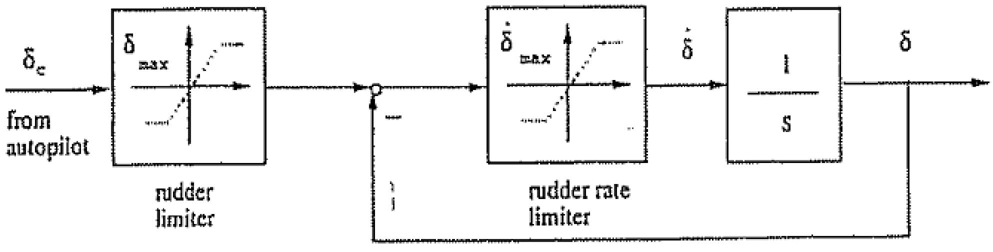
2.4 Aktuator Kendali pada Kapal Kargo

Curtis D. Johnson mendefinisikan aktuator merupakan komponen yang menterjemahkan sinyal energi kecil dari kontroler menjadi energi besar aksi pada proses. Pada penelitian ini aktuator yang digunakan pada kapal kargo adalah *rudder*. Model aktuator berdasarkan teori dari Van Amerongen dimana aktuator pada kapal biasanya dikontrol dengan sistem *on-off.* Sinyal *on-off* dari *rudder* kontroler digunakan untuk membuka dan menutup *port* dan *starboard valve* dari sistem telemotor hidrolik seperti ditunjukkan pada Gambar 2.5



Gambar 2.5. Diagram Hidrolik Aktuator Kapal Kargo

Pada posisi awal Telemotor dan *floating lever* berada pada kondisi (a). telemotor dapat berpindah ke posisi (b) dengan membuka *port valve*. Andaikata *rudder* tetap berada pada posisi awal terkait pada posisi (b) maka hal ini menyebabkan *steering cylinder valve* terbuka dan konsekuensinya *floating lever* bergerak pada posisi (c) ketika sudut *rudder* yang diinginkan telah tercapai. Blok diagram dari aktuator kapal kargo ditampilkan pada Gambar 2.6



Gambar 2.6. Blok Diagram *Rudder Control Loop* dihubungkan *Commanded Rudder angle* δ_c

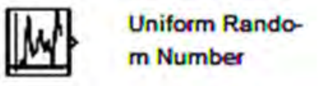
2.5 Gangguan (*disturbance*) pada Sistem

Selama beroperasi pada permukaan laut terdapat tiga jenis gangguan yang dialami oleh kapal kargo yaitu : arus, angin dan gelombang laut. Karakteristik dari setiap gangguan berbeda bergantung pada daerah operasional kapal. Kami berasumsi bahwa dengan massa kapal yang cukup besar yaitu ± 16622 ton maka gangguan utama yang berpotensi mengganggu operasional kapal adalah arus laut,

sedangkan gelombang dan angin kurang begitu mempengaruhi operasional kapal. Berikut ini merupakan review dari setiap gangguan yang terjadi pada kapal kargo.

2.5.1 Arus Laut (*Ocean Current*)

Arus laut adalah sistem sirkulasi horizontal dan vertikal dari air laut yang dihasilkan oleh gravitasi, gesekan angin dan densitas air. Selain arus yang dibangkitkan oleh angin, pertukaran panas pada permukaan air laut yang diikuti dengan perubahan salinitas menyebabkan penambahan komponen pada arus laut yang disebut dengan arus *termohaline.* Untuk mensimulasikan arus laut dan efeknya pada gerak kapal maka akan diterapkan menggunakan blok simulink *uniform random number* pada *software* MATLAB seperti yang ditampilkan pada Gambar 2.7



Gambar 2.7. Blok Simulink *Uniform Random Number*

2.5.2 Angin Laut (*Ocean Wind*)

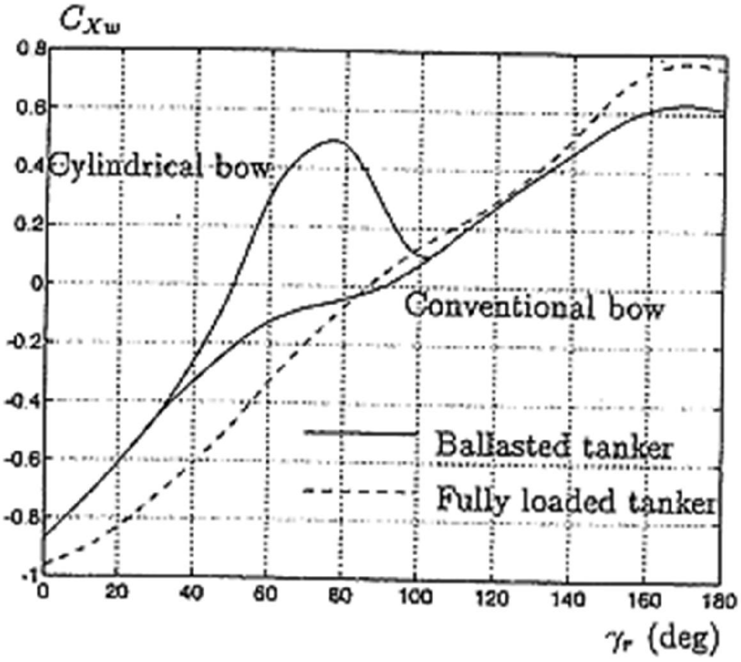
Angin didefinisikan sebagai pergerakan udara pada permukaan bumi. Model matematika dari gaya dan momen angin digunakan dalam kontrol kapal untuk meningkatkan perfoma dan *robustness* sistem dalam kondisi ekstrim. Isherwood menuliskan persamaan gaya angin untuk surge dan *yaw* pada kapal dengan bobot 150.000 sampai 500.000 ton sebagai berikut :

$$X_{wind} = C_{Xw}(\gamma_R)\frac{\rho_a}{7{,}6}V_R^2 A_T \tag{2.30}$$

$$N_{wind} = C_{Nw}(\gamma_R)\frac{\rho_a}{7{,}6}V_R^2 A_L L \tag{2.31}$$

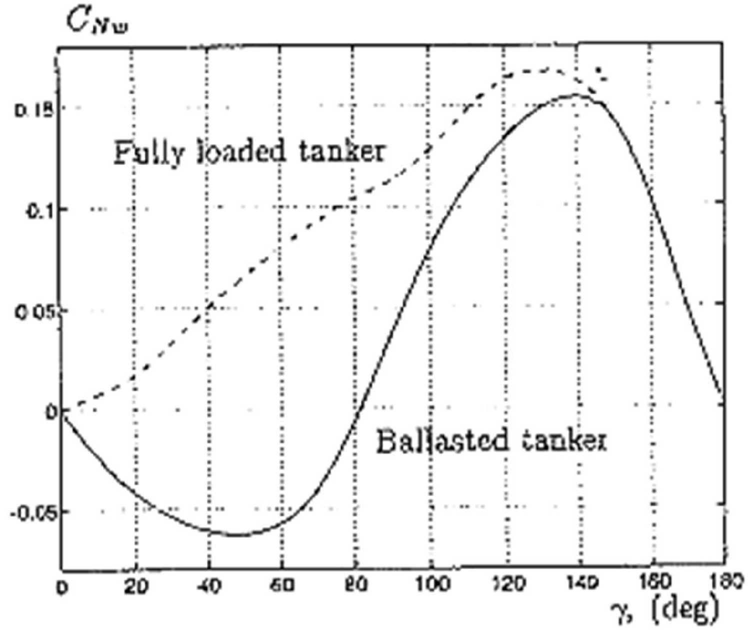
Dimana C_X adalah koefisien gaya, C_N koefisien moment, ρ_a densitas udara, L panjang kapal, γ_R sudut datang angin, A_L dan A_T merupakan daerah lateral dan transversal yang diproyeksikan.

Koefisien terpa angin untuk arah longitudinal C_{Xw} sebagai fungsi relatif sudut datang angin γ_R ditampilkan pada Gambar 2.8



Gambar 2.8. Koefisien C_{Xw} sebagai Fungsi *Relative* γ_R

Sedangkan koefisien momen *yaw* angin C_{Nw} sebagai fungsi relatif sudut datang angin γ_R ditampilkan pada Gambar 2.9



Gambar 2.9. Koefisien C_{Nw} sebagai Fungsi *Relative* γ_R

2.5.3 Gelombang Laut (*Ocean Wave*)

Untuk kapal yang bekerja pada permukaan laut gelombang (*ocean wave*) disimulasikan dengan mendefinisikan :

$$w_{wave} = [X_{waves}, Y_{waves}, N_{waves}]^T \tag{2.32}$$

Dimana X_{waves}, dan N_{waves} dibangkitkan menggunakan model gelombang berikut:

$$h(s) = \frac{K_w s}{s^2 + 2\lambda w_0 s + w_0^2} \tag{2.33}$$

Dimana $K_w = 2\lambda w_0 \sigma$

σ = konstanta intensitas gelombang

λ = koefisien damping

w_0 = frekuensi gelombang dominan

Yang kemudian di kembangkan menjadi :

$$X_{waves} = \frac{K_{w1} s}{s^2 + 2\lambda_1 w_{e1} s + w_{e1}^2} w_1 + d_1$$

$$N_{waves} = \frac{K_{w3} s}{s^2 + 2\lambda_3 w_{e3} s + w_{e3}^2} w_3 + d_3 \tag{2.34}$$

Disini $w_i (i = 1, \ldots, 3)$ dan *wave drift force* $d_1 (i = 1, \ldots, 3)$ adalah *Gaussian white noise*. Amplitudo X_{waves}, Y_{waves} dan N_{waves} disesuaikan dengan memilih konstanta $K_{wi} (i = 1, \ldots, 3)$.

2.6 *Track* (jalur) Sistem

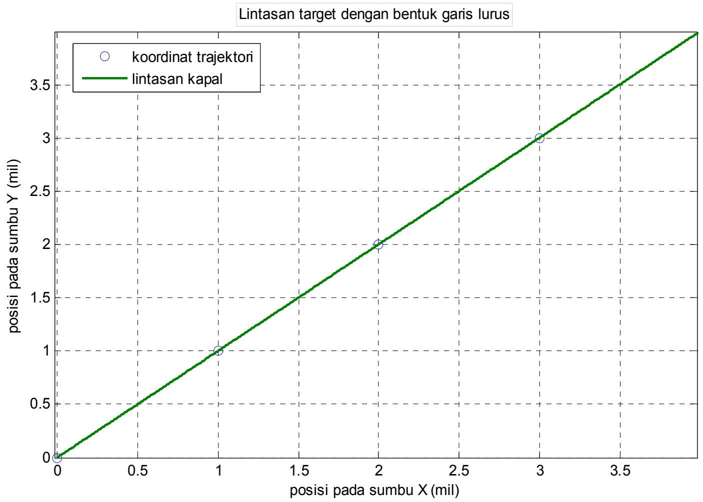
Untuk keperluan simulasi sistem, akan didesain empat jalur (*track*) yang akan diimplementasikan pada penelitian ini, yaitu : *track* garis lurus, *track* lingkaran, *track* persegi panjang dan *track* bebas. *Parameter* untuk setiap *track* dijelaskan sebagai berikut :

A. *Track* Garis Lurus (*Straight Line*)

- Merupakan *track* berbentuk garis lurus dengan koordinat (0,0), (1,1), (2,2), (3,3) dan berakhir pada koordinat (4,4)
- Panjang lintasan 5,6 mil

- Kecepatan kapal 8 knot
- Waktu tempuh 2545,6 s

Desain lintasan ini ditampilkan pada Gambar 2.10



Gambar 2.10. *Track* Garis Lurus

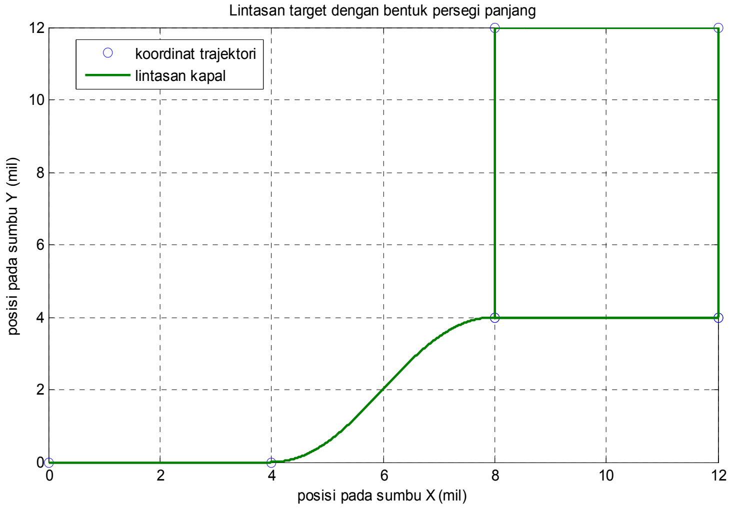
B. *Track* lingkaran (*Circular*)

- Merupakan *Track* dengan bentuk lingkaran (*circular*) searah jarum jam Memiliki radius 4 mil dan pusat pada koordinat (0,0)
- Kecepatan kapal 8 knot
- Waktu tempuh 14000 s

C. *Track* Persegi Panjang

- Merupakan *track* berbentuk persegi panjang dengan koordinat (0,0), (4,0), (8,4), (12,4), (12,12), (8,12), (8,4), (12,4)
- Panjang lintasan 27 mil
- Kecepatan kapal 8 knot
- Waktu tempuh 14600 s

Desain lintasan ini ditampilkan pada Gambar 2.11

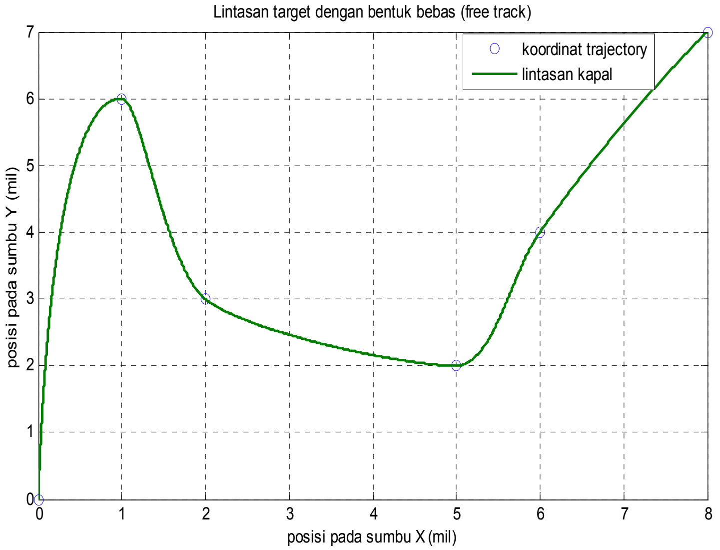


Gambar 2.11. *Track* Persegi Panjang

D. *Track* Bebas (*Free Track*)

- Merupakan *track* yang didesain meggunakan koordinat bebas yaitu : (0,1), (1,6), (2,3), (5,2), (6,4), (8,7)
- Panjang lintasan 20,5 mil
- Kecepatan kapal 8 knot
- Waktu tempuh 9218,3 s

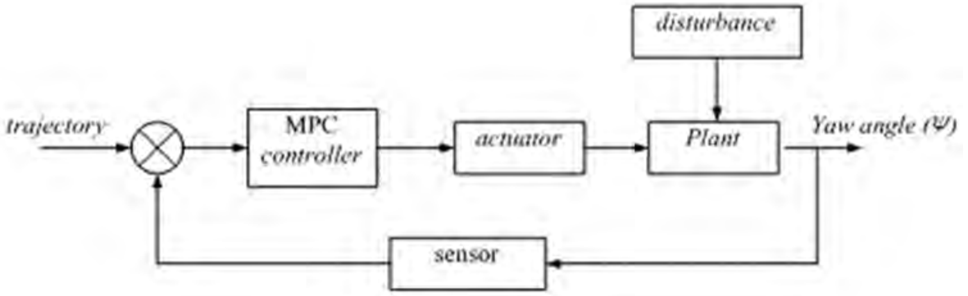
Desain lintasan ini ditampilkan pada Gambar 2.12



Gambar 2.12. *Track* Lintasan Bebas

BAB 3
PERANCANGAN SISTEM

Secara umum blok diagram penelitian ini ditampilkan pada Gambar 3.1 dimana tahapan-tahapan perancangan sistem yang terdiri dari perancangan *plant*, perancangan *trajectory* dan perancangan kontroler akan dibahas dalam bab ini. Ke tiga perancangan tersebut dilakukan dengan menggunakan *software* MATLAB.



Gambar 3.1. Diagram Blok Penelitian

3.1 Perancangan *Plant*

Seperti yang telah dijelaskan di bab sebelumnya bahwa *plant* yang digunakan dalam penelitian ini adalah *plant* kapal kargo. *plant* tersebut dirancang dan disimulasikan dengan menggunakan bantuan *software* MATLAB yaitu pada bagian m-file dan simulink.

Ada beberapa tahapan dalam perancangan *plant* tersebut yaitu :

1. Menurunkan model matematika *plant*

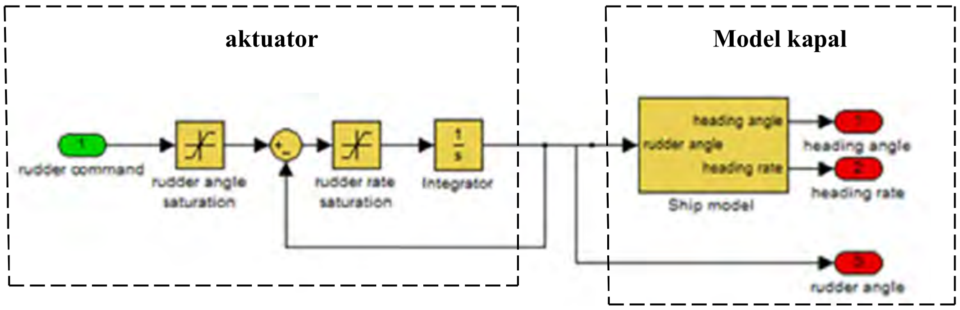
   Seperti yang dijelaskan pada bab 2 bahwa model matematika kapal kargo didapatkan dengan menurunkan model Nomoto 2^{nd} orde pada persamaan (2.26) sehingga diperoleh fungsi transfer (2.27)

2. Membuat simulasi *plant*

   Pada bagian perancangan simulasi ini terdiri atas blok model *plant* dan blok kinematika dari *plant*.

   Dari persamaan (2.26) dan fungsi transfer (2.27) dirancang model lengkap dari *plant* dengan menggunakan m-file dan simulink seperti pada Gambar 3.2. Blok diagram lengkap dari *plant* kapal kargo yang digunakan dalam

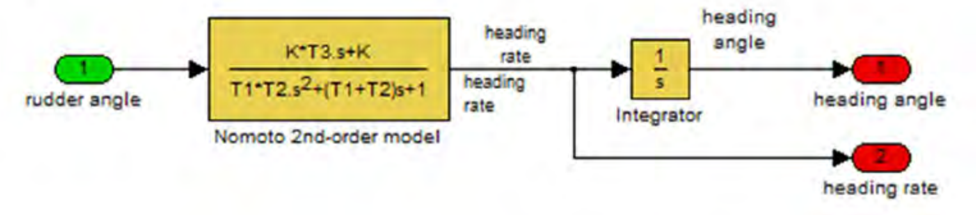
penelitian ini terbagi atas dua bagian utama yaitu blok model kapal dan blok aktuator.



Gambar 3.2. Blok Lengkap Model *Plant* Kapal Kargo

3.1.1 Blok model kapal

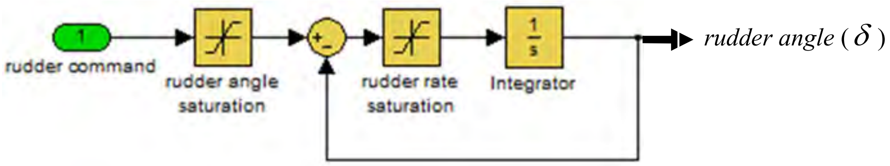
Isi dari blok model kapal ditampilkan pada Gambar 3.3 dimana terdiri atas *input* sudut *rudder* dan *output* berupa sudut *heading* serta kecepatan *heading*. Langkah selanjutnya adalah menghubungkan blok ini dengan blok kinematika kapal dan blok aktuator.



Gambar 3.3.Blok Model Kapal Kargo

3.1.2 Blok aktuator

Dalam simulasi computer untuk penelitian ini kami menggunakan representasi aktuator dari teori Van Amerongen untuk *Tracking control* yang ditunjukan pada Gambar 3.4



Gambar 3.4.Blok Diagram Aktuator Model Kapal Kargo

Dimana secara umum sudut *rudder* dan *rudder rate limiter* pada Gambar 3.4 diset dalam *range* berikut :

Sudut *rudder* : $\delta_{\max} = 30\,(deg)$ (3.1)

Rudder rate limiter: $2\frac{1}{3}(\deg/s) \le \dot{\delta}_{\max} < 7(\deg/s)$ (3.2)

Kecepatan perubahan sudut *rudder* antara 5 – 20 (deg/s) biasanya juga digunakan sebagai *rudder roll stabilization* (RRS) agar sistem beroperasi dengan baik.

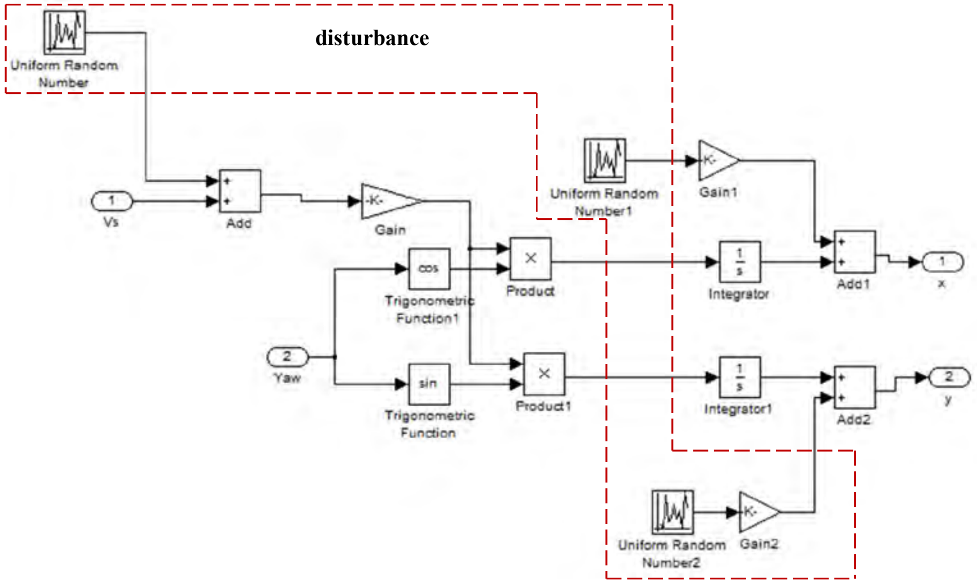
3.1.3 Blok Kinematika Kapal

Sesuai dengan uraian pada bab 2 kinematika *plant* menggunakan analisa vektor dalam bidang *xy* untuk gerak translasi pada sumbu – x (*surge*) dan gerak rotasi pada sumbu – z (*yaw*) hal ini didasarkan pada gerak real kapal kargo saat beroperasi pada permukaan laut dengan asumsi :

1. massa kapal terdistribusi *homogen*
2. bidang pada sumbu - x dan sumbu – z simetri
3. *gerak heave, roll* dan *pitch* konstan

Dalam perancangan kinematika kapal pada penelitian ini kami juga menyertakan blok gangguan (*disturbance*) dari lingkungan yang berupa arus laut dengan kecepatan maksimum 2 knot.

Representasi dari kinematika kapal kargo menggunakan simulink pada *software* MATLAB ditampilkan pada Gambar 3.5



Gambar 3.5. Kinematika Model *Plant* Kapal Kargo

3.2 Memasukan nilai-nilai *parameter Plant*

Proses selanjutnya yaitu memasukan *parameter plant* kapal kargo kedalam model simulink dan m-file yang digunakan untuk simulasi. Data-data mengenai *parameter plant* diambil dari buku karya Thor.I.Fossen. *Parameter plant* kapal kargo ditampilkan pada Tabel 3.1

Tabel 3.1. *Parameter* Simulasi *Plant* Kapal Kargo

Notasi	Keterangan	Ukuran
L	Panjang kapal	161 (m)
∇	Massa kapal	16622 (dwt)
K (1/s)	Gain	0.185
T_1	Propeller thrust time constant 1	118 (s)
T_2	Propeller thrust time constant 2	7.8 (s)
T_3	Propeller thrust time constant 3	18.5(s)

3.3 Perancangan Kontroler

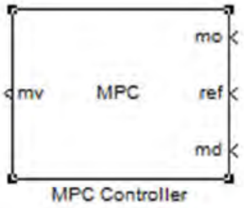
Kontroler yang digunakan untuk *Tracking* pada *plant* kapal kargo adalah kontroler *Model Predictive Control* (MPC) dengan *constrain* dari aktuator yaitu kecepatan sudut *rudder* maksimum.

untuk merancangan kontroler MPC digunakan simulink dari *software* MATLAB dengan beberapa tahapan yang harus dilakukan yaitu :

1. Mendesain struktur MPC.

Untuk mendesain Struktur MPC maka dilakukan langkah-langkah sebagai berikut :

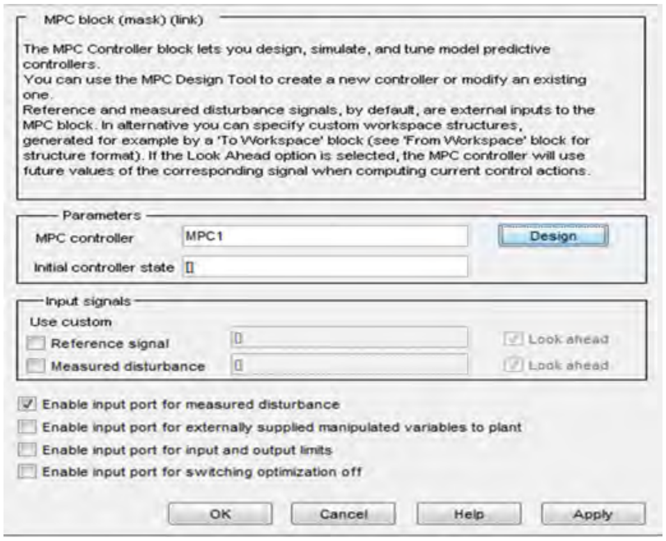
a. Klik *icon* simulink pada *toolbar* MATLAB kemudian pilih blok MPC kontroler pada kolom *libraries* seperti ditampilkan pada Gambar 3.6



Gambar 3.6.Blok MPC Kontroler pada Simulink MATLAB

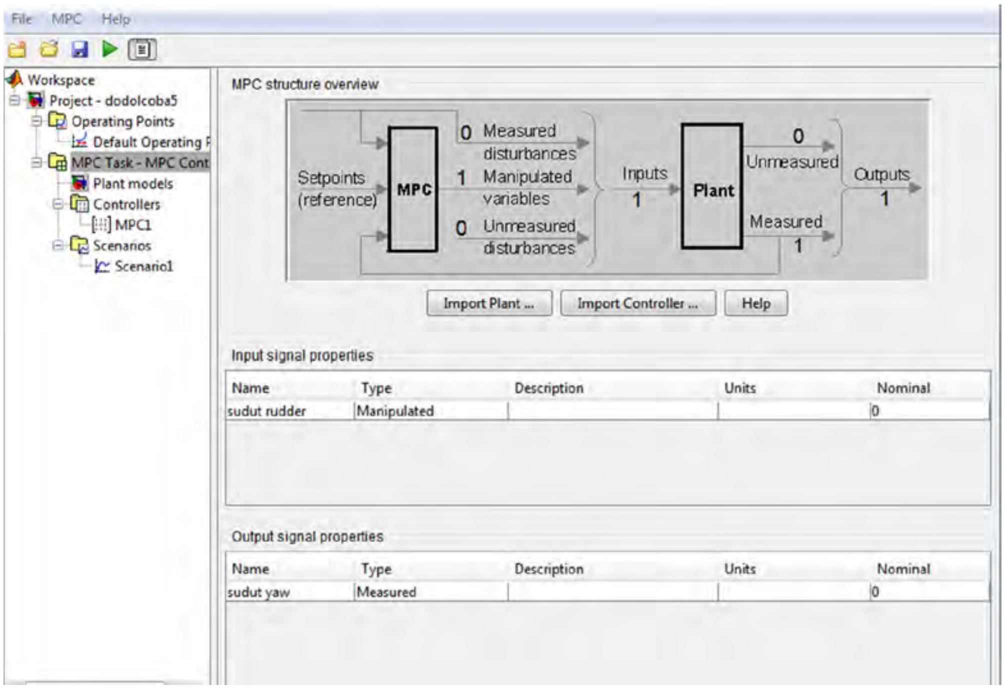
b. Klik *icon new model* pada *toolbar* kemudian tempatkan blok MPC kontroler pada *new model window*.

c. Pilih dan *double-klik* blok MPC kontroler. Maka akan muncul menu utama MPC *controller desainer* seperti pada Gambar 3.7



Gambar 3.7. Menu Utama MPC kontroler Desainer MATLAB

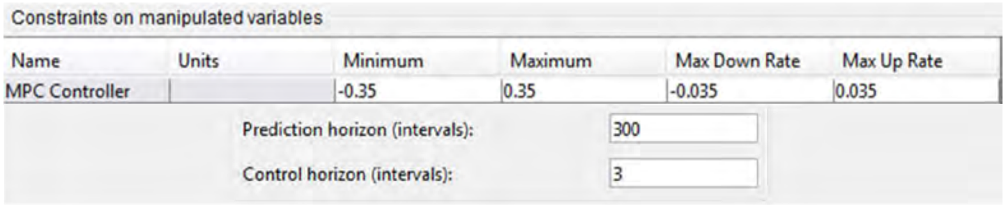
d. Langkah selanjutnya klik tombol desain pada menu utama MPC *controller desainer* maka kita akan masuk kedalam menu *control and estimation tool manager* seperti pada Gambar 3.8 berikut



Gambar 3.8.*Control and Estimation Tool Manager* MPC

e. Pada menu *control and estimation tool manager* terdapat dua kolom. Pada kolom sebelah kanan kita dapat menentukan *input* dan *output* untuk MPC kontroler yaitu sudut *rudder* (δ) sebagai *input* dan sudut *yaw* (ψ) sebagai *output*.

f. Pada kolom sebelah kiri terdapat beberapa mode diantaranya mode controller dan mode scenario. Dalam mode controllers klik MPC 1, maka kita dapat menentukan model dan *horizon* untuk *parameter* MPC yang terdiri atas *control horizon* (N_c) yaitu lintasan kontrol yang akan datang, *prediction horizon* (N_p) atau panjang *window* optimisasi serta *constrain* berupa kecepatan perubahan sudut *rudder*. *Parameter* MPC pada penelitian ini ditampilkan pada Gambar 3.9

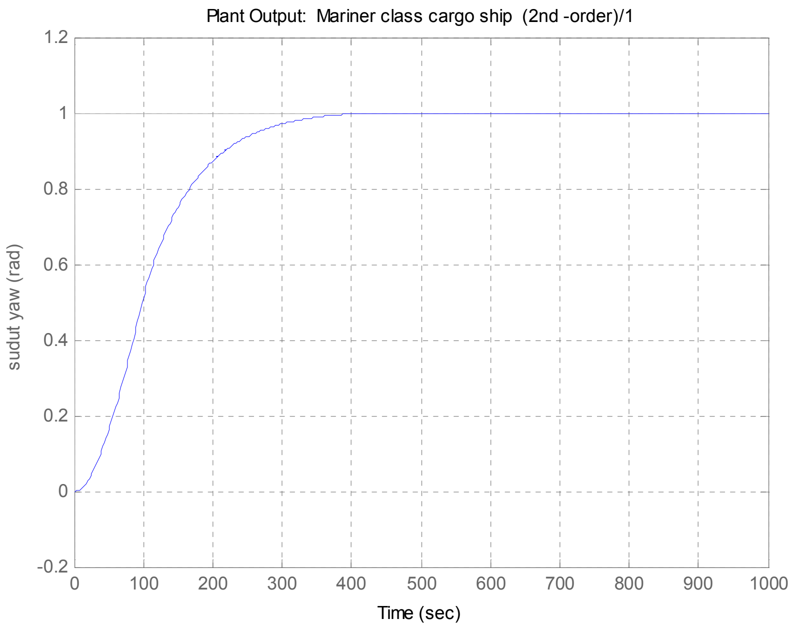


Gambar 3.9. *Parameter* MPC

g. Mode terakhir adalah *mode scenarios*. Mode ini kita gunakan untuk mensimulasikan dan mengetahui hasil dari MPC kontroler yang telah didesain seperti pada Gambar 3.10

Gambar 3.10. *Mode Scenarios* MPC

h. Dengan cukup dengan men-klik tombol *simulate,* maka hasil *output* dari kontroler *Model predictive Control* dengan *constrain* ditampilkan pada Gambar 3.11



Gambar 3.11. MPC Kontroler *Output*

3.4 Perancangan *Trajectory*

Perancangan *trajectory* dilakukan dengan bantuan m-file pada *software* MATLAB. Terdapat dua tahap dalam proses perancangan *trajectory* yaitu tahap menentukan titik koordinat target kapal dan tahap target *trajectory tracking*.

3.4.1 Tahap Menetukan Koordinat Target

Pada tahap ini ditentukan koordinat lintasan (x,y) yang akan dilalui oleh kapal seperti yang telah dibahas pada BAB II. Selanjutnya untuk merealisasikan hal tersebut dituliskan script program yang ditampilkan pada Lampiran A.1

3.4.2 Tahap *Trajectory Tracking*

Setelah koordinat target telah terealisasi tahap selanjutnya adalah men*tracking* koordinat target tersebut sehingga diharapkan *plant* dapat memenuhi atau mencapai target sesuai dengan lintasan yang telah didesain sebelumnya. Untuk merealisasikan tahap ini dilakukan dengan menulis *script* program pada m-file yang ditampilkan pada Lampiran A.2:

Halaman ini sengaja dikosongkan

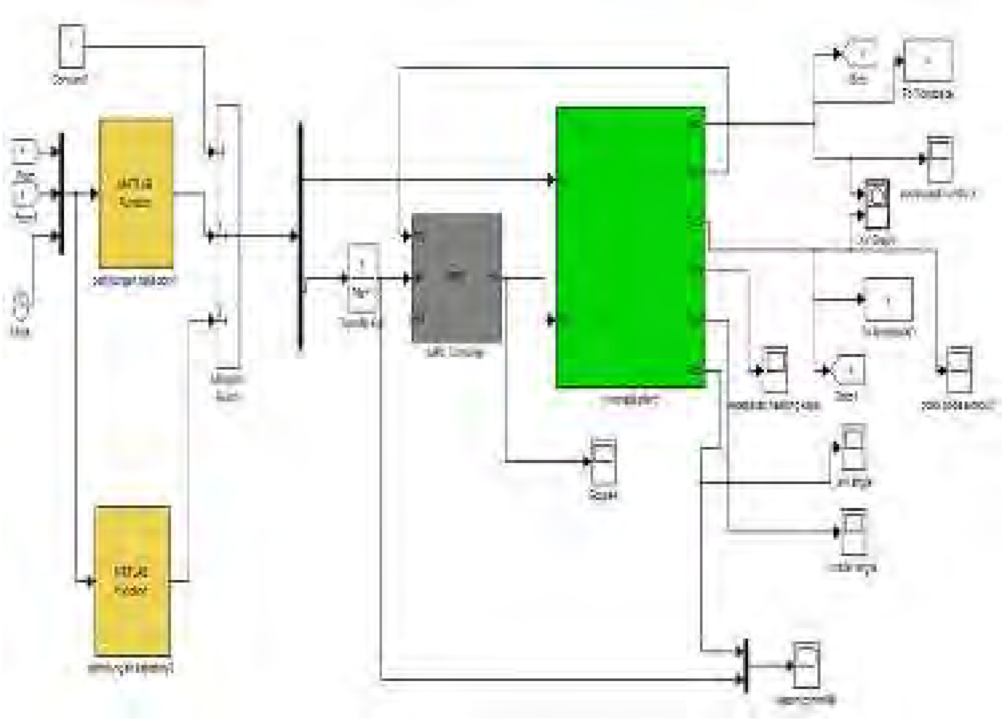
BAB 4
HASIL DAN PEMBAHASAN

Berdasarkan perancangan sistem yang telah dilakukan pada bab tiga, maka langkah selanjutnya adalah melakukan pengujian dan analisis data. Pengujian ini dilakukan untuk mengetahui apakah sistem berjalan sesuai dengan harapan dalam melakukan kinerja dan menjawab tuntutan yang ada.

Setelah melakukan pengujian maka dapat diketahui nilai perfoma dan kemampuan dari sistem yang telah dibuat. Pada bab ini, akan dijelaskan mengenai hasil simulasi sistem dan analisisnya.

4.1 Hasil simulasi sistem

Bentuk keseluruhan sistem *Model Predictive Control* (MPC) dengan *constrain* untuk pengaturan *tracking* pada kapal cargo ditampilkan pada Gambar 4.1



Gambar 4.1 Bentuk Keseluruhan Sistem

4.2 Hasil Pengujian simulasi sistem

Pengujian *tracking* dilakukan pada 4 bentuk lintasan yaitu lintasan lurus, lintasan persegi panjang, lintasan lingkaran dan lintasan bebas (*free track*) hasil dari pengujian tersebut ditampilkan sebagai berikut :

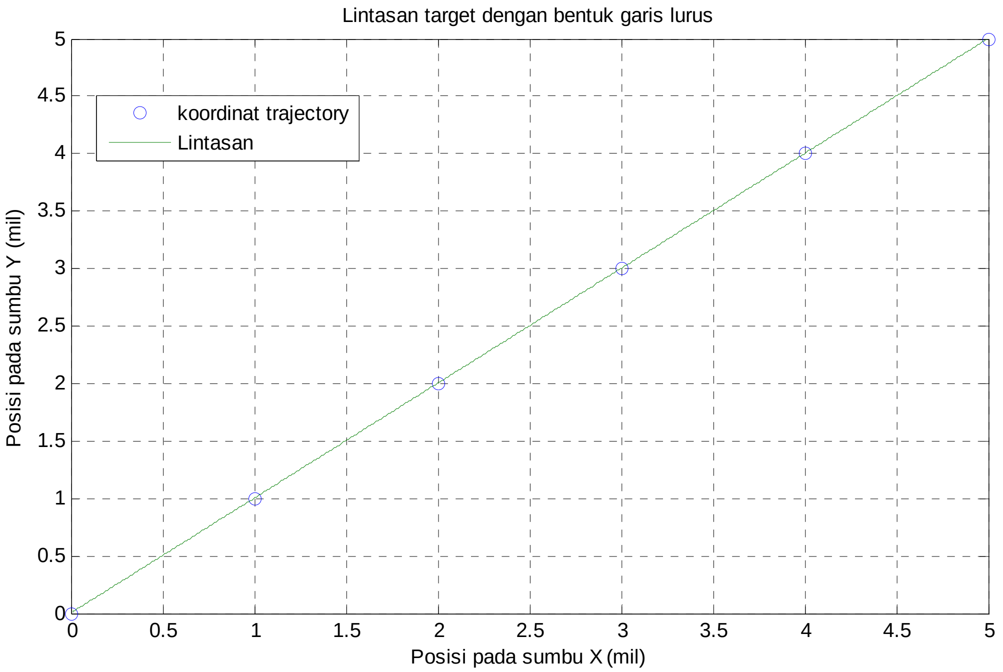
a. Lintasan Lurus

Parameter uji untuk lintasan garis lurus ditampilkan pada Tabel 4.1

Tabel 4.1 *Parameter* Uji Simulasi Lintasan Garis Lurus

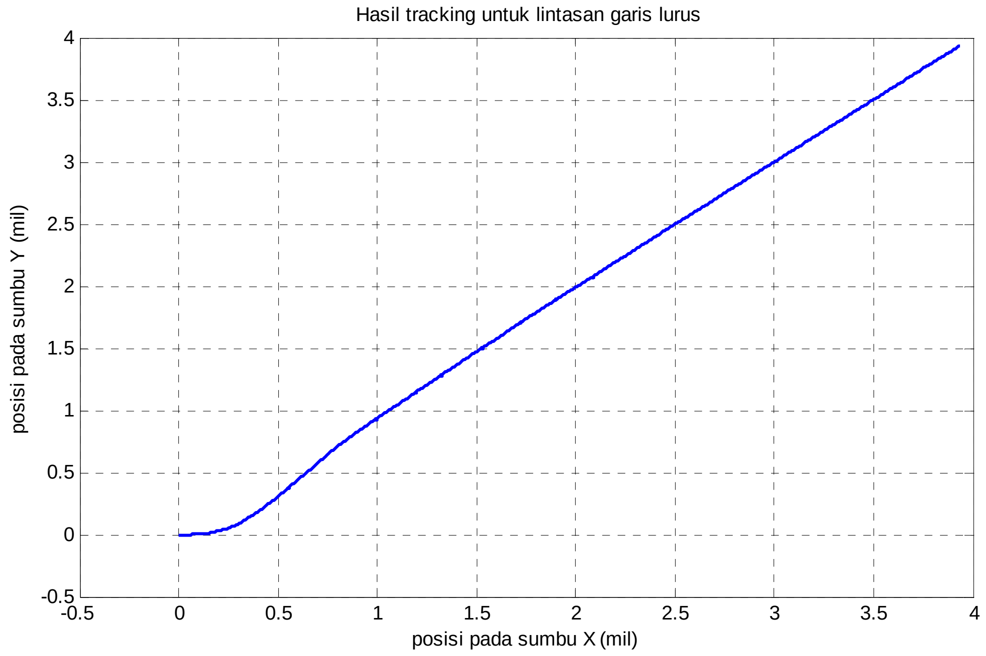
Parameter Uji	
Koordinat tujuan	(0,0), (1,1), (2,2), (3,3), (4,4)
Kecepatan kapal	8 knot
Jarak lintasan	5,65 mil
Waktu tempuh	2545 s

Dengan menginput data pada Tabel 4.1 pada m-file *software* MATLAB diperoleh hasil simulasi untuk lintasan target dengan bentuk garis lurus seperti pada Gambar 4.2



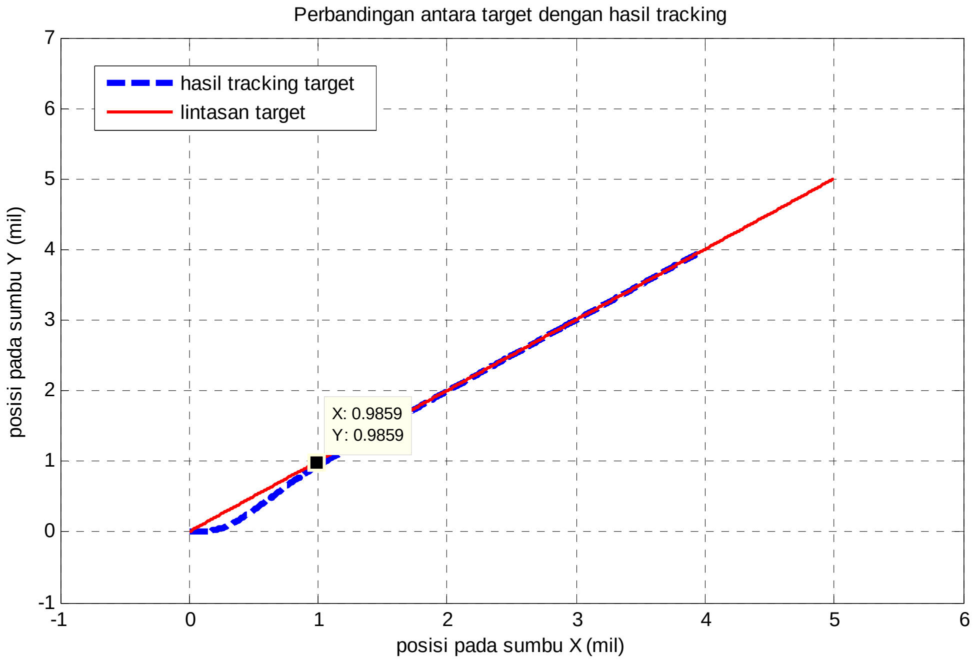
Gambar 4.2 Lintasan Target Garis Lurus

Simulasi hasil *tracking* dari lintasan target untuk bentuk garis lurus ditampilkan pada Gambar 4.3 berikut ini.



Gambar 4.3 Hasil Simulasi *Tracking* Lintasan Garis Lurus

Langkah selanjutnya adalah membandingkan hasil simulasi dari target dan hasil *tracking* dari lintasan garis lurus yang ditampilkan pada Gambar 4.4



Gambar 4.4 Perbandingan Hasil Simulasi Target dan *Tracking* pada Lintasan Garis Lurus

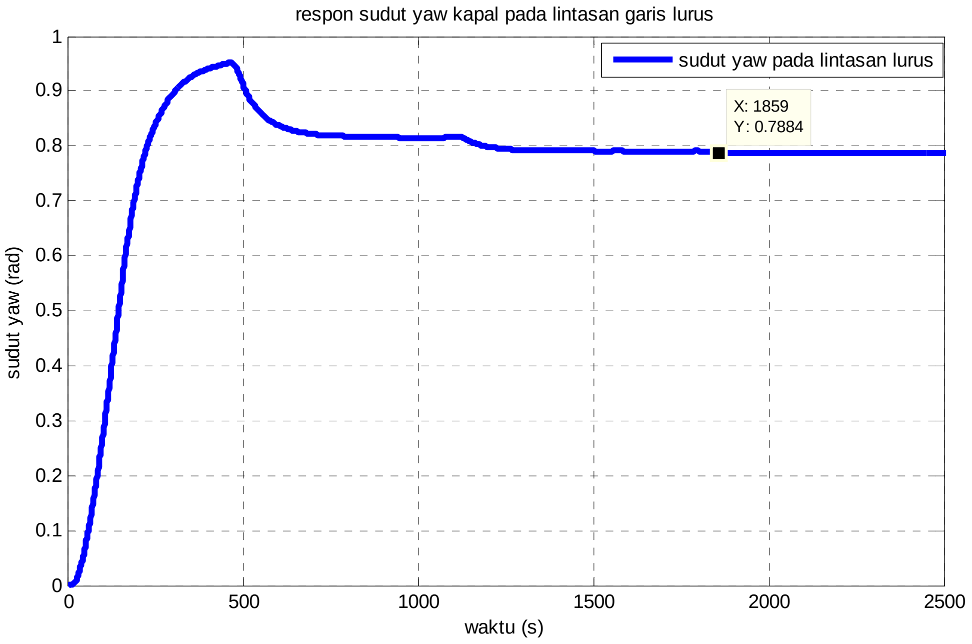
dari Gambar 4.4 diketahui bahwa *plant* mulai memasuki lintasan target pada koordinat (0.9859,0.9859) mil dengan analisa *error* untuk lintasan garis lurus ditampilkan pada Tabel 4.2 berikut.

Tabel 4.2 Analisa *Error Tracking* pada Uji Simulasi Lintasan Lurus

No	X_{target}	$X_{tracking}$	*Error* trajectory (mil)	Y_{target}	$Y_{tracking}$	*Error* Trajectory (mil)
1	0	-0,00112	-0,00112	0	-0,00112	-0,00112
2	0	0,000786	0,000786	0	-0,00112	-0,00112
3	0	0,002696	0,002696	0	-0,00112	-0,00112
4	0	0,004606	0,004606	0	-0,00112	-0,00112
5	0	0,006516	0,006516	0	-0,00112	-0,00112
6	0	0,008426	0,008426	0	-0,00112	-0,00112
7	0	0,010336	0,010336	0	-0,00112	-0,00112
8	0	0,012246	0,012246	0	-0,00112	-0,00112
9	0	0,014155	0,014155	0	-0,00112	-0,00112
10	0	0,016065	0,016065	0	-0,00111	-0,00111
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2541	4,970173	3,91938	-1,04765	4,967033	3,916507	-1,05053
2542	4,970173	3,921273	-1,04765	4,970173	3,918402	-1,05177
2543	4,970173	3,923165	-1,04765	4,973312	3,920297	-1,05302
2544	4,970173	3,925057	-1,04765	4,976452	3,922192	-1,05426
2545	4,970173	3,926695	-1,04765	4,979592	3,924087	-1,0555
2546	4,970173	3,928842	-1,04765	4,982732	3,925982	-1,05675
2547	4,970173	3,930735	-1,04765	4,985871	3,927877	-1,05799
2548	4,970173	3,9	-1,04765	4,989011	3,929772	-1,05924
2549	4,970173	3,91938	-1,04765	4,992151	3,931667	-1,06048
2550	4,970173	3,91938	-1,04765	4,99529	3,933562	-1,06173
2551	4,970173	3,91938	-1,04765	4,9984	3,936737	-1,06169
$\sum_{total}$	39181,25	4951,149	969,889	3981,25	4832,004	850,7546
***Error* rerata sumbu X (mil)**			**0,38**	***Error* rerata sumbu Y(mil)**		**0,33**
% *error* rerata			**5,7**	**%*error* rerata**		**6,3**

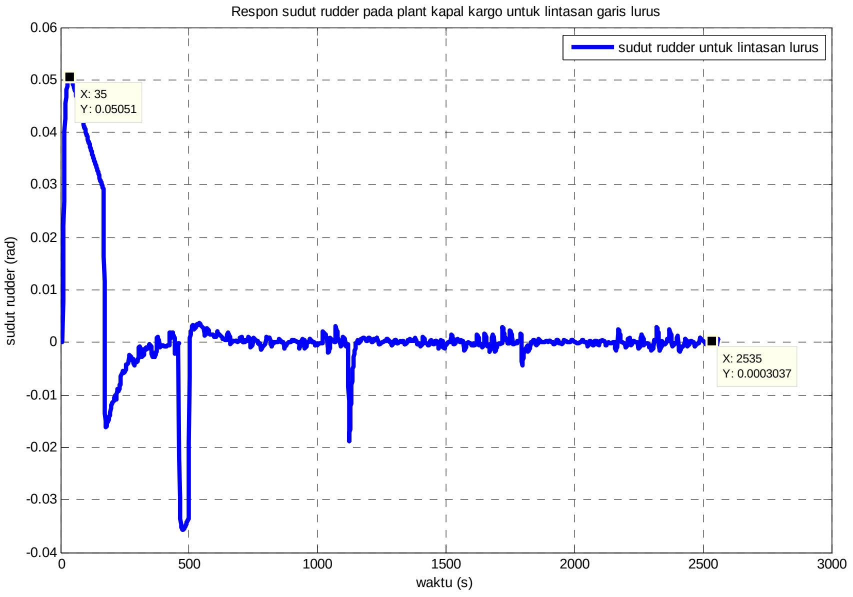
Berdasarkan analisa diperoleh nilai *error* rerata pada lintasan garis lurus dengan panjang lintasan 5,25 mil untuk sumbu X sebesar 0,38 mil atau 5,7 % dan untuk sumbu Y sebesar 0,33 mil atau 6,3 %

Respon sudut *yaw* (ψ) kapal untuk *tracking* pada lintasan garis lurus ditampilkan pada Gambar 4.5 dimana *plant* mengalami *steady state* pada waktu 18959 detik.



Gambar 4.5 Hasil Simulasi Respon Sudut *Yaw* untuk Lintasan Lurus

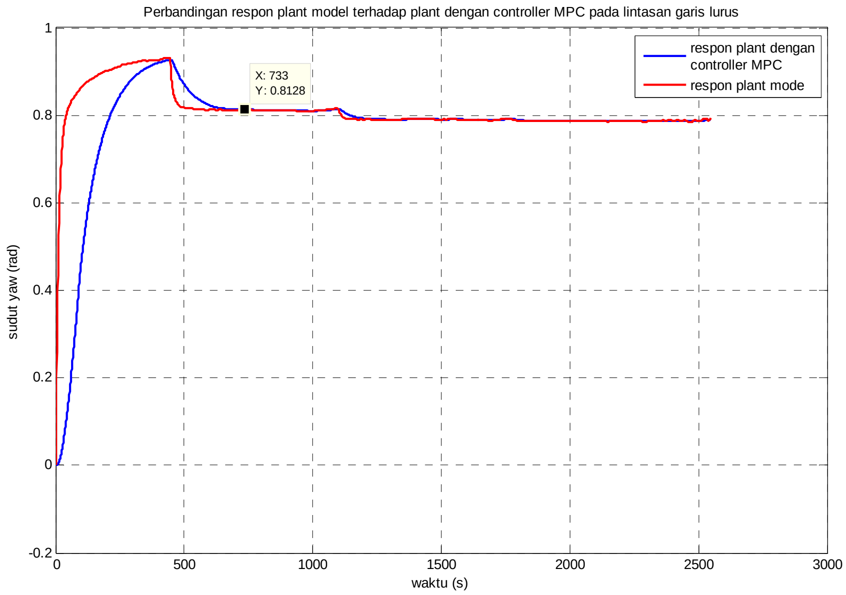
Respon sudut *rudder* (δ) pada simulasi *tracking* ditampilkan pada Gambar 4.6 berikut.



Gambar 4.6 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Lurus

Gambar 4.6 menunjukkan sudut *rudder* berfluktuasi dengan *overshoot* 0,05 rad, hal ini disebabkan oleh adanya *disturbance* berupa arus laut dengan kecepatan 2 knot. Sudut *rudder* mengalami kondisi *steady state* pada waktu 2535 detik.

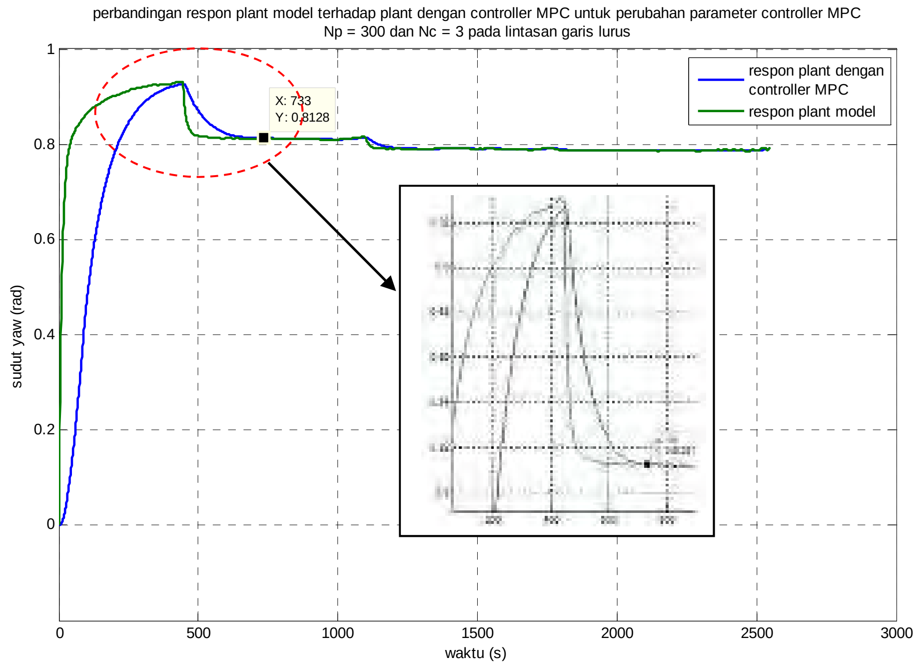
Perbandingan respon *plant* model terhadap *plant* dengan kontroler MPC pada lintasan lurus ditampilkan pada Gambar 4.7 dimana *plant* dengan kontroler MPC mampu mengikuti respon dari *plant* model dan *steady state* pada waktu 733 detik



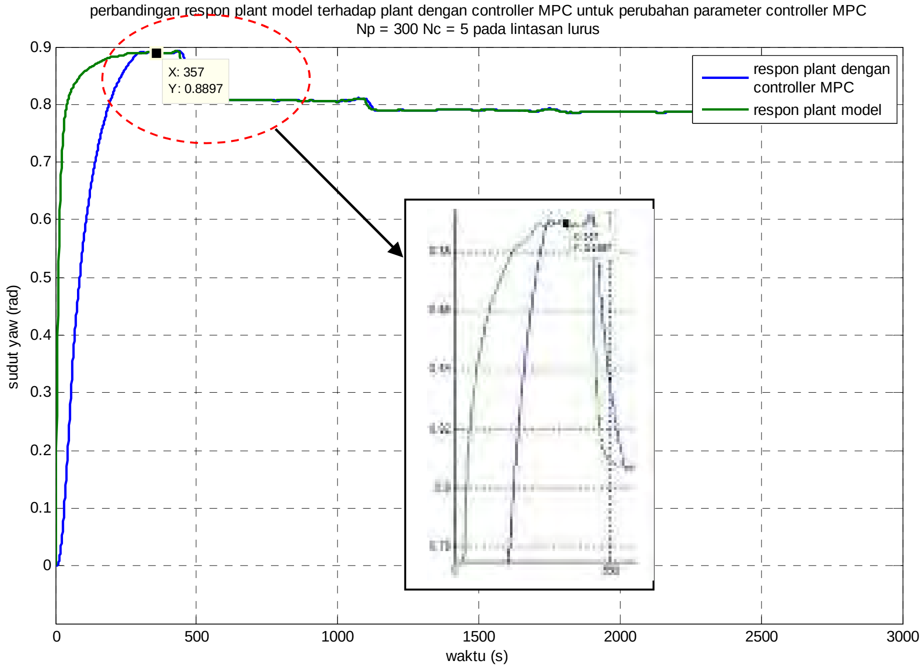
Gambar 4.7 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk Lintasan Garis Lurus

4.3 Perubahan *Parameter* Kontroler

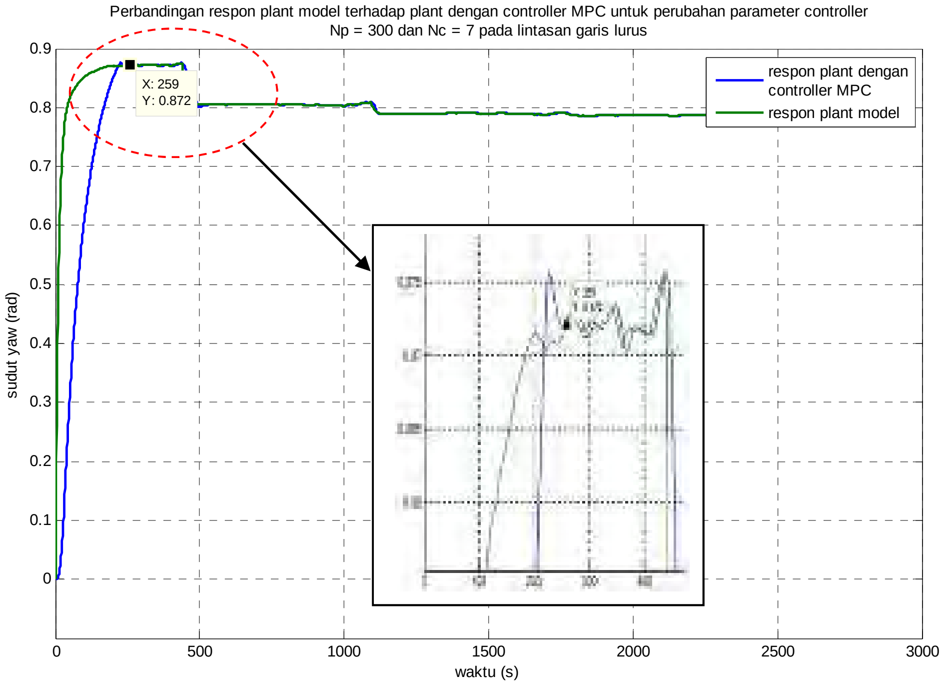
Di dalam MPC terdapat dua *parameter* kontroler yang mempengaruhi performa dari kontroler tersebut. *Parameter* tersebut yaitu *prediction horizon* N_p dan *control horizon* N_c. Pengujian dilakukan dengan mengubah *parameter prediction horizon* N_p dengan nilai N_p = 300, 400 dan 500 serta menetapkan *parameter* yang lain yaitu N_c = 3,5,7 dan 9. Pengujian dilakukan pada lintasan lurus, lintasan persegi panjang dan lintasan bebas. Hasil simulasi sistem dengan perubahan *prediction horizon* N_p dengan nilai N_p = 300 dan N_c = 3,5,7 dan 9 pada lintasan garis lurus disajikan pada Gambar 4.8 - 4.11 berikut.



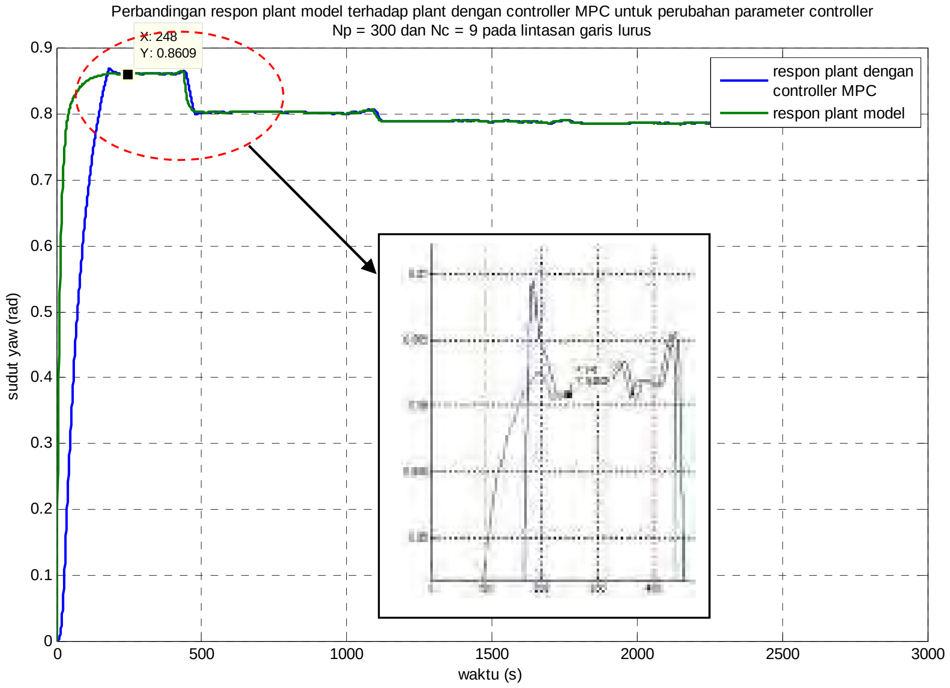
Gambar 4.8 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroller MPC untuk N_p = 300 dan N_c = 3 pada Lintasan Garis Lurus



Gambar 4.9 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p= 300 dan N_c = 5 pada Lintasan garis Lurus



Gambar 4.10 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 7 pada Lintasan Garis Lurus



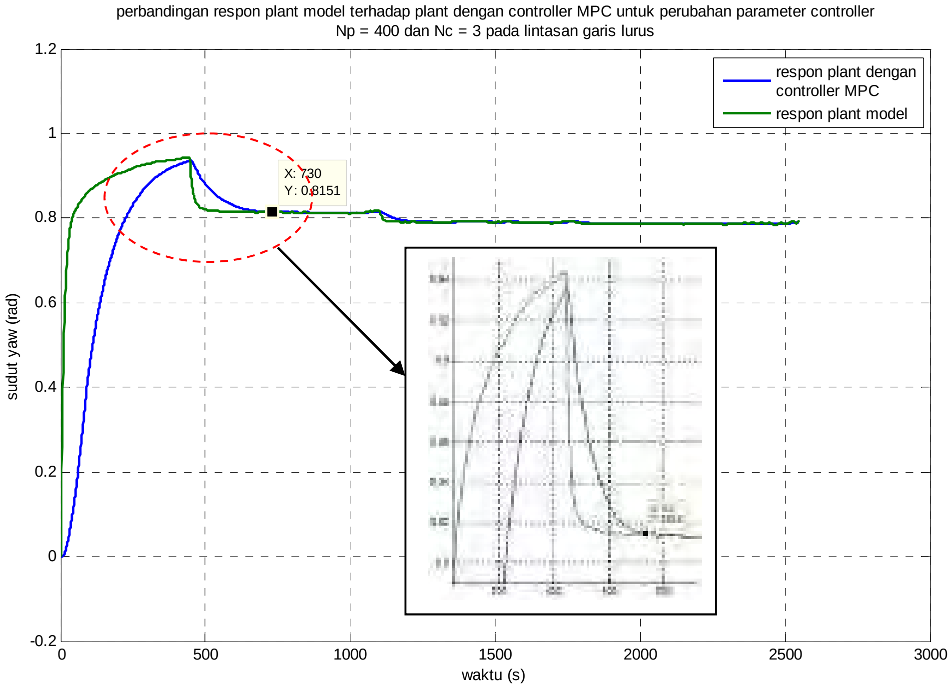
Gambar 4.11 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 9 pada Lintasan Garis Lurus

Berdasarkan hasil pengujian pada Gambar 4.8, 4.9, 4.10 dan 4.11 respon kontroler untuk mengendalikan *Plant* dengan *parameter* N_p = 300 membutuhkan waktu yang berangsur – angsur semakin cepat yaitu 248 detik pada N_c = 9. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model ditampilkan pada Tabel 4.3 berikut

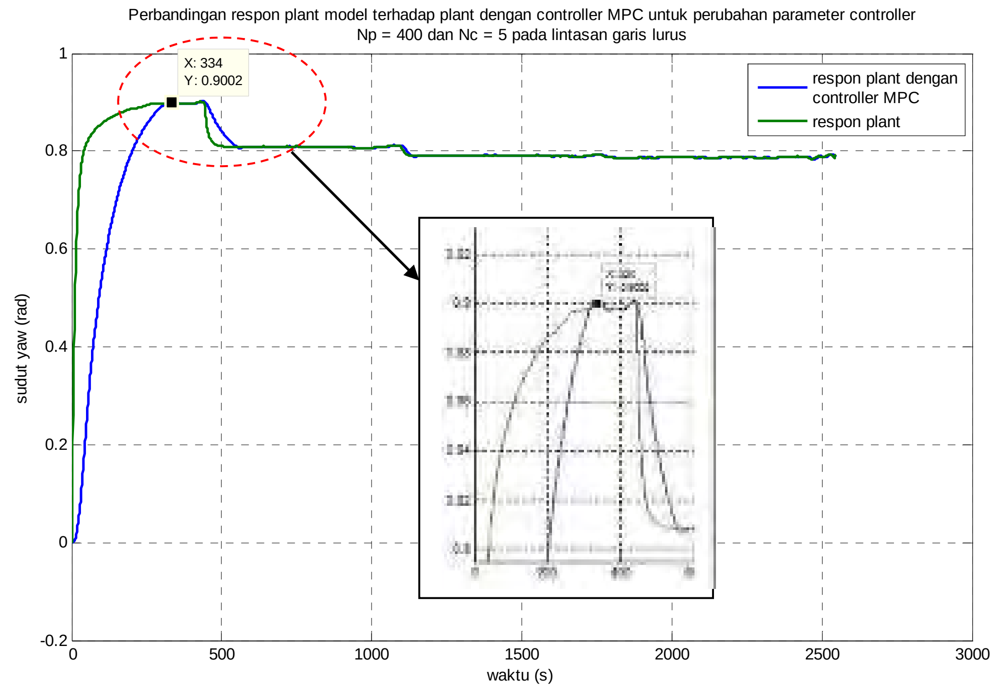
Tabel 4.3 Analisa *Response Time* Kontroler MPC dengan *horizon prediction* N_p = 300 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Lurus

Horizon control (N_c)	*Response time* Kontroler MPC
3	733
5	357
7	259
9	248

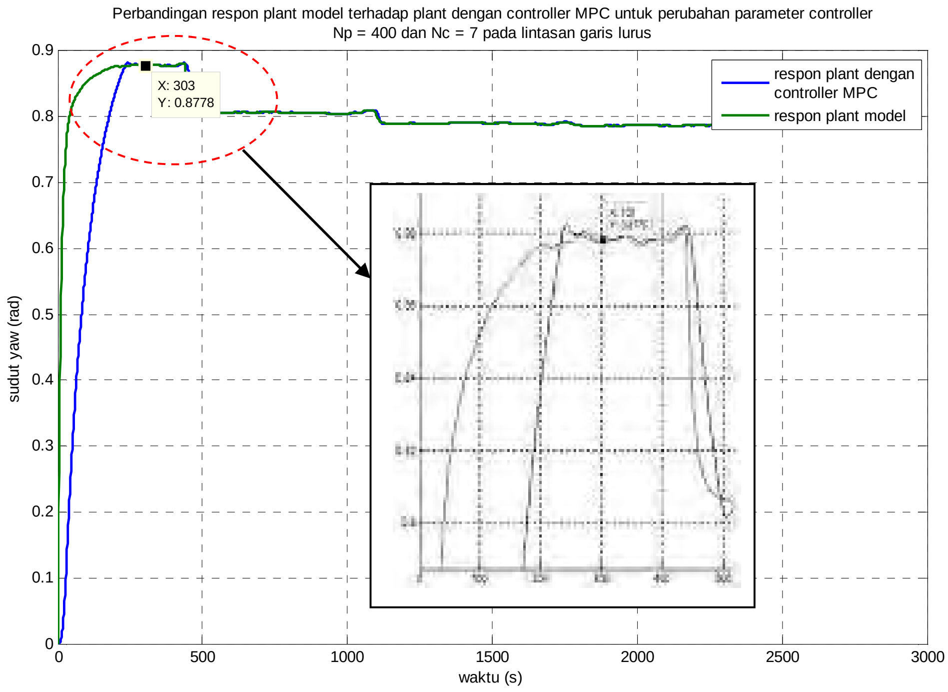
untuk perubahan *parameter* N_p = 400 dan N_c = 3,5,7 dan 9 disajikan pada Gambar 4.12 – 4.15



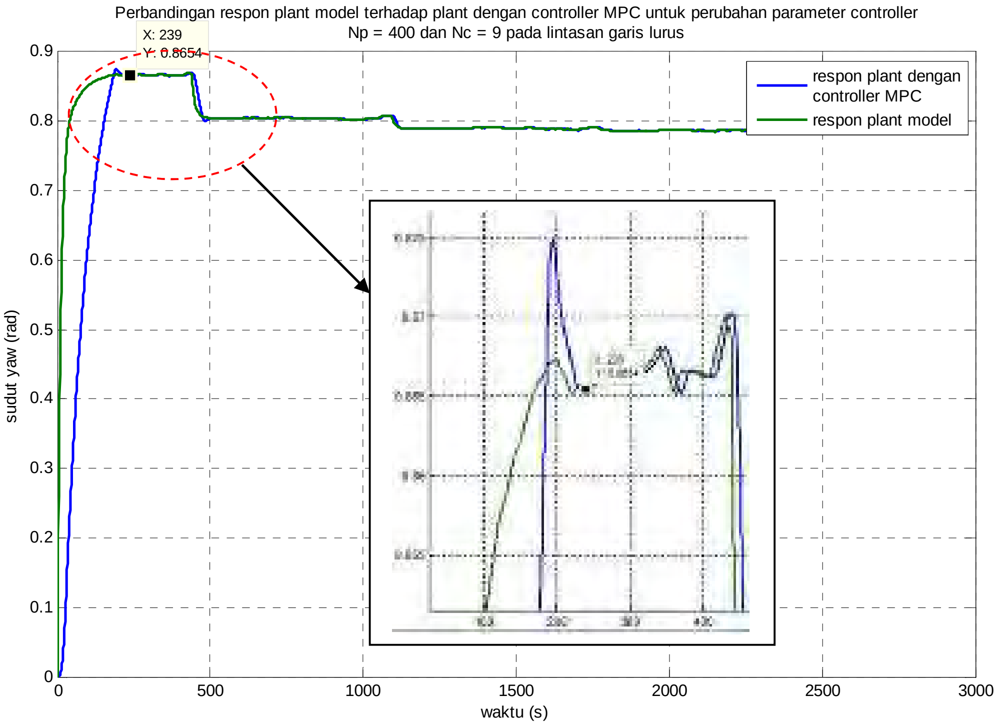
Gambar 4.12 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 3 pada Lintasan Garis Lurus



Gambar 4.13 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 5 pada Lintasan Garis Lurus



Gambar 4.14 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 7 pada Lintasan Garis Lurus



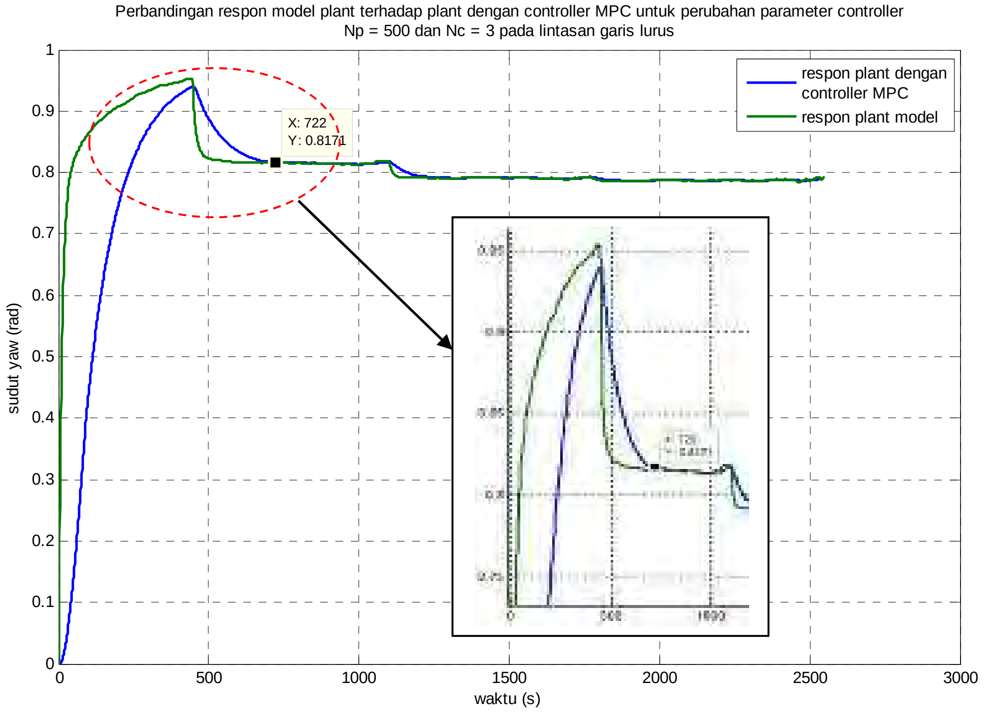
Gambar 4.15 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroller MPC untuk N_p = 400 dan N_c = 9 pada Lintasan Garis Lurus

Hasil pengujian pada Gambar 4.12, 4.13, 4.14 dan 4.15 menunjukkan respon kontroler MPC untuk mengendalikan *plant* dengan *parameter* N_p = 400 membutuhkan waktu lebih cepat dibandingkan dengan N_p = 300 yaitu sebesar 239 detik pada N_c = 9. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model ditampilkan pada Tabel 4.4 berikut

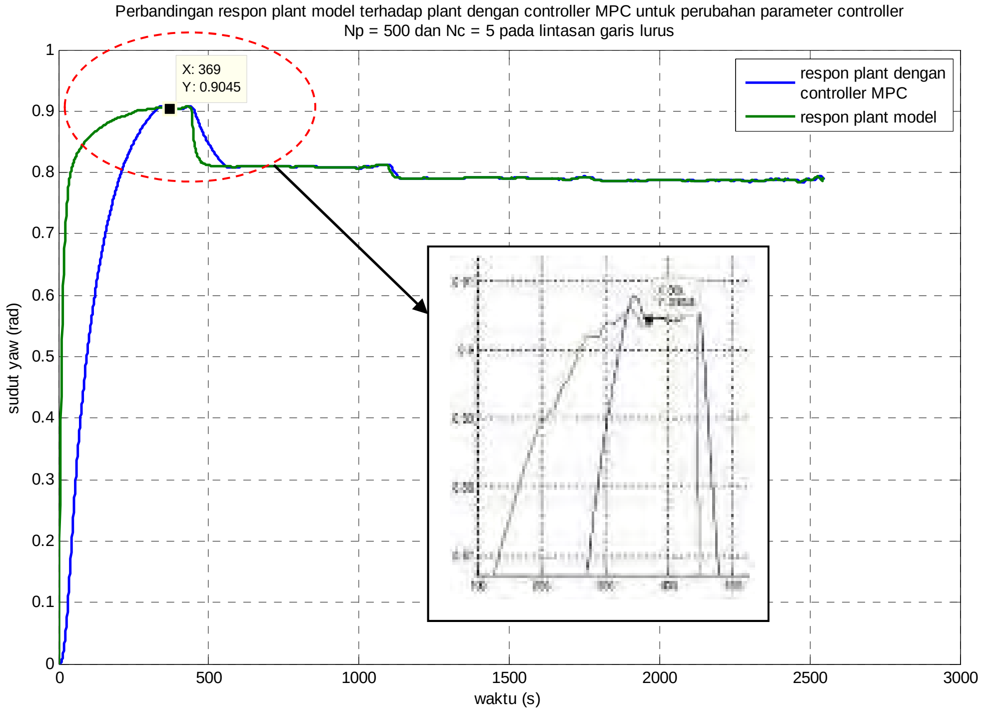
Tabel 4.4 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 400 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Lurus

Horizon control (N_c)	*Response time* Kontroler MPC
3	730
5	334
7	303
9	239

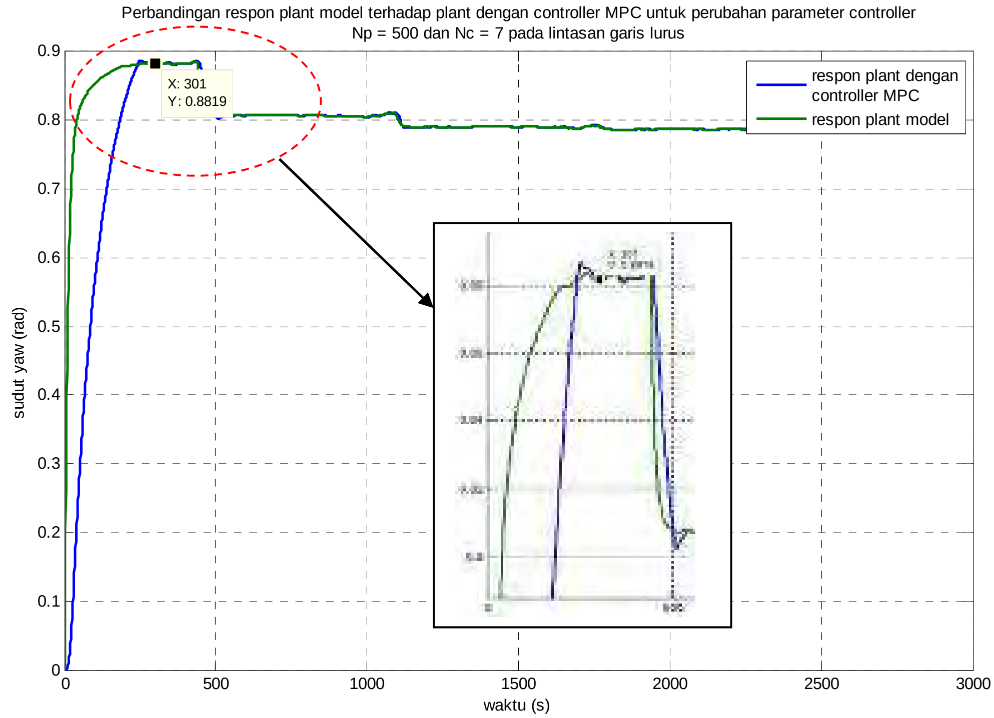
sedangkan $N_p = 500$ berturut-turut disajikan pada Gambar 4.16 – 4.19



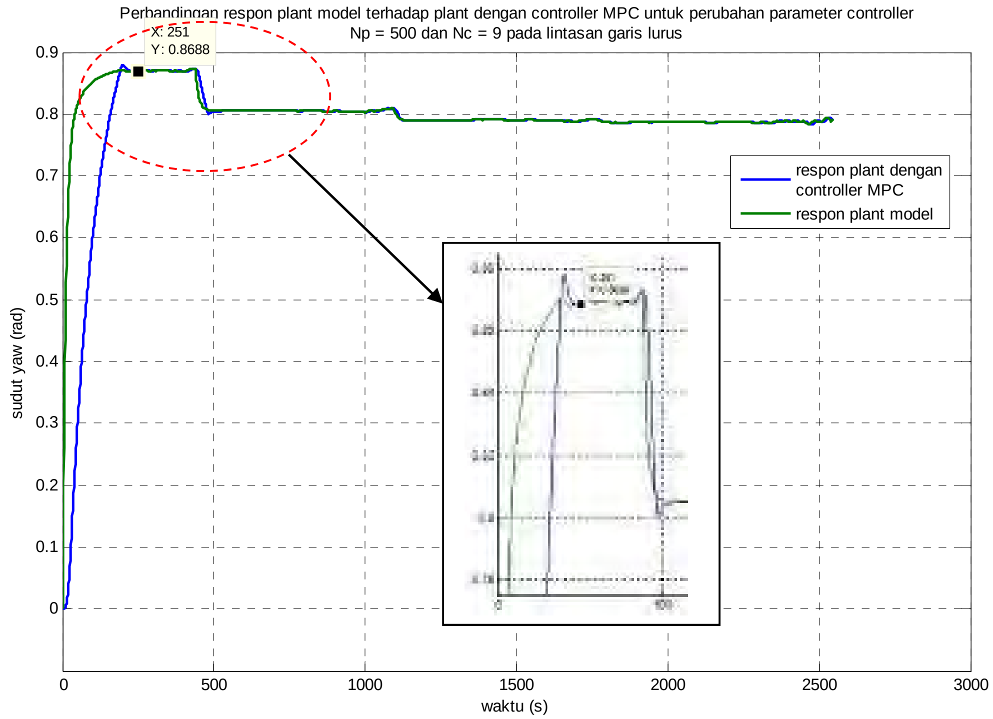
Gambar 4.16 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 3 pada Lintasan Garis Lurus



Gambar 4.17 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 5 pada Lintasan Garis Lurus



Gambar 4.18 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 7 pada Lintasan Garis Lurus



Gambar 4.19 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 9 pada Lintasan Garis Lurus

Hasil pengujian dengan *parameter* N_p = 500 pada Gambar 4.16, 4.17, 4.18 dan 4.19 menunjukkan fenomena bahwa respon kontroler MPC dalam mengendalikan *plant* model membutuhkan waktu lebih lama dibandingkan dengan N_p = 300 dan N_p = 400.

Respon kontroler tercepat pada N_p = 500 ditunjukkan pada N_c = 3 dan 7 yaitu sebesar 722 detik dan 301 detik. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model ditampilkan pada Tabel 4.5 berikut

Tabel 4.5 Analisa *Response time* Kontroler MPC dengan *Horizon Prediction* N_p = 500 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Lurus

Horizon control (N_c)	*Response time* kontroler MPC
3	722
5	369
7	301
9	251

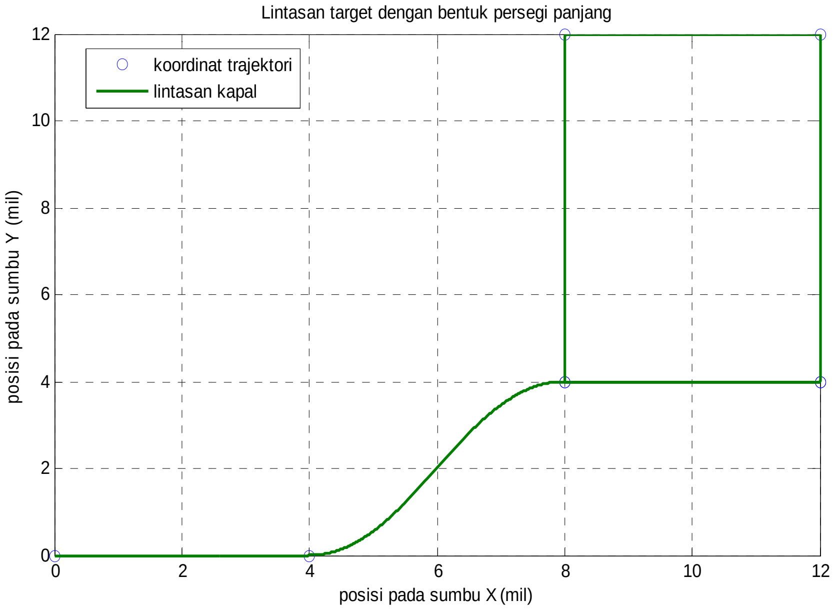
b. Lintasan persegi panjang

Parameter uji untuk lintasan persegi panjang ditampilkan pada Tabel 4.6 berikut

Tabel 4.6 *Parameter* Uji Simulasi Lintasan Persegi Panjang

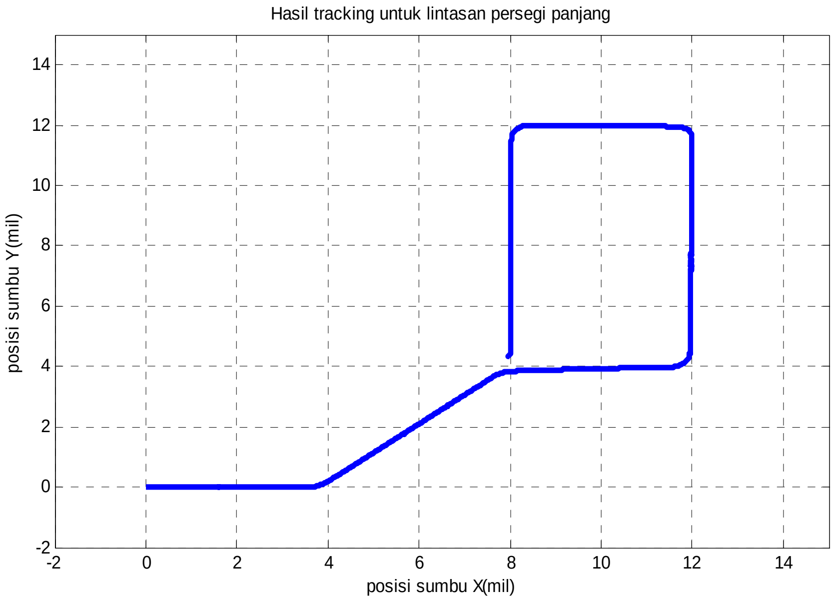
Parameter Uji	
Koordinat tujuan	(0,0), (4,0), (8,4), (12,4), (12,12), (8,12), (8,4), (12,4)
Kecepatan kapal	8 knot
Jarak lintasan	36,9 mil
Waktu tempuh	14600 s

Setelah menginput data pada Tabel 4.6 pada m-file *software* MATLAB diperoleh hasil simulasi untuk lintasan target dengan bentuk garis persegi panjang seperti pada Gambar 4.20



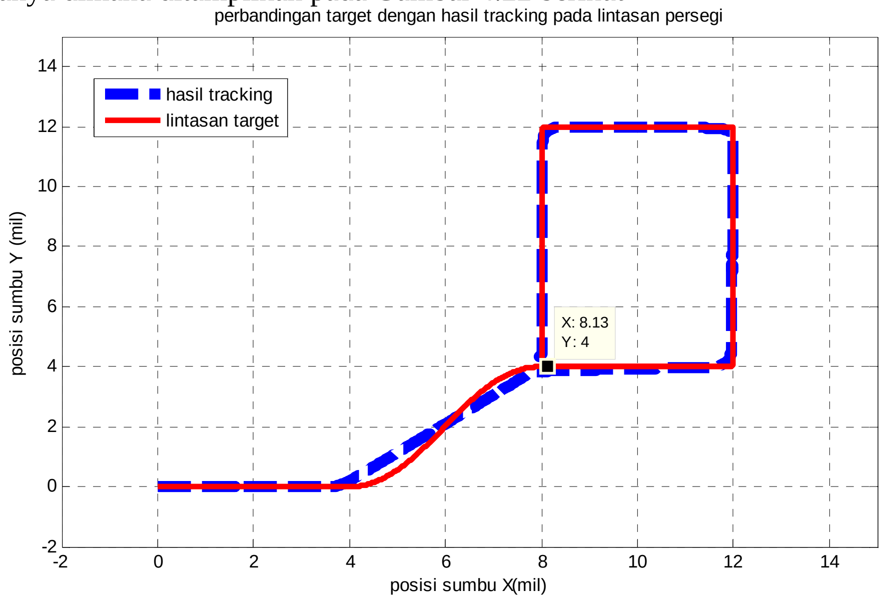
Gambar 4.20 Bentuk Lintasan Target pada Lintasan Persegi Panjang

Langkah selanjutnya adalah melakukan simulasi *tracking* dari target pada lintasan persegi panjang dimana hasil *tracking* tersebut ditampilkan pada Gambar 4.21



Gambar 4.21 Hasil Simulasi *Tracking* Target pada Lintasan Persegi Panjang

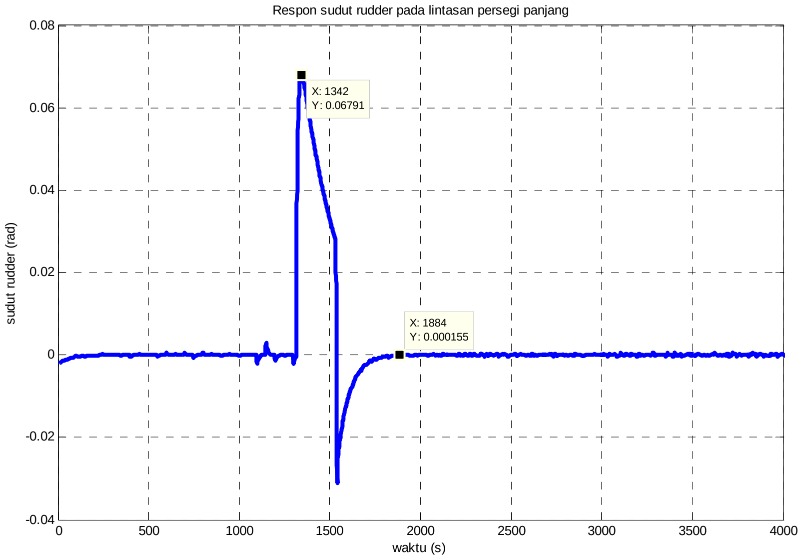
Setelah memperoleh hasil simulasi dari target dan hasil *tracking* dari lintasan persegi panjang, langkah selanjutnya adalah membandingkan hasil keduanya dimana ditampilkan pada Gambar 4.22 berikut



Gambar 4.22 Perbandingan Hasil Simulasi Target dan *Tracking* pada Lintasan Persegi Panjang

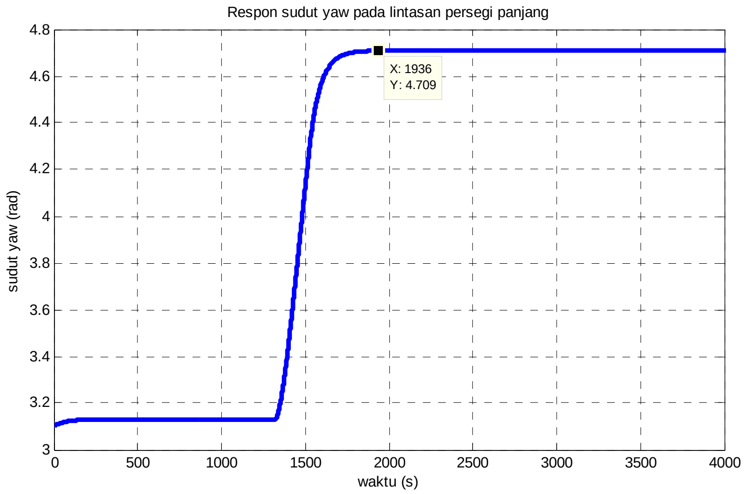
Dari gambar 4.22 diketahui bahwa *plant* mulai memasuki lintasan target pada koordinat (8,13) mil dengan analisa *error* hasil *tracking* ditampilkan pada Tabel 4.7.

Sedangkan untuk respon sudut *rudder* (δ) pada simulasi *tracking* lintasan persegi panjang ditampilkan pada Gambar 4.23 berikut.



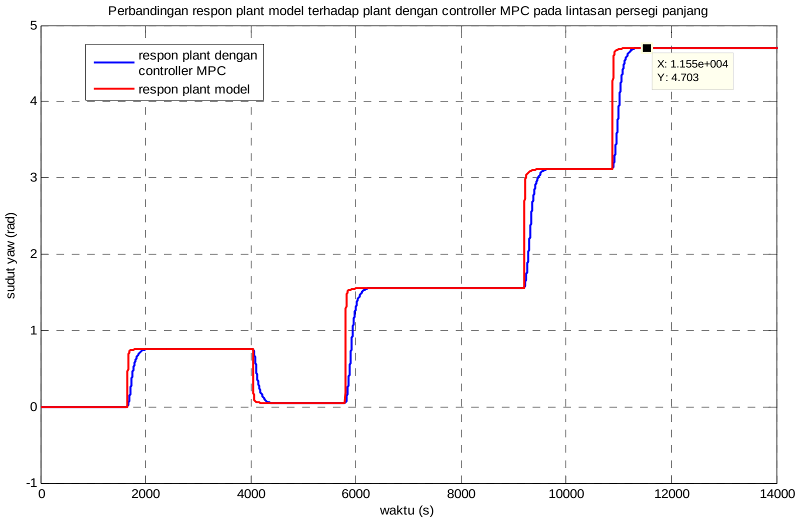
Gambar 4.23 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Persegi Panjang

Gambar 4.23 menunjukkan sudut *rudder* berfluktuasi dengan *overshoot* 0.06 rad hal ini disebabkan oleh adanya *disturbance* berupa arus laut dengan kecepatan 2 knot. Sudut *rudder* mengalami kondisi *steady state* pada waktu 1884 detik. Respon sudut *yaw* (ψ) *Plant* kapal kargo untuk lintasan persegi panjang ditampilkan pada Gambar 4.24 berikut dimana sudut *yaw* mengalami *steady state* pada waktu 1936 detik



Gambar 4.24 Hasil Simulasi Respon sudut *Rudder* untuk Lintasan Persegi Panjang

Perbandingan respon *plant* model terhadap *plant* dengan kontroler MPC untuk lintasan persegi panjang ditampilkan pada Gambar 4.25 dimana respon *plant* dengan kontroler MPC mampu mengikuti respon dari *plant* model dan *steady state* pada waktu 11550 detik

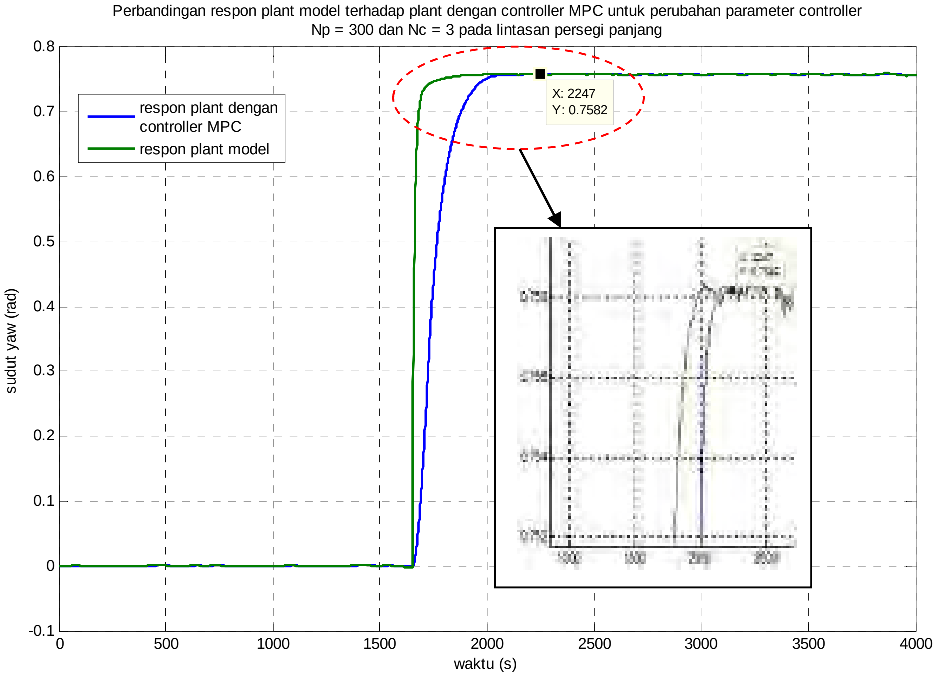


Gambar 4.25 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC pada Lintasan Persegi Panjang

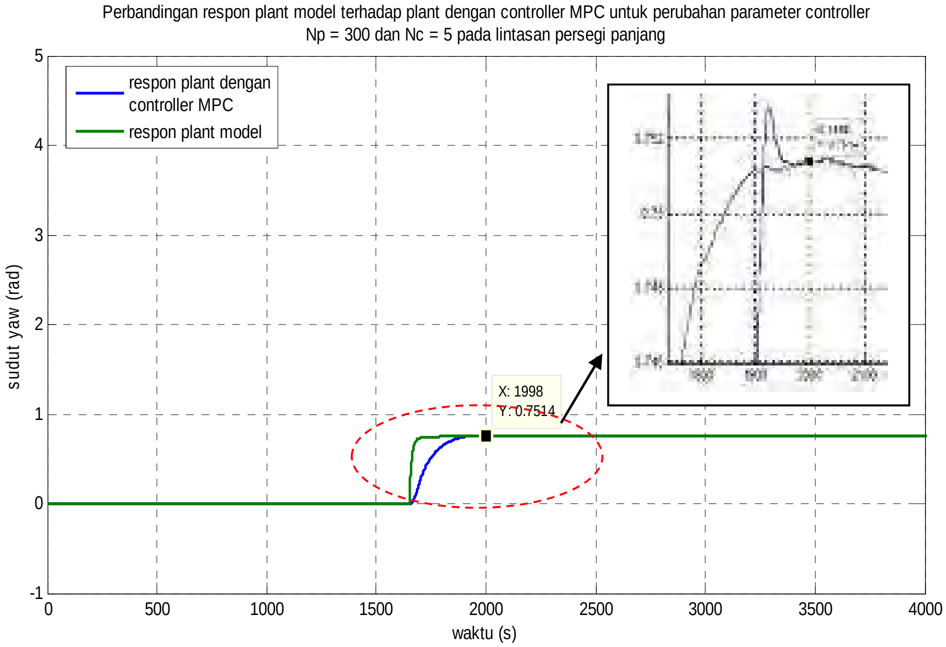
Tabel 4.7 Analisa *Error Tracking* pada Uji Simulasi Lintasan Persegi Panjang

No	X_{target}	$X_{tracking}$	*Error* trajectory (mil)	Y_{target}	$Y_{tracking}$	*Error* Trajectory (mil)
1	0	-0,00112	-0,00112	0	-0,00112	-0,00112
2	0	0,000768	0,000768	0	-0,00112	-0,00112
3	0	0,002696	0,002696	0	-0,00112	-0,00112
4	0	0,004606	0,004606	0	-0,00112	-0,00112
5	0	0,006516	0,006516	0	-0,00112	-0,00112
6	0	0,008426	0,008426	0	-0,00112	-0,00112
7	0	0,010336	0,010336	0	-0,00112	-0,00112
8	0	0,012246	0,012246	0	-0,00112	-0,00112
9	0	0,014155	0,014155	0	-0,00112	-0,00112
10	0	0,016065	0,016065	0	-0,00112	-0,00112
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1000	0	2,150308	2,150308	0	0,001483	0,001483
1001	0	2,153706	2,153706	0	0,000205	0,000205
1002	0	2,155961	2,155961	0	0,000203	0,000203
1003	0	2,158216	2,158216	0	0,000201	0,000201
1004	0	2,160472	2,160472	0	0,0002	0,0002
1005	0	2,162727	2,162727	0	0,000198	0,000198
⋮	⋮	⋮	⋮	⋮	⋮	⋮
14595	11,9465	7,97456	-3,97197	4	4,343115	0,343115
14596	11,9524	7,97346	-3,97902	4	4,341228	0,341228
14597	11,9584	7,97234	-3,98609	4	4,339355	0,339355
14598	11,9643	7,97120	-3,99318	4	4,337495	0,337495
14599	11,9703	7,97004	-4,00029	4	4,33565	0,33565
14600	11,9762	7,96885	-4,00742	4	4,333821	0,333821
14601	11,9822	7,96763	-4,01457	4	4,332006	0,332006
14602	11,9881	7,96640	-4,02174	4	4,330207	0,330207
14603	11,9940	7,96514	-4,02893	4	4,328425	0,328425
14604	12	7,96441	-4,03558	4	4,32832	0,32832
$\sum_{total}$	72380,4	121322,8	48942,38	49348	87927,03	38579,03
***Error* rerata sumbu X (mil)**			**3,3**	***Error* rerata sumbu Y(mil)**		**2,6**
% *error* rerata			**9**	**% *error* rerata**		**7,1**

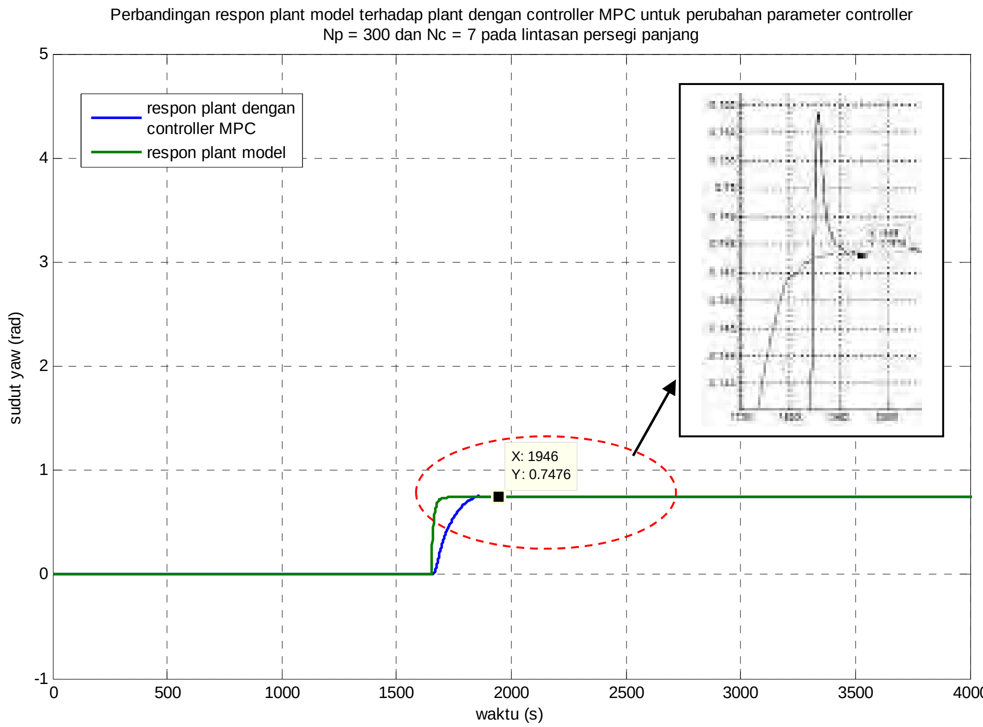
Hasil simulasi sistem dengan perubahan *prediction horizon* N_p dengan nilai N_p = 300 dan N_c = 3,5,7 dan 9 pada lintasan persegi panjang disajikan di Gambar 4.26 - 4.29



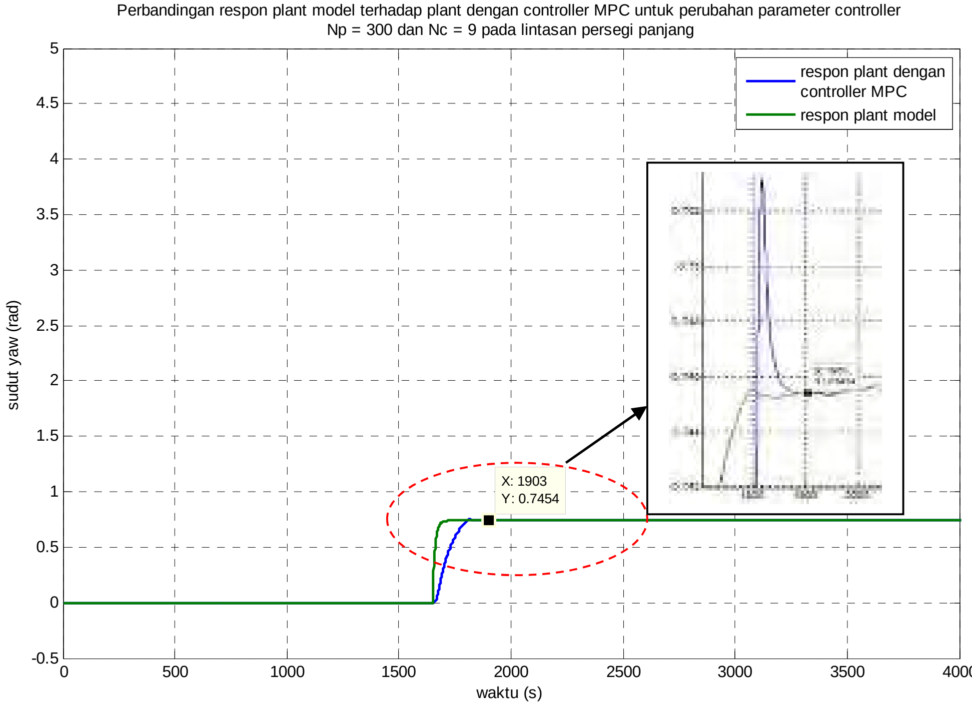
Gambar 4.26 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 3 pada Lintasan Persegi Panjang



Gambar 4.27 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 5 pada Lintasan Persegi Panjang



Gambar 4.28 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 7 pada Lintasan Persegi Panjang



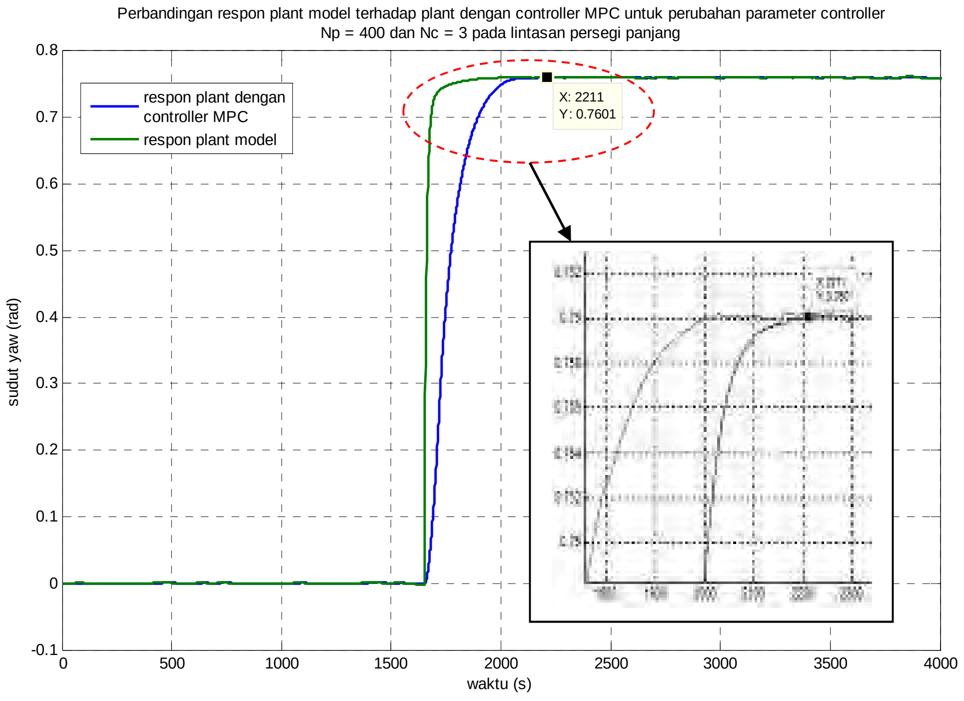
Gambar 4.29 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 9 pada Lintasan Persegi Panjang

Berdasarkan hasil pengujian pada Gambar 4.26, 4.27, 4.28 dan 4.29 respon kontroler untuk mengendalikan *plant* model dengan *parameter* N_p = 300 membutuhkan waktu yang berangsur – angsur semakin cepat yaitu 1903 detik pada N_c = 9. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model untuk lintasan persegi panjang ditampilkan pada Tabel 4.8 berikut

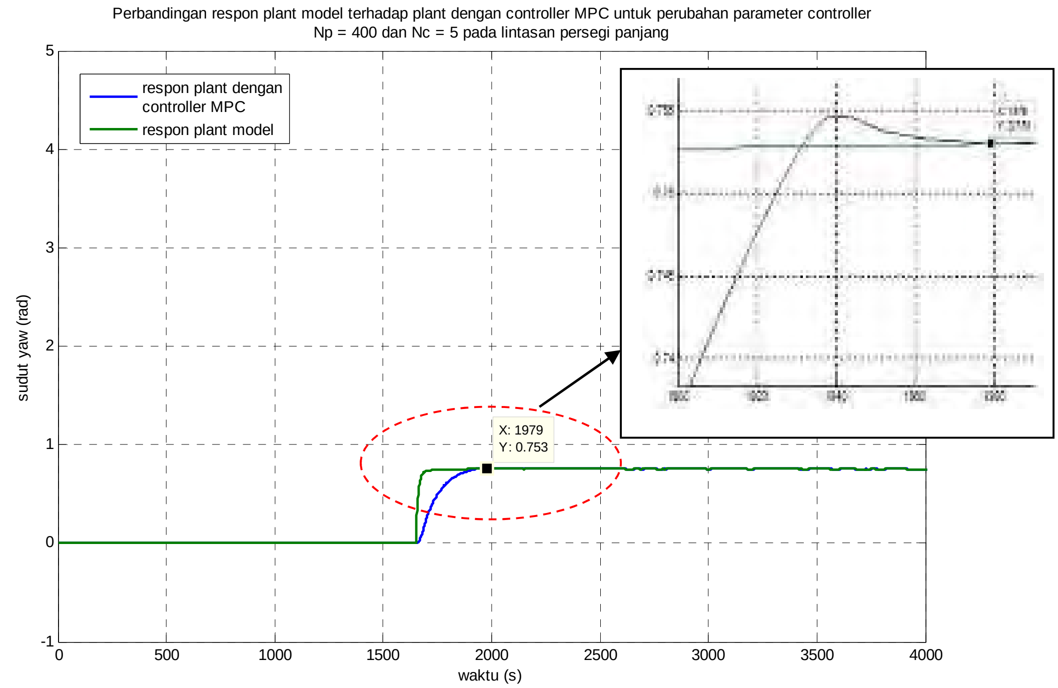
Tabel 4.8 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 300 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Persegi Panjang

Horizon control (N_c)	*Response time* Kontroler MPC
3	2247
5	1998
7	1946
9	1903

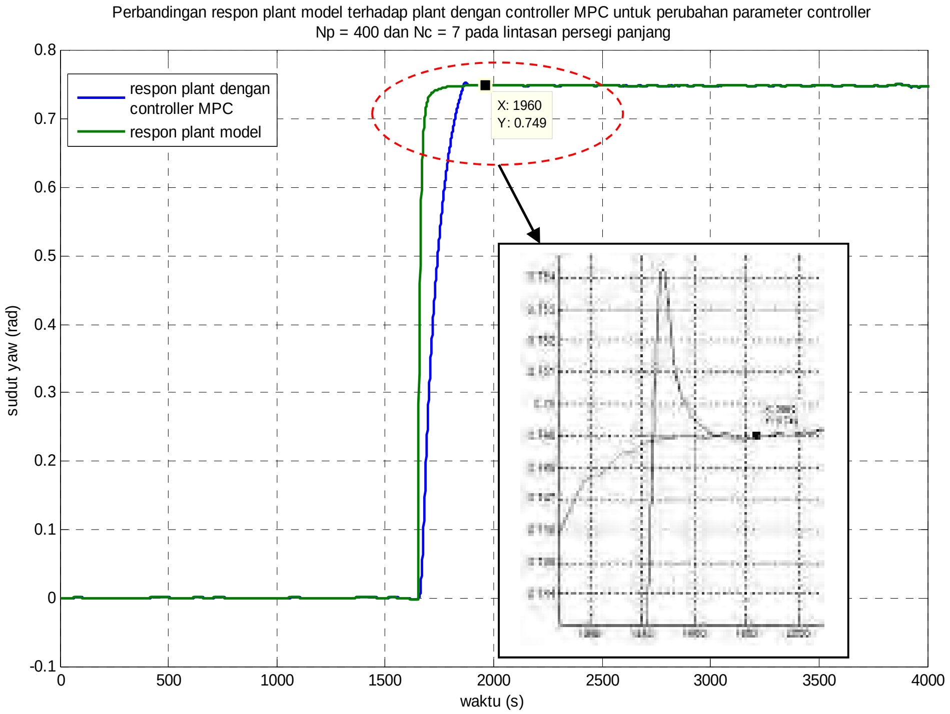
untuk perubahan *parameter* N_p = 400 dan N_c = 3,5,7 dan 9 disajikan pada Gambar 4.30 – 4.33



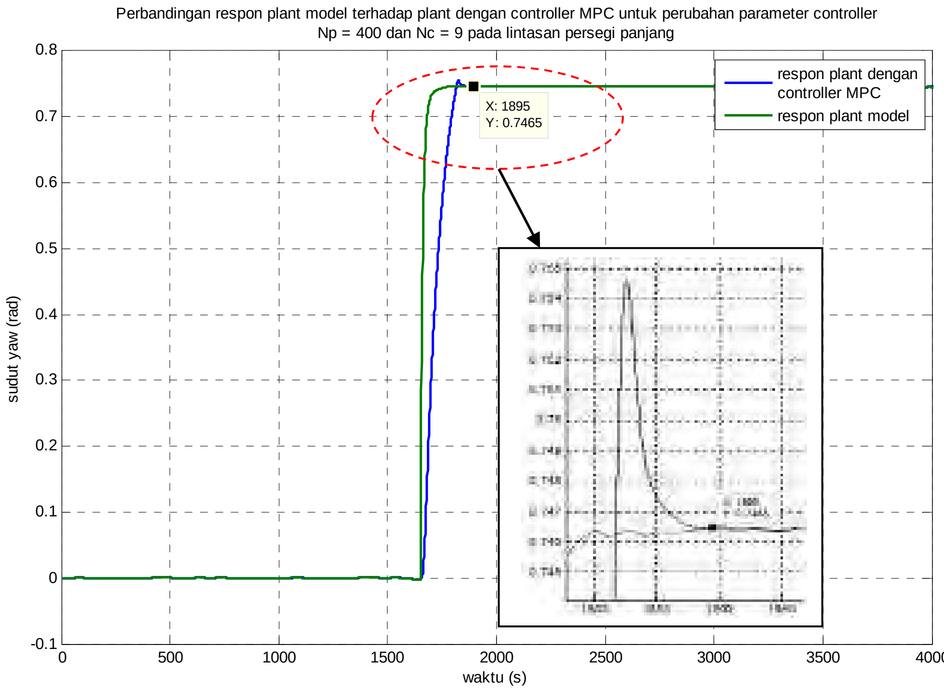
Gambar 4.30 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 3 pada Lintasan Persegi Panjang



Gambar 4.31 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 5 pada Lintasan Persegi Panjang



Gambar 4.2 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 7 pada Lintasan Persegi Panjang



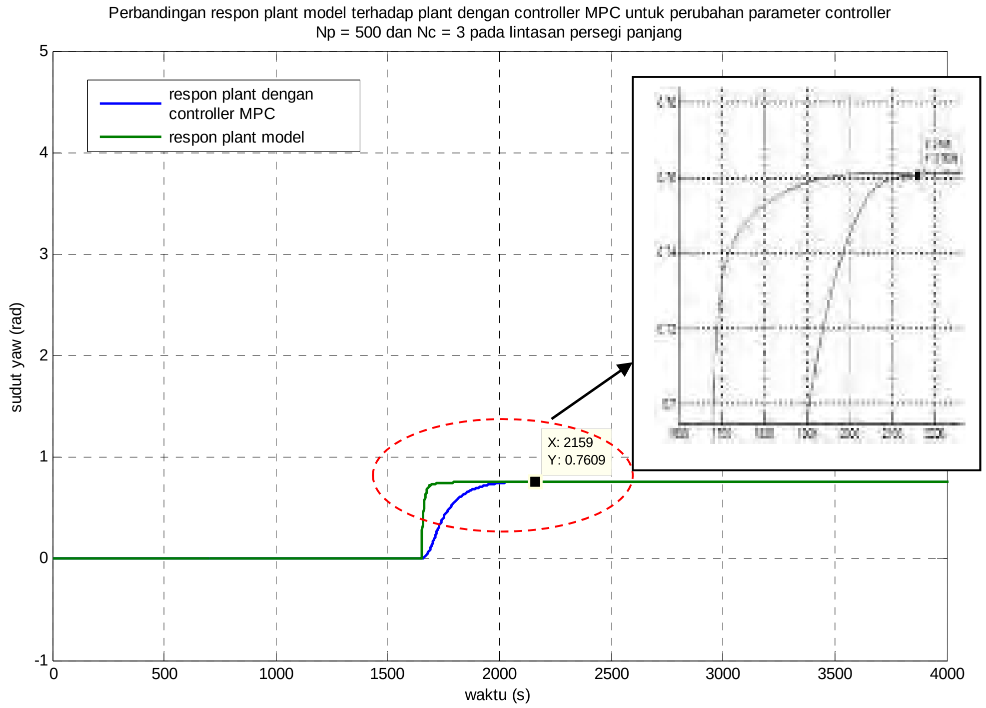
Gambar 4.33 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 9 pada Lintasan Persegi Panjang

Hasil pengujian pada Gambar 4.30, 4.31, 4.32 dan 4.33 menunjukkan respon kontroler MPC untuk mengendalikan *plant* dengan *parameter* N_p = 400 membutuhkan waktu lebih cepat dibandingkan dengan N_p = 300 yaitu sebesar 1895 detik pada N_c = 9. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model ditampilkan pada Tabel 4.9 berikut

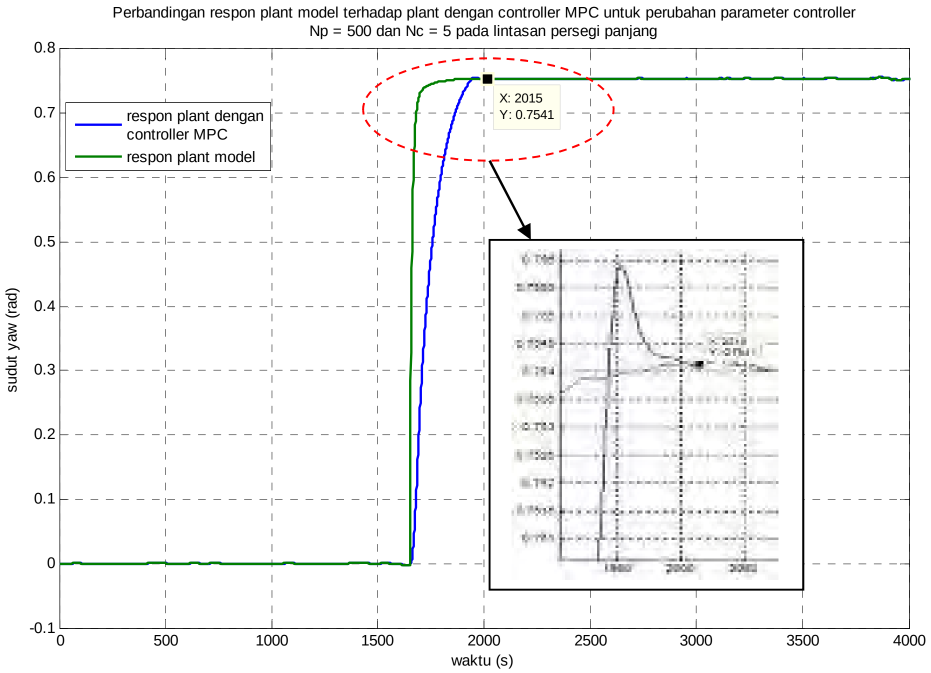
Tabel 4.9 Analisa *Response Time* Kontroler MPC dengan Horizon Prediction N_p = 400 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Persegi Panjang

Horizon control (N_c)	*Response time* Kontroler MPC
3	2211
5	1979
7	1960
9	1895

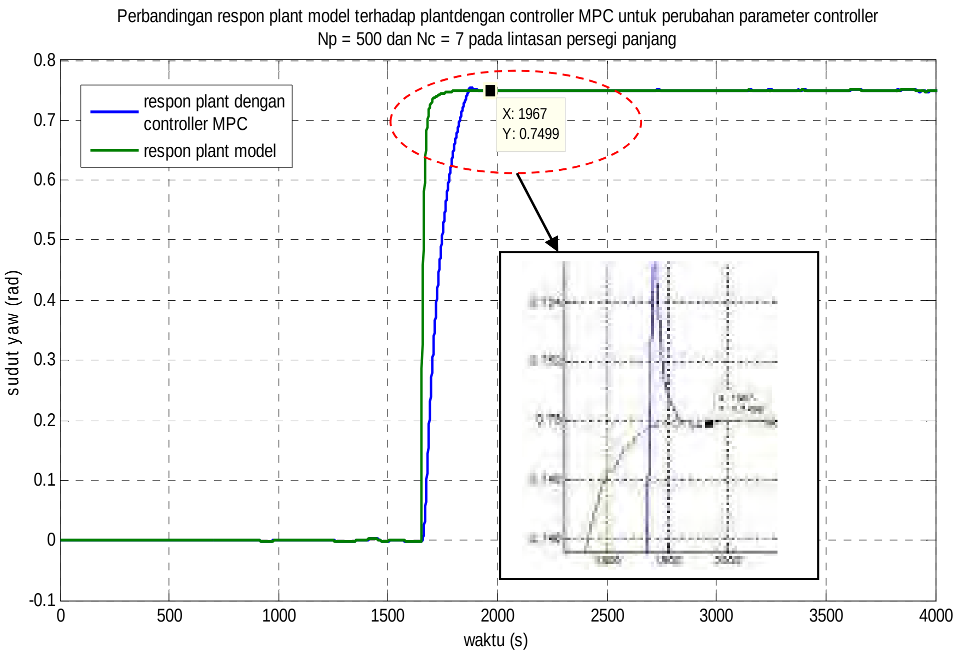
sedangkan N_p = 500 berturut-turut disajikan pada Gambar 4.34 – 4.37



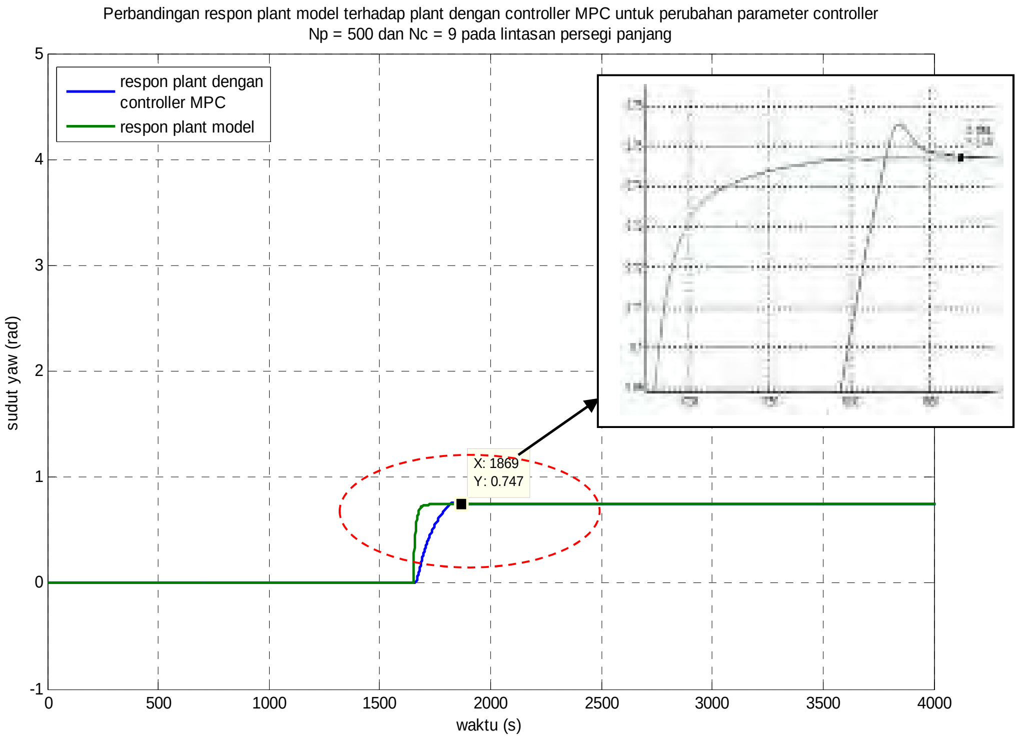
Gambar 4.34 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 3 pada Lintasan Persegi Panjang



Gambar 4.35 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 5 pada Lintasan Persegi Panjang



Gambar 4.36 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 7 pada Lintasan Persegi Panjang



Gambar 4.37 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 9 pada Lintasan Persegi Panjang

Hasil pengujian dengan *parameter* N_p = 500 pada Gambar 4.34, 4.35, 4.36 dan 4.37 menunjukkan fenomena bahwa respon kontroler MPC dalam mengendalikan *plant* model membutuhkan waktu lebih cepat dibandingkan dengan N_p = 300 dan N_p = 400. Yaitu pada N_c = 9 sebesar 1869 detik.

Respon kontroler terlama pada N_p = 500 ditunjukkan pada N_c = 5 dan 7 yaitu sebesar 2015 detik dan 1967 detik. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model ditampilkan pada Tabel 4.10 berikut

Tabel 4.10 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 500 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Persegi Panjang

Horizon control (Nc)	*Response time* Kontroler MPC
3	2159
5	2015
7	1967
9	1869

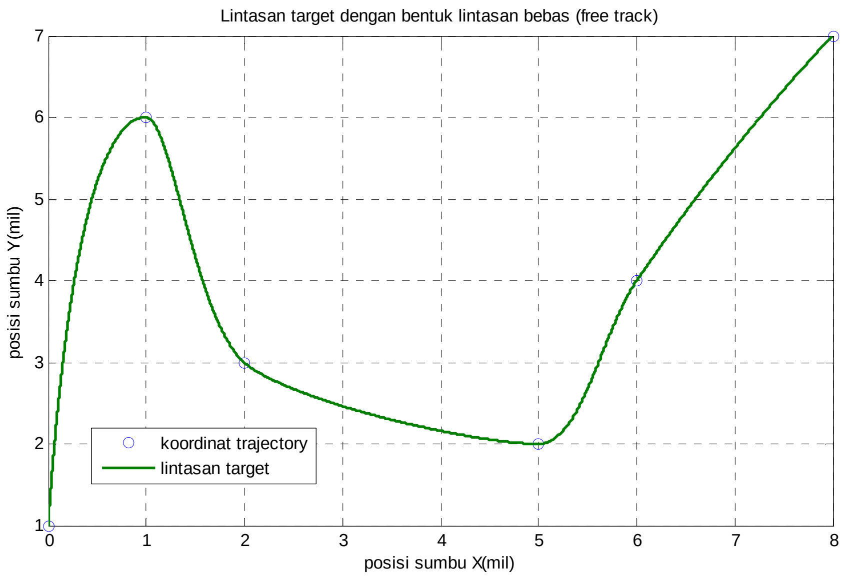
c. Lintasan Bebas (*Free Track*)

Parameter uji untuk lintasan bebas (*free track*) ditampilkan pada Tabel 4.11 berikut

Tabel 4.11 *Parameter* Uji Simulasi Lintasan Bebas (*Free Track*)

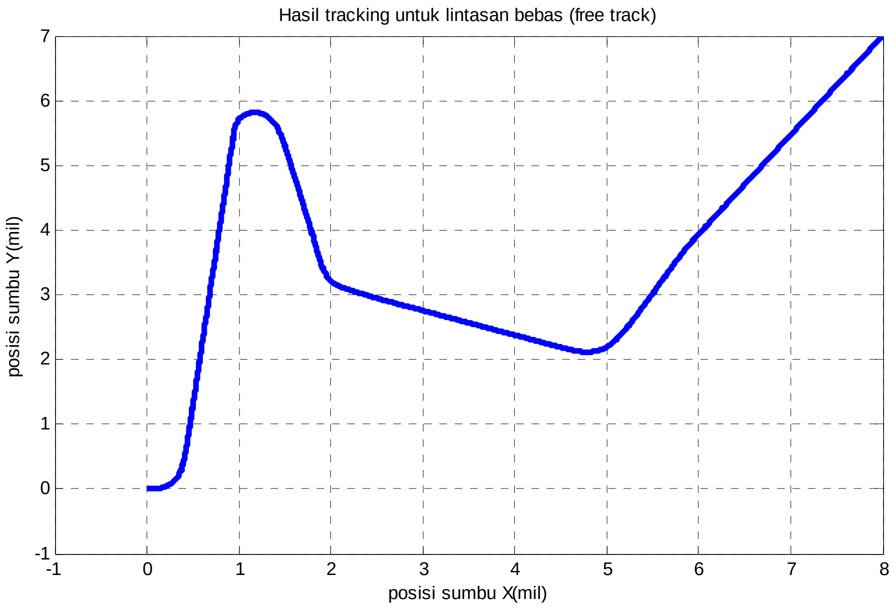
Parameter uji	
Koordinat tujuan	(0,1), (1,6), (2,3), (5,2), (6,4), (8,7)
Kecepatan kapal	8 knot
Jarak lintasan	20.5 mil
Waktu tempuh	8000 s

Dengan menginput data Tabel 4.11 pada m-file *software* MATLAB diproleh hasil simulasi target untuk bentuk lintasan bebas (*Free Track*) yang ditampilkan pada Gambar 4.38



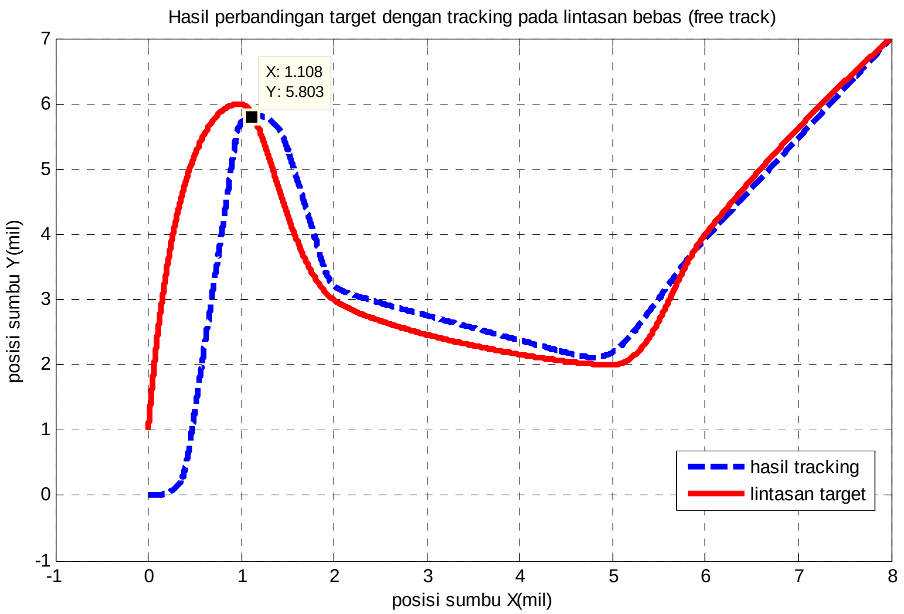
Gambar 4.38 Bentuk Lintasan Target untuk Lintasan Bebas (*Free Track*)

Langkah berikutnya adalah melakukan simulasi *tracking* dari target pada lintasan bebas (*free track*) dimana hasil *tracking* tersebut ditampilkan pada Gambar 4.39 berikut



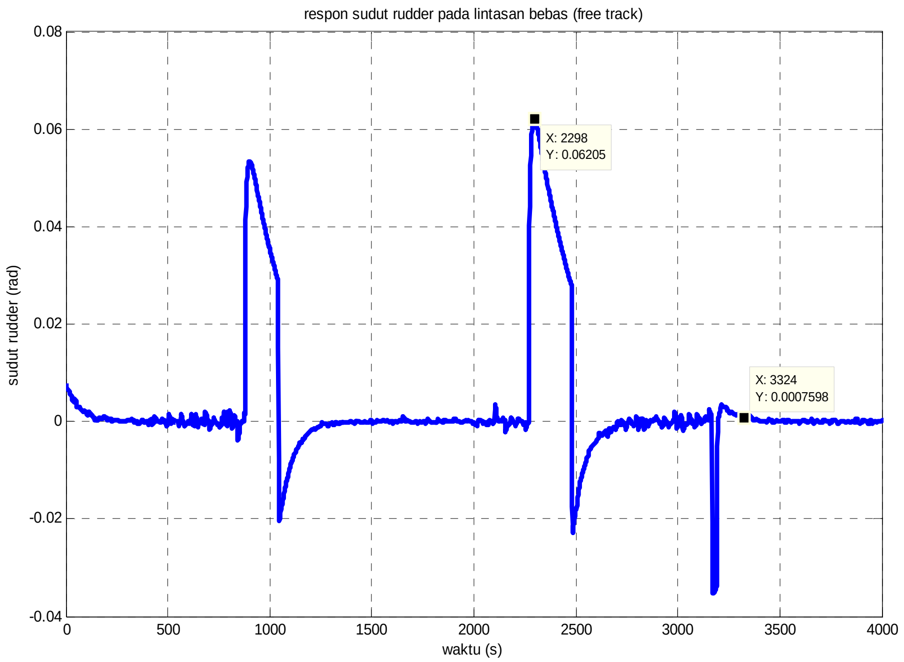
Gambar 4.39 Hasil Simulasi *Tracking* pada Lintasan Bebas (*free Track*)

Setelah memperoleh hasil simulasi dari target dan hasil *tracking* dari lintasan bebas (*free track*), langkah selanjutnya adalah membandingkan hasil keduanya dimana ditampilkan pada Gambar 4.40 berikut



Gambar 4.40 Perbandingan Hasil Simulasi Target dan Hasil *Tracking* pada Lintasan Bebas

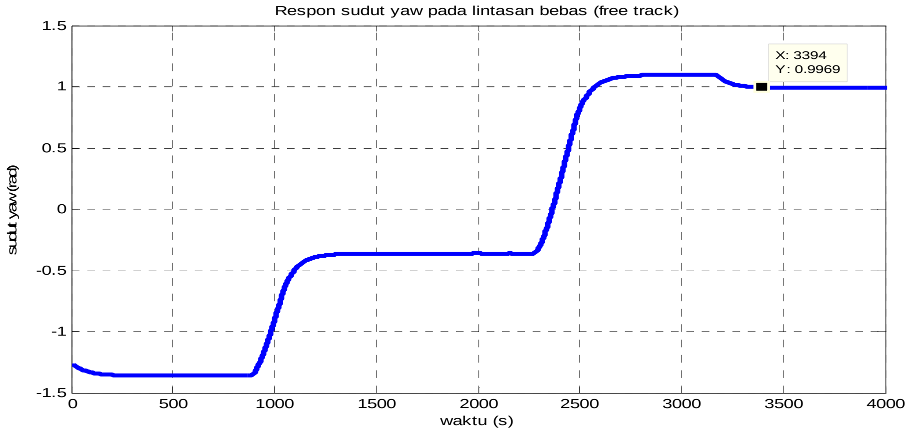
Dari gambar 4.16 diketahui bahwa *plant* mulai memasuki lintasan target pada koordinat (8.13,4) mil dengan analisa *error* hasil *tracking* ditampilkan pada Tabel 4.12. Sedangkan untuk respon sudut *rudder* (δ) pada simulasi *tracking* lintasan bebas (*free track*) ditampilkan pada Gambar 4.41 berikut.



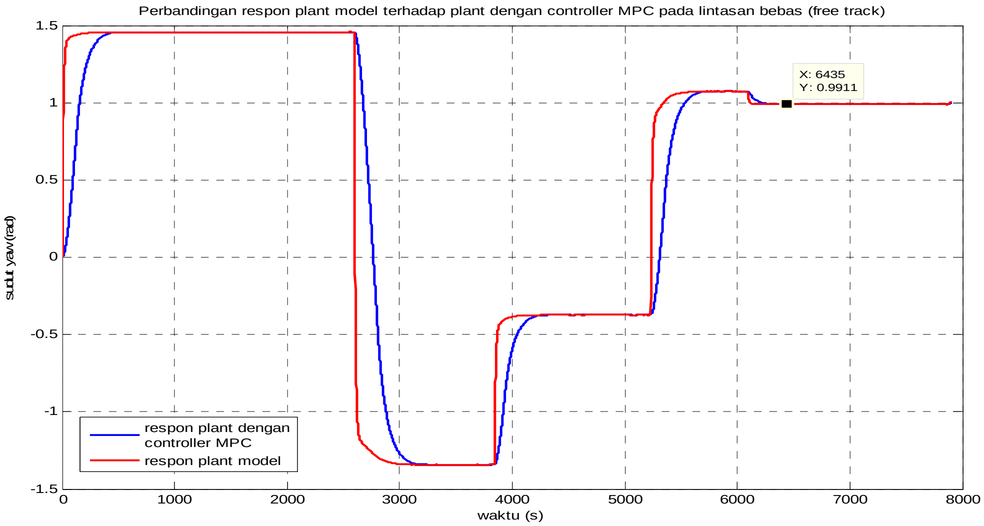
Gambar 4.41 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Bebas (*Free Track*)

Gambar 4.41 menunjukkan sudut *rudder* berfluktuasi dengan *overshoot* 0.06 rad hal ini disebabkan oleh adanya *disturbance* berupa arus laut dengan kecepatan 2 knot. Sudut *rudder* mengalami kondisi *steady state* pada waktu 3324 detik. Respon sudut *yaw* (ψ) kapal untuk *tracking* pada lintasan bebas (*free track*) ditampilkan pada Gambar 4.42 dimana *plant* mengalami *steady state* pada waktu 3394 detik.

Perbandingan respon *plant* model terhadap *plant* dengan kontroler MPC untuk lintasan bebas (*free track*) ditampilkan pada Gambar 4.43 dimana respon *plant* dengan kontroler MPC mampu mengikuti respon dari *plant* model dan *steady state* pada waktu 6435 detik



Gambar 4.42 Hasil Simulasi Respon Sudut *Yaw* pada Lintasan Bebas (*Free Track*)



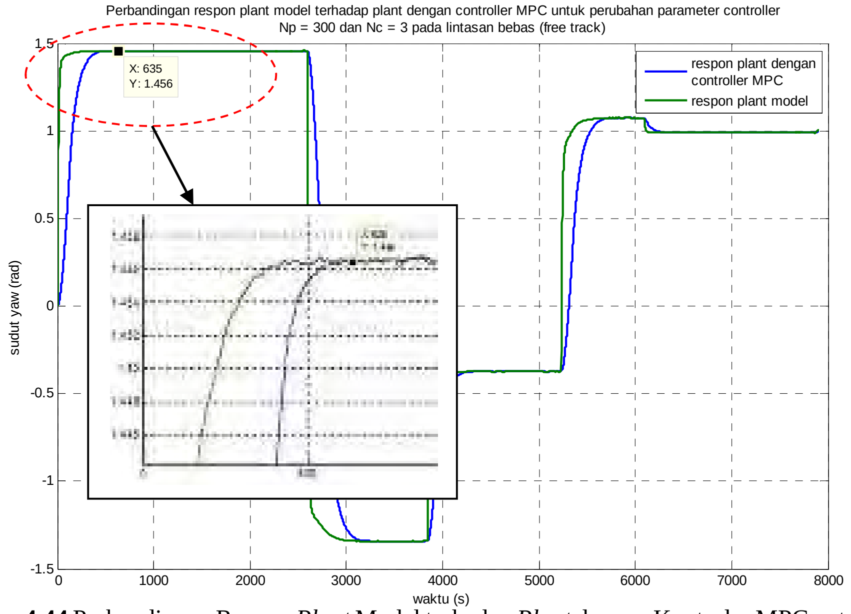
Gambar 4.43 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC pada Lintasan Bebas (*Free Track*)

Tabel 4.12 Analisa *Error Tracking* pada Uji Simulasi Lintasan Bebas (*Free Track*)

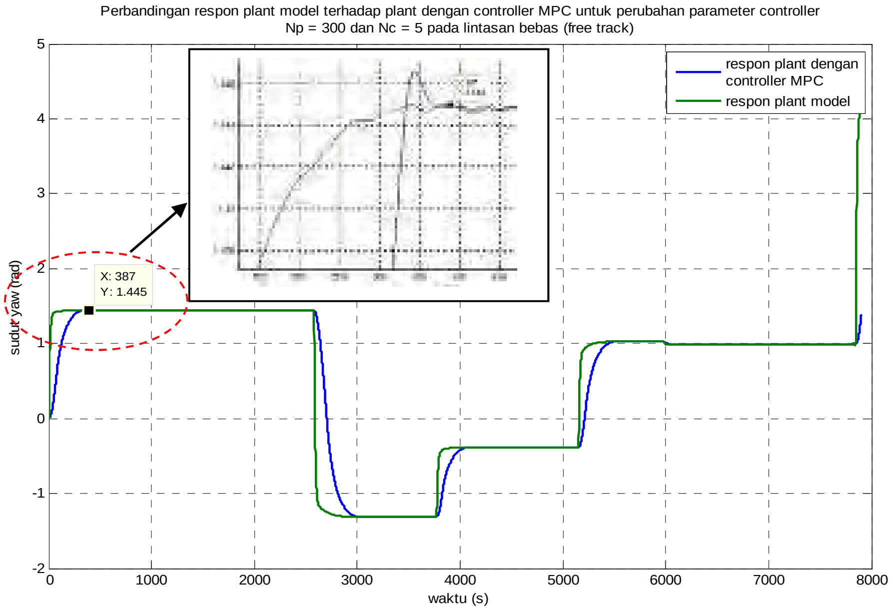
No	X_{target}	$X_{tracking}$	*Error* trajectory (mil)	Y_{target}	$Y_{tracking}$	*Error* Trajectory (mil)
1	0	-0,00112	-0,00112	0	-0,00112	-0,00112
2	0	0,000786	0,000786	0	-0,00112	-0,00112
3	0	0,002696	0,002696	0	-0,00112	-0,00112
4	0	0,004606	0,004606	0	-0,00112	-0,00112
5	0	0,006516	0,006516	0	-0,00112	-0,00112
6	0	0,008426	0,008426	0	-0,00112	-0,00112
7	0	0,010336	0,010336	0	-0,00112	-0,00112
8	0	0,012246	0,012246	0	-0,00112	-0,00112
9	0	0,014155	0,014155	0	-0,00111	-0,00111
10	0	0,016065	0,016065	0	-0,00111	-0,00111
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1000	0	0,564707	0,564707	0	1,906472	1,906472
1001	0	0,566561	0,566561	0	1,906911	1,906911
1002	0	0,566799	0,566799	0	1,909154	1,909154
1003	0	0,567037	0,567037	0	1,911397	1,911397
1004	0	0,567274	0,567274	0	1,913639	1,913639
1005	0	0,567512	0,567512	0	1,915882	1,915882
1006	0	0,56775	0,567750	0	1,918124	1,918124
1007	0	0,567988	0,567988	0	1,920367	1,920367
1008	0	0,568226	0,568226	0	1,922610	1,922610
1009	0	0,568464	0,568464	0	1,924852	1,924852
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7994	7,971063	7,984404	0,01334	6,96294	6,973117	0,010176
7995	7,973816	7,985826	0,012011	6,966472	6,97533	0,008858
7996	7,976569	7,987249	0,010681	6,970004	6,977544	0,00754
7997	7,979322	7,988673	0,00935	6,973534	6,979757	0,006223
7998	7,982077	7,990097	0,00802	6,977065	6,98197	0,004905
7999	7,984832	7,991521	0,006688	6,980594	6,984183	0,003588
8000	7,987588	7,992945	0,005356	6,984124	6,986395	0,002272
8001	7,990345	7,994368	0,004023	6,987635	6,988608	0,000956
8002	7,993103	7,995791	0,002689	6,991181	6,990822	-0,00036
8003	7,995861	7,997214	0,001353	6,994709	6,993036	-0,00167
8004	7,998620	7,997505	-0,00112	6,998236	6,998743	0,000507
$\sum_{total}$	12435,4	25241,18	12805,78	15969,06	28803,91	12834,85
***Error* rerata sumbu X (mil)**			**1,6**	***Error* rerata sumbu Y(mil)**		**1,6**
***% error* rerata**			**7,8**	***% error* rerata**		**7,8**

Dari Tabel 4.8 diperoleh hasil *error* rerata pada lintasan bebas (*free track*) untuk sumbu X sebesar 1,6 mil atau 7,8 % dan sumbu Y sebesar 1,6 mil atau 7,8 % dengan jarak tempuh lintasan 20.5 mil.

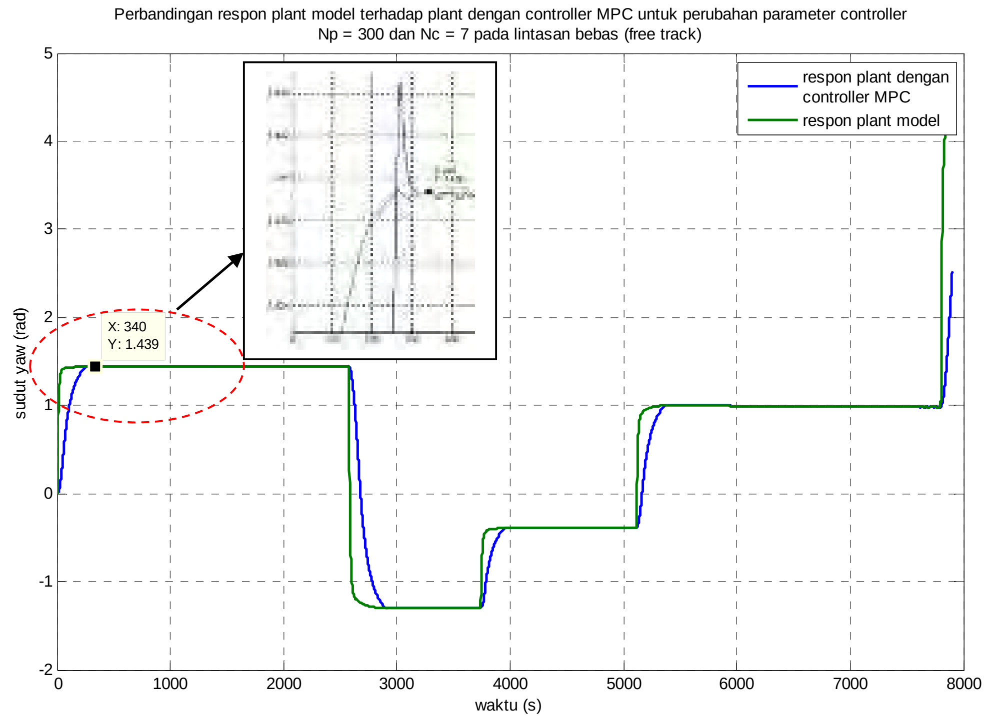
Hasil simulasi sistem dengan perubahan *prediction horizon* N_p dengan nilai $N_p = 300$ dan $N_c = 3{,}5{,}7$ dan 9 pada lintasan bebas (*free track*) disajikan di Gambar 4.44 - 4.47



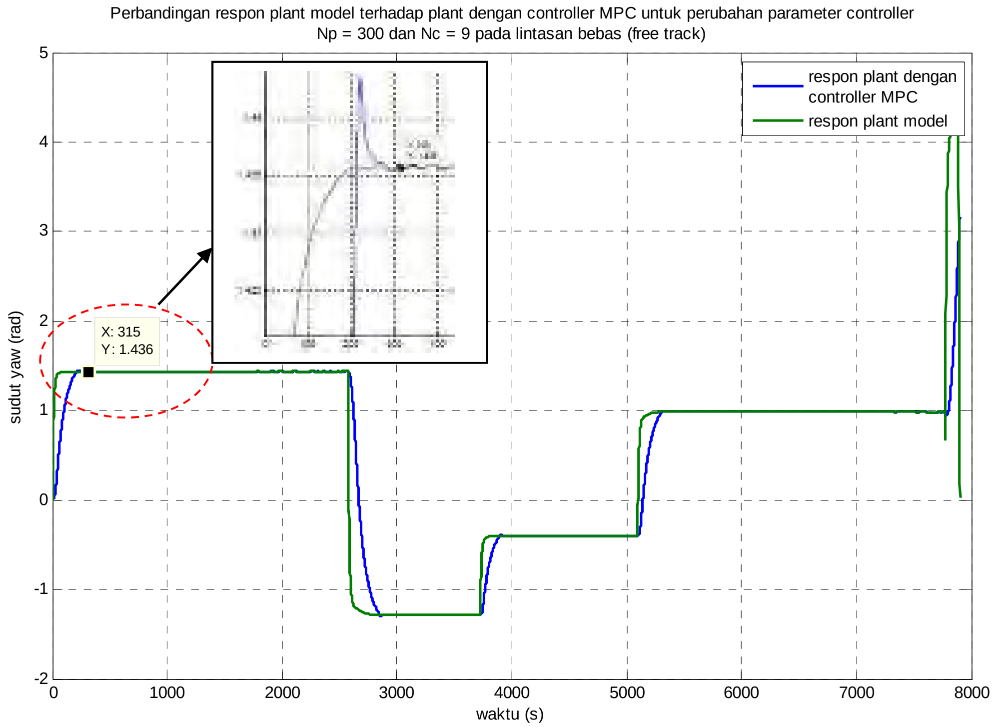
Gambar 4.44 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 3 pada Lintasan Bebas



Gambar 4.45 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 5 pada Lintasan Bebas



Gambar 4.46 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 7 pada Lintasan Bebas



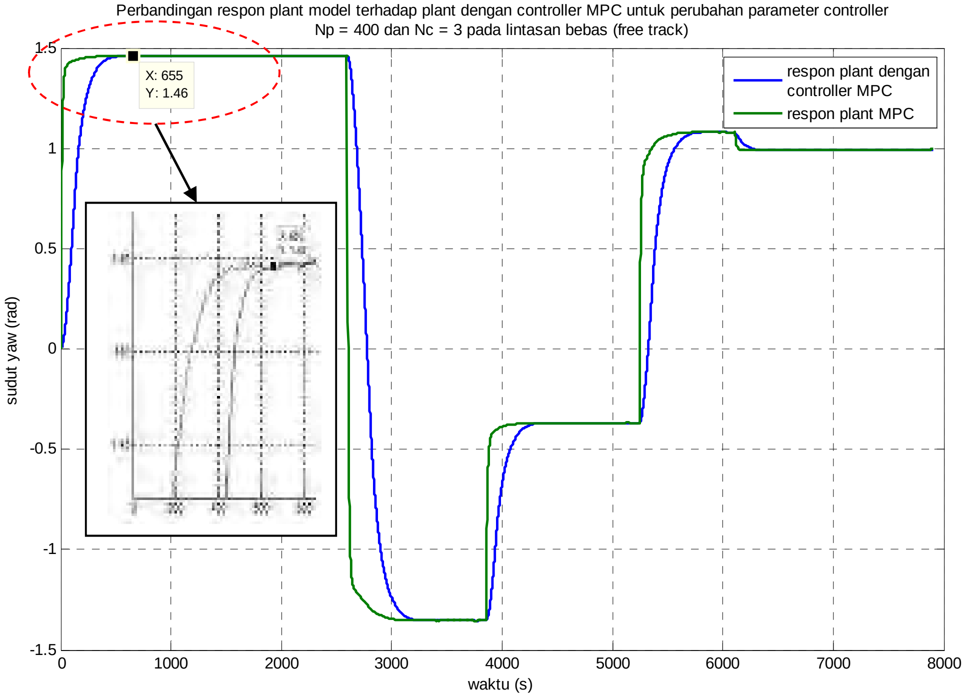
Gambar 4.47 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 300 dan N_c = 9 pada Lintasan Bebas

Berdasarkan hasil pengujian pada Gambar 4.26, 4.27, 4.28 dan 4.29 respon kontroler untuk mengendalikan *plant* model dengan *parameter* N_p = 300 membutuhkan waktu yang berangsur – angsur semakin cepat yaitu 315 detik pada N_c = 9. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model untuk lintasan bebas (*free track*) ditampilkan pada Tabel 4.13 berikut

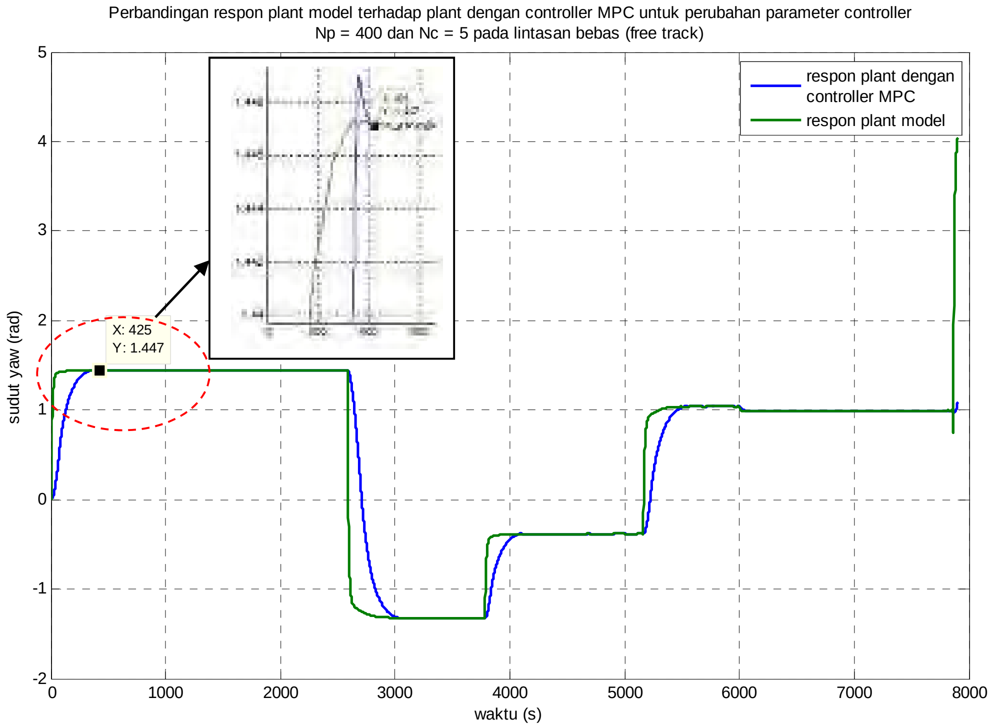
Tabel 4.13 Analisa *Response time* Kontroler MPC dengan *Horizon Prediction* N_p = 300 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Bebas (*Free Track*)

Horizon control (N_c)	*Response time* Kontroler MPC
3	635
5	387
7	340
9	315

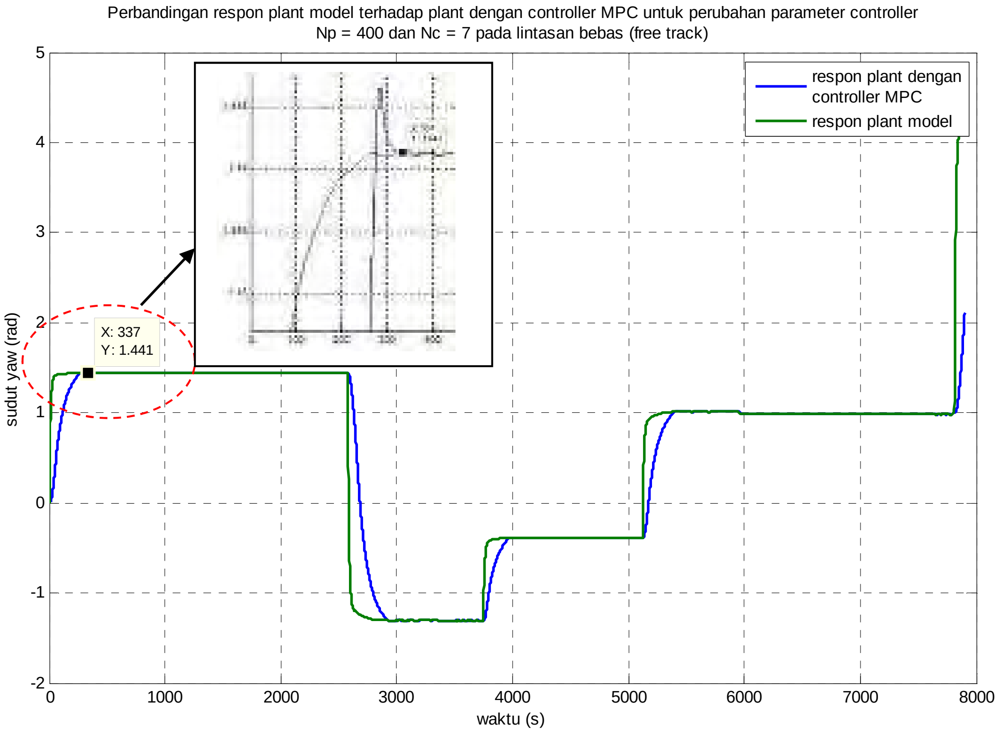
untuk perubahan *parameter* N_p = 400 dan N_c = 3,5,7 dan 9 pada lintasan bebas (*free track*) disajikan pada Gambar 4.48 – 4.51



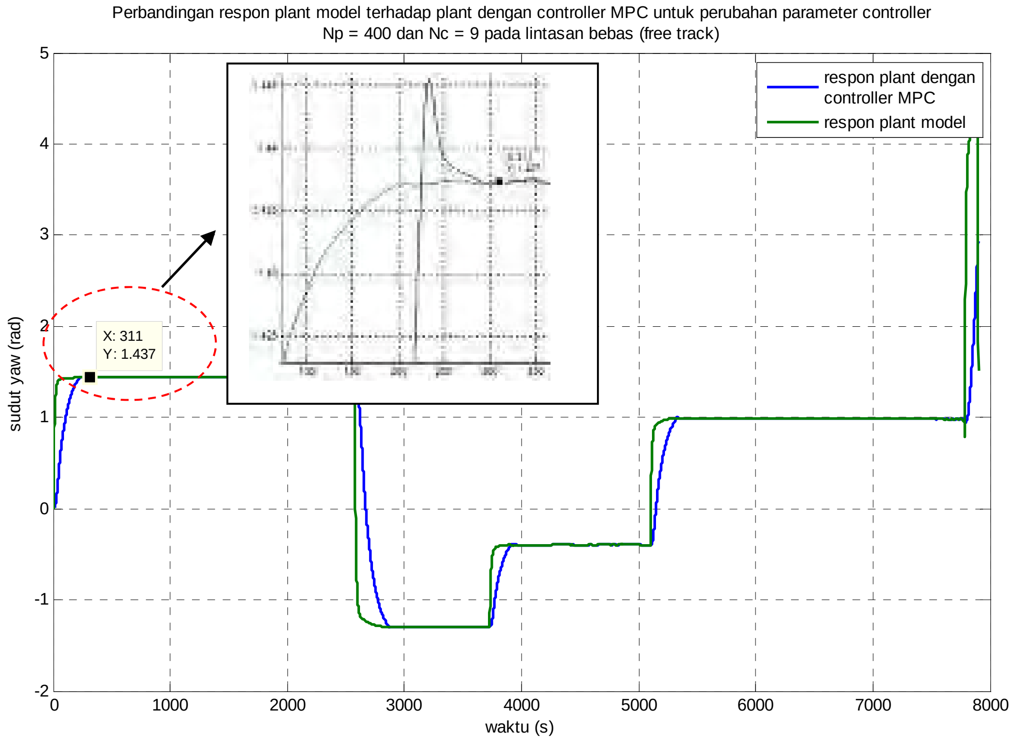
Gambar 4.48 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 3 pada Lintasan Bebas



Gambar 4.49 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 5 pada Lintasan Bebas



Gambar 4.50 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroller MPC untuk N_p = 400 dan N_c = 7 pada Lintasan Bebas



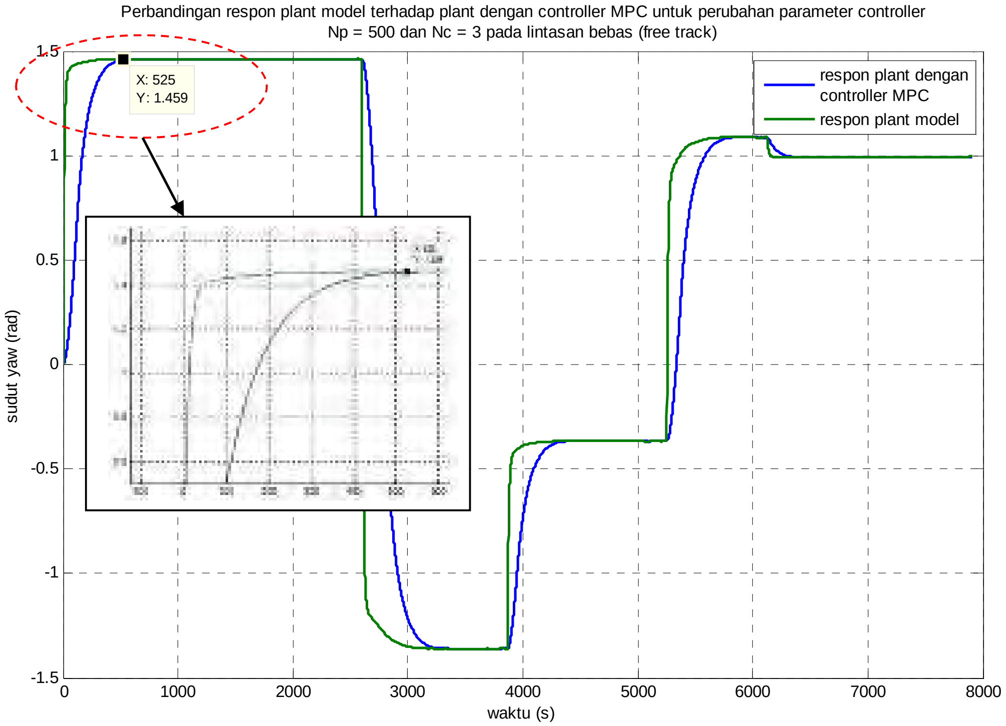
Gambar 4.51 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 400 dan N_c = 9 pada Lintasan Bebas

Hasil pengujian pada Gambar 4.48, 4.49, 4.50 dan 4.51 menunjukkan respon kontroler MPC untuk mengendalikan *plant* dengan *parameter* N_p = 400 membutuhkan waktu lebih cepat dibandingkan dengan N_p = 300 yaitu sebesar 311 detik pada N_c = 9. Sedangkan pada N_c = 3,5 dan 7 membutuhkan waktu yang relative lebih lama. Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model ditampilkan pada Tabel 4.14 berikut

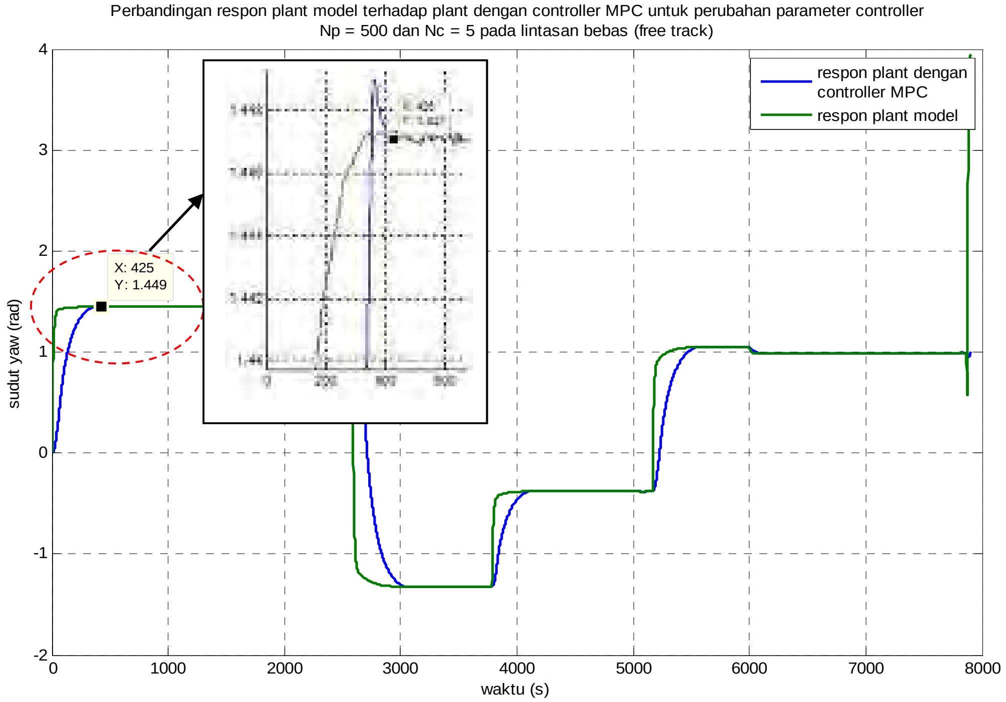
Tabel 4.14 Analisa Response Time Kontroler MPC dengan Horizon Prediction N_p = 400 terhadap Pengendalian *Plant* Model pada Uji Simulasi Lintasan Lurus

Horizon control (N_c)	*Response time* Kontroler MPC
3	655
5	425
7	337
9	311

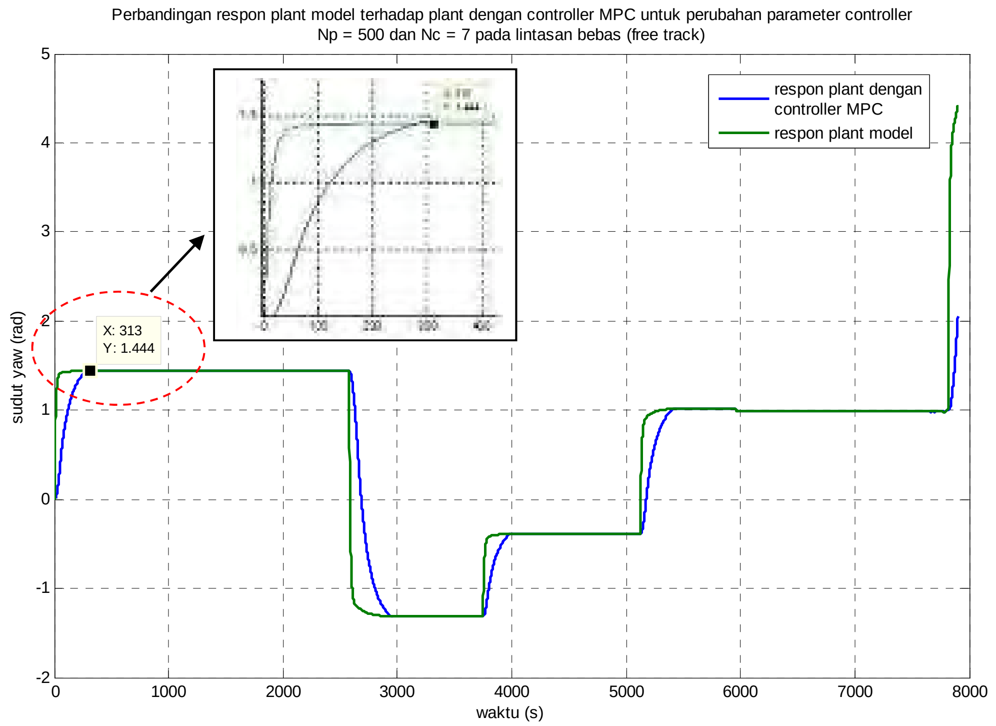
sedangkan Np = 500 berturut-turut disajikan pada Gambar 4.52 – 4.55



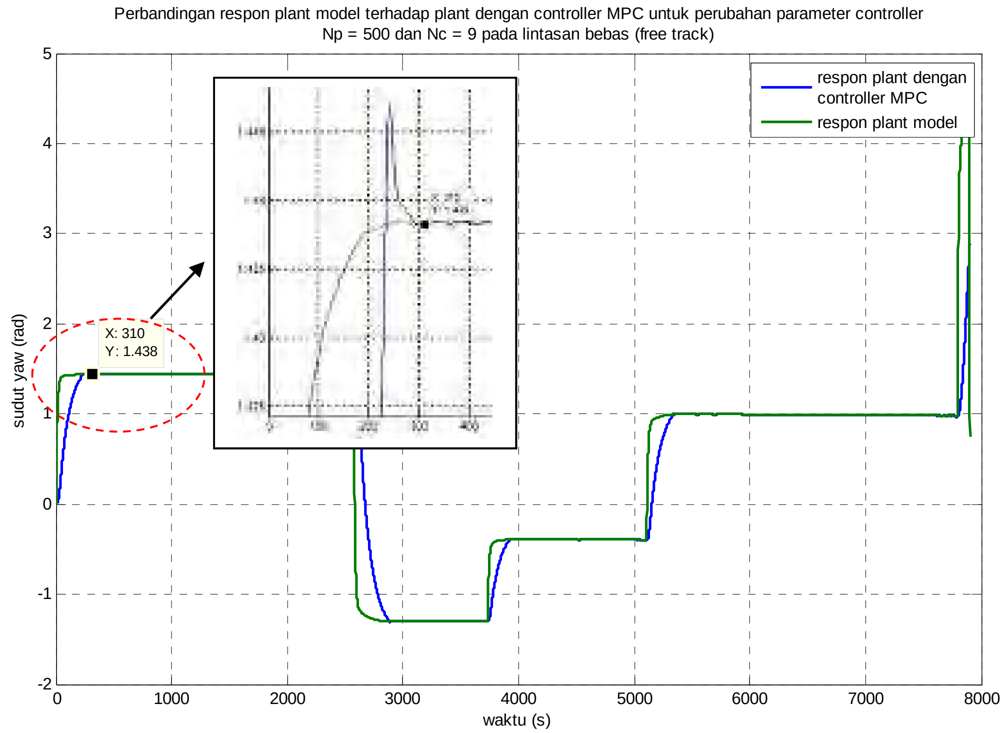
Gambar 4.52 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 3 pada Lintasan Bebas



Gambar 4.53 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 5 pada Lintasan Bebas



Gambar 4.54 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 7 pada Lintasan Bebas



Gambar 4.55 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC untuk N_p = 500 dan N_c = 9 pada Lintasan Bebas

Hasil pengujian dengan *parameter* N_p = 500 pada Gambar 4.52, 4.53, 4.54 dan 4.55 menunjukkan fenomena bahwa respon kontroler MPC dalam mengendalikan *plant* model membutuhkan waktu lebih cepat dibandingkan dengan N_p = 300 dan N_p = 400. Yaitu sebesar 310 detik pada N_c = 9

Adapun data lengkap dari respon kontroler terhadap pengendalian *plant* model ditampilkan pada Tabel 4.15 berikut

Tabel 4.15 Analisa *Response Time* Kontroler MPC dengan *Horizon Prediction* N_p = 500 terhadap pengendalian *Plant* Model pada Uji Simulasi Lintasan Bebas (*Free Track*)

Horizon control (N_c)	*Response time* Kontroler MPC
3	525
5	425
7	313
9	310

d. Lintasan lingkaran

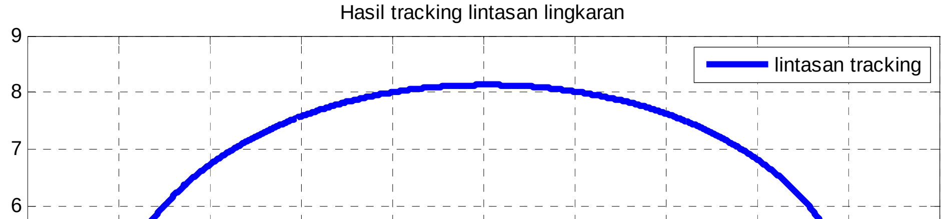
Parameter uji untuk lintasan lingkaran ditampilkan pada Tabel 4.16 berikut

Tabel 4.16 *Parameter* Uji Simulasi Lintasan Lingkaran

Parameter Uji	
Koordinat pusat	(0,0)
Radius	4 mil
Kecepatan kapal	8 knot
Waktu tempuh	11500 s

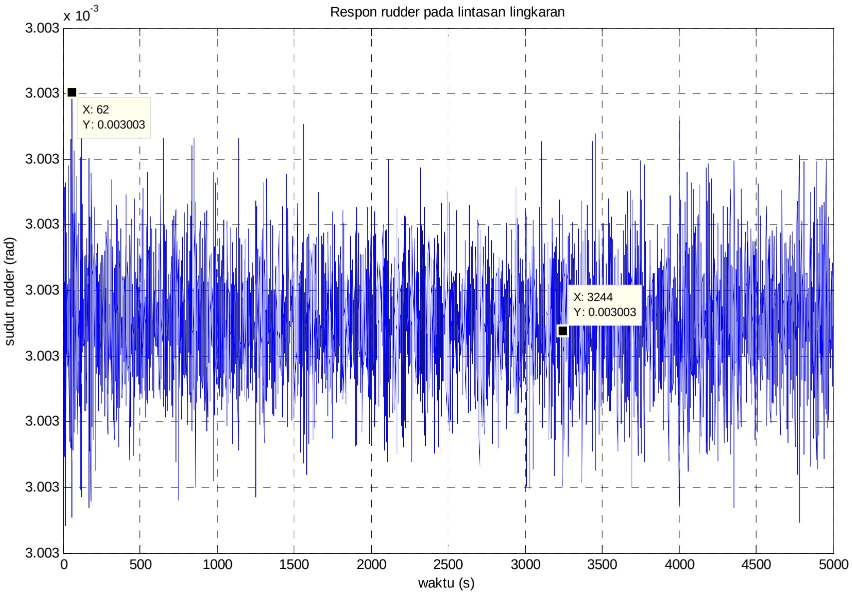
Dengan menginput data pada Tabel 4.16 pada m-file *software* MATLAB maka diperoleh hasil *tracking* untuk lintasan lingkaran dengan radius (R) sebesar 4 mil yang ditampilkan pada Gambar 4.56 berikut.

Dari gambar tampak *plant* dapat men*tracking* target sesuai dengan radius yang telah diinputkan tanpa ada *error tracking*.



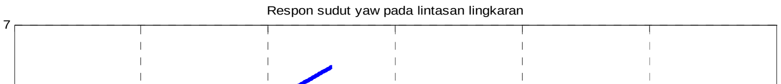
Gambar 4.56 Hasil Simulasi *Tracking* Lintasan Lingkaran

Respon sudut *rudder* (δ) pada simulasi *tracking* lintasan lingkaran ditampilkan pada Gambar 4.57 berikut



Gambar 4.57 Hasil Simulasi Respon Sudut *Rudder* untuk Lintasan Lingkaran

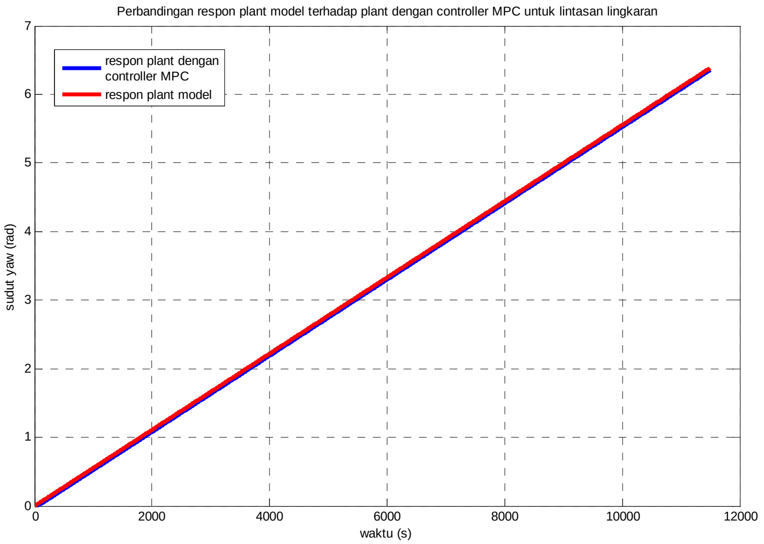
Gambar 4.57 menunjukkan sudut *rudder* sangat fluktuatif dengan *overshoot* 0.003 rad, hal ini disebabkan oleh dua hal yaitu adanya *disturbance* berupa arus laut dengan kecepatan 2 knot serta sudut *yaw* kapal yang terus bertambah mengikuti



bentuk lintasan target berupa lingkaran dengan jari-jari 4 mil. Respon sudut *yaw* (ψ) kapal untuk *tracking* pada lintasan lingkaran ditampilkan pada Gambar 4.58

Gambar 4.58 Hasil Simulasi Respon Sudut *Yaw* pada *Tracking* Lintasan Lingkaran

Perbandingan respon *plant* model terhadap *plant* dengan kontroler MPC untuk lintasan lingkaran ditampilkan pada Gambar 4.59 dimana respon *plant* dengan controller MPC mampu mengikuti respon dari *plant* model



Gambar 4.59 Perbandingan Respon *Plant* Model terhadap *Plant* dengan Kontroler MPC pada Lintasan Lingkaran

Berdasarkan hasil pengujian yang telah dilakukan, diperoleh beberapa fenomena berikut :

1. Pada simulasi untuk lintasan garis lurus, lintasan persegi panjang dan lintasan bebas (*free track*) *plant* dapat men*tracking trajectory* dengan baik hal ini

ditunjukan oleh hasil simulasi yang sesuai dengan *trajectory* target. Sedangkan pada lintasan lingkaran hasil *tracking* menunjukan hasil yang sesuai dengan radius target yaitu 4 mil.

2. Hasil respon sudut *rudder* (δ) pada setiap lintasan uji ditampilkan dalam Tabel 4.17 berikut

Tabel 4.17 Respon Sudut *Rudder* (δ) dari Sistem

Bentuk lintasan	**Sudut *Rudder* (rad)**
Lintasan lurus	0,05
Lintasan persegi panjang	0,06
Lintasan bebas (*free Track*)	0,06
Lintasan lingkaran	0,003

3. Berdasarkan tabel 4.17 respon sudut *rudder* (δ) hasil simulasi untuk setiap lintasan uji menunjukan nilai yang minimum hal ini menunjukan MPC controller bekerja cukup baik dengan mampu mempertahankan simpangan dari *disturbance*
4. Hasil analisa kesalahan (*error*) dari simulasi *tracking* terhadap target ditampilkan pada Tabel 4.18 dimana persentase *error* < 10 %

Tabel 4.18 Analisa Kesalahan (*Error*) dari Simulasi *Tracking* terhadap Target

Bentuk lintasan	**Jarak (mil)**	***Error* (mil)**		**Persentase *error* (%)**	
		X	**Y**	**X**	**Y**
Lintasan lurus	5,25	0,38	0,33	5,7	6,3
Lintasan persegi panjang	36,9	3,3	2,6	9	7
Lintasan bebas (*free track*)	20,5	1,6	1,6	7,8	7,8

5. Perubahan *parameter* controller MPC yaitu pada *horizon* prediksi N_p dan *horizon control* N_c menyebabkan perubahan kecepatan pada respon Kontroler dalam pengendalian *plant* model
6. Hasil variasi perubahan *parameter* N_p dan N_c dari setiap lintasan ditampilkan pada Tabel 4.19 – 4.21 berikut

Tabel 4.19 Hasil Variasi Perubahan *Parameter* Kontroler pada Lintasan Lurus

Uji pada Lintasan Garis Lurus		
N_p	N_c	*Response time* MPC untuk Pengendalian *Plant* Model (s)
	3	733

300	5	357
	7	259
	9	248
N_p	N_c	*Response time* MPC untuk Pengendalian *Plant* Model (s)
400	3	730
	5	334
	7	303
	9	239
N_p	N_c	*Respon time* MPC untuk Pengendalian *Plant* Model (s)
500	3	722
	5	369
	7	301
	9	251

Tabel 4.20 Hasil Variasi Perubahan *Parameter* Kontroler pada Lintasan Persegi Panjang

Uji pada Lintasan Persegi Panjang		
N_p	N_c	*Response time* MPC untuk Pengendalian *Plant* Model (s)
300	3	2247
	5	1998
	7	1946
	9	1903
N_p	N_c	*Response time* MPC untuk Pengendalian *Plant* Model (s)
400	3	2211
	5	1979
	7	1960
	9	1895
N_p	N_c	*Response time* MPC untuk Pengendalian *Plant* Model (s)
500	3	2159
	5	2015
	7	1967
	9	1869

Tabel 4.21 Hasil Variasi Perubahan *Parameter* Kontroler pada Lintasan Bebas

Uji pada Lintasan Bebas		
N_p	N_c	*Respon time* MPC untuk Pengendalian *Plant* Model (s)
300	3	635
	5	387
	7	340
	9	315

N_p	N_c	*Response time* MPC untuk Pengendalian *Plant* Model (s)
400	3	655
	5	425
	7	337
	9	311
N_p	N_c	*Response time* MPC untuk Pengendalian *Plant* Model (s)
500	3	525
	5	425
	7	313
	9	310

Halaman ini sengaja dikosongkan

BAB 5
PENUTUP

Berdasarkan hasil analisis yang telah diperoleh pada bab 4, maka dapat diambil kesimpulan dan saran mengenai perancangan dan simulasi penelitian ini.

5.1 Kesimpulan

1. untuk setiap lintasan uji respon *plant* menggunakan kontroler MPC mampu mengikuti respon pada *plant* model hal ini menunjukan MPC kontroler dapat bekerja dengan baik dan mampu mengikuti perubahan dinamika *plant*.
2. Dengan menggunakan MPC kontroler dihasilkan nilai simpangan sudut *rudder* yang minimum hal ini menunjukan bahwa MPC kontroler mampu mempertahankan simpangan sudut *rudder* dari *disturbance* seperti ditunjukan pada Tabel 4.17
3. Kesalahan (*error*) dari *tracking* terhadap target untuk setiap jenis lintasan kurang dari 10 % seperti ditunjukkan pada Tabel 4.18 dimana *error* terkecil terjadi pada *tracking* untuk lintasan garis lurus sebesar 5,7 % pada sumbu X dan 6,3 % pada sumbu Y sedangkan *error* terbesar terjadi pada *tracking* untuk lintasan persegi panjang yaitu sebesar 9 % pada sumbu X dan 7% pada sumbu Y
4. Kontroler MPC sangat baik untuk *tracking* berbasis radius. Hal ini ditunjukkan pada *tracking* untuk lintasan lingkaran dimana mampu membentuk lintasan sesuai dengan radius target yang diberikan yaitu 4 mil
5. Perubahan *horizon* prediksi N_p dan *horizon control* N_c pada kontroler MPC mempengaruhi respon pada sistem.
6. Semakin besar *horizon control* N_c yang digunakan maka respon *plant* dengan kontroler MPC semakin cepat dalam mengendalikan *plant* model seperti ditunjukkan pada Tabel 4.19 sampai dengan Tabel 4.21. berdasarkan tabel tersebut *horizon control* N_c yang paling sesuai untuk pengendalian *plant* model yaitu $N_c = 9$.

7. Berikut ini adalah *horizon control* N_c dan *horizon* prediksi N_p terbaik untuk pengendalian *plant* model pada lintasan garis lurus, lintasan persegi panjang dan lintasan bebas (*free track*)

   Pada lintasan garis lurus $N_c = 9$, $N_p = 400$ dengan *response time* 239 detik

   Pada lintasan persegi panjang $N_c = 9$, $N_p = 500$ dengan *response time* 1869 detik

   Pada lintasan bebas (*free track*) $N_c = 9$, $N_p = 500$ dengan *response time* 310 detik

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya antara lain:

1. Untuk penelitian selanjutnya sebaiknya menggunakan model *plant* dengan 3 DOF atau lebih sehingga dapat memperoleh dinamika *plant* yang lebih mendekati kondisi real serta diuji dengan bentuk lintasan yang lebih bervariasi.

LAMPIRAN

A.1 Menetukan Koordinat Target

```
clear all
close all
wptx = [0 1 2 5 6 8]; % koordinat input untuk sumbu x
wpty = [1 6 3 2 4 7]; % koordinat input untuk sumbu y
j=length(wptx);
wptime(1)=0;
for i=2:j
    dxy=sqrt( (wptx(i)-wptx(i-1))^2+(wpty(i)-wpty(i-1))^2);
    wptime(i)=wptime(i-1)+ceil(dxy/8*3600)
end

ts=2;
t = 0:ts:max(wptime)      %time
x_p = pchip(wptime,wptx,t)
y_p = pchip(wptime,wpty,t)
jj=length(x_p);
vs(1)=8;
for i=2:jj
    vs(i,1)=sqrt( ( x_p(i)-x_p(i-1))^2 + ( y_p(i)-y_p(i-
1))^2)*3600/ts;
end
vss=[t' vs];
xxp=[t' x_p'] % output target pada sumbu x
yyp=[t' y_p'] % output target pada sumbu y
plot(wptx,wpty,'o',x_p,y_p) %output trajectory lintasan kapal
;
```

A.2 Tahap *Trajectory Tracking*

Untuk bentuk lintasan garis lurus, persegi dan lintasan bebas

```
function out=trajek01(in)
global wptx wpty p dx dy dxy jp
x=in(1);
y=in(2);
tt=in(3);
if tt==0
wptx = [0 1 2 5 6 8]; % koordinat input untuk sumbu x
wpty = [1 6 3 2 4 7];    % koordinat input untuk sumbu y
dx=0;
dy=0;
p = 1;
jp=length(wptx);
     end
     dxy=sqrt(dx^2+dy^2);
     if dxy<0.5
         p=p+1;
     end
     if p>jp
         p=jp;
     end
```

```
        dx=wptx(p)-x;
        dy=wpty(p)-y;
        if dx==0
            dx=0.00001;
        end
        if dy==0
            dy=0.00001;
        end
        if and(dx>0,dy>0)
            yaw=atan(dy/dx);
        end
        if and(dx<0,dy>0)
            yaw=pi+atan(dy/dx);
        end
        if and(dx<0,dy<0)
            yaw=pi+atan(dy/dx);
        end
        if and(dx>0,dy<0)
            yaw=atan(dy/dx);
        end
    vs=8;
    out=[vs;yaw];
```

Untuk bentuk lintasan lingkaran

```
function out=trajek02(in)
global r x y k vs yaw
x=in(1);
y=in(2);
tt=in(3);
if tt==0
    r=4; $ radius lingkaran
    k=0;
    vs=8;
end
wr=vs/(3600*r);
yaw=wr*k;
out=[vs;yaw];

k=k+1
```

DAFTAR PUSTAKA

[1] Steiner Ardal, "*Robust Control applied on a cargo ship*", University of California, USA, 2003.

[2] R.Ferreiro Garcia, "*Multivariabel fuzzy cascade compensation*" Dept. Industrial Engineering University of a Coruna, Spain, 2006.

[3] Yanseng Yang, *"Robust adaptive fuzzy control and its application to ship roll stabilization",* Institute of Nautical Technology, Dalian Maritime University, China, 2008, pp. 177 – 194.

[4] Zhen Li et al, "*Disturbance Compensating Model predictive Control with Applicaton to Ship Heading Control*", IEEE Transaction on Control *Sistem* Technology, 2012, pp. 257 – 265.

[5] Leghmizi et al, "*Modelling Design and Control a Ship Carried 3 DOF Stabilized Platform*", College of Automation, Harbin Engineering University, China, 2012, pp. 3843 – 3851.

[6] J.M.Maciejowski, "*Predictive Control with Constraints*", Prentice Hall, 2000.

[7] Thor I. Fossen, "*Guidance and Control of Ocean Vehicle*" University of Trondheim, Norway, 1999, pp. 181 – 185.

[8] Aulia Siti et all "Fuzzy Control for Optimizing Ship Tracing in Karang Jamuang – Tanjung Perak", Journal for Technology and Science, 2012, pp. 118 – 124.

[9] Curtis D Johnson, *"Process Control Instrumentation Technology"*, University of Houston, USA, 2003.

[10] Van Amoregen, *"Adaptive Steering of ship-A Model Reference Approach to Improve Maneuvering and Economical Course Keeping",* PhD Thesis, Delft University of Technology, Netherland, 1982.

[11] Liuping Wang, "*Model Predictive Control Sistem Design and Implementation Using MATLAB",* School of Electrical and Computer Engineering, Australia, 2009.

Halaman ini sengaja dikosongkan

RIWAYAT PENULIS

Aji Brahma Nugroho dilahirkan di Surabaya, 30 Januari 1986. Merupakan putra pertama dari pasangan Sugeng Rawuhipun S.pd dan Mursiati. Lulus dari SDN Patrang IV Jember tahun 1998, kemudian melanjutkan ke SMPN 6 Jember dan lulus pada tahun 2001. Selanjutnya penulis melanjutkan pendidikan di SMAN 2 Jember dan lulus pada tahun 2004. Setelah lulus pendidikan SMA, penulis melanjutkan pendidikan ke perguruan tinggi strata-1 di Universitas Airlangga Surabaya jurusan Fisika dan lulus pada tahun 2010. Di tahun yang sama setelah lulus pendidikan strata-1, penulis melanjutkan pendidikan Pascasarjana strata-2 di Institut Teknologi 10 Nopember Surabaya jurusan Teknik Elektro dengan bidang keahlian teknik sistem pengaturan. Pada bulan Desember 2015 penulis mengikuti seminar dan ujian tesis sebagai salah satu syarat untuk memperoleh gelar Magister Teknik (MT).