



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 091399**

***TRACTION CONTROL PADA PARALLEL HYBRID  
ELECTRIC VEHICLE DENGAN METODE  
GENERALIZED PREDICTIVE CONTROL***

**Ilmiah Elrosa Citra Resmi  
NRP 2210 100 125**

**Dosen Pembimbing  
Ir. Rusdhianto Effendi AK, MT.  
Eka Iskandar, ST., MT.**

**JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2014**



**FINAL PROJECT - TE 091399**

**TRACTION CONTROL IN PARALLEL HYBRID ELECTRIC  
VEHICLE BY USING GENERALIZED PREDICTIVE  
CONTROL**

Ilmiyah Elrosa Citra Resmi  
NRP 2210 100 125

Advisor  
Ir. Rusdhianto Effendi AK,MT.  
Eka Iskandar, ST., MT.

ELECTRICAL ENGINEERING DEPARTEMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2014



**TRACTION CONTROL PADA PARALLEL HYBRID ELECTRIC  
VEHICLE DENGAN METODE  
GENERALIZED PREDICTIVE CONTROL**

**Nama** : Ilmiyah Elrosa Citra Resmi  
**Pembimbing I** : Ir. Rusdhianto Effendi AK,MT.  
**Pembimbing II** : Eka Iskandar, ST., MT

**ABSTRAK**

Dalam *Hybrid Electric Vehicle*, terutama pada *Parallel Hybrid Electric Vehicle* (PHEV) terdapat suatu masalah, yaitu lambatnya respons *Internal Combustion Engine* (ICE) apabila terjadi peningkatan kecepatan. Saat keadaan akselerasi ICE membutuhkan bantuan dari motor listrik. Karena jika tidak ada bantuan dari motor listrik dan terjadi secara terus menerus dapat mengurangi kinerja dari ICE. Oleh karena itu dibutuhkan pengaturan traksi motor listrik untuk membantu ICE dalam mencapai tingkat akselerasi yang diinginkan.

Pada Tugas Akhir ini digunakan metode *Generalized Predictive Control* untuk mengatasi masalah kenaikan kecepatan. Dengan metode ini yang memanfaatkan informasi masa depan, dapat menghasilkan respon yang mendekati model referensi. Dengan itu dapat membantu ICE dalam menanggung beban yang diberikan pada saat akselerasi. Berdasarkan hasil pengujian sistem pengaturan dengan metode GPC dapat memperbaiki respon yaitu peningkatan *rise time* dari 0,624s ke 0,35s, hal ini membuktikan bahwa kontroler GPC cocok untuk meningkatkan akselerasi

**Kata kunci:** *Hybrid Electric Vehicle, Traction Control, GPC*



*[Halaman ini sengaja dikosongkan]*

## **TRACTION CONTROL IN PARALLEL HYBRID ELECTRIC VEHICLE BY USING GENERALIZED PREDICTIVE CONTROL**

**Name** : Ilmiyah Elrosa Citra Resmi  
**Advisor I** : Ir. Rusdhianto Effendi AK,MT.  
**Advisor II** : Eka Iskandar, ST., MT

### **ABSTRACT**

*In Hybrid Electric Vehicle, especially for Parallel Hybrid Electric Vehicle (PHEV), there is a problem, decreasing Internal Combustion Engine(ICE) response when increasing of velocity happen. In the acceleration mode, ICE need a help from electric motor. If the electric motor doesn't help, and it happen all time, that can decrease ICE's performance. Hence, traction control is needed for electric motor to help ICE reach acceleration.*

*In this final project, Generalized Predictive Control is used for solve the increasing of velocity. With that method, using the future information can generate response that closed to reference model. With this method, can help ICE for bearing the load in acceleration mode. The testing result, control system using GPC can increasing rise time from 0,624s to 0,35s. it proves that GPC controller appropriate for increasing acceleration.*

**Keyword** : Hybrid Electric Vehicle, Traction Control, GPC



[Halaman ini sengaja dikosongkan]

**TRACTION CONTROL PADA PARALLEL HYBRID  
ELECTRIC VEHICLE DENGAN METODE  
GENERALIZED PREDICTIVE CONTROL**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Teknik Sistem Pengaturan  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I,

Dosen Pembimbing II,

  
Ir. Rusdhianto Effendi AK, M.T.  
NIP. 19570424 198502 1 001

  
Eka Iskandar S.T., M.T.  
NIP. 19800528 200812 1 001

**SURABAYA  
JANUARI, 2014**

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini dengan judul “*Traction Control pada Parallel Hybrid Electric Vehicle dengan Metode Generalized Predictive Control*”. Tugas Akhir ini disusun guna memenuhi persyaratan untuk mencapai Gelar Sarjana Teknik pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Pada kesempatan yang penulis juga mengucapkan terima kasih kepada

1. Keluarga saya atas doa dan dukungannya sehingga penulis dapat termotivasi untuk menyelesaikan Tugas Akhir.
2. Bapak Rusdhianto Effendi dan Bapak Eka Iskandar selaku pembimbing Tugas Akhir.
3. Teman-teman Bidang Studi Sistem Pengaturan yang selalu menemani penulis dalam pengerjaan Tugas Akhir dan *thanks to all of my teammates* Wid, Gopar, Chacul.
4. Mas Rama dan mas Fahrul yang telah banyak membantu dalam pengerjaan *software* dan *hardware*.
5. Serta berbagai pihak yang turut membantu dalam pengerjaan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa isi buku laporan Tugas Akhir ini masih jauh dari sempurna baik dalam penyusunan, maupun dalam pembahasan dan analisa permasalahan. Oleh karena itu, penulis mengucapkan mohon maaf yang sebesar-besarnya atas segala kekurangan dalam buku ini. Penulis juga berharap masukan dari pembaca agar penulis dapat mengurangi kesalahan pada penulisan berikutnya.

Besar harapan penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Jurusan Teknik Elektro pada khususnya.

Surabaya, Januari 2014

Penulis



*[Halaman ini sengaja dikosongkan]*

# DAFTAR ISI

HALAMAN

Halaman Judul .....	i
Pernyataan Keaslian.....	iii
Lembar Pengesahan .....	v
Abstrak .....	vii
<i>Abstract</i> .....	ix
Kata Pengantar .....	xi
Daftar Isi .....	xiii
Daftar Gambar .....	xv
Daftar Tabel .....	xix
BAB 1 Pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan.....	2
1.4 Batasan Masalah .....	2
1.5 Sistematika Penulisan .....	2
1.6 Relevansi .....	3
BAB 2 Tinjauan Pustaka.....	5
2.1 <i>Hybrid Electric Vehicle (HEV)</i> .....	5
2.1.1 HEV Seri .....	5
2.1.2 HEV Paralel .....	6
2.1.3 HEV Seri Paralel.....	7
2.2 <i>Internal Combustion Engine (ICE)</i> .....	7
2.3 Motor Listrik .....	10
2.4 Rem Elektromagnetik .....	14
2.5 Sistem Penyesuaian Kecepatan.....	15
2.6 Tachogenerator .....	16
2.7 Arduino Uno .....	17
2.8 MATLAB2010a.....	18
2.9 Permodelan Sistem.....	20
2.9.1 Metode Viteckova .....	20
2.9.2 Metode Latzel .....	21
2.9.3 Identifikasi Strejc.....	21
2.9.4 <i>Root Mean Square Error</i> .....	24
2.10 Kontrol Prediktif.....	24
2.11 <i>Generalized Predictive Control (GPC)</i> .....	26
2.12 <i>State Observer</i> .....	32

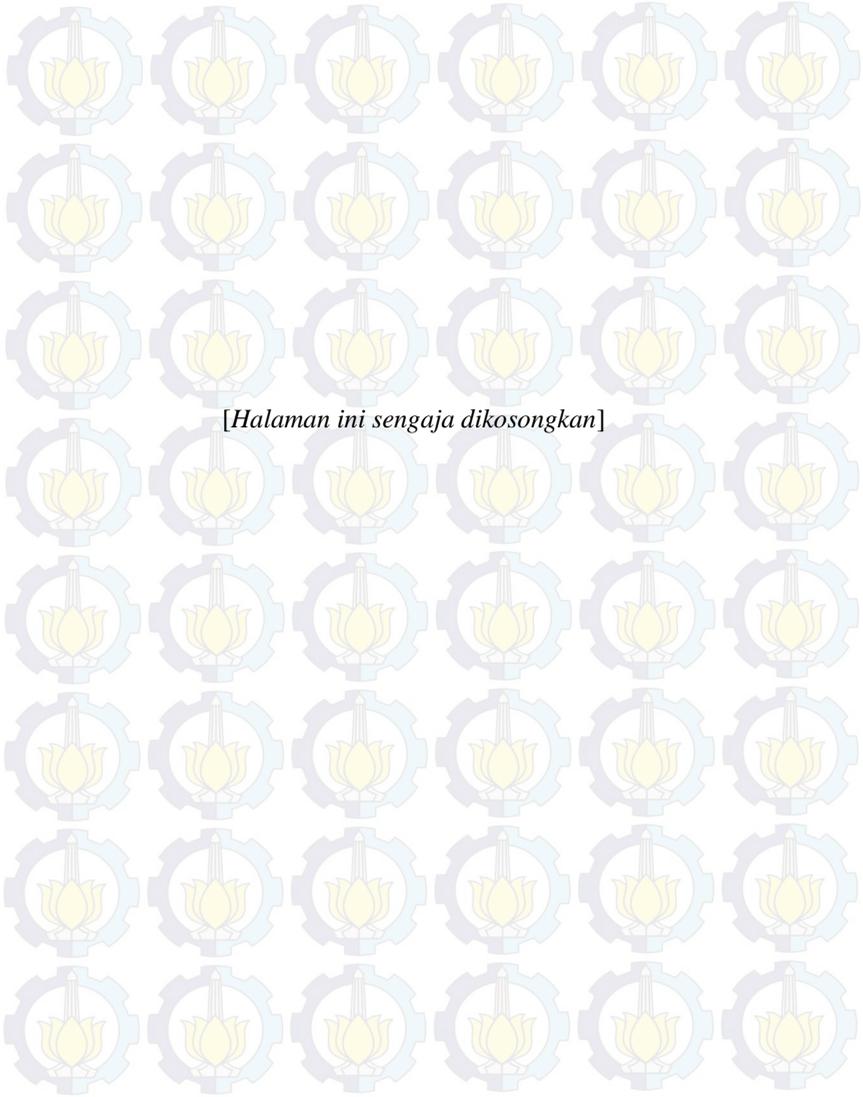
BAB 3 Perancangan Sistem .....	35
3.1 Gambaran Umum Sistem .....	35
3.2 Perancangan Perangkat Keras .....	36
3.2.1 Perancangan Mekanik .....	37
3.2.2 Perancangan Elektronik.....	43
3.3 Perancangan Perangkat Lunak.....	48
3.3.1 MATLAB2010a.....	48
3.3.2 <i>Software</i> Arduino.....	49
3.4 Pengujian <i>Open Loop</i> Kecepatan Motor DC .....	50
3.5 Pengujian Sensor .....	51
3.5.1 Potensiometer .....	51
3.5.2 Tachogenerator.....	52
3.6 Identifikasi dan Permodelan Sistem .....	53
3.6.1 Metode Pembebanan <i>Plant</i> .....	53
3.6.2 Metode Identifikasi dan Permodelan .....	54
3.6.3 Permodelan ICE.....	55
3.6.4 Permodelan Motor DC.....	60
3.6.5 Pengujian dan Validasi .....	62
3.6.6 Model Referensi.....	67
3.7 Desain Sistem Pengaturan Traksi .....	68
3.7.1 Desain Generalized Predictive Control .....	68
3.7.2 Desain Prediktor .....	70
3.7.3 Desain <i>Observer</i> .....	72
BAB 4 Pengujian dan Analisis .....	77
4.1 Gambaran Umum Pengujian Sistem.....	77
4.2 Simulasi Sistem .....	77
4.2.1 Diagram Blok Simulasi Sistem .....	77
4.2.2 Pengujian Respons .....	79
4.3 Implementasi Sistem .....	94
4.3.1 Pengujian Kontroler GPC Satu Langkah Kedepan.....	95
4.3.2 Pengujian Kontroler GPC Tiga Langkah Kedepan .....	95
BAB 5 Penutup.....	97
5.1 Kesimpulan .....	97
5.2 Saran.....	97
DAFTAR PUSTAKA .....	99
LAMPIRAN .....	101
RIWAYAT PENULIS .....	105

## DAFTAR GAMBAR

Gambar 2.1 Konfigurasi HEV Seri .....	5
Gambar 2.2 Konfigurasi HEV Paralel.....	6
Gambar 2.3 Konfigurasi HEV Seri-Paralel.....	7
Gambar 2.4 Komponen dalam Piston .....	8
Gambar 2.5 Prinsip Kerja ICE 2 Tak .....	10
Gambar 2.6 Langkah Pembakaran ICE 2 Tak.....	10
Gambar 2.7 Arah Putaran Motor DC .....	11
Gambar 2.8 Rangkaian dan Kurva Karakteristik Motor DC Seri.....	12
Gambar 2.9 Rangkaian dan Kurva Karakteristik Motor DC Shunt.....	13
Gambar 2.10 Rangkaian dan Kurva Karakteristik Motor DC Kompon	13
Gambar 2.11 Rem Elektromagnetik.....	15
Gambar 2.12 Konfigurasi Gir Beda Poros .....	16
Gambar 2.13 Konfigurasi Gir Satu Poros .....	16
Gambar 2.14 <i>Instrument Control Toolbox</i> .....	19
Gambar 2.15 Penarikan Garis pada Metode Strejce .....	22
Gambar 2.16 Sinyal yang Dimanipulasi, Referensi, dan Sinyal Kontrol melalui Aksi Kontrol .....	25
Gambar 2.17 Blok Diagram Sistem dan <i>Full Order State Observer</i> .....	34
Gambar 3.1 Bentuk Fisik Simulator HEV .....	35
Gambar 3.2 Diagram Blok Sistem .....	35
Gambar 3.3 Konfigurasi Perangkat Keras Simulator HEV .....	36
Gambar 3.4 Bentuk Fisik Mesin Pemotong Rumput .....	38
Gambar 3.5 Bentuk Fisik Motor DC .....	38
Gambar 3.6 Gir Transmisi .....	40
Gambar 3.7 Bentuk Fisik Potensiometer.....	41
Gambar 3.8 Bentuk Fisik Rem Elektromagnetik .....	42
Gambar 3.9 Konstruksi Rem Elektromagnetik .....	42
Gambar 3.10 Bentuk Fisik Tachogenerator .....	43
Gambar 3.11 Bentuk Fisik Rangkaian <i>Driver</i> Motor DC .....	43
Gambar 3.12 Rangkaian <i>Driver</i> Motor DC .....	44
Gambar 3.13 Pin Arduino pada Sistem Pengaturan HEV .....	45
Gambar 3.14 Bentuk Fisik Rangkaian Pembagi Tegangan.....	46
Gambar 3.15 Rangkaian Pembagi Tegangan .....	46
Gambar 3.16 Rangkaian Penyearah Gelombang Penuh .....	47
Gambar 3.17 Bentuk Fisik Rangkaian Penyearah Gelombang Penuh .	47
Gambar 3.18 Blok Simulink MATLAB untuk Pengambilan Data Open Loop.....	48

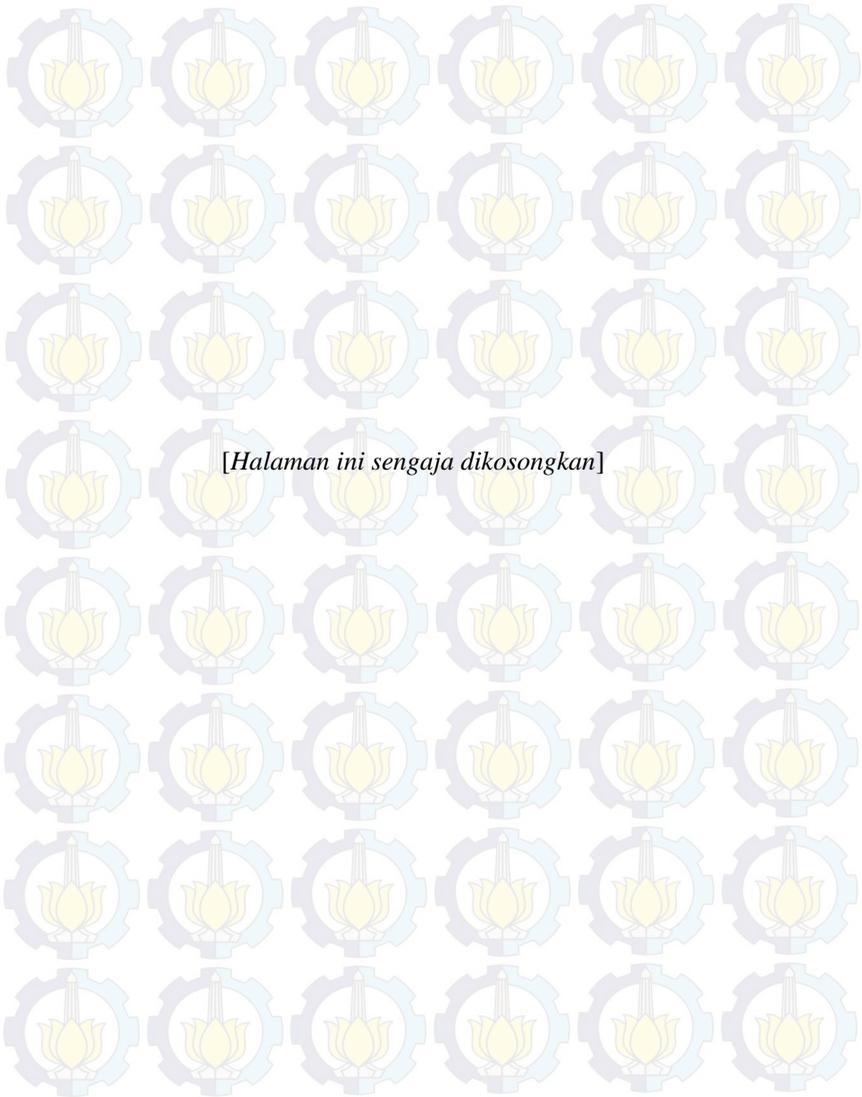
Gambar 3.19 Blok Simulink MATLAB untuk Pengambilan Data dengan Kontroler .....	49
Gambar 3.20 Tampilan <i>Software</i> Arduino .....	49
Gambar 3.21 Tegangan <i>Input</i> Motor DC dan Kecepatan Putar .....	51
Gambar 3.22 Pijakan pedal .....	52
Gambar 3.23 Respon Kecepatan Putar ICE .....	52
Gambar 3.24 Tegangan <i>Output</i> Tachogenerator terhadap Kecepatan Putar .....	53
Gambar 3.25 Respons ICE dengan Beban Minimal .....	56
Gambar 3.26 Respons ICE dengan Beban Nominal .....	58
Gambar 3.27 Respons ICE dengan Beban Maksimal .....	59
Gambar 3.28 Respons Kecepatan Motor DC .....	60
Gambar 3.29 Hasil Identifikasi dan Pemodelan ICE Beban Minimal ...	63
Gambar 3.30 Hasil Identifikasi dan Pemodelan ICE Beban Nominal ...	64
Gambar 3.31 Hasil Identifikasi dan Pemodelan ICE Beban Maksimal ...	65
Gambar 3.32 Hasil Identifikasi dan Pemodelan Motor DC .....	66
Gambar 4.1 Diagram Blok Simulasi Kontrol dengan GPC .....	77
Gambar 4.2 Diagram Blok Kontroler GPC .....	78
Gambar 4.3 Diagram Blok Prediktor ICE .....	78
Gambar 4.4 Diagram Blok Prediktor Model Referensi .....	78
Gambar 4.5 Diagram Blok Observer .....	79
Gambar 4.6 Respons ICE Terhadap Sinyal Uji Step .....	79
Gambar 4.7 Sinyal <i>Error</i> antara Kecepatan ICE dan Model Referensi ...	80
Gambar 4.8 Respons ICE terhadap Sinyal Step dengan Perubahan Amplitudo .....	80
Gambar 4.9 Sinyal <i>Error</i> antara Kecepatan ICE dan Model Referensi dengan Sinyal Uji Step dengan Perubahan Amplitudo ...	81
Gambar 4.10 Motor DC dengan Kontroler GPC Saat Informasi Masa Depan Diketahui .....	82
Gambar 4.11 Motor DC dengan Kontroler GPC Saat Informasi Masa Depan Tidak Diketahui .....	82
Gambar 4.12 Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan <i>Input</i> Sinyal Step .....	83
Gambar 4.13 Sinyal Kontrol GPC 3 Prediksi Kedepan dengan <i>Input</i> Sinyal Step .....	84
Gambar 4.14 <i>Output</i> Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan <i>Input</i> Sinyal Step .....	84
Gambar 4.15 Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan <i>Input</i> Sinyal Step dan Informasi Masa Depan. ...	85

Gambar 4.16 Sinyal Kontrol GPC 3 Prediksi Kedepan dengan <i>Input</i> Sinyal Step dan Informasi Masa Depan.....	86
Gambar 4.17 <i>Output</i> Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan <i>Input</i> Sinyal Step dan Informasi Masa Depan.....	86
Gambar 4.18 Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step .....	87
Gambar 4.19 Sinyal Kontrol GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step .....	88
Gambar 4.20 <i>Output</i> Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step.....	88
Gambar 4.21 Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step dan Infomasi Masa Depan .....	89
Gambar 4.22 Sinyal Kontrol GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step dan Informasi Masa Depan .....	90
Gambar 4.23 <i>Output</i> Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step .....	90
Gambar 4.24 Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan <i>Input</i> Sinyal Step.....	91
Gambar 4.25 Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step .....	92
Gambar 4.26 Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan <i>Input</i> Sinyal Step dan Infomasi Masa Depan.....	93
Gambar 4.27 Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan Perubahan Amplitudo <i>Input</i> Sinyal Step dan Infomasi Masa Depan .....	94
Gambar 4.28 Respons Hasil Implementasi dengan Kontroler GPC dengan Prediksi Satu Langkah Kedepan .....	95
Gambar 4.29 Respons Hasil Implementasi dengan Kontroler GPC dengan Prediksi 3 Langkah Kedepan .....	96



## DAFTAR TABEL

Tabel 2.1 Tabel Parameter Latzel .....	21
Tabel 2.2 Nilai Konstanta T .....	23
Tabel 2.3 Parameter Identifikasi Metode Strejc .....	23
Tabel 3.1 Perbandingan Tegangan <i>Input</i> dan Kecepatan Putar Motor DC .....	50
Tabel 3.2 Perbandingan Pijakan Pedal, Tegangan Potensiometer, <i>Output</i> Kecepatan ICE .....	51
Tabel 3.3 Perbandingan Nilai Tegangan dan Kecepatan Putar .....	53
Tabel 3.4 Permodelan dengan Beberapa Metode .....	54
Tabel 3.5 Persamaan Model dan Validasi <i>Plant</i> ICE Beban Minimal ..	63
Tabel 3.6 Persamaan Model dan Validasi <i>Plant</i> ICE Beban Nominal ..	64
Tabel 3.7 Persamaan Model dan Validasi <i>Plant</i> ICE Beban Maksimal	65
Tabel 3.8 Persamaan Model dan Validasi <i>Plant</i> Motor DC .....	66
Tabel 3.9 Pemilihan Model <i>Plant</i> ICE .....	67
Tabel 4.1 Nilai <i>Rise Time</i> dan <i>Error Output</i> HEV dengan Prediksi 1 hingga 5 Langkah Kedepan dan Input Sinyal Step .....	91
Tabel 4.2 Nilai <i>Rise Time</i> dan <i>Error Output</i> HEV dengan Prediksi 1 hingga 5 Langkah Kedepan dan Input Sinyal Step dengan Perubahan Amplitudo .....	92
Tabel 4.3 Nilai <i>Rise Time</i> dan <i>Error Output</i> HEV dengan Prediksi 1 hingga 5 Langkah Kedepan, Input Sinyal Step dan Informasi Masa Depan .....	93
Tabel 4.4 Nilai <i>Rise Time</i> dan <i>Error Output</i> HEV dengan Prediksi 1 hingga 5 Langkah Kedepan, Input Sinyal Step dengan Perubahan Amplitudo dan Informasi Masa Depan .....	94



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam kehidupan nyata, terdapat berbagai macam masalah yang sampai saat ini belum bisa diselesaikan. Salah satu masalah tersebut adalah masalah transportasi. Semakin meningkatnya jumlah alat transportasi mengakibatkan peningkatan tingkat emisi gas yang diakibatkan oleh asap asap kendaraan bermotor. Hal ini mengakibatkan memburuknya iklim di dunia (*global warming*). Ditambah lagi dengan penggunaan bahan bakar minyak (BBM) yang tergolong sumber daya alam yang tidak dapat diperbaharui, sehingga jika digunakan secara terus menerus akan terjadi kelangkaan..

Kendaraan listrik muncul untuk menyelesaikan permasalahan tersebut. Keunggulan kendaraan ini yaitu tidak mengeluarkan emisi karbon, tidak menggunakan BBM, dan tidak bising. Kelemahan kendaraan ini yaitu rendahnya kapasitas baterai, sehingga membuat waktu berkendara rendah dan membutuhkan waktu pengisian yang lama.

Salah satu solusi untuk menyelesaikan permasalahan diatas adalah dengan menggunakan kendaraan hybrid (*Hybrid Electric Vehicle*). Kendaraan hybrid merupakan kendaraan yang bekerja dengan menggunakan dua jenis penggerak. Pada umumnya penggerak yang digunakan adalah *Internal Combustion Engine* (ICE) dan motor listrik. Motor listrik didesain untuk membantu motor bakar pada keadaan transien dan membantu ICE beroperasi lebih konstan, sehingga dapat meningkatkan efisiensi penggunaan bahan bakar dan rendah emisi. Alat untuk Implementas Tugas Akhir ini merupakan simulator HEV konfigurasi paralel yang telah dirancang sebelumnya oleh Galih Satrio dan Wahyu Sriwidodo [1][2].

### 1.2 Rumusan Masalah

Pada *Hybrid Electric Vehicle* terutama pada HEV konfigurasi paralel, ICE merupakan penggerak utama. Hal ini disebabkan kapasitas *charge* baterai tidak terlalu besar, sehingga tidak dapat menggerakkan motor listrik secara terus menerus. Dalam penggunaannya, kendaraan tidak terlepas dari adanya kenaikan atau penurunan kecepatan. Oleh karena itu dibutuhkan pengaturan traksi untuk mengatasi kenaikan kecepatan. Ketika kecepatan dinaikkan ICE akan meningkatkan

kinerjanya sehingga untuk menstabilkan kinerja ICE motor listrik aktif untuk membantu motor bakar mencapai kecepatan yang diinginkan.

Metode GPC akan digunakan untuk pengaturan traksi dalam penelitian ini, metode ini digunakan untuk memprediksi adanya *error* sehingga dapat menghasilkan sinyal kontrol yang sesuai sehingga menghasilkan output yang mengikuti model referensi. Selain itu metode ini berguna untuk menjaga kestabilan motor bakar dalam mencapai kecepatan yang diinginkan.

### **1.3 Tujuan**

Dengan digunakannya kontroler GPC pada HEV, diharapkan dapat meningkatkan performansi HEV saat keadaan akselerasi. Dengan bantuan motor listrik, maka dapat menghemat penggunaan bahan bakar pada ICE dan mengurangi emisi gas. Hal ini dapat dicapai karena kontroler GPC memiliki kemampuan untuk memprediksi nilai *output* masa depan, sehingga dapat menghasilkan sinyal kontrol yang sesuai.

Implementasi metode ini diharapkan dapat memberikan kontribusi nyata terhadap dunia industri terutama industri otomotif. Selain itu juga dapat memberikan kontribusi pada bidang pendidikan dan dapat digunakan untuk referensi penelitian lebih lanjut.

### **1.4 Batasan Masalah**

Permasalahan pada tugas akhir ini dibatasi oleh beberapa hal antara lain:

- a. ICE dioperasikan secara manual dan tidak diberikan aksi kontrol.
- b. Motor listrik diberikan tegangan dari jala jala PLN, tidak menggunakan baterai sehingga tidak ada mekanisme *charging* dan *discharging*.
- c. Pengaturan traksi simulator HEV dilakukan oleh motor listrik saat keadaan akselerasi.
- d. Kecepatan putar ICE merupakan kecepatan yang terukur pada poros motor DC, sehingga kecepatan terukur sebesar 25% kecepatan aktual

### **1.5 Sistematika Penulisan**

Tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

## **BAB I : PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika penulisan, dan relevansi.

## **BAB II : TINJAUAN PUSTAKA**

Bab ini berisi tentang teori yang menunjang penelitian, berupa teori tentang HEV dan komponennya, serta metode yang digunakan untuk pengaturan HEV.

## **BAB III : PERANCANGAN SISTEM**

Bab ini berisi tentang perancangan perangkat keras, perangkat lunak, dan perancangan kontroler.

## **BAB IV : IMPLEMENTASI DAN ANALISA**

Bab ini berisi tentang hasil simulasi kontroler dan analisisnya. Selain itu berisi tentang hasil implementasi kontroler pada Simulator HEV beserta analisa hasil implementasi.

## **BAB V : KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan hasil penelitian dan saran untuk penelitian selanjutnya.

### **1.6 Relevansi**

Hasil dari tugas akhir ini diharapkan dapat berkontribusi dalam pengembangan penelitian tentang HEV serta diharapkan dapat meningkatkan teknologi otomotif dalam negeri.



*[Halaman ini sengaja dikosongkan]*

## BAB 2

### TINJAUAN PUSTAKA

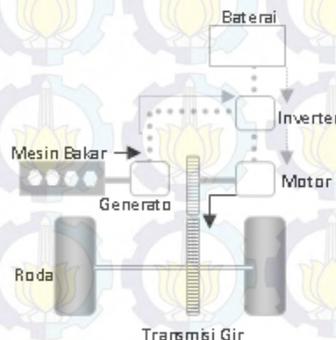
#### 2.1 *Hybrid Electric Vehicle (HEV)* [3]

HEV merupakan kendaraan yang melibatkan kombinasi dari dua sumber energi yaitu *Internal Combustion Engine (ICE)* dan motor listrik. Hal ini yang membedakan HEV dengan kendaraan konvensional yang beroperasi dengan ICE atau mesin bakar. Pada umumnya kendarannya bekerja dengan bahan bakar fosil (bensin atau diesel), sedangkan HEV menggunakan bahan bakar dan listrik. Energi listrik tersebut dialirkan pada motor listrik. Dalam hal ini motor listrik dirancang untuk menangani variasi energi *transien* dan membantu mesin untuk beroperasi lebih konstan dengan efisiensi tinggi dan emisi yang lebih rendah.

Ada beberapa jenis konfigurasi ICE dan motor listrik berdasarkan penghubungannya, yaitu HEV seri, HEV paralel. Dan HEV seri paralel.

##### 2.1.1 HEV Seri

Dalam konfigurasi ini ICE tidak terhubung secara langsung ke rangkaian kereta penggerak, tetapi generator listrik sebagai gantinya. Konfigurasi HEV seri dapat dilihat pada Gambar 2.1. Generator ini terpisah dengan motor listrik kecil yang dipasang pada tiap roda. Dengan ini dapat memungkinkan pengaturan energi yang disalurkan pada tiap roda dan mempermudah dalam pengaturan traksi. Keunggulan dari tipe ini adalah fleksibilitas yang diberikan karena tidak adanya hubungan mekanis antara ICE dan roda.



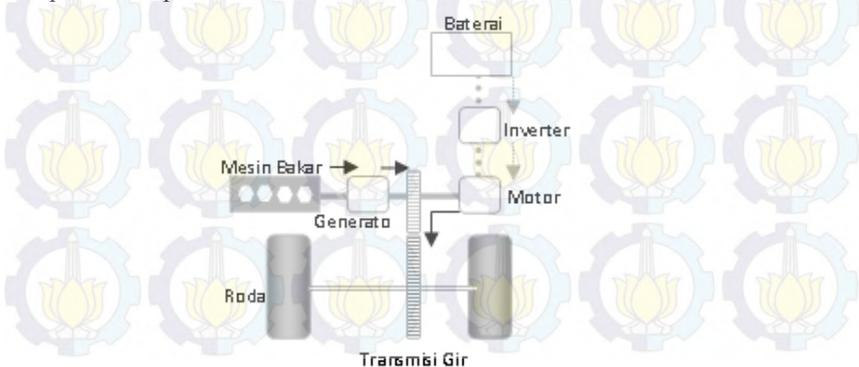
**Gambar 2.1** Konfigurasi HEV Seri

Terdapat kelemahan dan kelebihan dari konfigurasi ini. Kelemahan dari HEV seri adalah membutuhkan pemisahan porsi antara motor dan generator yang dapat dikombinasikan dalam HEV paralel. Efisiensi gabungan dari ICE dan generator juga akan lebih rendah dibandingkan dengan transmisi konvensional. Namun penggunaan HEV konfigurasi seri ini lebih efisien pada kendaraan yang sering berhenti dan mulai kembali, seperti kendaraan pengirim, bus kota, dan lain sebagainya.

### 2.1.2 HEV Paralel

Sistem ini menghubungkan ICE dan motor listrik langsung ke kereta penggerak. HEV ini dapat diklasifikasikan kembali berdasarkan rasio kontribusi energi. Dalam beberapa kasus mesin bakar adalah komponen yang dominan untuk memasok energi dan menggunakan energi baterai hanya ketika adanya percepatan. Selain itu ada pula kendaraan yang hanya menggunakan motor listrik. Pada umumnya HEV paralel menggabungkan generator listrik dan motor dalam satu unit yang terletak antara ICE dan gir transmisi untuk menggantikan motor starter konvensional dan alternator.

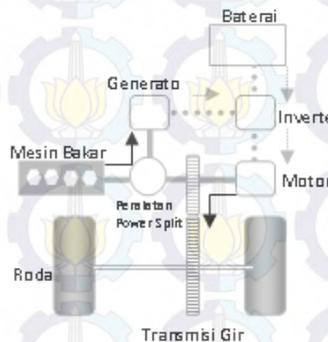
Baterai yang besar diperlukan untuk memberikan tegangan yang lebih tinggi 12V dari kendaraan normal. Perlengkapan tambahan seperti *power steering* dan AC mendapatkan energi dari motor listrik, sehingga perlengkapan itu dapat terus berfungsi walaupun ICE dihentikan. Dengan ini dapat meningkatkan efisiensi dari sistem dibandingkan dengan dijalankan langsung dengan ICE. Konfigurasi HEV paralel ini dapat dilihat pada Gambar 2.2



**Gambar 2.2** Konfigurasi HEV Paralel

### 2.1.3 HEV Seri Paralel

Kendaraan berjenis ini menggabungkan keunggulan dari kedua konfigurasi sebelumnya untuk menciptakan sistem yang efisien. Mode pertama, motor listrik digunakan untuk mengontrol kecepatan mesin sedangkan ICE memberikan sisa energi yang diperlukan untuk meningkatkan akselerasi. Mode kedua, penggunaan motor listrik dapat dikurangi. Masing masing motor digunakan untuk mengatur kecepatan mesin dalam kondisi yang berbeda. Dengan ini efisiensi dapat ditingkatkan dengan mengurangi sirkulasi energi melalui penggunaan gigi tetap. Konfigurasi HEV Seri Paralel dapat dilihat pada Gambar 2.3

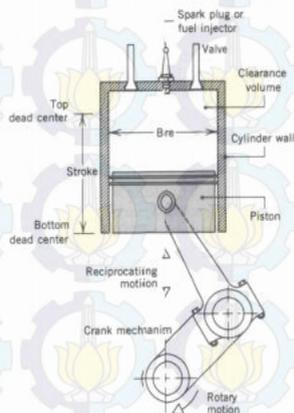


**Gambar 2.3** Konfigurasi HEV Seri-Paralel

### 2.2 *Internal Combustion Engine (ICE)* [4]

ICE sering digunakan dalam alat transportasi pribadi atau masal. Pada dasarnya ada dua jenis ICE yaitu mesin yang membutuhkan busi dan mesin yang mengandalkan kompresi fluida. Mesin yang membutuhkan busi mengambil campuran bahan bakar dan udara, mengkompresnya dan memicunya dengan menggunakan busi. Gambar 2.4 menunjukkan komponen dalam piston. Silinder piston pada dasarnya menggunakan mekanisme engkol silinder. Piston bergerak naik turun dengan gerakan berputar dari dua lengan. Lengan itu berputar dengan digerakkan oleh poros engkol. Piston ini terletak pada ruang pembakaran. Katup yang terletak diatas merupakan katup induksi dan pembuangan yang digunakan untuk memasukan campuran udara - bahan bakar dan pembuangan dari ruang residu. Pada mesin *spark ignition* diperlukan lecutan listrik untuk menyalakan campuran bahan bakar dan udara. Dalam mesin kompresi pengapian campuran menyatu pada suhu

dan tekanan yang tinggi. Titik terendah yang dicapai piston disebut titik mati bawah dan titik tertinggi yang dapat dicapai piston disebut titik mati atas. Rasio antara titik mati atas dan bawah disebut sebagai rasio kompresi. Rasio kompresi ini sangat penting dalam banyak aspek antara kompresi dan pengapian.



**Gambar 2.4** Komponen dalam Piston

Kompresi dari mesin pengapian mengambil udara luar, mengkompresinya ke temperatur dan tekanan tinggi pada saat pembakaran terjadi. Mesin ini dibagi menjadi 2 yaitu ICE 2 tak dan 4 tak. Pada ICE 4 tak, piston melakukan empat *stroke* yang berbeda untuk tiap perputaran poros engkol 2 kali, sedangkan pada ICE 2 tak, piston melakukan dua *stroke* yang berbeda dalam satu putaran.

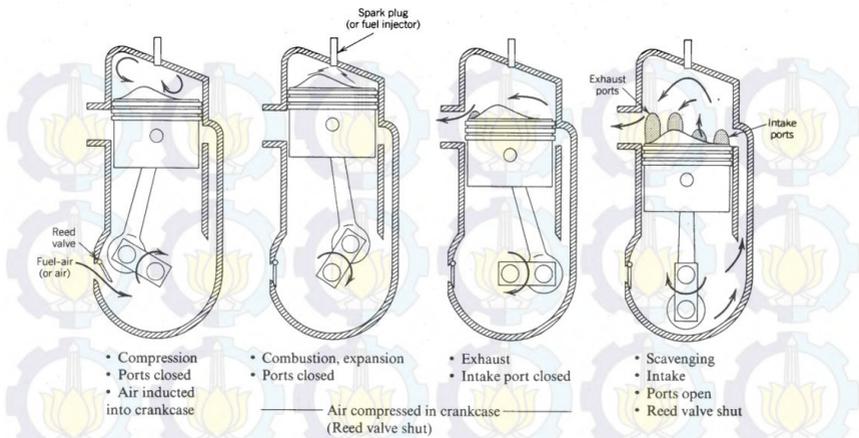
Ketika piston berada di titik mati bawah (TMB) katup *intake* terbuka, campuran bahan bakar dan air kemudian dikompresi ke titik mati atas (TMA), di mana busi digunakan untuk membakar campuran. Hal ini disebut *stroke* kompresi. Setelah memukul TMA campuran udara dan bahan bakar mengalami pembakaran. kekuatan dari *stroke* menyediakan energi yang dibutuhkan untuk mendorong poros engkol. Setelah piston kembali ke TMB maka katub buang akan membuka. *Stroke exhaust* adalah tempat residu untuk meninggalkan ruang pembakaran. Dengan ini agar *residu exhaust* dapat meninggalkan ruang pembakaran, tekanan ruang pembakaran harus lebih tinggi dari tekanan atmosfer. Kemudian piston bergerak ke TMA, sehingga katub buang menutup. *Stroke* berikutnya adalah langkah isap. Selama langkah isap

katup intake terbuka sehingga memungkinkan campuran udara dan bahan bakar masuk ke ruang bakar dan mengulangi proses yang sama.

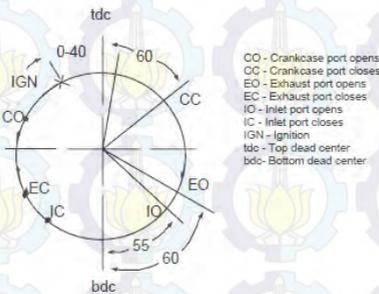
ICE 2 tak terbuat dari mesin pengapian kompresi. *Output* daya mesin ini biasanya tidak lebih dari 100W. Terdapat pula jenis lain dari ICE 2 tak yang memiliki *output* daya lebih dari 100W yaitu mesin *spark ignition*. Beberapa dari mesin yang berdaya tinggi ini relatif terhadap berat mereka. Mesin ini digunakan pada sepeda motor, gergaji mesin, mesin pemotong rumput, dan generator kecil.

Prinsip kerja ICE 2 tak ditunjukkan pada Gambar 2.5. Proses pembakaran untuk ICE 2 tak ini melalui langkah langkah sebagai berikut:

- a. Pada  $60^\circ$  sebelum memukul TMB, piston membuka *exhaust port* (tempat pembuangan gas) dan *exhaust* meninggalkan silinder menuju tekanan atmosfer.
- b. Pada  $5-10^\circ$ , *inlet port* akan membuka dan muatan yang telah dikompres pada *crankcase* akan mengalir ke ruang utama dan bercampur dengan *residu exhaust*. Beberapa muatan akan meninggalkan *exhaust port*. Deflektor akan membantu dengan mengalihkan aliran dari *inlet port* ke *exhaust port*.
- c. Pada sekitar  $55^\circ$  setelah TMB, ketika piston bergerak naik, *inlet port* akan menutup. Oleh karena itu terdapat aliran kembali muatan dari *inlet port* ke *crankcase*.
- d. Pada  $60^\circ$  setelah TMB, *exhaust port* akan menutup dan piston akan menekan muatan melalui gerakan ke atas.
- e. Pada  $60^\circ$  sebelum TMA, *port crankcase* akan terbuka dan muatan akan mengalir ke dalamnya. Muatan itu mengalir karena tekanan dalam *crankcase* berada dibawah tekanan ambien.
- f. Ketika piston pada  $10-40^\circ$  sebelum TMA, muatan akan dikompresi hingga bertemperatur tinggi. Kemudian pembakaran akan mengikuti pengapian dari busi. Proses ini dilakukan oleh mesin yang menggunakan campuran udara dan bahan bakar. Daya *stroke* dimulai ketika piston memukul TMA terus menerus hingga *exhaust port* terbuka, seperti pada langkah a.



**Gambar 2.5** Prinsip Kerja ICE 2 Tak



CO - Crankcase port opens  
 CC - Crankcase port closes  
 EO - Exhaust port opens  
 EC - Exhaust port closes  
 IO - Inlet port opens  
 IC - Inlet port closes  
 IGN - Ignition  
 tdc - Top dead center  
 bdc - Bottom dead center

**Gambar 2.6** Langkah Pembakaran ICE 2 Tak

### 2.3 Motor Listrik[5]

Motor merupakan mesin yang mengubah energi listrik menjadi energi gerak atau mekanik. Terdapat dua jenis motor listrik, yaitu motor AC dan motor DC. Pada subbab ini hanya motor DC yang akan dibahas.

Motor DC merupakan salah satu mesin yang mengubah energi listrik yang berupa tegangan DC ke energi mekanik. Awalnya mesin ini dibuat dan diuji oleh Michael Faraday yang merumuskan konsep elektromagnetik.

Desain Faraday dikembangkan dengan cepat, banyak mesin DC yang dibuat pada tahun 1880 ketika DC merupakan bentuk utama dari pembangkit tenaga listrik. Dengan munculnya AC 60Hz sebagai standar energi listrik di *United States* dan 50Hz di Eropa. Dengan

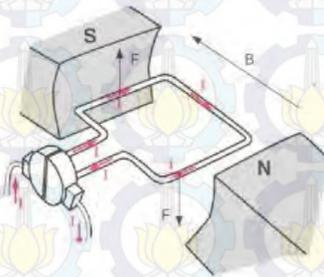
penemuan motor induksi dengan biaya produksi yang rendah, penggunaan motor DC menjadi berkurang. Dalam beberapa tahun terakhir, penggunaan motor DC meningkat. Hal ini disebabkan karena karakteristik motor dc yang unik contohnya torsi starting yang tinggi untuk aplikasi traksi motor dan hampir sebagian besar peralatan yang mudah dibawa bekerja dengan sumber tegangan DC (seperti baterai).

Jika arus listrik melalui konduktor dan diletakkan pada medan magnet, terdapat gaya yang bekerja di atasnya. Nilai gaya adalah fungsi kerapatan fluks magnetik ( $B$ ), arus listrik ( $I$ ), dan vektor arah medan dan arus. Gaya pada konduktor dapat ditulis dengan persamaan

$$F = BLI [N: T, m, A] \quad (2.1)$$

di mana  $L$  adalah panjang konduktor. Dari persamaan diatas dapat diketahui bahwa semakin besar arus dalam kawat atau semakin besar medan magnet maka semakin kawat akan bergerak semakin cepat.

Arah perputaran dari motor DC berdasarkan pada kaidah tangan kiri yang berbunyi “Apabila tangan kiri diletakkan di antara kutub utara dan selatan, sehingga garis gaya yang memiliki arah sesuai jari telunjuk dan arus di dalam kawat mengalir searah dengan jari tengah, maka kawat itu akan mendapat gaya yang arahnya sesuai dengan arah ibu jari.”. Dari kaidah ini arah putaran motor DC dapat ditunjukkan pada Gambar 2.7



**Gambar 2.7** Arah Putaran Motor DC

Pada dasarnya, Motor DC terdiri dari kumparan jangkar yang disebut rotor dan kumparan jangkar yang disebut stator. Stator terletak pada luar bagian dari motor. Biasanya stator terdiri dari dua atau lebih potongan tiang magnet permanen atau elektromagnet. Rotor merupakan bagian dalam motor yang berotasi. Rotor dibentuk dari lilitan jangkar

yang terhubung ke rangkaian eksternal melalui komutator. Stator dan rotor terbuat dari bahan feromagnetik dan keduanya dipisahkan oleh gap udara.

Motor DC diklasifikasikan berdasarkan rangkaian listrik dari kumparan jangkar dan kumparan medan. Perbedaan rangkaian ini mengakibatkan mesin beroperasi dengan karakteristik yang berbeda. Kumparan medan dapat melalui penguatan sendiri atau penguatan terpisah, yaitu lilitan terhubung dengan tegangan *input* atau disuplai dari sumber tegangan terpisah. Penguatan sendiri terjadi jika lilitan terhubung langsung dengan terminal sumber tegangan *input* baik secara seri atau paralel.

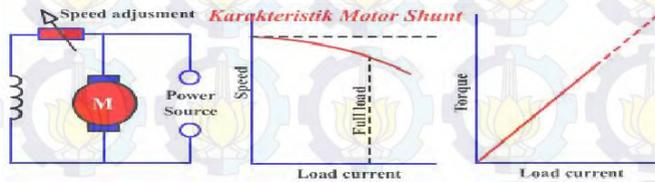
Motor DC seri. Ketika kumparan medan dan jangkar terhubung secara seri dengan sumber tegangan. Pada motor ini medan magnet dihasilkan oleh arus yang melalui kumparan jangkar, hasilnya medan magnet lemah ketika beban motor ringan (kumparan jangkar mengalirkan sedikit arus). Medan magnet akan kuat saat beban berat (kumparan jangkar mengalirkan arus maksimum). Akibatnya kecepatan motor DC seri ditentukan oleh arus beban. Kecepatan rendah pada beban berat dan sangat cepat saat tanpa beban. Sehingga saat beroperasi tanpa beban, motor akan beroperasi dengan cepat dan akan merusak komponen didalamnya. Torsi pada motor DC tergantung pada perkalian arus jangkar dan medan magnet. Sehingga torsi akan besar pada arus jangkar tinggi dan pada awal pengaktifan. Motor seri dapat beradaptasi dengan beban inersia yang berat dan berguna sebagai penggerak pada bus listrik, kereta, dan aplikasi traksi beban berat. Dibandingkan dengan motor shunt, motor DC seri mempunyai torsi starting tinggi dan regulasi kecepatan rendah. Rangkaian dan kurva karakteristik motor DC seri ditunjukkan pada Gambar 2.8.



**Gambar 2.8** Rangkaian dan Kurva Karakteristik Motor DC Seri

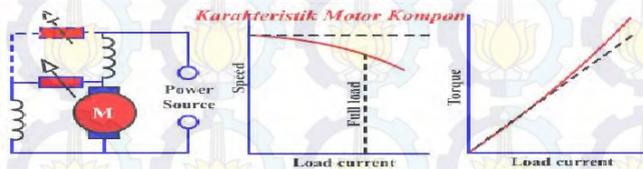
Motor DC shunt. Ketika kumparan medan dan jangkar dihubungkan secara shunt atau paralel terhadap sumber tenaga. Dengan itu tegangan yang melewati dua kumparan sama tetapi kumparan

jangkar mengalirkan arus lebih dari kumparan medan. Torsi dihasilkan dari interaksi arus pada kumparan jangkar dengan medan magnet yang dihasilkan kumparan medan. Jika tegangan dc konstan, tegangan jangkar dan tegangan jangkar konstan. Regulasi kecepatan cukup baik, kecepatan adalah fungsi dari arus jangkar dan tidak selalu konstan. Jangkar berotasi dalam medan magnet, EMF diinduksi. EMF memiliki arah yang bertolak belakang dengan sumber EMF and disebut counter EMF (CEMF), yang mengakibatkan perubahan kecepatan putar. Arus yang melalui kumparan jangkar adalah selisih antara EMF dan CEMF. Ketika beban meningkat, kecepatan motor menurun dan sedikit CEMF diinduksi, yang terus meningkatkan arus jangkar dan meningkatkan torsi untuk kenaikan beban. Rangkaian dan kurva karakteristik motor DC shunt ditunjukkan pada Gambar 2.9.



**Gambar 2.9** Rangkaian dan Kurva Karakteristik Motor DC Shunt

Motor DC kompon. Motor dengan menggabungkan karakteristik operasi dari motor seri dan paralel. Kemampuan torsi yang tinggi dari motor DC seri yang mengakibatkan kecepatan yang berlebihan. Kekurangan itu dapat dikurangi dengan penambahan medan shunt, dihubungkan sedemikian hingga membantu medan seri. Motor dapat menjadi mesin kompon kumulatif. Rangkaian dan kurva karakteristik motor DC kompon ditunjukkan pada Gambar 2.10.



**Gambar 2.10** Rangkaian dan Kurva Karakteristik Motor DC Kompon

## 2.4 Rem Elektromagnetik [6]

Ketika fluks magnet melalui konduktor, arus eddy dibangkitkan pada konduktor tersebut. Arus eddy bersirkulasi dalam konduktor menghasilkan medan magnet yang memiliki polaritas yang berlawanan. Interaksi dari dua medan magnet menyebabkan gaya yang menghambat perubahan fluks magnet. Namun karena hambatan dalam dari material konduktif, arus eddy akan didisipasikan menjadi panas dan gaya akan berkurang. Selama arus eddy didisipasi, energi dipindahkan dari sistem dan mengakibatkan efek peredaman.

Salah satu sifat yang berguna pada peredaman arus eddy adalah memindahkan energi sistem tanpa bersentuhan langsung dengan struktur. ada 2 tipe dasar dari rem arus eddy yaitu rem arus eddy sirkular dan rem arus eddy linier

### a. Rem arus eddy sirkular.

Rem elektromagnetik ini sama dengan motor listrik, *disk* metal non feromagnetik (rotor) dihubungkan pada kumparan putar dan medan magnet antara rotor dan kumparan membentuk hambatan yang digunakan untuk menghasilkan listrik atau panas. Ketika menggunakan elektromagnet, pengaturannya aksi pengereman dibuat dengan memvariasikan kekuatan dari medan magnet. Gaya pengereman terjadi ketika arus listrik melalui elektromagnet. Pergerakan metal melalui medan magnet menghasilkan arus eddy pada *disk*. Arus eddy menimbulkan medan magnet yang berlawanan yang kemudian menghambat perputaran disk.

### b. Rem arus eddy linier

Rem ini terdiri dari beban magnetik dengan kumparan listrik yang diletakkan sepanjang lintasan beban magnetik ini menjadi magnet alternatif kutub utara dan selatan. Magnet ini tidak menyentuh lintasan, tetapi berjarak cukup kecil dari lintasan. Ketika magnet digerakkan melalui lintasan, akan menghasilkan medan magnet non stasioner pada ujung lintasan yang menghasilkan tagangan listrik (hukum induksi faraday) dan menyebabkan arus eddy. Dengan gangguan dari medan magnet, medan magnet memberikan gaya yang berlawanan sehingga menciptakan gaya horizontal yang melawan gerak magnet.

Rem elektromagnet biasa diletakkan dekat dengan bagian yang bergerak. Rem ini bekerja pada kondisi yang dingin dan memenuhi persyaratan energi pengereman kecepatan tinggi karena tanpa adanya

gesekan. Poros baling baling dibedakan dan disesuaikan dengan *sliding universal joint* dan dihubungkan ke *coupling flange* pada rem, rem disesuaikan dengan sasis kendaraan dan dengan pemasangan anti getar. Letak rem dalam kendaraan mencengah tumbukkan langsung dari udara yang disebabkan oleh gerakan kendaraan. Setiap aliran udara dalam sasis memiliki efek yang besar pada lairan udara sekitar ban dan suhu cakram depan dan belakang. Bentuk rem elektromagnetik ditunjukkan pada Gambar 2.11



**Gambar 2.11** Rem Elektromagnetik

## 2.5 Sistem Penyesuai Kecepatan [1][2]

Dalam mengkombinasikan dua mesin penggerak yang memiliki *range* kerja yang berbeda, maka dibutuhkan suatu sistem untuk mengubah kecepatan sudut dan torsi. Gir merupakan alat yang digunakan untuk mentransmisikan gaya dan momen. Secara fisik, gir berbentuk lingkaran dengan gerigi pada bagian luarnya.

Ada 2 jenis konfigurasi gir yang biasa digunakan yaitu gir beda poros dan gir satu poros. Bentuk konfigurasi gir beda poros (*reduction gear*) ditunjukkan pada Gambar.2.12. Pada penggunaannya konfigurasi gir ini biasa dikombinasikan dengan *timing belt* karena jika terhubung secara langsung membutuhkan dimensi tempat yang berdekatan. Berikut adalah aturan yang berlaku pada gir yang berbeda poros.

$$V_A = V_B \quad (2.2)$$

$$\omega_A R_A = \omega_B R_B \quad (2.3)$$

$$\frac{\omega_A}{\omega_B} = \frac{R_B}{R_A}$$

$$\frac{\omega_B}{\omega_A} = \frac{R_A}{R_B}$$

$$F_A = F_B \quad (2.4)$$

$$\frac{\tau_A}{R_A} = \frac{\tau_B}{R_B} \quad (2.5)$$

$$\frac{\tau_A}{\tau_B} = \frac{R_A}{R_B}$$

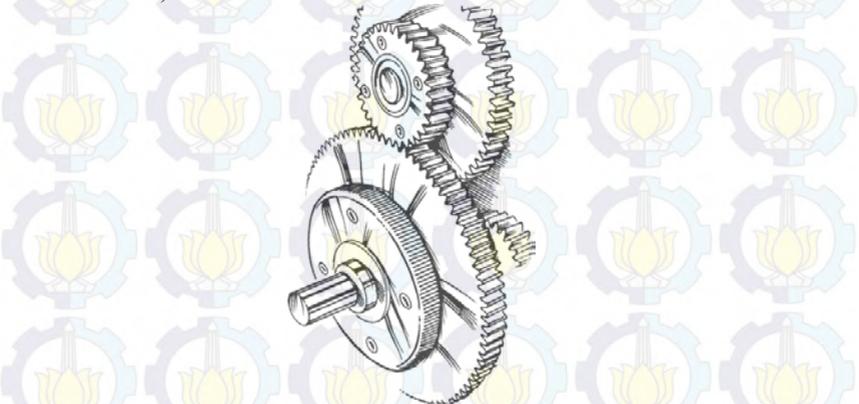
$$\frac{\tau_B}{\tau_A} = \frac{R_B}{R_A}$$

dengan  $V$ ,  $R$ ,  $\omega$ ,  $F$ , dan  $\tau$  adalah kecepatan tangensial gir (m/s), jari jari gir (m), kecepatan sudut (rad/s), gaya tagensial (N), dan torsi (Nm).



**Gambar 2.12** Konfigurasi Gir Beda Poros

Konfigurasi gir yang kedua adalah gir satu poros (*compound gear*). Gir memiliki kecepatan yang sama pada porosnya dan biasa digunakan untuk menghubungkan dua gir agar mengurangi dimensi gir dan tetap dapat menjalankan fungsi yang sama dalam mengubah kecepatan atau torsi. Konfigurasi gir satu poros dapat dilihat pada Gambar 2.13,



**Gambar 2.13** Konfigurasi Gir Satu Poros

## 2.6 Tachogenerator

Alat ini berupa generator elektromekanik yang dapat memproduksi energi listrik dari energi mekanik. Generator akan menghasilkan tegangan yang proporsional dengan kecepatan batang berputar. Dengan desain dan konstruksinya, tachogenerator dapat

menghasilkan tegangan untuk *range* kecepatan tertentu, sehingga alat ini cocok untuk pengukuran kecepatan putar peralatan mekanik.

Tachogenerator juga dapat menunjukkan arah putaran dengan melihat polaritas dari tegangan *output*. Ketika magnet permanen pada tachogenerator berputar dengan arah yang berkebalikan, polaritas tegangan *output* akan berpindah.

Jangkar tachogenerator (rotor) ini terhubung dengan mesin penggerak yang kecepataannya akan dideteksi. Kumparan jangkar berputar pada medan magnet permanen, tegangan AC yang diinduksi dari kumparan jangkar. Tegangan ini akan dialirkan pada komutator dengan *brush* khusus dan pada prosesnya merubah putaran ke tegangan DC.

Tachogenerator DC menghasilkan tegangan dengan *ripple* yang kecil, frekuensi dan amplitudo tergantung pada kecepataannya, jumlah kutub magnet, jumlah slot jangkar dan jumlah segmen komutator. Kesalahan instalasi tachogenerator pada mesin penggerak dapat meningkatkan *ripple*.

## 2.7 Arduino Uno[7]

Arduino Uno adalah *board* mikrokontroler berbasis Atmega328. Alat ini mempunyai 14 pin *input/output* digital dan 6 diantaranya dapat digunakan sebagai *output* PWM, 6 *input* analog, resonator keramik 19MHz, koneksi USB, soket listrik, ICSP *header*, dan tombol reset. Dengan kabel USB alat ini mudah dihubungkan ke komputer dan dapat diaktifkan dengan baterai atau adaptor AC ke DC. Spesifikasi Arduino Uno adalah sebagai berikut :

- Mikrokontroler : Atmega328
- Tegangan operasi : 5V
- Tegangan *input* : 7-12V  
(direkomendasikan)
- Tegangan *input* (batasan) : 6-20V
- Pin *input/output* digital : 14 (6 diantaranya *output* PWM)
- Pin *input* analog : 6
- Arus DC tiap pin I/O : 40mA
- Arus DC untuk pin 3,3 V : 50mA

14 pin *Input* dan *output* Arduino Uno dapat digunakan dengan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Tiap pin memiliki tegangan operasi 5V dan dapat menerima arus maksimum 40mA. Beberapa pin memiliki fungsi khusus

- a. *Serial* : 0(RX) dan 1(TX). Digunakan untuk menerima (RX) dan mengirim (TX) data serial TTL. Pin ini terhubung dengan pin pada Atmega8U2 USB ke cip USB ke TTL serial
- b. *External Interrupts*: 2 dan 3.pin ini digunakan sebagai trigger gangguan pada nilai yang rendah, menaikkan dan menurunkan nilai, atau merubah nilai.
- c. PWM: 3,5,6,9,10, dan 11. Menyediakan *output* PWM 8bit dengan fungsi *analogWrite()*.
- d. SPI (*Serial Peripheral Interface*): 10 (*Slave Select*), 11(*Master Out Slave In*), 12(*Master In Slave Out*), 13(*Serial Clock*). Pin ini mendukung komunikasi SPI.
- e. LED :13, LED ini terhubung dengan pin 13. Ketika pin memiliki nilai yang tinggi LED akan *on* dan ketika pin memiliki nilai yang rendah, LED akan *off*

Pada Arduino Uno terdapat 6 *input* analog, dengan nama A0 hingga A5. Pin tersebut memiliki resolusi 10bits (1024 nilai yang berbeda). *Input* analog ini berupa tegangan dari ground ke 5V, walaupun begitu terdapat kemungkinan untuk mengubah batas atas dan bawah dengan menggunakan pin AREF dan fungsi *analogReference()*.

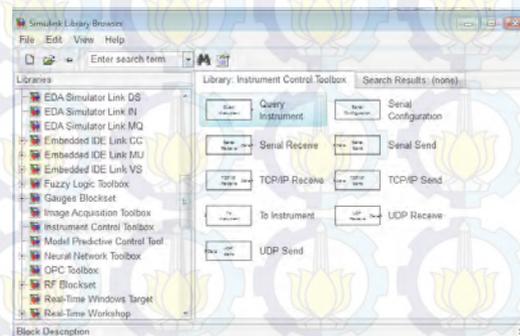
## 2.8 MATLAB2010a[8]

MATLAB2010a merupakan paket program dengan bahasa pemrograman yang tinggi untuk mengembangkan algoritma, visualisasi data, dan komputasi numerik. Program MATLAB ini dapat digunakan untuk menyelesaikan masalah komputasi dengan lebih cepat dibandingkan dengan bahasa pemrograman tradisional, seperti C, C++, dan Fortran. MATLAB digunakan untuk banyak aplikasi seperti *signal and image processing*, desain kontrol, pengujian dan pengukuran, permodelan, dan analisis.

Simulink merupakan bagian dari MATLAB untuk memodelkan, mensimulasikan, dan menganalisa sistem dinamik. Simulink dapat membentuk model dari awal atau memodifikasi model yang sudah ada sesuai dengan apa yang diinginkan. Selain itu simulink juga mendukung sistem linier dan non linier, permodelan waktu kontinyu atau diskrit, atau gabungan. Simulink ini dapat digunakan sebagai media untuk menyelesaikan masalah dalam industri nyata meliputi kedirgantaraan

dan pertahanan, otomotif, komunikasi, elektronik dan pemrosesan sinyal,

Salah satu modul dalam simulink yang dapat digunakan untuk komunikasi perangkat keras adalah *Instrument Control Toolbox*, seperti terlihat pada Gambar 2.14. Modul ini merupakan kumpulan fungsi M-file yang dibangun dilingkungan komputasi teknis MATLAB. *Toolbox* ini menyediakan kerangka kerja untuk komunikasi instrumen yang mendukung GPIB *interface*, standar VISA, TCP/IP, dan protokol UDP. *Toolbox* ini memperluas fitur dasar *serial port* yang ada dalam MATLAB. Selain itu *toolbox* ini berfungsi untuk komunikasi data antara workspace MATLAB dan peralatan lainnya. Data tersebut dapat berbentuk biner atau *text*.



**Gambar 2.14** *Instrument Control Toolbox* [8]

Komunikasi serial merupakan protokol dasar tingkat rendah untuk komunikasi antara dua peralatan atau lebih. Pada umumnya satu komputer dengan modem, printer, mikrokontroler, atau peralatan lainnya. *Serial port* mengirim dan menerima informasi *bytes* dengan hubungan seri. *bytes* tersebut dikirimkan menggunakan format biner atau karakter ASCII (*American Standard Code for Information Interchange*). Dalam komunikasi serial MATLAB, agar data ASCII dapat diproses *real time*, maka digunakan ASCII *encode* dan *decode* yang terdapat pada *xPC Target Library for RS232*. ASCII *encode* merupakan blok dalam simulink yang digunakan untuk mengubah data *bytes* menjadi karakter ASCII. Sedangkan ASCII *decode* merupakan blok simulink yang digunakan untuk mengubah karakter ASCII menjadi data *bytes* yang kemudian dapat dikonversi sesuai kebutuhan.

## 2.9 Permodelan Sistem

Dalam dunia industri segala sesuatu harus dikontrol agar berjalan dengan aman dan efisien. Contohnya untuk mendesain regulator, dibutuhkan model dari proses, terdapat beberapa bentuk model dari sistem, antara lain grafik, tabel, dan model matematika. Diagram bode dari sistem merupakan salah satu contoh bentuk grafik dan contoh model matematika adalah persamaan diferensial dan persamaan beda. Model matematika ini cocok digunakan untuk analisa, prediksi, dan desain sistem, regulator, dan filter.

Model matematika dari sebuah sistem dapat diperoleh melalui dua cara yaitu permodelan fisik dan identifikasi sistem. Permodelan fisik merupakan pendekatan analitik berdasarkan hukum fisika seperti hukum newton dan hukum kesetimbangan. Permodelan ini menjelaskan dinamika dalam sistem. Yang kedua adalah identifikasi sistem. Hal ini dilakukan dengan pendekatan eksperimental. Model ini berdasarkan data dari eksperimen yang kemudian didapatkan nilai parameter sistem. Pada beberapa kasus yang sangat kompleks, sangat sulit untuk menentukan model berdasarkan pemahaman fisik.[9]

### 2.9.1 Metode Viteckova[10]

Metode Viteckova dapat didekati dengan model orde satu dan orde dua dengan delay. Fungsi alih orde 1 dapat dituliskan sebagai berikut

$$G_{v1}(s) = \frac{K}{\tau_{v1}s + 1} e^{-T_{dv1}s} \quad (2.6)$$

Dengan

$$T_{dv1} = 1,498t_{33} - 0,498t_{70}$$

$$\tau_{v1} = 1,245(t_{70} - t_{33})$$

Fungsi alih orde dua dapat dituliskan sebagai berikut

$$G_{v2}(s) = \frac{K}{(\tau_{v2}s + 1)^2} e^{-T_{dv2}s} \quad (2.7)$$

Dengan

$$T_{dv2} = 1,937t_{33} - 0,937t_{70}$$

$$\tau_{v1} = 0,794(t_{70} - t_{33})$$

### 2.9.2 Metode Latzel[10]

Metode Latzel merupakan metode yang berdasarkan pendekatan sistem dengan fungsi alih sebagai berikut

$$G_L(s) = \frac{K}{(\tau_L s + 1)^n} \quad (2.8)$$

di mana

$$\tau_L = \alpha_{10}t_{10} + \alpha_{50}t_{50} + \alpha_{90}t_{90}$$

Dengan  $t_{10}$ ,  $t_{50}$ ,  $t_{90}$  adalah respons step ketika mencapai 10%, 50% dan 90% dari nilai akhir.

Paramater untuk menentukan permodelan dengan metode Latzel salah satunya adalah  $\mu$  yang ditemukan dengan persamaan berikut

$$\mu = \frac{t_{10}}{t_{90}} \quad (2.9)$$

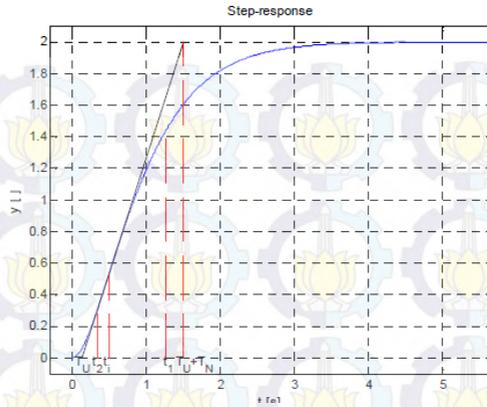
Kemudian dientukan parameter  $\mu_a$  yang mendekati parameter  $\mu$ , orde faktor n, konstanta  $\alpha_{10}$ ,  $\alpha_{50}$ ,  $\alpha_{90}$  dapat ditemukan pada Tabel 2.1

**Tabel 2.1** Tabel Parameter Latzel

$\mu_a$	n	$\alpha_{10}$	$\alpha_{50}$	$\alpha_{90}$	$\mu_a$	n	$\alpha_{10}$	$\alpha_{50}$	$\alpha_{90}$
0,137	2	1,880	0,596	0,257	0,456	11	0,142	0,094	0,065
0,174	2,5	1,245	0,460	0,216	0,472	12	0,128	0,086	0,060
0,207	3	0,907	0,374	0,188	0,486	13	0,116	0,079	0,056
0,261	4	0,573	0,272	0,150	0,499	14	0,106	0,073	0,053
0,304	5	0,411	0,214	0,125	0,512	15	0,097	0,068	0,050
0,340	6	0,317	0,176	0,108	0,523	16	0,090	0,064	0,047
0,370	7	0,257	0,150	0,095	0,533	17	0,084	0,060	0,045
0,396	8	0,215	0,130	0,085	0,543	18	0,078	0,057	0,042
0,418	9	0,184	0,115	0,077	0,552	19	0,073	0,054	0,040
0,438	10	0,161	0,103	0,070	0,561	20	0,069	0,051	0,039

### 2.9.3 Identifikasi Strejc [10]

Metode Strejc merupakan metode untuk identifikasi *plant* dengan cara menarik garis singgung pada titik belok respons step *plant* dalam domain waktu. Penarikan garis ini ditunjukkan pada Gambar 2.15. Metode ini diterapkan terhadap *plant* orde tinggi yang memiliki respons *overdamped* ( $\zeta > 1$ ) dan respons sistem memiliki *time delay* sebelum mencapai *steady state*.



**Gambar 2.15** Penarikan Garis pada Metode Strejc

Pada metode strejc penarikan garis berdasar pada titik belok (inflection point) dari grafik respons. Dengan ini didapatkan nilai  $T_u$  dan  $T_N$ .  $T_u$  merupakan waktu tunda sistem dan  $T_N$  merupakan waktu saat respons melewati waktu tunda hingga mencapai batas penarikan garis terhadap titik belok respons. Nilai  $T_u$  dan  $T_N$  ini berpengaruh terhadap orde pant dan untuk memodelkan respons *plant* berdasarkan tabel Strejc. Dengan diketahui nilai  $T_u$  dan  $T_N$ , maka dapat dicari nilai  $\tau$ ,

$$\tau = \frac{T_u}{T_N} \tag{2.10}$$

Ketika nilai  $\tau < 0$ , dilakukan pendekatan orde 2 dengan persamaan

$$G_{ST1}(s) = \frac{K}{(\tau_{ST1}s + 1)(\tau_{ST2}s + 1)} \tag{2.11}$$

Nilai  $\tau_{ST1}$  dan  $\tau_{ST2}$  ditentukan dengan menemukan nilai  $t_1$  dan  $t_2$ .  $t_1$  adalah waktu saat respons sistem mencapai  $0.72y_\infty$ . Dengan diketahui nilai  $t_1$ , maka didapatkan nilai :

$$\tau_{ST1} + \tau_{ST2} = \frac{t_1}{1,2564} \tag{2.12}$$

dengan nilai  $\tau_{ST1} + \tau_{ST2}$ , dapat diketahui nilai dari  $t_2$

$$t_2 = 0,3574(\tau_{ST1} + \tau_{ST2}) \quad (2.13)$$

Setelah mengetahui nilai  $t_2$ , maka nilai  $y(t_2)$  dapat diketahui dengan melihat respons sistem. Dengan melihat Tabel 2.2 dapat diketahui nilai T. Setelah mendapatkan nilai T, maka dapat diketahui nilai

$$T = \frac{\tau_{ST1}}{\tau_{ST2}} \quad (2.14)$$

Untuk  $\tau \geq 0$  sistem didekati dengan orde ke n dengan persamaan

$$G_{ST2}(s) = \frac{K}{(\tau_{ST} s + 1)^n} \quad (2.15)$$

**Tabel 2.2** Nilai Konstanta T

$y(t_2)$	T	$y(t_2)$	T	$y(t_2)$	T
0,3	0	0,25	0,105	0,2	0,264
0,29	0,023	0,24	0,128	0,19	0,322
0,28	0,043	0,23	0,154	0,18	0,403
0,27	0,063	0,22	0,183	0,17	0,538
0,26	0,084	0,21	0,219	0,16	1

Berdasarkan nilai  $\tau$  dapat ditentukan nilai  $y_t$  berdasarkan nilai yang ada pada Tabel 2.3. Dengan nilai  $y_t$ , maka nilai  $t_i$  dapat ditemukan dengan melihat respons sistem. Jika nilai  $t_i$  sudah diketahui maka dapat ditemukan nilai  $\tau_{ST}$ .

$$\tau_{ST} = \frac{t_i}{n - 1} \quad (2.16)$$

**Tabel 2.3** Parameter Identifikasi Metode Strejc

n	$\tau$	$y_t$
2	0,104	0,264
3	0,218	0,327
4	0,319	0,359
5	0,41	0,371
6	0,493	0,384
7	0,57	0,394
8	0,642	0,401
9	0,709	0,407
10	0,773	0,413

### 2.9.4 Root Mean Square Error

Persamaan permodelan digunakan pada banyak bidang ilmu, di mana variabelnya harus diukur dengan ada tidaknya *error*. Tujuannya adalah agar nilai pada permodelan mendekati nilai real sistem.

Untuk melakukan validasi model dapat digunakan beberapa tolok ukur. Salah satunya adalah RMSE. Dalam penggunaannya, harus ditentukan terlebih dahulu jumlah data dan interval toleransi RMSE. Jika nilai RMSE kecil, menunjukkan bahwa nilai parameter sesuai dengan apa yang diinginkan sebelumnya.

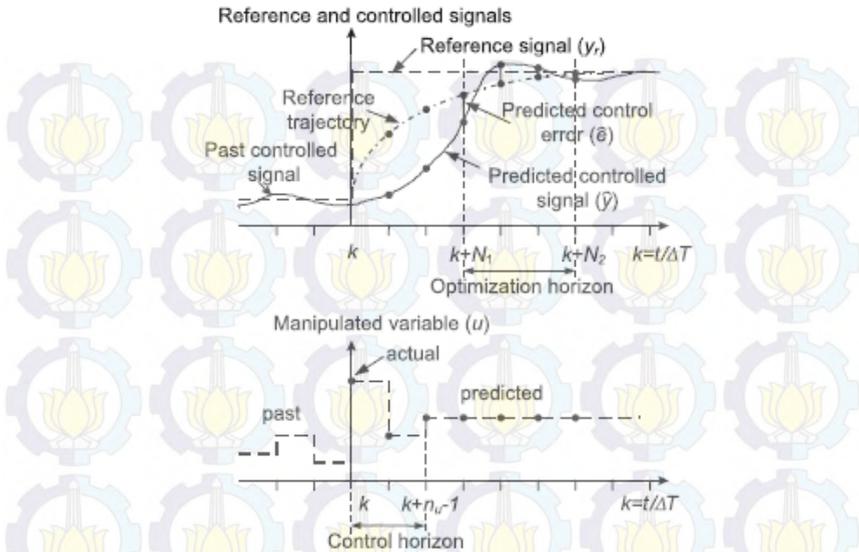
Dengan  $e$  adalah *error* antara nilai estimasi dan nilai sebenarnya dan  $n$  adalah jumlah data.

$$RMSE = \sqrt{\frac{\sum e^2}{n}} \quad (2.17)$$

### 2.10 Kontrol Prediktif [11]

Kontrol prediktif berdasarkan model merupakan metode baru pada teknik pengaturan. Pada dasarnya metode ini memperhitungkan dalam optimasi variabel, tidak pada saat ini tapi juga nilai dimasa mendatang. Tujuan yang ingin dicapai yaitu memilih variabel dan mensimulasikan tujuan proses. Jika sinyal kontrol tidak menghasilkan respons yang memuaskan, maka urutan variabel yang dimanipulasi akan mengubah aksi kontrol hingga mencapai hasil yang sesuai. Terdapat perbedaan yang mendasar antara kontrol prediktif dan kontrol konvensional. Kontroller konvensional hanya mengamati variabel proses pada saat ini dan melihat pada masa lalu. Sedangkan kontroler prediktif dapat mengamati variabel proses saat ini dan masa depan serta juga melihat variabel di masa lampau.

Kontrol prediktif dapat menjalankan beberapa tugas. Pertama kontrol prediktif dapat meminimalkan kesalahan kontrol beberapa langkah kedepan dari saat ini (antara  $k+N1$  dan  $k+N2$ ). Kedua yaitu menghilangkan nilai kontrol beberapa langkah kedepan ( $n-1$ ) dari saat ini. Dan ketiga yaitu memperhitungkan keterbatasan dalam kontrol dan variabel dalam perhitungan. Gambar 2.16 menunjukkan nilai yang dimanipulasi, referensi, dan sinyal kontrol dalam pengaturan.



**Gambar 2.16** Sinyal yang Dimanipulasi, Referensi, dan Sinyal Kontrol melalui Aksi Kontrol

Dengan  $u$  adalah variabel yang dimanipulasi,  $y$ ,  $y_r$  adalah sinyal referensi dan kontrol,  $\hat{y}$  adalah sinyal kontrol prediktif,  $t$ ,  $k$  adalah waktu kontinu dan diskrit,  $\Delta T$  adalah waktu sampling, dan  $d$  adalah *dead time* relatif terhadap waktu sampling.

Ada beberapa kriteria yang dapat dicapai kontrol prediktif yaitu kontrol cepat, kontrol cepat tanpa *overshoot* pada sinyal kontrol, dan kontrol tersepat dengan keterbatasan sinyal dimanipulasi. Kriteria lain pada kontrol prediktif adalah dengan meminimalkan *cost function* kuadrat dari *error* kontrol dan kenaikan variabel manipulasi sesuai dengan *prediction horizon*.

*Cost function* kuadrat dari kasus SISO adalah

$$J = \sum_{i=N_1}^{N_2} \lambda_{y_i} [y_r(k+i) - \hat{y}(k+i|k)]^2 + \sum_{j=1}^{n_u} \lambda_{u,j} \Delta u^2(k+j-1) \rightarrow \min_{\Delta u}$$

(2.18)

dengan  $y_r(k+i|k)$  merupakan sinyal referensi  $i$  langkah kedepan dari waktu saat ini,  $\hat{y}(k+i|k)$  merupakan sinyal *output* prediksi (dan terkontrol)  $i$  langkah kedepan, dan  $\Delta u(k+1)$  merupakan kenaikan *output* terkontrol  $i$  langkah kedepan. Di mana  $(k+i|k)$  menunjukkan sinyal akan datang yang diprediksi pada basis informasi yang tersedia hingga waktu tertentu.

Parameter yang akan ditala pada algoritma kontrol adalah

- $N_1$  yang merupakan titik pertama dari *prediction horizon* berdasarkan waktu saat ini
- $N_2$  yang merupakan titik terakhir dari *prediction horizon* berdasarkan waktu saat ini
- $n_u$  yang merupakan panjang *control horizon* (jumlah perubahan yang dikehendaki berturut turut dalam sinyal kontrol)
- $\lambda_{y,N_1}, \dots, \lambda_{y,N_2}$  yang merupakan faktor pembebanan dari *error* kontrol, biasanya diasumsikan bernilai 1 ( $\lambda_y = 1$  pada kasus SISO).
- $\lambda_{u1}, \dots, \lambda_{u,n_u}$  yang merupakan faktor pembebanan dari kenaikan kontrol, biasanya diasumsikan bernilai sama (dan didefinisikan dengan  $\lambda_u$ ).

Variabel manipulasi dipengaruhi oleh sinyal kontrol hanya setelah dead time, nilai pertama dan terakhir dari *error* kontrol horizon disebut  $n_{u1}$  dan  $n_{u2}$ . Dengan nilai  $n_{u1} = N_1 - d - 1$  dan  $n_{u2} = N_2 - d - 1$ .

Dengan dinotasikan kembali menjadi

$$J = \sum_{n_c=n_{c1}}^{n_{c2}} [y_r(k+d+1+n_c) - \hat{y}(k+d+1+n_c|k)]^2 + \sum_{j=1}^{n_u} \lambda_u \Delta u^2(k+j-1) \rightarrow \min_{\Delta u_f} \quad (2.19)$$

*Cost function* mengandung dua bagian yaitu *cost* untuk mengontrol *error* selama horizon *error* kontrol biasa disebut optimasi atau *prediction horizon* dan *cost* untuk menghilangkan kenaikan sinyal kontrol selama horizon variabel manipulasi yang disebut *control horizon*.

## 2.11 Generalized Predictive Control (GPC) [11][12]

GPC yang telah diperkenalkan oleh Clarke *et al* [12] telah banyak digunakan pada proses dalam industri. Hal ini disebabkan karena

kemampuan pengaturan yang stabil untuk sistem dengan parameter, *dead time*, dan orde model yang berubah ubah.

Ketika memperhatikan titik operasi, *plant* non linier umumnya menggunakan model linier pada daerah tertentu

$$A(q^{-1}) = B(q^{-1})u(t - 1) + x(t) \quad (2.20)$$

dengan  $u(t)$  *input* kontrol,  $y(t)$  variabel yang terukur atau *output*, dan  $x(t)$  gangguan. Di mana A dan B adalah polinomial

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + b_{nb}q^{-nb} \end{aligned}$$

Jika *plant* tidak mempunyai *dead time* maka polinomial B bernilai nol.  $x(t)$  dianggap rata rata bergerak dengan

$$x(t) = C(q^{-1})\xi(t) \quad (2.21)$$

di mana

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc}$$

Pada persamaan ini,  $\xi(t)$  adalah rangkaian nilai acak berkorelasi. Dengan menggabungkan Persamaan 2.20 dan 2.21 maka akan diperoleh model CARMA (*Controlled Auto-Regresive Moving Average*),

$$A(q^{-1}) = B(q^{-1})u(t - 1) + C(q^{-1})\xi(t) \quad (2.22)$$

Walaupun sebagian besar teori *self tuning* berdasarkan pada model ini. Teori ini tidak tepat untuk banyak aplikasi industri yang gangguanya tidak seimbang. Pada prakteknya terdapat dua jenis gangguan yang sering ditemui yaitu nilai acak pada waktu yang acak (contohnya perubahan kualitas material) dan *brownian motion* (ditemukan pada *plant* yang mengandalkan keseimbangan energi). Pada kedua kasus ini model yang sesuai yaitu

$$x(t) = C(q^{-1}) \frac{\xi(t)}{\Delta} \quad (2.23)$$

di mana  $\Delta$  adalah operator differensial  $1 - q^{-1}$ . Jika digabung dengan Persamaan 2.10 akan menghasilkan model CARIMA (*Controlled Auto-Regresive Integrated Moving Average*).

$$A(q^{-1}) = B(q^{-1})u(t - 1) + C(q^{-1})\frac{\xi(t)}{\Delta} \quad (2.24)$$

Sinyal gangguan  $\xi(t)$  berupa sinyal deterministik atau stokastik. Pada perkembangan GPC,  $\xi(t)$  diasumsikan berupa sinyal stokastik.

Dengan model linier, yang membatasi kriteria kontrol optimal kuadrat diajukan dalam kenaikan *input output plant*. Fungsi harga yang diminimumkan adalah

$$J(N_1, N_2) = E\left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j) [\Delta u(t+j-1)]^2 \right\} \quad (2.25)$$

di mana  $N_1$  adalah *costing horizon* minimal,  $N_2$  adalah *costing horizon* maksimal,  $\lambda(j)$  adalah urutan kontrol pembebanan.

Fungsi harga diatas dikondisikan pada data pada waktu t dengan asumsi sinyal kontrol diterapkan dalam loop terbuka. Kontrol pertama diterapkan dan diminimalisasi hingga sampel berikutnya. Hukum kontrol yang dihasilkan dikenal sebagai *open loop feedback optimal control* dapat dilihat bahwa fungsi harga dipengaruhi oleh seluruh *input* masa depan yang mempengaruhi *output*.

Sinyal *output* prediksi  $\hat{y}(k+d+1+n_c)$  dituliskan sebagai penjumlahan dari *free response* dan *force response*. Persamaan *force response* adalah sebagai berikut :

$$\begin{aligned} \hat{y}_{force}(k+d+1+n_c|k) &= \frac{B(q^{-1})}{A(q^{-1})} u(k+1+n_c|k) \\ &= \frac{B(q^{-1})}{(1-q^{-1})A(q^{-1})} (1-q^{-1}) \times \\ &\quad u(k+1+n_c|k) \end{aligned}$$

$$\begin{aligned}
&= \frac{B(q^{-1})}{A_{\Delta}(q^{-1})} \Delta u(k+1+n_c|k) \\
&= H(q^{-1}) \Delta u(k+1+n_c|k) \quad (2.26)
\end{aligned}$$

dengan

$$\begin{aligned}
\frac{B(q^{-1})}{A(q^{-1})} &= G(q^{-1}) \\
\frac{B(q^{-1})}{A_{\Delta}(q^{-1})} &= H(q^{-1})
\end{aligned}$$

$G(q^{-1})$  adalah fungsi alih pulsa dari proses tanpa *dead time* fisik dan  $H(q^{-1})$  adalah fungsi alih pulsa dari proses yang diperpanjang dengan integrator tanpa *dead time* fisik.

*Free Response* merupakan respons hasil pengaruh dari *input* lampau pada *output* yang akan datang dengan asumsi aksi kontrol dari waktu saat ini

$$\hat{y}_{free}(k+d+1+n_c|k) = \frac{B(q^{-1})}{A(q^{-1})} u(k+1+n_c|k-1)$$

$$\Delta u(k+i) = 0; i = 0, 1, 2, \dots \quad (2.27)$$

di mana

$$\begin{aligned}
\mathbf{y}_r &= [y_r(k+d+1+n_{c1}|k), \dots, y_r(k+d+1+n_{c2}|k)]^T, \\
\hat{\mathbf{y}} &= [\hat{y}(k+d+1+n_{c1}|k), \dots, \hat{y}(k+d+1+n_{c2}|k)]^T, \\
\hat{\mathbf{y}}_{force} &= [\hat{y}_{force}(k+d+1+n_{c1}|k), \dots, \hat{y}_{force}(k+d+1+n_{c2}|k)]^T, \\
\hat{\mathbf{y}}_{free} &= [\hat{y}_{free}(k+d+1+n_{c1}|k), \dots, \hat{y}_{free}(k+d+1+n_{c2}|k)]^T,
\end{aligned}$$

Vektor dari *output* prediksi adalah penjumlahan dari respons prediksi paksa dan bebas

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_{force} + \hat{\mathbf{y}}_{free} \quad (2.28)$$

*Output force response* dapat ditunjukkan dengan

$$\begin{bmatrix} \hat{y}_{force}(k+d+n_{c1}+1|k) \\ \hat{y}_{force}(k+d+n_{c1}+2|k) \\ \vdots \\ \hat{y}_{force}(k+d+n_{c2}+1|k) \\ h_{n_{c1}+1} & h_{n_{c1}} & \dots & h_{n_{c1}-n_u+2} \\ h_{n_{c1}+2} & h_{n_{c1}+1} & \dots & h_{n_{c1}-n_u+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_{c2}+1} & h_{n_{c1}} & \dots & h_{n_{c2}-n_u+2} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+n_u-1) \end{bmatrix} = \quad (2.29)$$

atau

$$\hat{\mathbf{y}}_{force} = \mathbf{H}\Delta\mathbf{u}_f$$

di mana

$$\Delta\mathbf{u}_f = [\Delta\mathbf{u}(k|k), \Delta\mathbf{u}(k+1|k), \dots, \Delta\mathbf{u}(k+n_{c2}-1|k)]^T,$$

Variabel manipulasi untuk saat ini dan akan datang disebut  $\Delta u(k+i)$ ,  $i \geq 0$  didesain dengan waktu diskrit, dan bisa ditulis  $\Delta u(k+i|k)$ . Untuk lebih mudahnya penulisan  $\Delta u(k+i|k)$  hanya ditulis  $\Delta u(k+i)$ .  $\mathbf{H}$  merupakan koefisien respons unit step

$$\mathbf{H} = \begin{bmatrix} h_{n_{c1}+1} & h_{n_{c1}} & \dots & h_{n_{c1}-n_u+2} \\ h_{n_{c1}+2} & h_{n_{c1}+1} & \dots & h_{n_{c1}-n_u+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_{c2}+1} & h_{n_{c1}} & \dots & h_{n_{c2}-n_u+2} \end{bmatrix}$$

Respons step didefinisikan sebagai waktu diskrit positif, elemen matrix segitiga atas  $\mathbf{H}$  bernilai nol. Cost function kuadrat dapat ditulis

$$J(\Delta\mathbf{u}_f) = (\mathbf{y}_r - \hat{\mathbf{y}})^T \mathbf{y}(\mathbf{y}_r - \hat{\mathbf{y}}) + \Delta\mathbf{u}_f^T \mathbf{u} \Delta\mathbf{u}_f \rightarrow \min_{\Delta\mathbf{u}_f} \quad (2.30)$$

Mensubstitusikan vektor  $\hat{\mathbf{y}}$  dengan penjumlahan hasil *forced response* dan *free response*

$$J = (\mathbf{y}_r - \mathbf{H}\Delta\mathbf{u}_f - \hat{\mathbf{y}}_{free})^T \mathbf{y}(\mathbf{y}_r - \mathbf{H}\Delta\mathbf{u}_f - \hat{\mathbf{y}}_{free}) + \Delta\mathbf{u}_f^T \mathbf{u} \Delta\mathbf{u}_f \rightarrow \min_{\Delta\mathbf{u}_f} \quad (2.31)$$

Minimalisasi tanpa batasan dari *cost function* berdasarkan persamaan kenaikan *input* menyebabkan

$$\begin{aligned} \frac{dJ(\Delta \mathbf{u}_f)}{d\Delta \mathbf{u}_f} &= -\mathbf{H}^T \begin{bmatrix} \mathbf{y}^T & \mathbf{y} \end{bmatrix} (\mathbf{y}_r - \mathbf{H}\Delta \mathbf{u}_f - \hat{\mathbf{y}}_{\text{free}}) \\ &+ \begin{bmatrix} \mathbf{y}^T & \mathbf{y} \end{bmatrix} \Delta \mathbf{u}_f = 0 \end{aligned} \quad (2.32)$$

Dengan hasil

$$\begin{aligned} \Delta \mathbf{u}_f &= \left[ \mathbf{H}^T \begin{bmatrix} \mathbf{y}^T & \mathbf{y} \end{bmatrix} \mathbf{H} + \begin{bmatrix} \mathbf{y}^T & \mathbf{y} \end{bmatrix} \right]^{-1} \\ &\quad \mathbf{H}^T \begin{bmatrix} \mathbf{y}^T & \mathbf{y} \end{bmatrix} (\mathbf{y}_r - \hat{\mathbf{y}}_{\text{free}}) \end{aligned} \quad (2.33)$$

matriks pembebanan berbentuk simetris dan umumnya diagonal, walaupun *error* pengaturan dan kenaikan kontrol berbentuk fungsi *square* dari waktu dan bukan hasil perkalian *cross* ke nilai antara waktu yang berbeda.

$$\begin{aligned} \Lambda_y^T &= \text{diag}(\lambda_{y,n_{c1}}, \lambda_{y,n_{c1+1}}, \dots, \lambda_{y,n_{c2}}); \\ \Lambda_y^T &= \text{diag}(\lambda_{u0}, \lambda_{u1}, \dots, \lambda_{u,n_u-1}) \end{aligned} \quad (2.34)$$

Sebagai matriks diagonal, nilai matriks transpose sama dengan matriks semula,  $\Lambda_y^T = \Lambda_y$  dan  $\Lambda_u^T = \Lambda_u$ , persamaan sinyal kontrol menjadi

$$\Delta \mathbf{u}_f = \left[ \mathbf{H}^T \begin{bmatrix} \mathbf{y} & \mathbf{u} \end{bmatrix} \mathbf{H} + \Lambda_u \right]^{-1} \mathbf{H}^T \begin{bmatrix} \mathbf{y} & \mathbf{u} \end{bmatrix} (\mathbf{y}_r - \hat{\mathbf{y}}_{\text{free}}) \quad (2.35)$$

Biasanya nilai konstanta diagonal matriks pembebanan bernilai sama  $\Lambda_y = \lambda_y \mathbf{I}$  dan  $\Lambda_u = \lambda_u \mathbf{I}$  (di mana  $\mathbf{I}$  adalah matriks identitas) dengan ukuran yang sama. Nilai  $\lambda_y \neq \lambda_u$ , dengan

$$\Delta \mathbf{u}_f = \left[ \lambda_y \mathbf{H}^T \mathbf{H} + \lambda_u \mathbf{I} \right]^{-1} \mathbf{H}^T (\mathbf{y}_r - \hat{\mathbf{y}}_{\text{free}}) \quad (2.36)$$

Aksi kontrol dipengaruhi hubungan antara  $\lambda_y/\lambda_u$ , sehingga ketika salah satu faktor pembebanan diatur bernilai 1, contohnya  $\lambda_y = 1$  maka

$$\Delta \mathbf{u}_f = \left[ \mathbf{H}^T \mathbf{H} + \lambda_u \mathbf{I} \right]^{-1} \mathbf{H}^T (\mathbf{y}_r - \hat{\mathbf{y}}_{\text{free}}) \quad (2.37)$$

$\lambda_y = 1$  diasumsikan dengan kontrol *single input single output* (SISO), jika tidak ditetapkan sebelumnya.

Berdasarkan strategi *receding horizon* hanya kenaikan *input* pertama  $\Delta u(k)$  digunakan dan dihitung dan diulang ke langkah kontrol selanjutnya,

$$\Delta \mathbf{u}(\mathbf{k}) = [1, 0, \dots, 0] [\mathbf{H}^T \mathbf{H} + \lambda_u \mathbf{I}]^{-1} \mathbf{H}^T (\mathbf{y}_r - \hat{\mathbf{y}}_{\text{free}}) \quad (2.38)$$

di mana vektor  $[1, 0, \dots, 0]$  memiliki  $n_u - 1$  nilai nol.

### 2.12 State Observer [13]

Pada sistem pengaturan dengan pendekatan *pole placement* dilakukan ketika semua variabel dapat terbaca sebagai umpan balik. Pada kenyataannya, tidak semua *state* variabel dapat terbaca, sehingga dibutuhkan estimasi dari *state* variabel yang tidak terukur yang disebut *observer*. Terdapat beberapa cara menemukan *observer* antara lain *full order state observer*, *reduced order state observer*, dan *minimum order state observer*.

Pada *Full order state observer* orde *state observer* sama dengan orde *plant*. Jika persamaan *plant* adalah sebagai berikut :

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \quad (2.39)$$

Model matematika *observer* didefinisikan sebagai berikut :

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_e(\mathbf{y} - \mathbf{C}\tilde{\mathbf{x}}) \\ &= (\mathbf{A} - \mathbf{K}_e\mathbf{C})\tilde{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_e\mathbf{y} \end{aligned} \quad (2.40)$$

Dengan menggunakan rumus Ackermann untuk *pole placement* untuk sistem diatas, ditemukan matriks K

$$\mathbf{K} = [0 \quad 0 \quad \dots \quad 0 \quad 1] \begin{bmatrix} \mathbf{B} \\ \mathbf{A}\mathbf{B} \\ \vdots \\ \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}^{-1} \phi(\mathbf{A}) \quad (2.41)$$

Dalam mendesain *full order state observer*, harus menyelesaikan *dual problem*, yang dapat diselesaikan dengan *pole placement* untuk dual sistem

$$\begin{aligned} \dot{z} &= A^*z + C^*v \\ n &= B^*z \end{aligned} \quad (2.42)$$

Berdasarkan rumus Ackermann untuk *pole placement* diubah menjadi

$$K = [0 \ 0 \ \dots \ 0 \ 1] \begin{bmatrix} C^* \\ A^*C^* \\ \vdots \\ (A^*)^{n-1}C^* \end{bmatrix}^{-1} \phi(A^*) \quad (2.43)$$

Sehingga matriks gain  $K_e$  diketahui dari nilai  $K^*$ , di mana  $K$  didapat dari Persamaan 2.43. sehingga

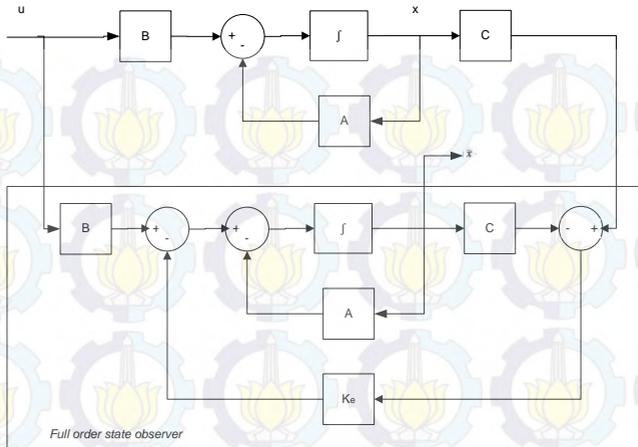
$$\begin{aligned} K_e &= K^* = \phi(A^*)^* \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-2} \\ CA^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\ &= \phi(A) \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-2} \\ CA^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (2.44)$$

di mana  $\phi(s)$  adalah karakteristik polinomial yang diinginkan dari *state observer* adalah

$$\phi(s) = (s + \mu_1)(s + \mu_2)(s + \mu_3)(s + \mu_4) \quad (2.45)$$

dengan  $\mu_1, \mu_2, \dots, \mu_n$  adalah *eigenvalue* yang diinginkan

Blok diagram sistem dan *full order state observer* ditunjukkan pada Gambar 2.17



**Gambar 2.17** Blok Diagram Sistem dan *Full Order State Observer*

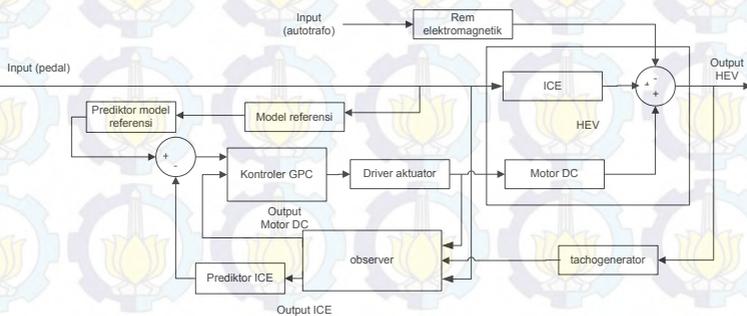
## BAB 3 PERANCANGAN SISTEM

### 3.1 Gambaran Umum Sistem [1][2]

Simulator HEV menggunakan penggerak yang terdiri dari ICE dan motor DC. Bentuk fisik simulator HEV ditunjukkan pada Gambar 3.1. ICE digunakan sebagai penggerak utama dan motor DC digunakan untuk membantu ICE dalam meningkatkan torsi kendaraan. Selain itu digunakan rem elektromagnetik untuk memberikan efek pembebanan. Pembebanan ini digunakan untuk menguji pengaruh motor DC dalam membantu motor bakar meningkatkan akselerasi. Diagram blok sistem ditunjukkan pada Gambar 3.2.



Gambar 3.1 Bentuk Fisik Simulator HEV

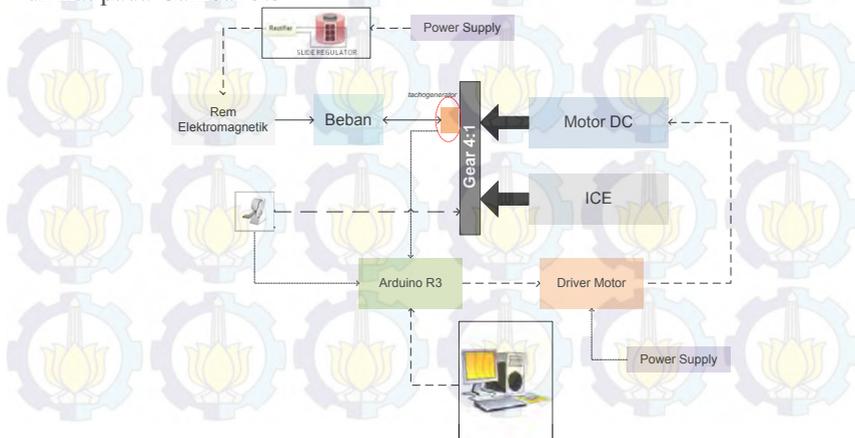


Gambar 3.2 Diagram Blok Sistem

Dalam sistem ini, motor DC merupakan komponen yang dikontrol agar mencapai *output* yang diinginkan. Hal ini dilakukan dengan membandingkan nilai model referensi dan nilai *output* tachogenerator. Hasil dari perbandingan tersebut berupa sinyal *error* yang kemudian akan diolah kontroler GPC. Kontroler ini akan mengeluarkan sinyal kontrol yang kemudian dikirimkan ke *driver* aktuator untuk menggerakkan motor DC. Dengan ini *output* yang dihasilkan diharapkan mempunyai nilai yang menyerupai model referensi.

### 3.2 Perancangan Perangkat Keras

Pada tahap perancangan ini, terdapat 2 jenis perancangan yang dilakukan, yaitu perancangan mekanik dan elektrik. Perancangan mekanik merupakan perancangan untuk *plant* dan sensor untuk *input* dan *output plant*. *Plant* berupa ICE, motor DC, dan rem elektromagnetik. Sensor *input* berupa potensiometer dan sensor *output* berupa tachogenerator. Perancangan elektronik merupakan perancangan untuk kontroler, aktuator, dan rangkaian pendukung lain. Kontroler akan diprogram pada Arduino. Arduino juga akan menerima tegangan dari sensor *input* dan *output*. Aktuator yang digunakan berupa *board driver* yang terdiri dari beberapa komponen elektronik. Rangkaian pendukung berupa rangkaian penyearah gelombang penuh dan rangkaian pembagi tegangan. Konfigurasi perangkat keras pada simulator HEV dapat dilihat pada Gambar 3.3



**Gambar 3.3** Konfigurasi Perangkat Keras Simulator HEV

### 3.2.1 Perancangan Mekanik

Pada simulator HEV ini terdapat beberapa peralatan yang digunakan, antara lain ICE dan motor DC sebagai tenaga penggerak. Untuk menggabungkan kedua mesin ini digunakan gir transmisi. Alat ini digunakan karena perbedaan *range* kerja dari kedua mesin. Selain itu terdapat rem elektromagnetik yang digunakan untuk memberikan efek pembebanan. Rem elektromagnetik diberikan sumber tegangan yang dihasilkan autotrafo yang telah melalui penyearah gelombang penuh. *Input* ICE berupa bukaan *throttle* yang direpresentasikan dengan potensiometer. *Output* kedua mesin diukur oleh tachogenerator yang *tercouple* dengan motor DC. Tachogenerator memberikan masukan ke arduino dan arduino melakukan aksi kontrol yang kemudian diaktuatori oleh *driver* ke motor DC.

#### a. *Internal Combustion Engine (ICE)*

ICE yang dipilih merupakan jenis mesin pemotong rumput. Alat ini dipilih karena memiliki karakteristik yang menyerupai ICE pada kendaraan 2 tak. Bentuk fisik mesin pemotong rumput yang digunakan dapat dilihat pada Gambar 3.4 Spesifikasi mesin pemotong rumput adalah sebagai berikut :

- Tipe : 2 tak
- Bahan bakar : Bensin dan oli samping
- Perbandingan bensin : oli : 25 :1
- Kapasitas silinder : 33,8 cc
- Pemasukan bahan bakar : Karburator
- *Starter* : *Recoil starter*
- Sistem pengapian : *Electronic ignition*
- Sistem kopling : Kering, sentrifugal
- Kecepatan putar maks. : 8000 rpm
- Daya keluaran : 1,8 HP



**Gambar 3.4** Bentuk Fisik Mesin Pemotong Rumput

**b. Motor DC**

Dalam simulator HEV, motor DC berfungsi sebagai penggerak sekunder. Dalam hal ini motor DC bertugas untuk membantu motor bakar dalam memberikan torsi ketika terjadi pembebanan. Motor DC yang digunakan berupa motor DC yang memiliki daya keluaran yang lebih rendah dari pada ICE. Bentuk fisik motor DC dapat dilihat pada Gambar 3.5. Spesifikasi motor DC adalah sebagai berikut :

- Tegangan Nominal : 190 V
- Kecepatan putar : 1750 rpm
- Tipe : Paralel
- Daya Keluaran : ½ HP



**Gambar 3.5** Bentuk Fisik Motor DC

### c. Sistem Penyesuaian Kecepatan

Pada simulator HEV terdapat dua jenis penggerak yang memiliki *range* kerja yang berbeda yaitu ICE dengan kecepatan 0-8000 rpm dan motor DC dengan kecepatan 0-2000 rpm. Untuk menggabungkan kerja kedua penggerak tersebut dibutuhkan gir transmisi. Dengan menggunakan teori perbandingan gir

$$\begin{aligned}\omega_{dc}R_{dc} &= \omega_{bk}R_{bk} & (3.1) \\ \frac{\omega_{dc}}{\omega_{bk}} &= \frac{R_{dc}}{R_{bk}} \\ \frac{R_{dc}}{R_{bk}} &= \frac{2000}{8000} = \frac{1}{4}\end{aligned}$$

dengan

$\omega_{dc}$  adalah kecepatan sudut motor DC

$\omega_{bk}$  adalah kecepatan sudut ICE

$R_{dc}$  adalah jari jari gir pada motor DC

$R_{bk}$  adalah jari jari gir pada ICE

Secara fisik perbandingan gir tersebut tidak efisien dalam hal penempatan, sehingga dilakukan modifikasi yaitu dengan menggunakan gir tengah yang menghubungkan kedua gir. Bentuk fisik gir ditunjukkan pada Gambar 3.6. Dengan perbandingan jari jari gir pada ICE dan gir tengah  $R_{bk}:R_{cbk} = 1:2$  dan perbandingan jari jari gir motor DC dan gir tengah  $R_{dc}:R_{cdc} = 2:1$ . sesuai dengan persamaan

$$\begin{aligned}\omega_{bk}R_{bk} &= \omega_{cbk}R_{cbk} \\ \omega_{dc}R_{dc} &= \omega_{cdc}R_{cdc}\end{aligned}$$

Karena seporos, maka  $\omega_{cbk} = \omega_{cdc}$   
Sehingga,

$$\begin{aligned}\frac{\omega_{bk}R_{bk}}{R_{cbk}} &= \frac{\omega_{dc}R_{dc}}{R_{cdc}} \\ \omega_{dc} &= \frac{R_{bk}}{R_{cbk}} \times \frac{R_{cdc}}{R_{dc}}\end{aligned}$$

$$\frac{\omega_{dc}}{\omega_{bk}} = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

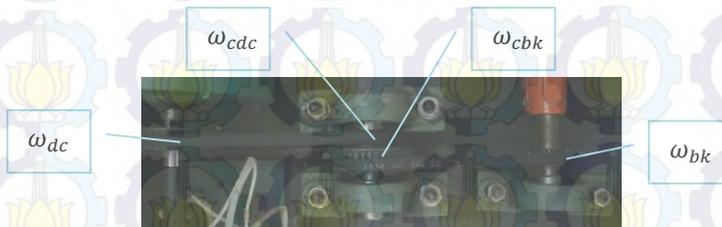
dimana

$\omega_{cdc}$  adalah kecepatan sudut gir tengah yang terhubung dengan motor DC

$\omega_{bk}$  adalah kecepatan sudut gir tengah yang terhubung dengan ICE

$R_{dc}$  adalah jari jari gir tengah yang terhubung dengan motor DC

$R_{bk}$  adalah jari jari gir tengah yang terhubung dengan ICE



**Gambar 3.6** Gir Transmisi

Dengan penggunaan gir tengah akan menghasilkan perbandingan gir yang sama dengan penggunaan gir yang terhubung secara langsung.

Penggunaan gir ini selain berpengaruh pada kecepatan putar kedua mesin juga berpengaruh pada torsi yang dihasilkan kedua mesin. Hal ini dapat terlihat dari persamaan

$$\frac{\tau_{dc}}{\tau_{bk}} = \frac{\omega_{bk}}{\omega_{dc}} \quad (3.2)$$

$$\frac{\tau_{dc}}{\tau_{bk}} = \frac{\omega_{bk}}{\omega_{dc}}$$

$$\frac{\tau_{dc}}{\tau_{bk}} = \frac{\omega_{bk}}{8000}$$

$$\frac{\tau_{dc}}{\tau_{bk}} = \frac{2000}{8000}$$

$$\frac{\tau_{dc}}{\tau_{bk}} = \frac{1}{4}$$

$$\tau_{dc} = 4\tau_{bk}$$

Dengan

$\tau_{dc}$  adalah torsi pada poros motor DC

$\tau_{bk}$  adalah torsi pada poros ICE

Sehingga pada poros motor DC memiliki torsi sebesar empat kali torsi pada ICE.

**d. Sensor Posisi**

Pada sistem HEV *input* diberikan pada ICE. *Input* tersebut diukur dengan menggunakan sensor posisi berupa potensiometer. Potensiometer ini diberi tegangan sebesar 5 volt. Nilai tegangan yang dihasilkan dapat merepresentasikan bukaan *throttle* ICE. Bukaan *throttle* berbanding lurus dengan tegangan yang melalui potensiometer. Bentuk fisik sensor posisi dapat dilihat pada Gambar 3.7.

Pedal dan potensiometer dihubungkan oleh gir reduksi yang memiliki perbandingan 4:1. Gir ini mengakibatkan setiap 1 derajat perubahan pijakan pedal mengakibatkan perubahan 4 derajat pada potensiometer.



potensiometer

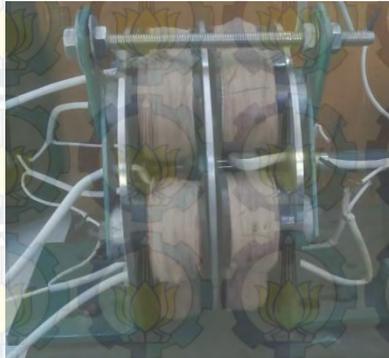
**Gambar 3.7** Bentuk Fisik Potensiometer

**e. Rem Elektromagnetik**

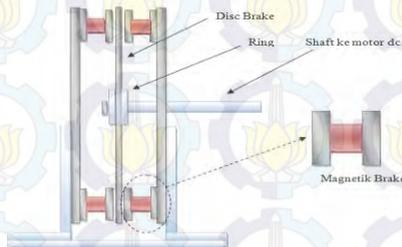
Alat ini digunakan untuk memberikan efek pembebanan bagi simulator HEV. Bentuk fisik rem elektromagnetik yang digunakan pada simulator HEV dapat dilihat pada Gambar 3.8. Gaya dari rem ini dihasilkan oleh beberapa kumparan yang dihubungkan secara seri dan diberikan sumber tegangan DC. Sumber ini didapatkan dari jala jala PLN yang dialirkan ke autotrafo dan dialirkan melalui rangkaian penyearah gelombang penuh.

Rem elektromagnetik ini terdiri dari 12 kumparan yang disusun secara seri. 12 kumparan ini dipisahkan menjadi dua bagian dan diberikan celah. Pada celah ini diberikan piringan besi yang telah terhubung dengan penggerak motor DC. Konstruksi rem elektromagnetik ini dapat dilihat pada Gambar 3.9.

Prinsip kerja rem elektromagnetik ini yaitu ketika kumparan yang dialiri arus listrik akan menghasilkan medan magnet. Medan magnet ini akan memberikan torsi yang berlawanan dengan arah putaran motor. Sehingga dengan meningkatkan tegangan *input* DC akan meningkatkan besar medan magnet yang dihasilkan dan meningkatkan efek pengereman.



**Gambar 3.8** Bentuk Fisik Rem Elektromagnetik



**Gambar 3.9** Konstruksi Rem Elektromagnetik

#### **f. Sensor Kecepatan**

Sensor kecepatan yang digunakan dalam simulator HEV ini adalah tachogenerator. Alat ini dipasang pada motor DC sehingga tegangan DC yang dikeluarkan tachogenerator proporsional dengan kecepatan putar dari motor DC. Bentuk fisik tachogenerator dapat dilihat pada Gambar 3.10. Spesifikasi tachogenerator yang digunakan adalah sebagai berikut :

- Tegangan Kerja : 6-24 V
- Kecepatan Maks : 13570 rpm
- Tipe : *Brush*
- Konstruksi : Magnet Permanen
- Torsi : 29,9 Nm
- Daya Keluaran : 7 - 35 W



**Gambar 3.10** Bentuk Fisik Tachogenerator

### 3.2.2 Perancangan Elektronik

Perancangan elektronik ini meliputi desain layout rangkaian dengan PCB, serta pengkabelan. Desain layout rangkaian dengan PCB dilakukan untuk merancang rangkaian *driver* motor DC, rangkaian pembagi tegangan, dan rangkaian penyearah gelombang penuh. Untuk rancangan pengkabelan dilakukan pada *board* mikrokontroler Arduino.

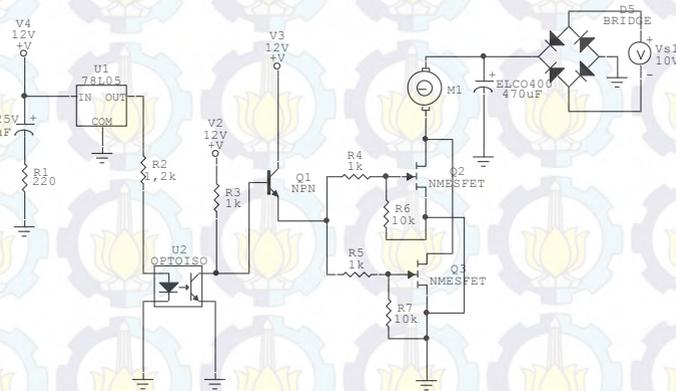
#### a. Rangkaian *Driver* Motor DC

Dalam sistem HEV, *driver* motor DC merupakan aktuator dari motor DC. Bentuk fisik rangkaian *Driver* motor DC dapat dilihat pada Gambar 3.11



**Gambar 3.11** Bentuk Fisik Rangkaian *Driver* Motor DC

Aktuator ini bertujuan untuk mengubah sinyal kontrol menjadi tegangan yang akan dioperasikan ke motor DC. Sumber tegangan diambil dari jala jala PLN. Tegangan tersebut dilewatkan ke autotrafo untuk menyesuaikan dengan tegangan maksimum motor DC. Setelah itu kemudian dilewatkan rangkaian penyearah gelombang penuh agar berubah ke tegangan DC. Tegangan DC yang keluar dari rangkaian penyearah gelombang penuh merupakan tegangan maksimum untuk menggerakkan motor DC. Sehingga untuk mengatur tegangan sesuai dengan sinyal kontrol yang telah dirancang maka digunakan sinyal PWM (*Pulse Width Modulation*). Sinyal PWM ini dihasilkan oleh Arduino. Dengan sinyal PWM maka dapat diatur *duty cycle* dari tegangan supply. Rangkaian elektronik driver ini ditunjukkan pada Gambar 3.12.



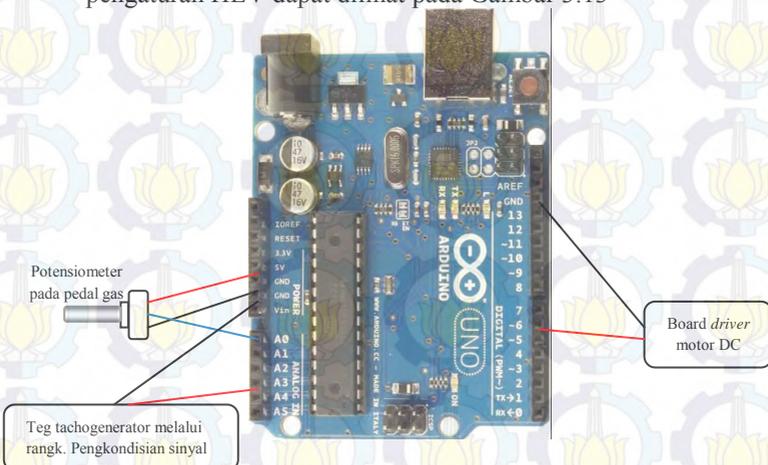
**Gambar 3.12** Rangkaian Driver Motor DC

Driver ini berkerja dengan diberikan tegangan 12V untuk mengaktifkan komponen komponen seperti regulator (IC7805), transistor (TR TIP31C), dan *optocoupler* (Opto P521). Cara kerjanya, ketika mikrokontroler memberikan tegangan dengan *duty cycle low stage*, maka terdapat beda potensial antara mikrokontroler dengan regulator sehingga *optocoupler* akan aktif. Saat itu, transistor akan tidak aktif karena tegangan dialirkan oleh transistor yang ada pada *optocoupler*. Saat mikrokontoler memberikan tegangan dengan *duty cycle high*

stage, tidak ada beda potensial antara regulator dan mikrokontroler sehingga *optocoupler* tidak aktif. Karena itu tegangan 12 V akan mengalir melalui basis dari transistor dan mengaktifkan transistor tersebut. tegangan dari kolektor mengalir ke emitor dan akan mengaktifkan basis dari dua MOSFET (IRFP460) yang dipasang paralel. Basis pada kedua MOSFET aktif, maka tegangan *supply* motor DC akan dapat mengalir ke kedua MOSFET tersebut dan mengaktifkan motor DC.

### b. Arduino

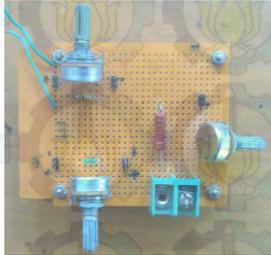
Pada sistem pengaturan simulator HEV, Arduino berfungsi sebagai *board* pengolah *input* dan penghasil *output* sinyal kontrol. Ouput sensor tachometer dihubungkan ke *input* analog arduino. Sedangkan tiga kaki sensor potensiometer harus dihubungkan dengan tegangan 5V, *input* analog, dan ground dari arduino. Tegangan yang masuk kearduino akan diubah menjadi 0-1023 bit dan akan diolah oleh program kontroler yang telah diupload sebelumnya. Kemudian kontroler tersebut menghasilkan output pwm yang dikeluarkan melalui pin *output* digital pwm (pin 3, 5, 6, 9, 10, atau 11). Pin yang digunakan pada sistem pengaturan HEV dapat dilihat pada Gambar 3.13



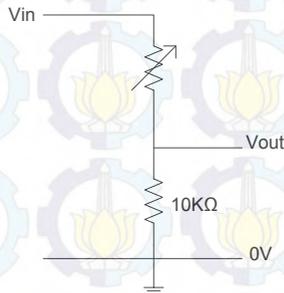
**Gambar 3.13** Pin Arduino pada Sistem Pengaturan HEV

### c. Rangkaian Pembagi Tegangan

Rangkaian ini digunakan untuk menyesuaikan tegangan *output* tachogenerator dengan tegangan *input* Arduino. Rangkaian ini sangat dibutuhkan karena tegangan *output* tachogenerator berkisar antara 0-6,3V sedangkan tegangan operasi arduino 0-5V. Hal ini dilakukan agar tegangan yang masuk arduino tidak melebihi 5V, karena jika melebihi batas arduino dapat mengakibatkan kerusakan pada sistem arduino.



**Gambar 3.14** Bentuk Fisik Rangkaian Pembagi Tegangan



**Gambar 3.15** Rangkaian Pembagi Tegangan

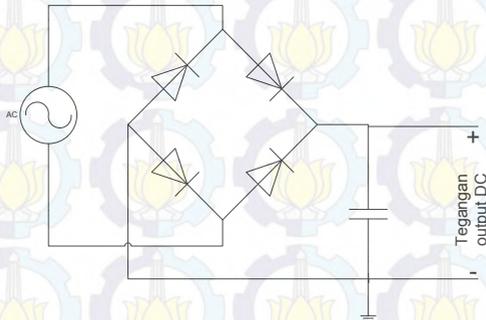
Rangkaian pembagi tegangan ini terdiri dari 1 buah resistor dan 1 buah potensio yang disusun secara seri. Bentuk fisik rangkaian pengkondisian sinyal dapat dilihat pada Gambar 3.14. Potensio digunakan untuk mengantisipasi terjadinya perubahan keluaran tachogenerator, agar dapat menghasilkan *output* tegangan yang sesuai dengan spesifikasi arduino. Resistor yang digunakan memiliki nilai resistansi 10K $\Omega$ . Sehingga

dengan menggunakan rangkaian pada Gambar 3.15 didapat resistansi potensio.

$$V_{out} = \frac{10K}{R_{potensio} + 10K} V_{in} \quad (3.3)$$

#### d. Rangkaian Penyearah Gelombang Penuh

Dalam pengaturan HEV, motor DC dan rem elektromagnetik membutuhkan sumber tegangan berupa tegangan DC. Tegangan ini didapatkan dari tegangan jala jala PLN yang dilewatkan ke rangkaian penyearah gelombang penuh. Dengan rangkaian ini tegangan AC berubah menjadi tegangan DC.



**Gambar 3.16** Rangkaian Penyearah Gelombang Penuh



**Gambar 3.17** Bentuk Fisik Rangkaian Penyearah Gelombang Penuh

Rangkaian ini terdiri dari empat buah dioda yang disusun menjadi rangkaian jembatan dan satu buah kapasitor. Bentuk rangkaian dapat dilihat pada Gambar 3.16. Pada rangkaian

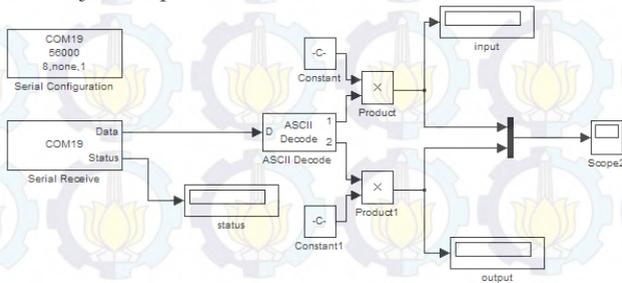
jembatan 2 dioda aktif untuk menghantarkan tegangan AC sedangkan 2 dioda lainnya sebagai isolator pada sikus yang sama. Agar tegangan yang dihasilkan menyerupai tegangan DC maka digunakan kapasitor. Fungsi kapasitor adalah untuk mengurangi *ripple* yang dihasilkan pada proses penyearahan gelombang AC. Bentuk fisik rangkaian penyearah gelombang dapat dilihat pada Gambar 3.17

### 3.3 Perancangan Perangkat Lunak

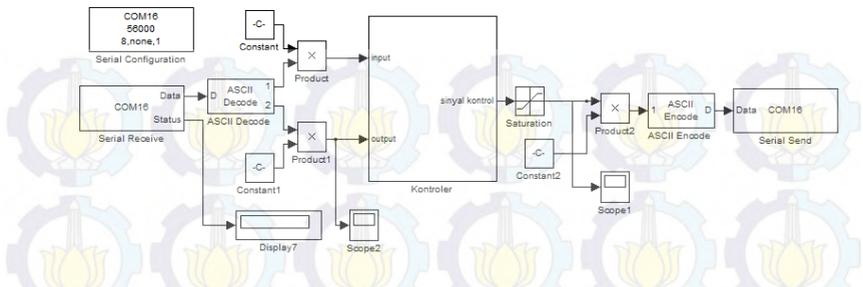
Dalam sistem pengaturan HEV, perangkat lunak digunakan untuk proses pengambilan data, perancangan kontroler, dan pengiriman data. Perangkat lunak yang digunakan yaitu MATLAB dan *software* Arduino.

#### 3.3.1 MATLAB2010a

Perangkat lunak digunakan sebagai antarmuka antara mesin dan pengguna, biasa disebut *Human Machine Interface* (HMI). HMI yang digunakan pada tugas akhir ini adalah MATLAB2010a. Perangkat lunak ini digunakan pada proses pengambilan data dan pengiriman data. Dengan menggunakan komunikasi serial dengan Arduino, Simulink MATLAB dapat mengolah data dari blok *serial receive* dan mengirimkannya kembali melalui blok *serial send*. Contoh diagram blok Simulink MATLAB untuk pengambilan data open loop ditunjukkan pada Gambar 3.18 dan diagram blok Simulink MATLAB dengan kontroler ditunjukkan pada Gambar 3.19



**Gambar 3.18** Blok Simulink MATLAB untuk Pengambilan Data Open Loop



**Gambar 3.19** Blok Simulink MATLAB untuk Pengambilan Data dengan Kontroler

### 3.3.2 Software Arduino

```

dua_analog_read | Arduino 1.0.5
File Edit Sketch Tools Help
dua_analog_read
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorvalue = 0;
  long memont = 0;
  sensorvalue = analogRead(0);
  sensor = analogRead(5);
  // print out the value you read:
  Serial.println("##");
  Serial.println(sensorvalue);
  Serial.println("");
  Serial.println(memont);
}

```

**Gambar 3.20** Tampilan Software Arduino [7]

Software ini merupakan software untuk memasukkan program pada arduino. Program ini digunakan membaca dan mengirim data. Data yang diterima berupa tegangan *output* tachogenerator (data output sistem) dan tegangan output potensiometer (data *input* sistem). Data yang dikirim berupa sinyal kontrol yang berupa PWM, data ini dikirim

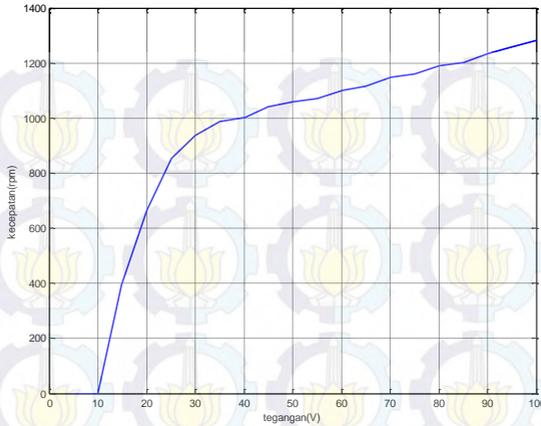
melalui output digital. Tampilan program pembacaan dua data *input* dapat dilihat pada Gambar 3.20

### 3.4 Pengujian *Open Loop* Kecepatan Motor DC

Pengujian ini dilakukan untuk melihat hubungan antara *input* tegangan dan *output* kecepatan pada motor DC. Tegangan yang diberikan berupa tegangan DC yang bernilai antara 0-100V. *Output* kecepatan didapatkan dari tachogenerator. Pengujian ini dilakukan tanpa menggunakan pembebanan. Hasil pengujian ini dapat dilihat pada tabel 3.1 dan grafik perbandingan tegangan *input* dan kecepatan putar motor DC ditunjukkan pada Gambar 3.21

Tabel 3.1 Perbandingan Tegangan *Input* dan Kecepatan Putar Motor DC

Tegangan DC <i>input</i> (V)	Kecepatan putar (rpm)	Tegangan DC <i>input</i> (V)	Kecepatan putar (rpm)
5	0	55,1	1071,486
10	0	60,2	1099,525
14,9	394,5473	65	1115,547
20	666,9251	70,1	1147,592
25,1	853,1834	75	1161,611
30	937,3001	80,1	1189,65
35	987,3696	85,1	1201,667
40,3	1001,389	90,1	1235,714
45	1041,445	95,1	1259,747
50	1059,47	100	1281,778



**Gambar 3.21** Tegangan *Input* Motor DC dan Kecepatan Putar

### 3.5 Pengujian Sensor

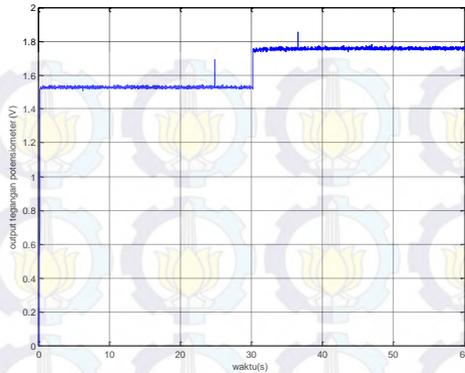
Pengujian ini dilakukan pada sensor *input* dan output sistem. *Input* sistem berupa sensor potensiometer dan output sistem berupa sensor tachogenerator.

#### 3.5.1 Potensiometer

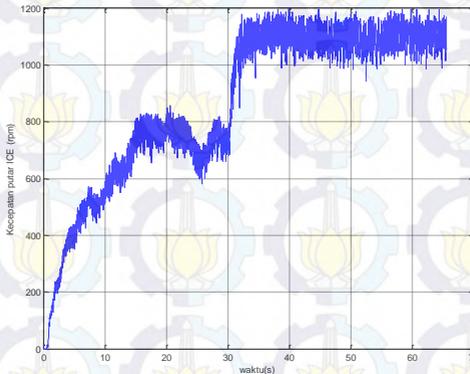
*Input* pada sistem HEV ini direpresentasikan dari pijakan pedal. Pijakan pedal ini dapat menunjukkan seberapa besar bukaan *throttle*. Pedal yang telah terhubung dengan potensiometer ini diberikan *input* tegangan 5V dari *board* Arduino. Pengujian ini dilakukan dengan menginjak pedal dengan kedalaman yang berbeda, sehingga didapatkan perbandingan pijakan pedal dengan *output* tegangan. Presentase pijakan pedal dan tegangan yang melauai potensiometer dapat dilihat pada Tabel 3.2. Grafik tegangan *output* potensiometer dan kecepatan putar ICE dapat dilihat pada Gambar 3.22 dan Gambar 3.23.

**Tabel 3.2** Perbandingan Pijakan Pedal, Tegangan Potensiometer, *Output* Kecepatan ICE

Pijakan pedal (%)	Tegangan Potensiometer (V)	<i>Output</i> ICE (rpm)
0	0	0
75	1,52	760,6
100	1,75	1095



**Gambar 3.22** Pijakan Pedal



**Gambar 3.23** Respon Kecepatan Putar ICE

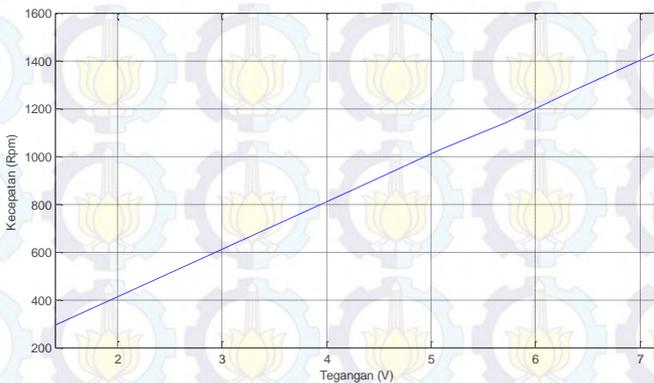
### 3.5.2 Tachogenerator

Pengukuran kecepatan pada simulator HEV melalui tachogenerator yang dipasang pada motor DC. Tachogenerator ini mengeluarkan tegangan DC yang nilainya proporsional dengan kecepatan putar motor DC. Pengujian terhadap sensor ini dilakukan dengan membandingkan tegangan *output* tachogenerator dan putaran motor yang diukur dengan tachometer digital. Nilai tegangan dan rpm ditunjukkan pada Tabel 3.3 dan grafik nilai tegangan output tachometer terhadap kecepatan putar dapat ditunjukkan pada Gambar 3.24. Dengan

menggunakan analisa regresi didapatkan persamaan linier yang menunjukkan kecepatan putar terhadap tegangan tachogenerator.

**Tabel 3.3** Perbandingan Nilai Tegangan dan Kecepatan Putar

Tegangan tacho (V)	Kecepatan (Rpm)
1,4	296
4,2	849
5,1	1029
5,7	1140
6,4	1282
7,2	1442



**Gambar 3.24** Tegangan *Output* Tachogenerator terhadap Kecepatan Putar

### 3.6 Identifikasi dan Permodelan Sistem

Identifikasi merupakan proses yang dilakukan untuk menentukan parameter dari sistem identifikasi ini dilakukan untuk melihat respons *plant*. Dengan respons yang didapatkan, dapat dibuat model matematika yang melambangkan sistem. Pada sistem ini, identifikasi dilakukan untuk mendapatkan model matematika dari ICE dan motor DC

#### 3.6.1 Metode Pembebanan *Plant*

Respons *plant* dapat dilihat dari beberapa cara. Salah satu cara yaitu dengan memberikan pembebanan. Pada simulator HEV pembebanan dilakukan dengan menggunakan rem elektromagnetik.

Pembeban dilakukan untuk mendapatkan model sistem dengan beban minimal, nominal, dan maksimal.

Model sistem dengan beban minimal didapatkan dari menjalankan ICE yang telah *tercouple* dengan motor DC yang tidak bekerja, sehingga diasumsikan motor DC merupakan beban minimal dari ICE. Model pembebanan nominal didapatkan ketika ICE yang bekerja pada kecepatan tertentu diberikan beban dan ICE dapat meningkatkan kerjanya hingga mencapai kecepatan yang sama sebelum diberikan beban. Setelah dilakukan pengujian, pembebanan nominal ini dilakukan dengan memberikan rem elektromagnetik tegangan sebesar 60VDC. Kemudian model pembebanan maksimal didapatkan ketika ICE yang bekerja pada kecepatan tertentu diberikan beban dan ketika ICE meningkatkan kinerjanya tidak dapat mencapai kecepatan yang sama sebelum diberikan pembebanan. Pembebanan maksimal ini dilakukan dengan memberikan tegangan rem elektromagnetik sebesar 90VDC.

### 3.6.2 Metode Identifikasi dan Permodelan

Identifikasi pada ICE dan motor DC dilakukan pada sistem *open loop* dengan identifikasi statis, yaitu dengan memberikan *input* berupa sinyal step. Pada ICE, sinyal step berupa injakan pedal yang langsung diarahkan kebatas maksimum pedal. Sedangkan pada motor DC dilakukan secara manual dengan memutar autotrafo. Pengambilan data ini dilakukan dengan menghubungkan tachogenerator dengan arduino yang telah terhubung ke software Simulink MATLAB. Dari data yang didapatkan dilakukan permodelan sistem dengan beberapa metode yaitu Viteckova, Latzel, Strejc,. Permodelan sistem dengan beberapa metode tersebut ditunjukkan pada Tabel 3.4.

**Tabel 3.4** Permodelan dengan Beberapa Metode

Metode	Fungsi Alih Plant	RMSE
Latzel	$G(s) = \frac{1}{1.852321s^2 + 2.722s + 1}$	63.68
Viteckova 1 <sup>th</sup> Order	$G(s) = \frac{1}{0.382s + 1} e^{-0.353s}$	29.6813
Viteckova 2 <sup>nd</sup> Order	$G(s) = \frac{1}{(0.244s + 1)^2} e^{-0.212s}$	18.3816
Strejc	$G(s) = \frac{1}{(0.098s + 1)(0.533s + 1)}$	16.3812

Dari permodelan diatas ditemukan permodelan yang memiliki nilai RMSE lebih kecil yaitu permodelan dengan metode Strejc.

### 3.6.3 Permodelan ICE

Permodelan ICE didapatkan dari permodelan statis dari sistem open loop. Dalam hal ini dilakukan 3 jenis permodelan yaitu permodelan dengan beban minimal, nominal, dan maksimal. Data yang telah didapatkan dari Simulink MATLAB kemudian dianalisis untuk menentukan paramter permodelan Strejc.

#### a. Beban Minimal

Grafik respons dengan beban minimal dapat dilihat pada Gambar 3.25. Dari grafik tersebut, dilakukan perhitungan untuk mendapatkan parameter untuk permodelan dengan metode Strejc seperti berikut

$$Y_{ss} = 1178 \text{ rpm}$$

$$X_{ss} = 914 \text{ rpm}$$

Dengan nilai  $X_{ss}$  dan  $Y_{ss}$  maka didapatkan nilai  $K$

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{1178}{914} = 1,29$$

Selanjutnya dicari nilai  $T_U$  dan  $T_N$  yang didapatkan dengan menarik garis miring yang memotong titik belok respons saat transien. dari nilai  $T_U$  dan  $T_N$  didapatkan nilai  $\tau$ .

$$T_U = 0,0953 \text{ s}$$

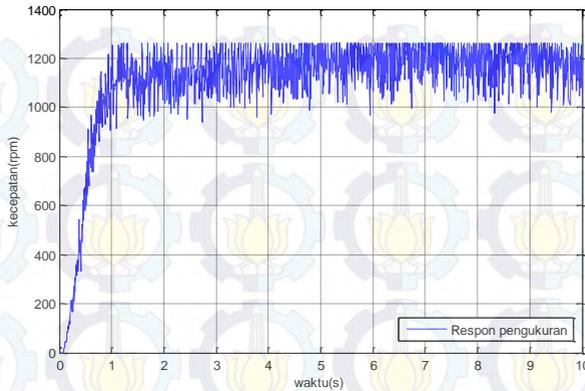
$$T_N = 0,7647 \text{ s}$$

$$\tau = \frac{T_U}{T_N} = \frac{0,0953}{0,7647} = 0,125$$

Setelah nilai  $\tau$  diketahui, dengan melihat Tabel 2.3, maka ditemukan nilai  $y_i$ . Nilai tersebut kemudian dikalikan dengan nilai  $Y_{ss}$ , dan ditemukan suatu nilai.

$$y_i = 0,264(1178)$$

$$y_i = 310,992 \text{ rpm}$$



**Gambar 3.25** Respons ICE dengan Beban Minimal

Dengan melihat grafik respons sistem yang ditunjukkan pada Gambar 3.24, maka ditemukan  $t_i$  saat  $y_i$

$$t_i = 0,3118 \text{ s}$$

Kemudian dengan menggunakan Persamaan 2.12, didapatkan nilai  $\tau_{ST}$

$$\tau_{ST} = \frac{0,3118}{2 - 1}$$

$$\tau_{ST} = 0,3118$$

Dari nilai parameter permodelan di atas, didapatkan fungsi alih ICE dengan beban minimal sebagai berikut.

$$G_{ST2}(s) = \frac{K}{(\tau_{ST}s + 1)^n}$$

$$G(s) = \frac{1,29}{(0,3118s + 1)^2}$$

b. Beban Nominal

Permodelan beban nominal dilakukan memberikan beban berupa rem elektromagnetik yang diberikan tegangan 60V. Grafik respons ICE dengan beban nominal dapat dilihat pada

Gambar 3.25. Dari grafik tersebut ditemukan parameter permodelan Strejc seperti berikut

$$\begin{aligned} Y_{ss} &= 914 \text{ rpm} \\ X_{ss} &= 914 \text{ rpm} \end{aligned}$$

Sehingga didapatkan nilai K

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{914}{914} = 1$$

Selanjutnya dicari nilai  $T_U$  dan  $T_N$  yang didapatkan dengan menarik garis miring yang memotong titik belok respons saat transien. dari nilai  $T_U$  dan  $T_N$  didapatkan nilai  $\tau$ .

$$\begin{aligned} T_U &= 0,06133 \text{ s} \\ T_N &= 0,5843 \text{ s} \\ \tau &= \frac{T_U}{T_N} = \frac{0,06133}{0,5843} = 0,1049 \end{aligned}$$

Setelah nilai  $\tau$  diketahui, dengan melihat Tabel 2.3, maka ditemukan nilai  $y_i$ . Nilai tersebut kemudian dikalikan dengan nilai  $Y_{ss}$ , dan ditemukan suatu nilai.

$$\begin{aligned} y_i &= 0,264(914) \\ y_i &= 241,296 \text{ rpm} \end{aligned}$$

Dengan melihat grafik respons sistem yang ditunjukkan pada Gambar 3.26, maka ditemukan  $t_i$  saat  $y_i$

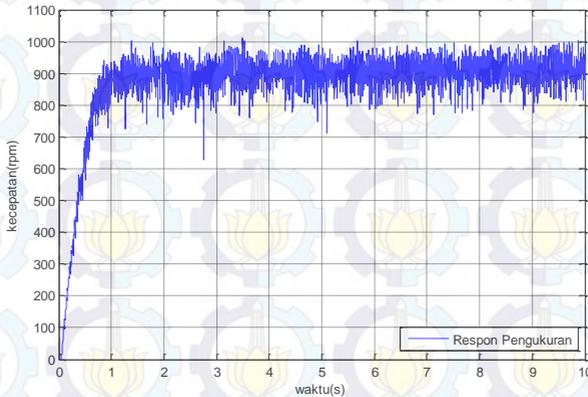
$$t_i = 0,1848 \text{ s}$$

Kemudian dengan menggunakan persamaan, didapatkan nilai  $\tau_{ST}$

$$\begin{aligned} \tau_{ST} &= \frac{0,1848}{2 - 1} \\ \tau_{ST} &= 0,1848 \end{aligned}$$

Dari perhitungan di atas didapatkan fungsi alih ICE dengan beban nominal sebagai berikut.

$$G(s) = \frac{1}{(0,1848s + 1)^2}$$



**Gambar 3.26** Respons ICE dengan Beban Nominal

c. Beban Maksimal

Pemodelan yang berikutnya adalah permodelan mesin bakar dengan beban maksimum. Beban ini berupa rem elektromagnetik yang diberikan tegangan sebesar 90V. Grafik respons ICE dengan beban maksimal dapat dilihat pada Gambar 3.27. Dari grafik tersebut ditemukan parameter permodelan Strejc sebagai berikut

$$Y_{ss} = 548,9 \text{ rpm}$$

$$X_{ss} = 914 \text{ rpm}$$

Sehingga didapatkan nilai K

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{548,9}{914} = 0,6$$

Selanjutnya dicari nilai  $T_U$  dan  $T_N$  yang didapatkan dengan menarik garis miring yang memotong titik belok respons saat transien. dari nilai  $T_U$  dan  $T_N$  didapatkan nilai  $\tau$ .

$$T_U = 0,104 \text{ s}$$

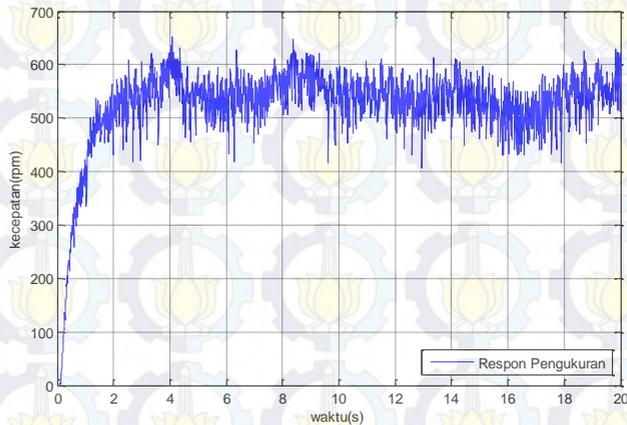
$$T_N = 0,891 \text{ s}$$

$$\tau = \frac{T_U}{T_N} = \frac{0,104}{0,891} = 0,1167$$

Setelah nilai  $\tau$  diketahui, dengan melihat Tabel 2.3, maka ditemukan nilai  $y_i$ . Nilai tersebut kemudian dikalikan dengan nilai  $Y_{SS}$ , dan ditemukan suatu nilai.

$$y_i = 0,264(548,9)$$

$$y_i = 144,9 \text{ rpm}$$



**Gambar 3.27** Respons ICE dengan Beban Maksimal

Dengan melihat grafik respons sistem yang ditunjukkan pada Gambar 3.26, maka ditemukan  $t_i$  saat  $y_i$

$$t_i = 0,289 \text{ s}$$

Kemudian dengan menggunakan Persamaan 2.12, didapatkan nilai  $\tau_{ST}$

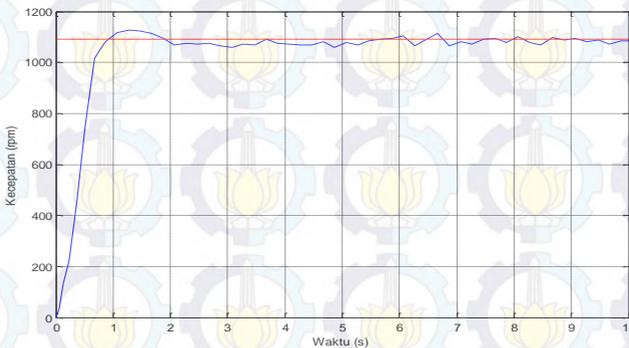
$$\tau_{ST} = \frac{0,289}{2 - 1}$$
$$\tau_{ST} = 0,289$$

Dari perhitungan di atas didapatkan fungsi alih ICE dengan beban maksimal sebagai berikut.

$$G(s) = \frac{0,6}{(0,289 + 1)^2}$$

### 3.6.4 Permodelan Motor DC

Proses identifikasi motor DC melalui tahapan yang hampir sama dengan motor bakar. Motor DC diberikan tegangan 75V dc dari autotrafo. Kemudian kecepatan motor diukur oleh tachometer dan melalui rangkaian pengkondisian sinyal memasuki *input* analog arduino. Arduino akan menampilkan data tersebut ke Simulink MATLAB. Grafik respons motor DC ditampilkan pada Gambar 3.28. Dari gambar tersebut terlihat bahwa terdapat *overshoot* pada respons sistem, sehingga dengan respons demikian permodelan Strejc tidak bisa dilakukan. Oleh karena itu dilakukan permodelan dengan pendekatan analitik sebagai berikut :



**Gambar 3.28** Respons Kecepatan Motor DC

Menentukan besar *overshoot* maksimum (persen) dengan rumus :

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

dengan nilai  $c(t_p) = 1126 \text{ rpm}$ , dan  $c(\infty) = 1090 \text{ rpm}$  didapatkan nilai  $M_p$  sebagai berikut

$$M_p = \frac{1126 - 1090}{1090} \times 100\% = 5,5\%$$

Selanjutnya adalah mencari nilai  $\omega_d$  yang merupakan frekuensi alami teredam dengan persamaan berikut

$$t_p = \frac{\pi}{\omega_d}$$

dengan nilai  $t_p = 1,287\text{s}$  maka didapatkan nilai  $\omega_d$  sebagai berikut

$$\omega_d = \frac{\pi}{t_p}$$
$$\omega_d = \frac{3,14}{1,287} = 2,44$$

Untuk menemukan nilai koefisien peredaman ( $\xi$ ), digunakan persamaan:

$$M_p = e^{-(\xi/\sqrt{1-\xi^2})\pi}$$

sehingga didapatkan nilai  $\xi$  sebagai berikut

$$0,055 = e^{-(\xi/\sqrt{1-\xi^2})\pi}$$
$$\ln 0,076 = \ln e^{-(\xi/\sqrt{1-\xi^2})\pi}$$
$$-2,9 = -(\xi/\sqrt{1-\xi^2}) \times \pi$$
$$8,4 = \frac{\xi^2}{1-\xi^2} \times \pi^2$$
$$\frac{8,4}{\pi^2} = \frac{\xi^2}{1-\xi^2}$$
$$0,85 - 0,85\xi^2 = \xi^2$$

$$\xi^2 + 0,85\xi^2 = 0,85$$

$$\xi^2 = \frac{0,85}{1,85} = 0,401$$

$$\xi = \sqrt{0,46} = 0,678$$

Setelah mendapatkan nilai  $\xi$ , maka dapat ditemukan nilai  $\omega_n$  dengan persamaan :

$$\omega_d = \omega_n \sqrt{1 - \xi^2}$$

sehingga dari persamaan diatas didapatkan nilai  $\omega_n$  sebagai berikut

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \xi^2}}$$

$$\omega_n = \frac{2,44}{\sqrt{1 - 0,678^2}} = 3,318$$

Respons sistem menunjukkan bahwa sistem tergolong redaman kurang atau *under damped* ( $0 < \xi < 1$ ), maka fungsi alih sistem dapat dituliskan sebagai berikut

$$G(s) = \frac{\omega_n^2}{(s + \xi\omega_n + j\omega_d)(s + \xi\omega_n - j\omega_d)}$$

$$G(s) = \frac{3,318^2}{(s + 2,25 + j2,44)(s + 2,25 - j2,44)}$$

$$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)}$$

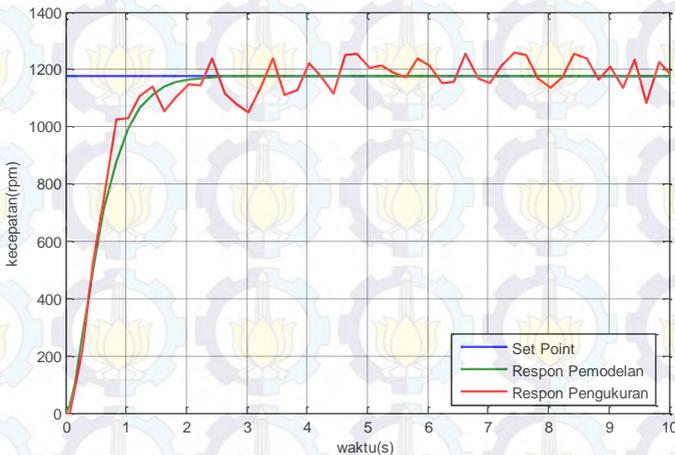
### 3.6.5 Pengujian dan Validasi

Dalam menemukan permodelan yang menyerupai respons sistem, maka dilakukan penyusunan beberapa persamaan model. Dari beberapa persamaan tersebut, dipilih persamaan yang memiliki RRMSE yang paling kecil. Nilai RRMSE didapatkan dari persamaan RMSE (2.11) yang dibandingkan dengan nilai  $y_{ss}$ .

Persamaan model *plant* ICE dengan beban minimal ditunjukkan pada Tabel 3.5 dan grafik perbandingan data sebenarnya dan permodelan ditunjukkan pada Gambar 3.29.

**Tabel 3.5** Persamaan Model dan Validasi *Plant* ICE Beban Minimal

No	Fungsi Alih <i>Plant</i>	RRMSE (%)
1	$G(s) = \frac{1,29}{(0,3118s + 1)^2}$	4,39
2	$G(s) = \frac{1,31}{(0,3133s + 1)^2}$	4,49
3	$G(s) = \frac{1,302}{(0,3128s + 1)^2}$	4,37
4	$G(s) = \frac{1,27}{(0,3107s + 1)^2}$	4,61
5	$G(s) = \frac{1,25}{(0,3088s + 1)^2}$	5,463

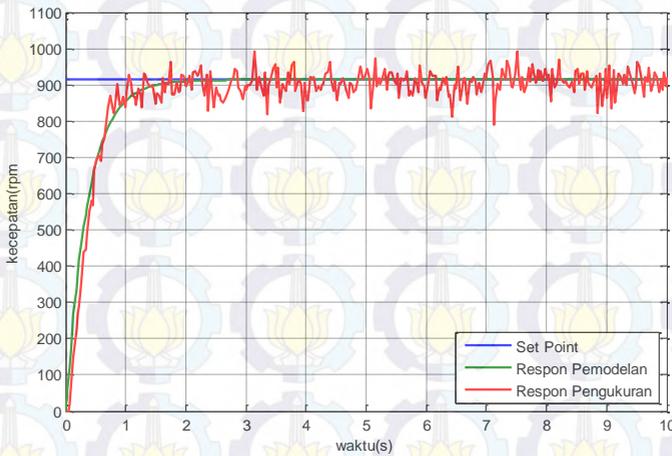


**Gambar 3.29** Hasil Identifikasi dan Permodelan ICE Beban Minimal

Persamaan model *plant* ICE dengan beban nominal ditunjukkan pada Tabel 3.6 dan grafik perbandingan data sebenarnya dan permodelan ditunjukkan pada Gambar 3.30

**Tabel 3.6** Persamaan Model dan Validasi *Plant* ICE Beban Nominal

No	Fungsi Alih <i>Plant</i>	RRMSE(%)
1	$G(s) = \frac{0,946}{(0,17s + 1)^2}$	5,397
2	$G(s) = \frac{0,974}{(0,18s + 1)^2}$	3,852
3	$G(s) = \frac{0,996}{(0,1843s + 1)^2}$	3,576
4	$G(s) = \frac{1}{(0,1848s + 1)^2}$	3,208
5	$G(s) = \frac{1,01}{(0,1858s + 1)^2}$	3,717

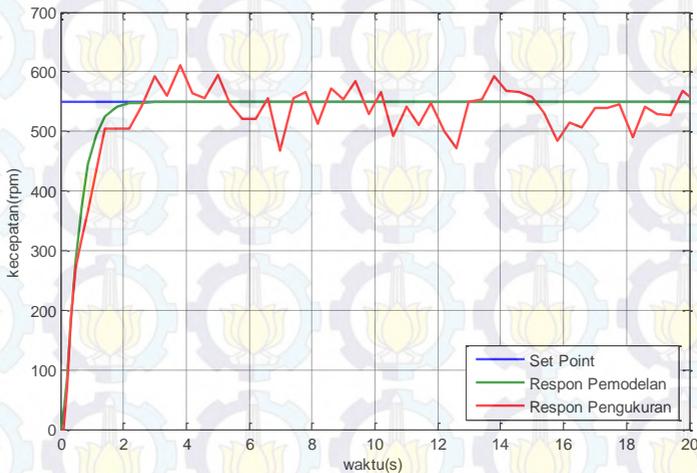


**Gambar 3.30** Hasil Identifikasi dan Pemodelan ICE Beban Nominal

Persamaan model *plant* ICE dengan beban maksimal ditunjukkan pada Tabel 3.7 dan grafik perbandingan data sebenarnya dan pemodelan ditunjukkan pada Gambar 3.31

**Tabel 3.7** Persamaan Model dan Validasi *Plant* ICE Beban Maksimal

No	Fungsi Alih <i>Plant</i>	RRMSE
1	$G(s) = \frac{0,580}{(0,286s + 1)^2}$	6,66
2	$G(s) = \frac{0,571}{(0,284s + 1)^2}$	7,3
3	$G(s) = \frac{0,578}{(0,2854s + 1)^2}$	6,83
4	$G(s) = \frac{0,601}{(0,289s + 1)^2}$	6,17
5	$G(s) = \frac{0,6113}{(0,291s + 1)^2}$	6,39

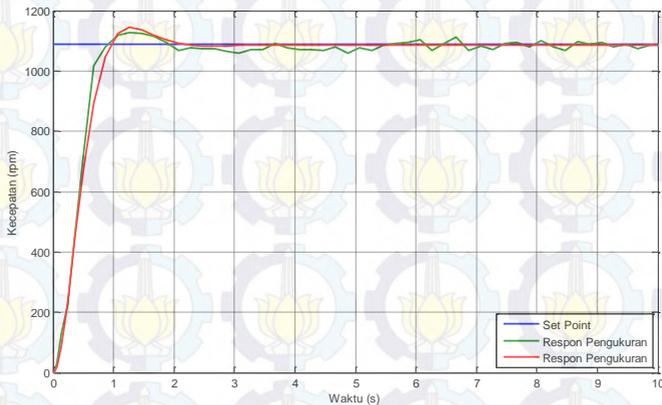


**Gambar 3.31** Hasil Identifikasi dan Pemodelan ICE Beban Maksimal

Persamaan model *plant* motor DC ditunjukkan pada Tabel 3.8 dan grafik perbandingan data sebenarnya dan pemodelan ditunjukkan pada Gambar 3.32

**Tabel 3.8** Persamaan Model dan Validasi *Plant* Motor DC

No	Fungsi Alih <i>Plant</i>	RRMSE(%)
1	$G(s) = \frac{10,98}{(s^2 + 4,8s + 10,975)}$	0,7355
2	$G(s) = \frac{11,09}{(s^2 + 4,52s + 11,087)}$	0,6129
3	$G(s) = \frac{11,2}{(s^2 + 4,49s + 11,17)}$	0,6248
4	$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)}$	0,6104
5	$G(s) = \frac{11,35}{(s^2 + 4,54s + 11,29)}$	0,6589



**Gambar 3.32** Hasil Identifikasi dan Pemodelan Motor DC

Dari data diatas maka dilakukan pemilikitah fungsi alih *plant* yang memiliki nilai yang mendekati respons sistem yang sebenarnya, fungsi alih ICE ditunjukkan pada Tabel 3.9.

**Tabel 3.9** Pemilihan Model *Plant* ICE

No	Beban	Fungsi Alih <i>Plant</i>	RRMSE
1	Minimal	$G(s) = \frac{1,302}{(0,3128s + 1)^2}$	4,37%
2	Nominal	$G(s) = \frac{1}{(0,1848s + 1)^2}$	3,208%
3	Maksimal	$G(s) = \frac{0,601}{(0,289s + 1)^2}$	6,17%

Untuk motor DC fungsi alih yang akan digunakan adalah

$$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)}$$

### 3.6.6 Model referensi

Model referensi merupakan model yang digunakan sebagai tujuan kontrol. Model ini digunakan agar respons sistem dapat sesuai dengan keinginan perancang. Sehingga jika respons sistem tidak sesuai dengan model referensi, kontroler akan menghasilkan sinyal kontrol yang sesuai agar respons sistem dapat mengikuti model referensi.

Tujuan pengaturan traksi adalah untuk mengatur simulator HEV saat keadaan akselerasi. Ketika ICE berada pada keadaan akselerasi, motor DC akan membantu ICE agar dapat mencapai model referensi. Model ini didesain dengan tidak memiliki *error steady state* dan memiliki *rise time* yang lebih cepat dibanding ICE. Dengan model matematika ICE sebagai berikut

$$G(s) = \frac{1}{0,034s^2 + 0,3696s + 1}$$

Dengan

$$\omega_n = 5,423$$

$$\xi = 1$$

Sistem tergolong *critically damped*, karena memiliki nilai  $\xi = 1$ . Dari respons ICE tersebut didapatkan *time constant* sebesar

$$\begin{aligned}\tau &= t[0,632 \times Y_{ss}] \\ \tau &= t[0,632 \times 914] \\ \tau &= t[577,648] \\ \tau &= 0,4\end{aligned}$$

Dengan *settling time* ( $\pm 5\%$ ) ICE sebesar 1,2. Maka ditentukan *settling time* model referensi lebih cepat yaitu 0,6s. Model referensi dirancang dengan model orde satu dengan nilai  $\tau$

$$\begin{aligned}T_s(\pm 5\%) &= 3\tau \\ \tau &= 0,2\end{aligned}$$

Sehingga didapatkan model referensi sebagai berikut

$$G(s) = \frac{1}{0,2s + 1}$$

### 3.7 Desain Sistem Pengaturan Traksi

Desain sistem pengaturan ini meliputi beberapa hal antara lain desain kontroler yang berupa kontroler GPC, desain prediktor, dan desain *observer*.

#### 3.7.1 Desain *Generalized Predictive Control*

Pada penelitian ini, *Generalized Predictive Control* (GPC) digunakan untuk mengatur motor DC. Pengaturan motor DC ini bertujuan untuk membantu motor bakar dalam meningkatkan akselerasi. Hal ini disebabkan karena respons motor bakar lambat.

Kontroler ini disusun berdasarkan kinerja motor DC. Dari model matematika motor DC, maka dilakukan permodelan yang sesuai dengan GPC yaitu ARMA (*AutoRegressive Moving Average*). Model matematika motor DC adalah sebagai berikut :

$$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)}$$

Bentuk ARMA didapatkan dari penyusunan blok identifikasi ARMAX pada simulink Matlab dengan modifikasi. Ditentukan orde sistem 2, orde *input* 2, *time sampling* sebesar 0.1, dengan dijalankan selama 100s maka didapatkan hasil permodelan ARMA sebagai berikut:

$$\frac{Y(z)}{U(z)} = \frac{0,047215z + 0,0406226}{z^2 - 1,548z + 0,63763}$$

$$y(k) = 1,548y(k-1) - 0,63763y(k-2) + 0,047215u(k-1) - 0,0406226u(k-2)$$

Dengan

$$\begin{aligned} A(q^{-1}) &= a_1q^{-1} + a_2q^{-2} \\ &= 0,047215q^{-1} + 0,0406226q^{-2} \\ B(q^{-1}) &= 1 - b_1q^{-1} + b_2q^{-2} \\ &= 1 - 1,548q^{-1} + 0,63763q^{-2} \end{aligned}$$

Setelah model ARMA ditemukan, langkah selanjutnya yaitu menentukan panjang *prediction horizon*. Penentuan ini harus disesuaikan dengan respons yang diharapkan. Apabila panjang horizon terlalu pendek akan menimbulkan respons mendahului model referensi, sedangkan apabila terlalu panjang respons sistem akan lambat walaupun tidak terdapat *overshoot*. Dari model referensi yang telah ditentukan sebelumnya, diharapkan respons lebih cepat dari pada sebelumnya, sehingga ditentukan panjang *prediction horizon* sebesar 3.

$$\begin{aligned} y(k) &= b_1y(k-1) + b_2y(k-2) + a_1u(k-1) + a_2u(k-2) \\ y(k+1) &= b_1y(k) + b_2y(k-1) + a_1u(k) + a_2u(k-1) \\ y(k+2) &= (b_1^2 + b_2)y(k) + b_1b_2y(k-1) + a_1u(k+1) + \\ &\quad (b_1a_1 + a_2)u(k) + b_1a_2u(k-1) \\ y(k+3) &= (b_1^3 + 2b_1b_2)y(k) + (b_1^2b_2 + b_2^2)y(k-1) \\ &\quad + a_1u(k+2) + (b_1a_1 + a_2)u(k+1) \\ &\quad + (b_1^2a_1 + b_2a_1 + b_1a_2)u(k) + \\ &\quad + (b_1^2b_2 + b_2^2)u(k-1) \end{aligned} \quad (3.4)$$

Dari ketiga Persamaan 3.4 diatas maka dapat dibentuk matriks sebagai berikut

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_{force} + \hat{\mathbf{y}}_{free}$$

$$\begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{bmatrix} = \mathbf{H} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \end{bmatrix} + \begin{bmatrix} \hat{y}_{free}(k+1) \\ \hat{y}_{free}(k+2) \\ \hat{y}_{free}(k+3) \end{bmatrix}$$

$$= \begin{bmatrix} h_1 & 0 & 0 \\ h_2 & h_1 & 0 \\ h_3 & h_2 & h_1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \end{bmatrix} + \begin{bmatrix} \hat{y}_{free}(k+1) \\ \hat{y}_{free}(k+2) \\ \hat{y}_{free}(k+3) \end{bmatrix} \quad (3.5)$$

Dengan

$$\begin{bmatrix} h_1 & 0 & 0 \\ h_2 & h_1 & 0 \\ h_3 & h_2 & h_1 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 \\ b_1 a_1 + a_2 & a_1 & 0 \\ b_1^2 a_1 + b_2 a_1 + b_1 a_2 & b_1 a_1 + a_2 & a_1 \end{bmatrix};$$

$$\begin{bmatrix} \hat{y}_{free}(k+1) \\ \hat{y}_{free}(k+2) \\ \hat{y}_{free}(k+3) \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & a_2 \\ b_1^2 + b_2 & b_1 b_2 & b_1 a_2 \\ b_1^3 + 2b_1 b_2 & b_1^2 b_2 + b_2^2 & b_1^2 b_2 + b_2^2 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ u(k-1) \end{bmatrix}$$

Sinyal kontrol didapatkan dengan

$$u(k) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [H^T H + \lambda_u I]^{-1} H^T (\mathbf{y}_r - \hat{\mathbf{y}}_{free}) \quad (3.6)$$

Dengan

$$\mathbf{y}_r = \begin{bmatrix} y_{r+1} \\ y_{r+2} \\ y_{r+3} \end{bmatrix}$$

### 3.7.2 Desain Prediktor

Prediktor merupakan formulasi suatu sistem agar dapat diketahui *output* sistem beberapa langkah kedepan. Dalam perancangan kontroler GPC prediktor ini sangat dibutuhkan untuk menemukan nilai dari matriks  $\mathbf{y}_r$  yang merupakan nilai referensi dari kontroler.

Pada sistem pengaturan traksi pada HEV, motor DC mendapatkan *input* berupa *error* antara model referensi dan ouput ICE. *Error* tersebut merupakan referensi dari kontroler, sehingga dibutuhkan prediktor untuk memprediksi *error* beberapa langkah kedepan. Prediktor ini dapat dibentuk dengan menyusun prediktor dari model referensi dan ICE.

Model referensi yang diharapkan dapat dicapai oleh sistem HEV, berbentuk persamaan orde 1 sebagai berikut :

$$G(s) = \frac{1}{0,2s + 1}$$

Dari persamaan tersebut, dibentuk permodelan diskrit dengan *time sampling* sebesar 0,1 dan didapatkan persamaan

$$G(z) = \frac{0,39346}{z - 0,60654}$$

Sehingga didapatkan

$$\begin{aligned} y_r(k) &= b_r y_r(k-1) + a_r x(k-1) \\ y_r(k) &= 0,60654 y_r(k-1) + 0,39346 x(k-1) \end{aligned}$$

Dari persamaan diatas dapat diturunkan persamaan  $y_r$  tiga langkah kedepan, sehingga didapatkan persamaan  $y_r(k+3)$ . Dalam hal ini nilai  $x(k)$  dianggap sama pada setiap waktu.

$$\begin{aligned} y_r(k+1) &= b_r y_r(k) + a_r x(k) \\ y_r(k+2) &= b_r^2 y_r(k) + (b_r a_r + b_r) x(k) \\ y_r(k+3) &= b_r^3 y_r(k) + (b_r^2 a_r + b_r a_r + a_r) x(k) \end{aligned} \quad (3.7)$$

Selain pada model referensi, prediktor juga harus didesain pada ICE, karena ICE merupakan pembanding model referensi dalam menemukan *error* yang akan menjadi *input* kontroler GPC. Persamaan ICE pada beban nominal adalah sebagai berikut :

$$G(s) = \frac{1}{0,034s^2 + 0,3696s + 1}$$

Dari persamaan diatas, dibentuk model diskrit dengan *time sampling* sebesar 0,1s. Sehingga didapatkan persamaan diskrit orde 2 sebagai berikut :

$$G(z) = \frac{0,10328z + 0,071766}{z^2 - 1,1622z + 0,33277}$$

Sehingga didapatkan persamaan

$$\begin{aligned} y_b(k) &= b_{b1}y(k-1) + b_{b2}y(k-2) + a_{b1}x(k-1) \\ &\quad + a_{b2}x(k-2) \\ &= 1,1622y(k-1) - 0,33722y(k-2) \\ &\quad + 0,10328x(k-1) + 0,071766x(k-2) \end{aligned}$$

Dari persamaan diatas maka dapat diturunkan nilai  $y_b(k)$  tiga langkah kedepan. sehingga didapatkan nilai  $y_b(k+3)$  Sama dengan model referensi nilai  $x(k)$  dianggap sama pada setiap waktu.

$$\begin{aligned} y_b(k+1) &= b_{b1}y(k) + b_{b2}y(k-1) + (a_{b1} + a_{b2})x(k) \\ y_b(k+2) &= (b_{b1}^2 + b_{b2})y(k) + b_{b1}b_{b2}y(k-1) \\ &\quad + (b_{b1} + 1)(a_{b1} + a_{b2})x(k) \\ y_b(k+3) &= (b_{b1}^3 + 2b_{b1}b_{b2})y(k) + (b_{b1}^2 + b_{b1}b_{b2}^2)y(k-1) \\ &\quad + (b_{b1}^2 + b_{b1} + b_{b2} + 1)(a_{b1} + a_{b2})x(k) \quad (3.8) \end{aligned}$$

Dari persamaan  $y_b(k+3)$  dan  $y_r(k+3)$  yang dihubungkan dengan model ICE dan model referensi, maka didapatkan  $e(k+3)$  yang menjadi model referensi untuk kontroler GPC. Dengan nilai *error* tersebut motor DC akan bekerja membantu ICE pada saat akselerasi.

### 3.7.3 Desain *Observer*

Pada sistem pengaturan HEV *output* sistem terbaca oleh tachogenerator yang terpasang pada motor DC. Ketika kedua mesin bekerja nilai yang terbaca merupakan *output* gabungan dari kedua mesin. Hal ini menyebabkan beberapa nilai *output* tidak diketahui. Oleh karena itu dibutuhkan matriks *observer* untuk mengetahui *output* dari ICE dan motor DC.

Dalam penyusunan matriks *observer* maka harus dibentuk representasi *state-space* dari kedua mesin. Dari model matematika hasil identifikasi ICE dimisalkan

$$\begin{aligned} x_1 &= y_1 \\ \dot{x}_1 &= \dot{y}_1 \\ x_2 &= \dot{y}_1 \\ \dot{x}_2 &= \ddot{y}_1 \end{aligned}$$

Sehingga diperoleh

$$s^2 Y_1(s) + 10,87s Y_1(s) + 29,412 Y_1(s) = 29,412 U_1(s)$$
$$\ddot{y}_1 + 10,87\dot{y}_1 + 29,412 y_1 = 29,412 u_1$$

Dengan melakukan substitusi persamaan didapatkan

$$\dot{x}_2 + 10,87x_2 + 29,412 = 29,412u_1$$
$$\dot{x}_2 = -10,87x_2 - 29,412 + 29,412u_1$$

Maka didapatkan representasi *state space* dari ICE

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -29,412 & -10,87 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 29,412 \end{bmatrix} u_1$$
$$y_1 = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Dari model matematika motor DC, dimisalkan :

$$x_3 = y_2$$
$$\dot{x}_3 = \dot{y}_2$$
$$x_4 = \dot{y}_2$$
$$\dot{x}_4 = \ddot{y}_2$$

Sehingga diperoleh

$$s^2 Y_2(s) + 4,5s Y_2(s) + 11,056 Y_2(s) = 11,01 U_2(s)$$
$$\ddot{y}_2 + 4,5\dot{y}_2 + 11,056 y_2 = 11,01 u_2$$

Dengan melakukan substitusi persamaan didapatkan

$$\dot{x}_4 + 4,5\dot{x}_3 + 11,056x_3 = 11,01u_2$$
$$\dot{x}_4 = -4,5\dot{x}_3 - 11,056x_3 + 11,01u_2$$

Maka didapatkan representasi *state space* dari motor DC

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -11,056 & -4,5 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 11,01 \end{bmatrix} u_2$$

$$y_2 = [1 \ 0] \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

Dari representasi *state space* kedua mesin, maka didapatkan representasi *state space* gabungan sebagai berikut :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -29,412 & -10,87 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -11,056 & -4,5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 29,412 & 0 \\ 0 & 0 \\ 0 & 11,01 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = [1 \ 0 \ 1 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Dimana

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -29,412 & -10,87 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -11,056 & -4,5 \end{bmatrix} = \mathbf{A}$$

$$\begin{bmatrix} 0 & 0 \\ 29,412 & 0 \\ 0 & 0 \\ 0 & 11,01 \end{bmatrix} = \mathbf{B}$$

$$[1 \ 0 \ 1 \ 0] = \mathbf{C}$$

Dari matriks A,B, dan C yang ditemukan, maka dilakukan pengujian observability matriks. Dengan persamaan

$$[C^T \ A^T C^T \ (A^T)^2 C^T \ (A^T)^3 C^T]$$

maka didapatkan rank = 4, hal ini membuktikan bahwa sistem tergolong observable.

Selanjutnya dilakukan perancangan *full-order state observer*, oleh karena itu harus ditemukan  $K_e$  yang merupakan matriks *gain observer*. Matriks ini dapat ditemukan dengan beberapa cara. Pada tugas akhir ini digunakan metode Ackermann.

Sebelum menentukan *gain observer*, langkah pertama yang dilakukan adalah mencari pole dari sistem yang diketahui dari *eigenvalue*.

$$|sI - A| = 0$$

$$(s + 5,0783)(s + 5,7917)(s + 2,05 - j2,63711)(s + 2,05 + j2,63711) = 0$$

Dari pole sistem, dari *trial-and-error* dicari persamaan karakteristik yang mendekati pole sistem yaitu

$$\begin{aligned} \phi(s) &= (s + 5 - j2)(s + 5 + j2)(s + 2 - j2)(s + 2 + j2) \\ &= (s + \mu_1)(s + \mu_2)(s + \mu_3)(s + \mu_4) \\ &= s^4 + \alpha_1 s^3 + \alpha_2 s^2 + \alpha_3 s + \alpha_4 \\ \phi(A) &= A^4 + \alpha_1 A^3 + \alpha_2 A^2 + \alpha_3 A + \alpha_4 \end{aligned}$$

Nilai  $K_e$  dapat ditemukan dari persamaan

$$K_e = \phi(A) \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.9)$$



*[Halaman ini sengaja dikosongkan]*

## BAB 4 PENGUJIAN DAN ANALISIS

### 4.1 Gambaran Umum Pengujian Sistem

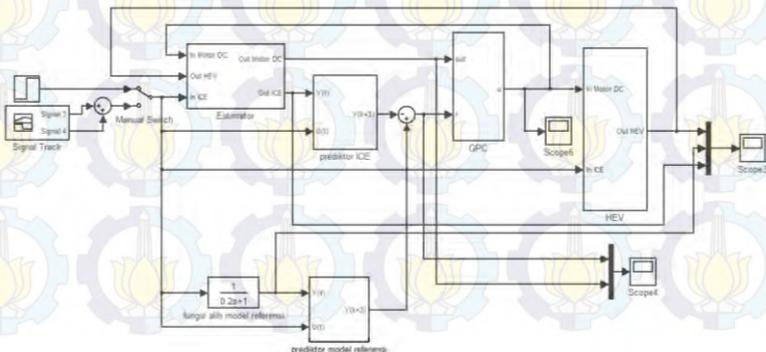
Pada pengerjaan Tugas Akhir ini dilakukan pengujian sistem dengan beberapa kondisi. Pengujian ini dilakukan dalam dua tahap yaitu simulasi dan implementasi. Simulasi dilakukan dengan software Matlab dan implementasi dilakukan dengan Simulator HEV. Beberapa kondisi yang diberikan saat pengujian antara lain :

1. Pengujian mesin bakar berbeban tanpa bantuan motor listrik terhadap *input* berupa sinyal step.
2. Pengujian mesin bakar berbeban tanpa bantuan motor listrik terhadap *input* dengan perubahan amplitudo.
3. Pengujian mesin bakar berbeban dengan bantuan motor listrik terhadap *input* berupa sinyal step.
4. Pengujian mesin bakar berbeban dengan bantuan motor listrik terhadap *input* dengan perubahan amplitudo.

### 4.2 Simulasi Sistem

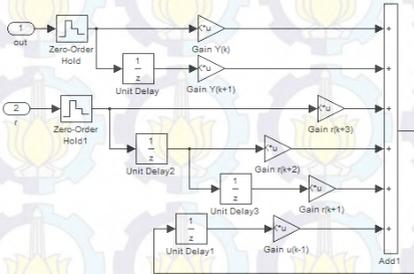
Simulasi merupakan salah satu hal yang harus dilakukan sebelum mengimplementasikan kontroler pada plant. Dengan hasil simulasi yang sesuai akan mempermudah dalam penerapan kontroler agar sistem mencapai kriteria yang diinginkan oleh perancang.

#### 4.2.1 Diagram Blok Simulasi Sistem



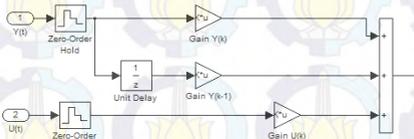
**Gambar 4.1**Diagram Blok Simulasi Kontrol dengan GPC

Kontroler GPC terhubung dengan *output* motor DC dan *error* antara model referensi dan ICE. Dari persamaan kontroler yang didapat pada Bab 3, maka blok diagram kontroler dapat disusun seperti terlihat pada Gambar 4.2

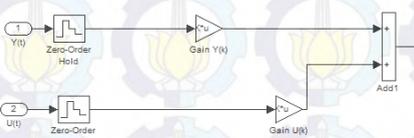


**Gambar 4.2** Diagram Blok Kontroler GPC

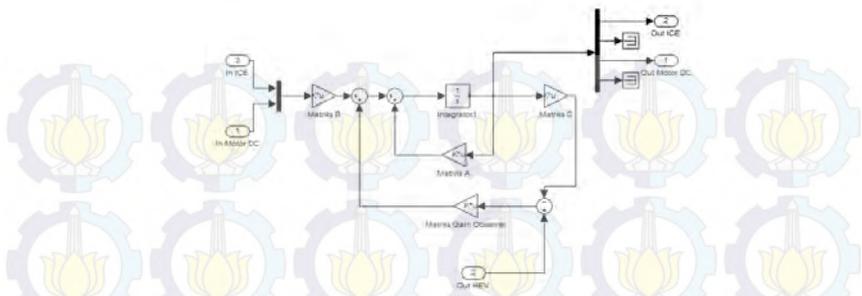
Pada simulasi kontroler GPC terdapat beberapa subsistem yang harus ditambahkan untuk menunjang penggunaan kontroler. Subsistem itu antara lain prediktor ICE, prediktor model referensi, dan Estimator. Blok diagram subsistem prediktor ICE, prediktor model referensi, dan Estimator ditunjukkan pada Gambar 4.3, Gambar 4.4, dan Gambar 4.5



**Gambar 4.3** Diagram Blok Prediktor ICE



**Gambar 4.4** Diagram Blok Prediktor Model Referensi

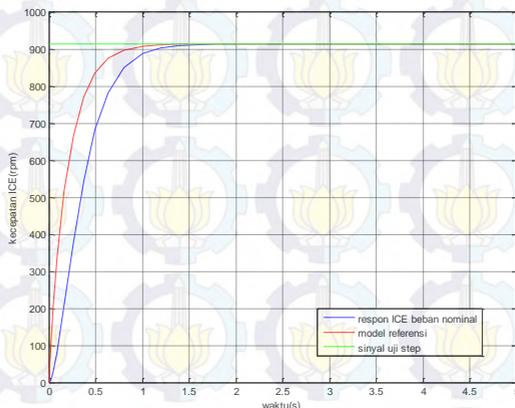


**Gambar 4.5** Diagram Blok *Observer/Estimator*

### 4.2.2. Pengujian Respons

Pengujian terhadap respons sistem dilakukan melalui pengujian ICE tanpa bantuan motor DC, pengujian motor DC dengan kontroler GPC, dan pengujian HEV dengan kontroler GPC. Pengujian ini dilakukan dengan memberikan input step dan input step dengan perubahan amplitudo

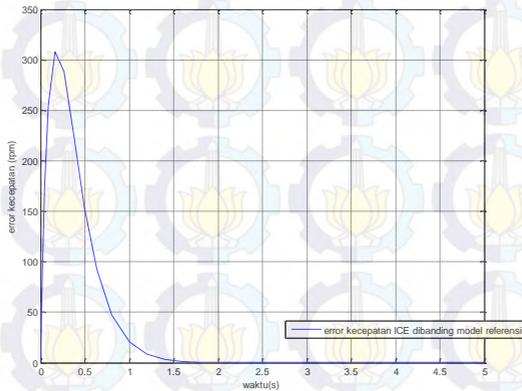
#### 4.2.2.1 Pengujian ICE Tanpa Bantuan Motor DC



**Gambar 4.6** Respons ICE Terhadap Sinyal Uji Step

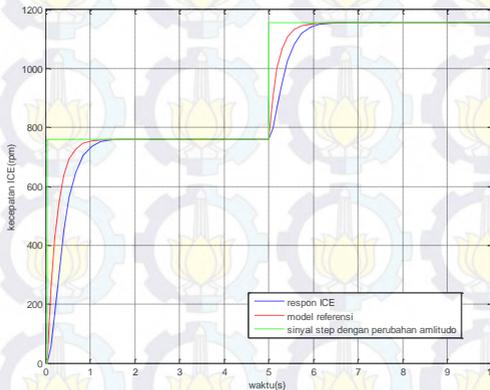
Simulasi pertama yaitu dengan menguji respons ICE terhadap set point berupa nilai step. Respons tersebut dapat dilihat pada Gambar 4.6. dari gambar tersebut terlihat bahwa terdapat *error* antara respons

kecepatan ICE dengan model referensi. *Error* terbesar yaitu bernilai 307,84 rpm. Grafik *error* ditunjukkan pada Gambar 4.7.



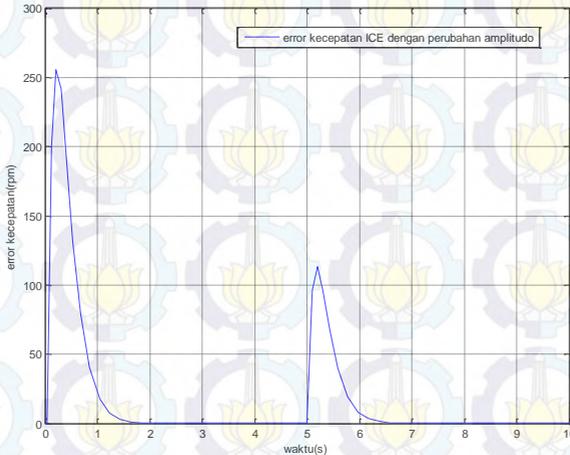
**Gambar 4.7** Sinyal *Error* antara Kecepatan ICE dan Model Referensi

Simulasi selanjutnya yaitu dengan memberikan sinyal uji dengan perubahan amplitudo. Hal ini dilakukan dengan memberikan set point yang berasal dari *signal builder* yang telah disesuaikan dengan pengukuran output ICE terhadap tegangan potensiometer. Respons ICE dengan perubahan amplitudo ditunjukkan pada Gambar 4.8



**Gambar 4.8** Respons ICE terhadap Sinyal Step dengan Perubahan Amplitudo

Dari pengujian dengan memberikan sinyal step dengan perubahan amplitudo, dapat terlihat bahwa terdapat *error*. *Error* terbesar bernilai 255 rpm. Dan ketika amplitudo berubah *error* terbesar bernilai 133,55rpm.

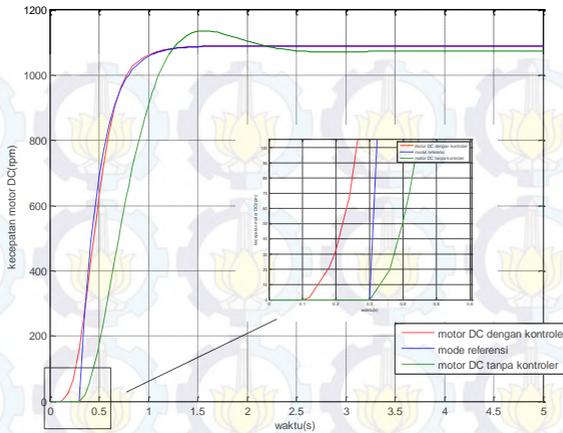


**Gambar 4.9** Sinyal *Error* antara Kecepatan ICE dan Model Referensi dengan Sinyal Uji Step dengan Perubahan Amplitudo

Dari kedua pengujian diatas dapat terlihat bahwa apabila ICE bekerja sendiri, ICE tidak dapat mencapai model referensi yang diinginkan. Oleh karena itu dibutuhkan motor DC untuk membantu ICE mencapai tingkat akselerasi yang menyerupai model referensi.

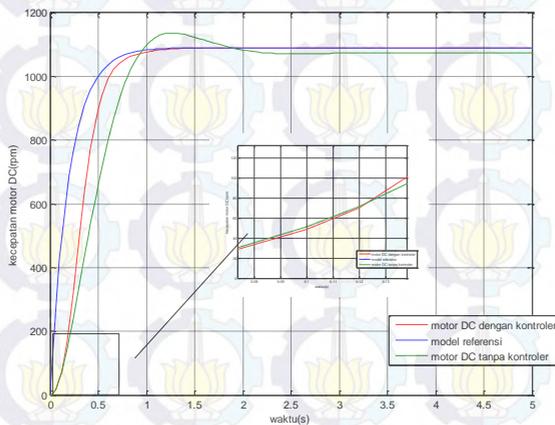
#### 4.2.2.2 Pengujian Motor DC dengan Metode GPC

Sebelum pengujian metode GPC terhadap plant hybrid, maka terlebih dahulu dilakukan pengujian terhadap motor DC. Pengujian dilakukan untuk melihat respons motor DC ketika diberikan kontroler GPC. Motor DC dapat membantu ICE dalam mencapai model referensi apabila ketika motor DC bekerja sendiri dapat mencapai model referensi. Terdapat dua cara untuk mengetahui kontroler GPC dapat mencapai respons yang diinginkan atau tidak. Cara yang pertama yaitu dengan menganggap sistem mengetahui respons yang akan datang, sehingga kontroler dapat memulai aksi kontrol beberapa detik sebelum sistem mulai. Respons motor DC dengan cara pertama ditunjukkan pada Gambar 4.10.



**Gambar 4.10** Motor DC dengan Kontroler GPC Saat Informasi Masa Depan Diketahui

Cara kedua yaitu dengan menganggap sistem tidak mengetahui respons yang akan datang, sehingga ketika motor DC bekerja kontroler belum mulai memprediksi kedepan. Kontroler akan mulai bekerja memprediksi ketika mencapai waktu satu *time sampling*. Respons motor DC dengan cara kedua ditunjukkan pada Gambar 4.11.

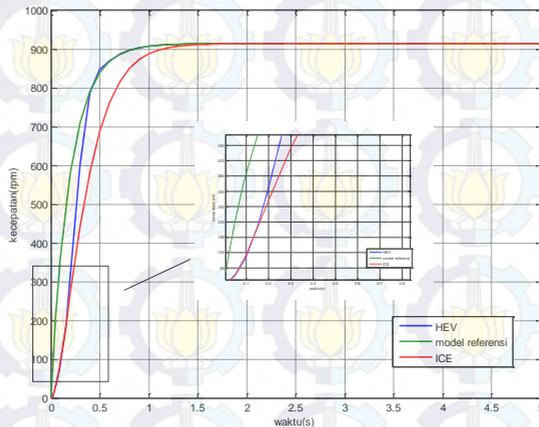


**Gambar 4.11** Motor DC dengan Kontroler GPC Saat Informasi Masa Depan Tidak Diketahui

Gambar 4.10 dan Gambar 4.11 menunjukkan bahwa kontroler GPC yang dirancang pada motor DC dapat mencapai model referensi yang diinginkan.

#### 4.2.2.3 Pengujian HEV dengan Metode GPC

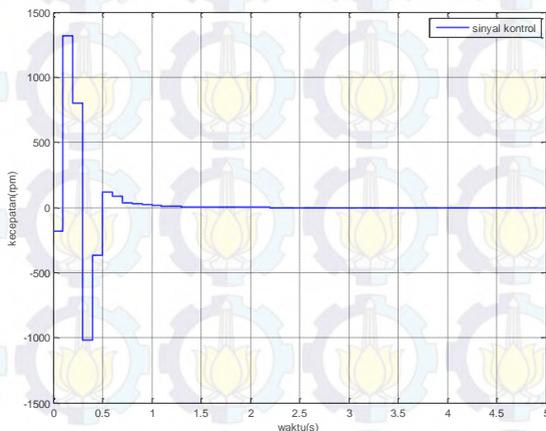
Sistem pengaturan traksi pada HEV melibatkan motor DC sebagai penggerak sekunder. Oleh karena itu motor DC akan bekerja ketika terdapat *error* antara kecepatan ICE dan model referensi. Pengujian ini dilakukan dengan memberikan kontroler GPC dengan 3 prediksi kedepan. Respons HEV dengan kontroler GPC 3 prediksi kedepan ditunjukkan pada Gambar 4.12. Gambar tersebut menunjukkan bahwa kontrol GPC yang dirancang sudah dapat melaksanakan aksi kontrol setelah menjalani *satu time sampling* (0,1s). Setelah melalui 0,1s, kontroler bekerja mendekati model referensi. Walau tertinggal diawal, respons plant yang telah diberi kontroler dapat mendahului model referensi setelah 0,4s dan mempunyai *rise time* sebesar 0,353s.



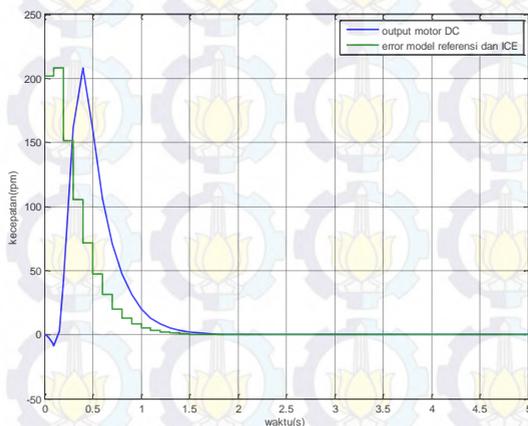
**Gambar 4.12** Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan *Input* Sinyal Step

Dari kontroler GPC tersebut menghasilkan sinyal kontrol berbentuk diskrit seperti yang ditunjukkan pada Gambar 4.13. Sinyal kontrol menunjukkan bahwa kontroler memberikan *input* kecepatan sebesar 1288rpm pada motor DC saat 0,1s. Saat 0,3s kontroler menghasilkan sinyal kontrol sebesar -968,43rpm. Hal ini menunjukkan bahwa kontroler bekerja secara maksimal setelah dapat melakukan prediksi.

*Output* motor DC ditunjukkan pada Gambar 4.14. gambar tersebut menunjukkan bahwa motor DC dapat mengikuti sinyal referensi dan motor DC akan berhenti saat kecepatan ICE sama dengan model referensi. Kecepatan maksimal yang dihasilkan motor DC sebesar 206,9s.

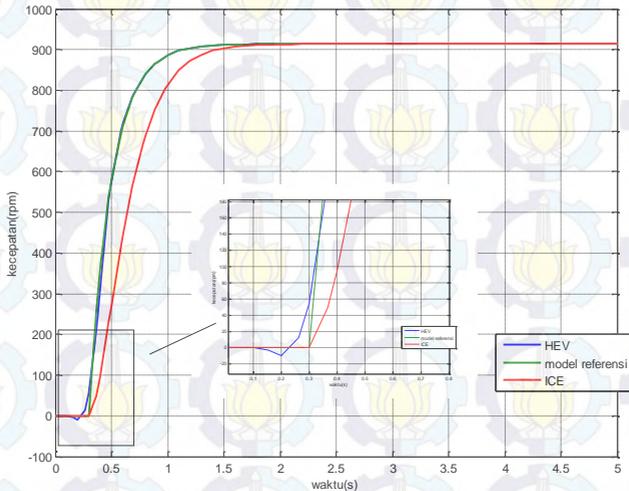


**Gambar 4.13** Sinyal Kontrol GPC 3 Prediksi Kedepan dengan *Input* Sinyal Step



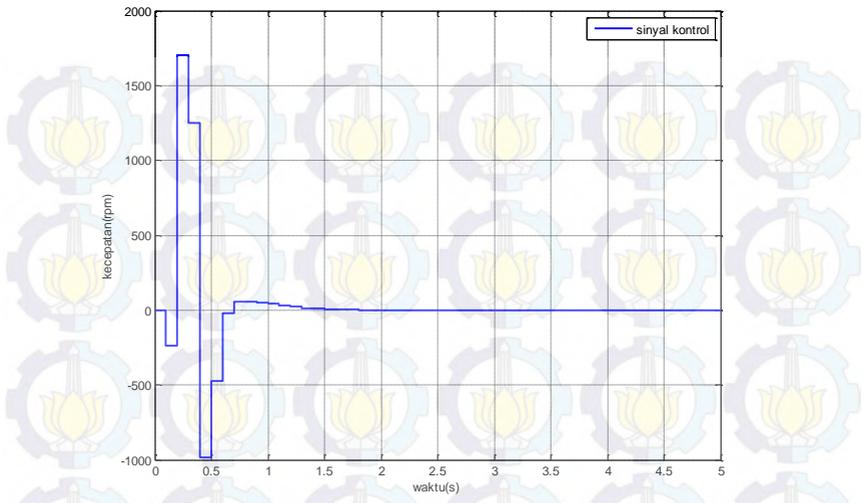
**Gambar 4.14** *Output* Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan *Input* Sinyal Step

Pada simulasi dengan *input* sinyal step juga dilakukan pengujian dengan mengganggu sistem mengetahui *output* yang akan datang sejak awal start. Respons sistem dengan pengujian seperti ini ditunjukkan pada Gambar 4.15. Gambar tersebut menunjukkan bahwa kontroler mengetahui respons 3 langkah kedepan, sehingga sebelum sistem start kontroler sudah memulai aksi kontrolnya dengan mengaktifkan motor DC. Dengan mengetahui informasi masa depan, respons sistem dapat mengikuti model referensi yaitu memiliki *rise time* sebesar 0,44s

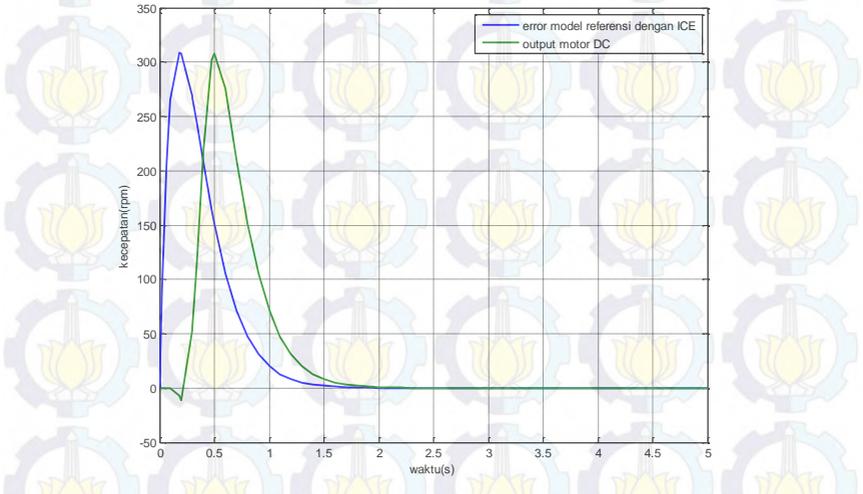


**Gambar 4.15** Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan *Input* Sinyal Step dan Informasi Masa Depan.

Kinerja motor DC dengan informasi masa depan ditunjukkan pada Gambar 4.17. gambar tersebut menunjukkan bahwa motor DC dapat bekerja mengikuti sinyal referensi yang berupa *error* antara model referensi dengan ICE. Kecepatan maksimum motor DC sebesar 306,3rpm. Untuk mencapai respons demikian motor DC diberikan sinyal kontrol dari kontroler GPC seperti yang ditunjukkan pada Gambar 4.16. Besar sinyal kontrol tersebut tertinggi sebesar 1664rpm dan terendah sebesar -926,28rpm.

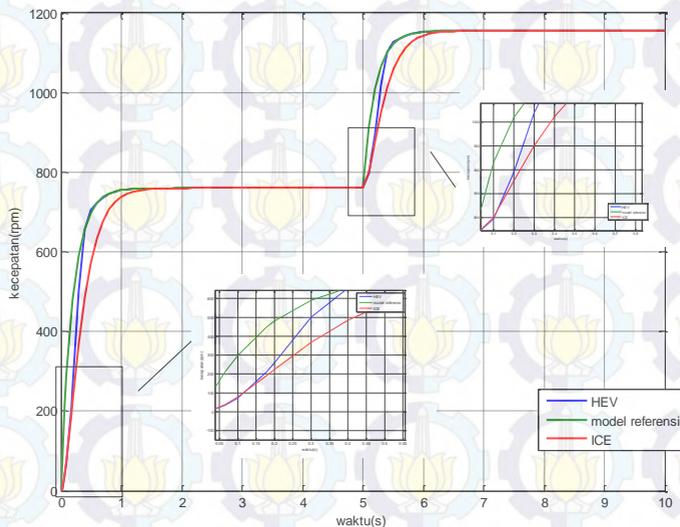


**Gambar 4.16** Sinyal Kontrol GPC 3 Prediksi Kedepan dengan *Input* Sinyal Step dan Informasi Masa Depan.



**Gambar 4.17** *Output* Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan *Input* Sinyal Step dan Informasi Masa Depan.

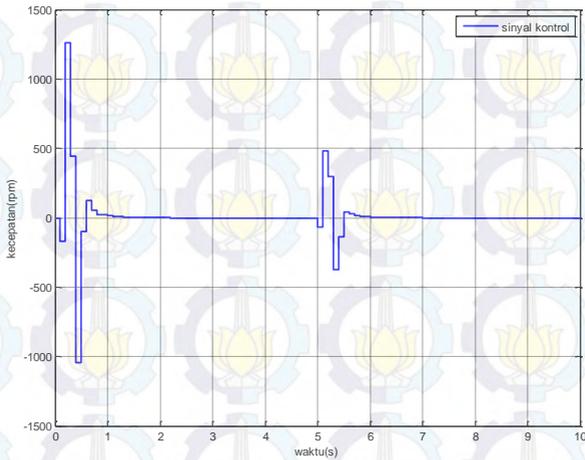
Saat diberikan sinyal uji berupa sinyal step dengan perubahan amplitudo, respons sistem ditunjukkan pada Gambar 4.18. Gambar tersebut menunjukkan bahwa sistem dapat mengikuti model referensi. Sama halnya ketika diberi sinyal uji step, setelah melalui 0,1s, kontroler bekerja mendekati model referensi. Walau tertinggal diawal, respons plant yang telah diberi kontroler dapat mendahului model referensi setelah 0,4s dan memiliki *rise time* sebesar 0,34s. Untuk kenaikan amplitudo plant yang telah diberi kontroler dapat mendahului model referensi setelah 0,4s dan memiliki *rise time* sebesar 0,35s sama seperti saat step pertama.



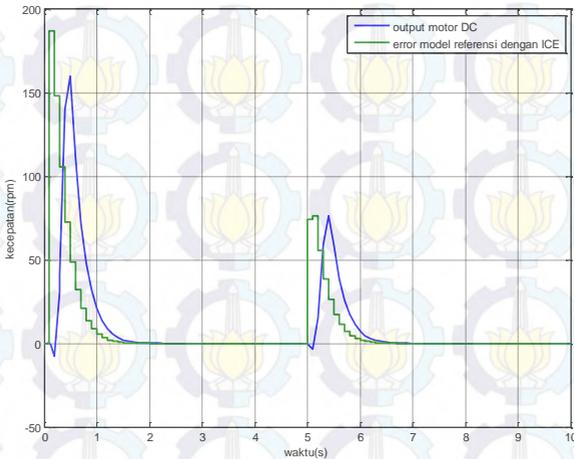
**Gambar 4.18** Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo *Input* Sinyal Step

Seperti halnya ketika diberikan sinyal uji step, saat diberikan sinyal step dengan amplitudo yang berbeda, sinyal kontrol akan menyesuaikan dengan kenaikan sinyal step. Sinyal kontrol mengalami kenaikan saat 0,1s setelah perubahan sinyal masukan dan bernilai 1071rpm. Saat 0,3s sinyal kontrol turun dan bernilai -805,26rpm. Hal ini menunjukkan bahwa kontroler membutuhkan waktu untuk menyesuaikan nilainya dengan model referensi. *Output* motor DC

menunjukkan bahwa motor DC membantu ICE saat keadaan akselerasi. *Output* motor DC ini ditunjukkan pada Gambar 4.17

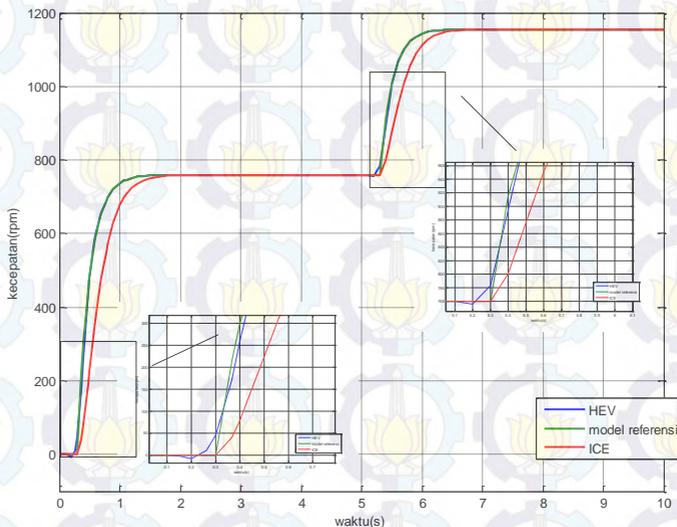


**Gambar 4.19** Sinyal Kontrol GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo *Input* Sinyal Step



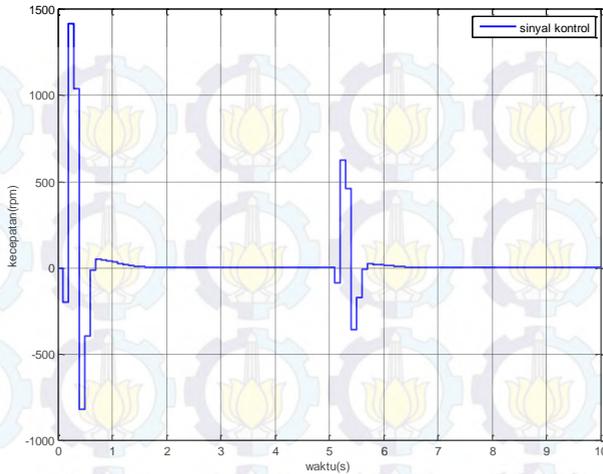
**Gambar 4.20** *Output* Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo *Input* Sinyal Step

Pengujian sistem dengan perubahan amplitudo juga dilakukan untuk sistem dengan pengetahuan terhadap informasi masa depan. Respons sistem dengan pengujian ini ditunjukkan pada Gambar 4.21. Gambar tersebut menunjukkan bahwa sistem HEV dapat mengikuti model referensi. Perubahan amplitudo tidak mempengaruhi respons sistem. Pada step yang pertama respons sistem memiliki *rise time* sebesar 0,44s serta pada step kedua pertama respons sistem memiliki *rise time* yang sama saat step pertama, yaitu sebesar 0,6s dan 0,44s

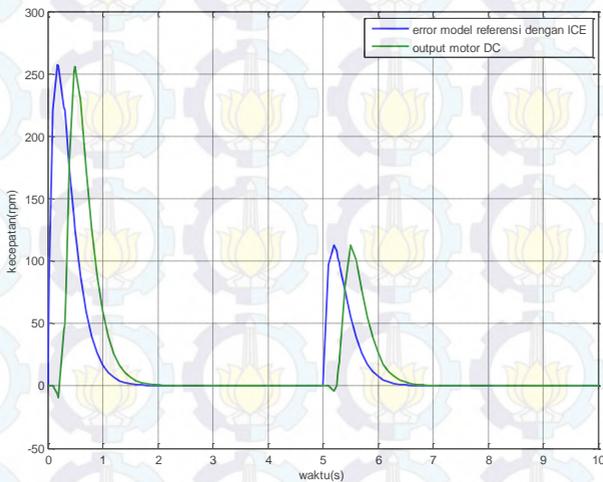


**Gambar 4.21** Respons HEV dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo *Input* Sinyal Step dan Informasi Masa Depan

Respons sinyal kontrol yang dihasilkan oleh kontroler GPC dengan informasi masa depan ditunjukkan pada Gambar 4.22. Pada step pertama, sinyal kontrol tertinggi sebesar 1383,65rpm dan terendah sebesar -770,2rpm, sedangkan pada step kedua sinyal kontrol tertinggi sebesar 719rpm dan terendah sebesar -400,2rpm.. Dengan sinyal kontrol tersebut motor DC bekerja pada step pertama dengan kecepatan maksimum sebesar 254,6 rpm dan pada step kedua dengan kecepatan maksimum sebesar 132,33 rpm, seperti ditunjukkan pada Gambar 4.23



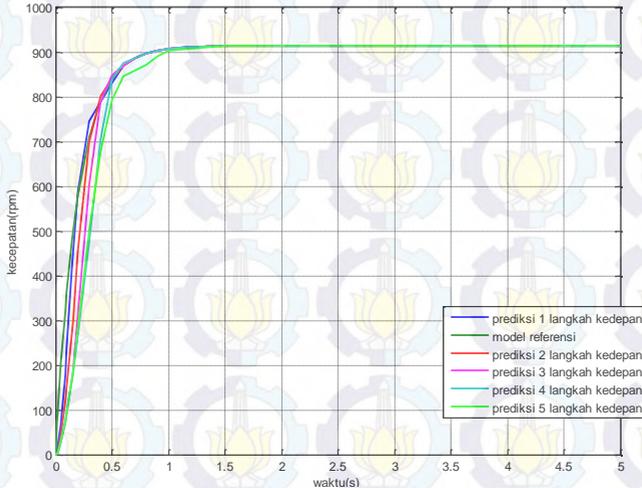
**Gambar 4.22** Sinyal Kontrol GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo *Input* Sinyal Step dan Informasi Masa Depan



**Gambar 4.23** *Output* Motor DC dengan Kontroler GPC 3 Prediksi Kedepan dengan Perubahan Amplitudo *Input* Sinyal Step

Dalam pengujian respons sistem, selain menggunakan kontroler GPC dengan prediksi 3 langkah kedepan, diuji pula dengan kontroler GPC

dengan prediksi 1, 2, 3, 4, dan 5 langkah kedepan. Hasil respons sistem dengan beberapa prediksi dengan *input* step ditunjukkan pada Gambar 4.34



**Gambar 4.24** Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan *Input* Sinyal Step

Gambar tersebut menunjukkan bahwa semakin rendah tingkat prediksi, maka semakin cepat mengikuti model referensi tapi memiliki nilai yang melebihi model referensi pada saat tertentu. *Rise time* dan *error* pada saat *transien* dari hasil respons dengan beberapa tingkat prediksi ditunjukkan pada Tabel 4.1

**Tabel 4.1** Nilai *Rise Time* dan *Error Output* HEV dengan Prediksi 1 hingga 5 Langkah Kedepan dan *Input* Sinyal Step

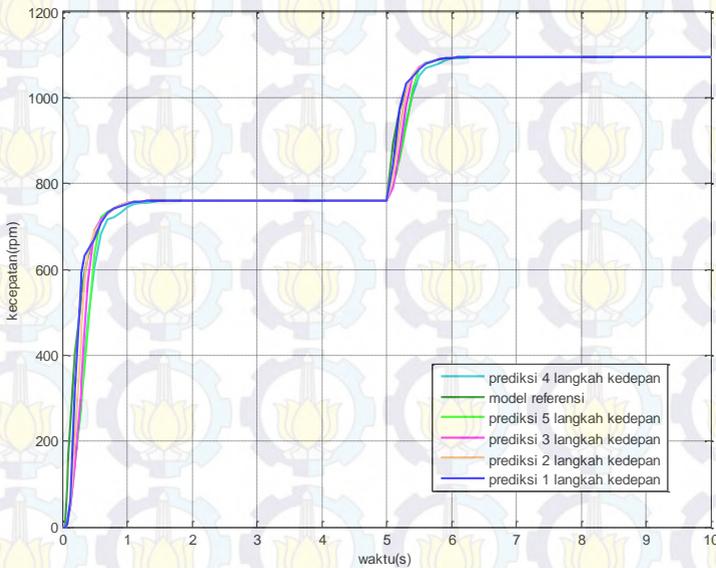
Tingkat prediksi	<i>Rise time</i> (s)	<i>Error transien</i> (%)
1	0,423	2,04
2	0,3787	2,99
3	0,3533	4,56
4	0,3806	5,42
5	0,4402	5,34

Setelah dilakukan pengujian dengan *input* step, selanjutnya dilakukan pengujian terhadap *input* step dengan perubahan amplitudo. Hasil respons pengujian dengan kontroler GPC 1,2,4, dan 5 langkah kedepan

ditunjukkan pada Gambar 4.25. respons tersebut memiliki *rise time* dan *error* pada keadaan *transien* yang ditunjukkan pada Tabel 4.2.

**Tabel 4.2** Nilai *Rise Time* dan *Error Output* HEV dengan Prediksi 1 hingga 5 Langkah Kedepan dan Input Sinyal Step dengan Perubahan Amplitudo

Tingkat prediksi	<i>Rise time</i> (s)	<i>Error transien</i> (%)	<i>Rise time</i> (s)	<i>Error transien</i> (%)
1	0,4067	2,50	0,43945	1,55
2	0,3464	3,48	0,3902	2,8
3	0,33755	4,93	0,3573	4,53
4	0,3927	5,71	0,39	5,49
5	0,4541	5,62	0,4561	5,4



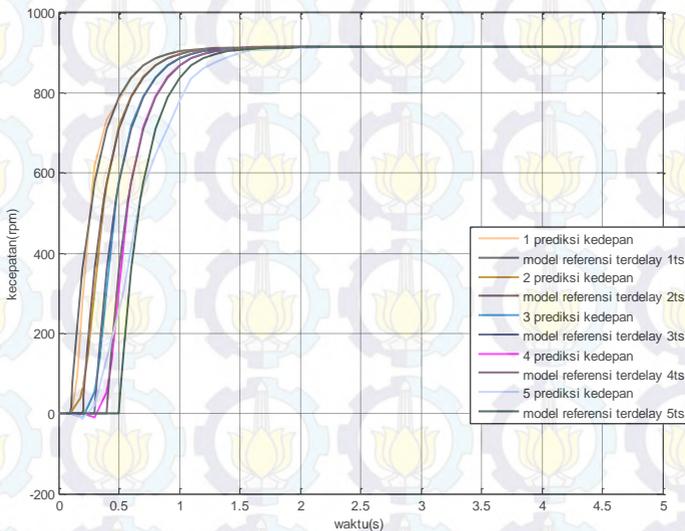
**Gambar 4.25** Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan Perubahan Amplitudo *Input Sinyal Step*

Pengujian sistem ini dengan kontroler GPC 1 hingga 5 langkah kedepan juga dilakukan pada sistem dengan pengetahuan terhadap informasi masa depan. Hasil respons pengujian dengan kontroler GPC 1,2,3,4, dan 5 langkah kedepan ditunjukkan pada Gambar 4.26. respons tersebut

memiliki *rise time* dan *error* pada keadaan *transien* yang ditunjukkan pada Tabel 4.2.

**Tabel 4.3** Nilai *Rise Time* dan *Error Output* HEV dengan Prediksi 1 hingga 5 Langkah Kedepan, Input Sinyal Step dan Informasi Masa Depan

Tingkat prediksi	<i>Rise time</i> (s)	<i>Error transien</i> (%)
1	0,4284	1,16
2	0,4404	1,30
3	0,4402	1,15
4	0,4407	1,21
5	0,7172	5,51

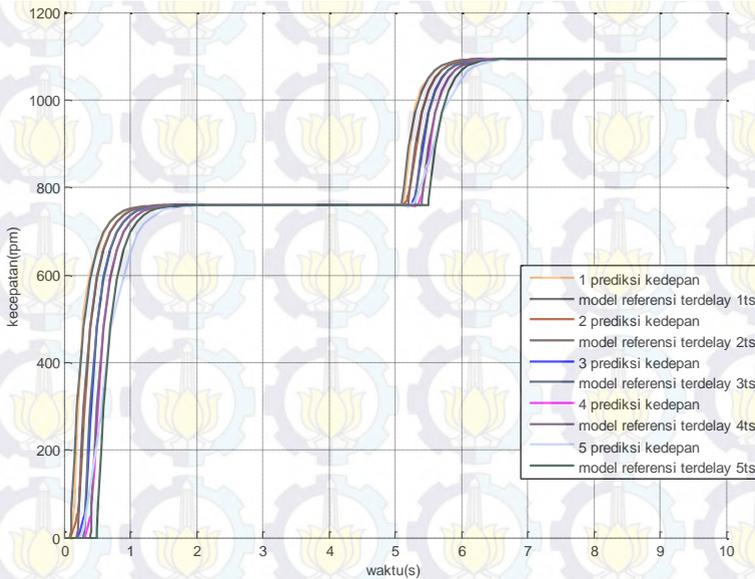


**Gambar 4.26** Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan *Input* Sinyal Step dan Informasi Masa Depan

Setelah dilakukan pengujian dengan *input* step, selanjutnya dilakukan pengujian terhadap *input* step dengan perubahan amplitudo. Hasil respons pengujian dengan kontroler GPC 1,2,3,4, dan 5 langkah kedepan ditunjukkan pada Gambar 4.27. respons tersebut

**Tabel 4.4** Nilai *Rise Time* dan *Error Output* HEV dengan Prediksi 1 hingga 5 Langkah Kedepan, Input Sinyal Step dengan Perubahan Amplitudo dan Informasi Masa Depan

Tingkat prediksi	<i>Rise time</i> (s)	<i>Error transien</i> (%)	<i>Rise time</i> (s)	<i>Error transien</i> (%)
1	0,422	1,15	0,4289	1,28
2	0,4575	1,29	0,4423	0,95
3	0,4393	1,15	0,4426	0,84
4	0,4397	1,21	0,4432	0,88
5	0,7078	5,52	0,7155	4,4



**Gambar 4.27** Respons HEV dengan Kontroler GPC 1 hingga 5 Prediksi Kedepan dengan Perubahan Amplitudo *Input* Sinyal Step dan Informasi Masa Depan

Dari pengujian dengan beberapa tingkat prediksi kedepan, dapat ditunjukkan bahwa kontroler dengan prediksi satu langkah kedepan menghasilkan *error transien* yang lebih kecil dibandingkan dengan tingkat prediksi yang lain. Tetapi kontroler dengan prediksi 3 langkah kedepan juga menunjukkan respons yang baik dalam hal *rise time*

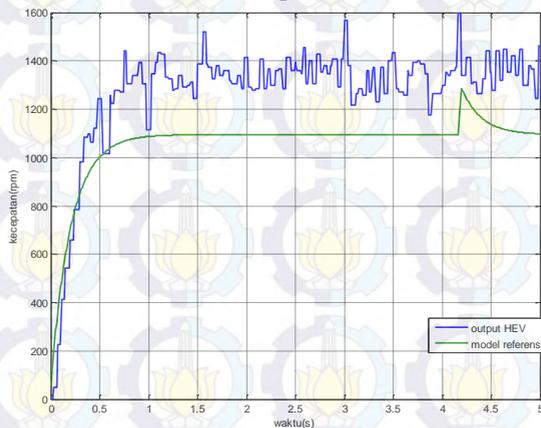
walaupun memiliki nilai *error transien* yang lebih tinggi dibandingkan dengan prediksi satu atau dua langkah kedepan.

## 4.2 Implementasi Sistem

Setelah mensimulasikan kontroler, langkah selanjutnya adalah mengimplementasikan kontroler pada simulator HEV. Implementasi dilakukan dengan menghubungkan potensiometer dan tachogenerator pada mikrokontroler arduino. Mikrokontroler mengolah data dan menampilkan ke Simulink Matlab. Selanjutnya matlab akan mengirimkan sinyal kontrol melalui komunikasi serial arduino dan arduino akan mengeluarkan sinyal kontrol berupa *output* PWM. *Output* PWM kemudian dihubungkan dengan *driver* motor DC. *Driver* ini menggerakkan motor DC berdasarkan sinyal kontrol yang dikehendaki.

### 4.2.2 Pengujian kontroler GPC Satu Langkah Kedepan

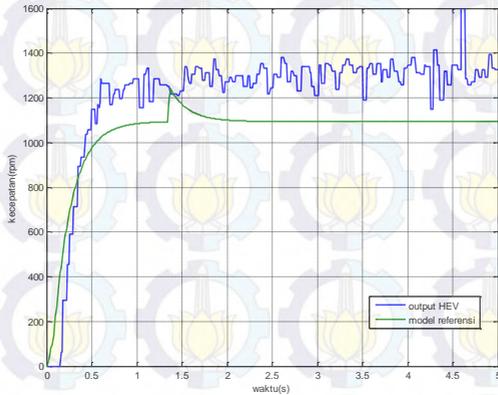
Implementasi kontroler GPC pada simulator HEV dilakukan dengan memberikan sinyal *input* step dari pedal. Pengujian pertama dilakukan dengan menggunakan kontroler GPC dengan prediksi 1 langkah kedepan, respons hasil implementasi dengan kontroler GPC 1 langkah kedepan ditunjukkan pada Gambar 4.28. Gambar tersebut menunjukkan bahwa *output* HEV dapat mengikuti model referensi pada saat *transien*. respons HEV memiliki *rise time* sebesar 0,345s dan memiliki *error* sebesar 7,2% terhadap model referensi.



**Gambar 4.28** Respons Hasil Implementasi dengan Kontroler GPC dengan Prediksi Satu Langkah Kedepan

### 4.2.3 Pengujian kontroler GPC Tiga Langkah Kedepan

Pengujian selanjutnya yaitu dengan memberikan kontroler GPC dengan prediksi 3 langkah ke depan. Respons hasil implementasi ditunjukkan pada Gambar 4.29. gambar tersebut menunjukkan bahwa *output* HEV dapat mengikuti model referensi pada saat *transien*. respons HEV memiliki *rise time* sebesar 0,29s dan memiliki *error* sebesar 11,5% terhadap model referensi.



**Gambar 4.29** Respons Hasil Implementasi dengan Kontroler GPC dengan Prediksi 3 Langkah Kedepan

## BAB 5 PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis, dapat diperoleh kesimpulan sebagai berikut :

- a. Kontroler GPC dengan prediksi 3 langkah kedepan memiliki *rise time* yang lebih cepat. Sehingga kontroler ini cocok untuk akselerasi
- b. Kontroler GPC tanpa informasi masa depan mulai mengikuti model referensi setelah mencapai 0,1s (satu *time sampling*) dan memiliki nilai yang sama dengan model referensi setelah 0,4s.
- c. Kontroler GPC dengan informasi masa depan mampu mengikuti model referensi dari awal hingga akhir dan memulai aksi kontrol beberapa saat sebelum *plant* bekerja.
- d. Dalam implementasi kontroler GPC dengan prediksi 3 langkah kedepan memiliki *rise time* sebesar 0,29s dan memiliki *error* sebesar 11,5%

### 5.2 Saran

Dalam melanjutkan penelitian yang berkaitan dengan Tugas Akhir ini, disarankan beberapa hal :

- a. Penggunaan kontroler GPC cocok untuk plant yang memiliki pengetahuan tentang informasi yang akan datang
- b. Perancangan kontroler harus diberikan batasan, agar sinyal kontrol selalu bernilai positif.



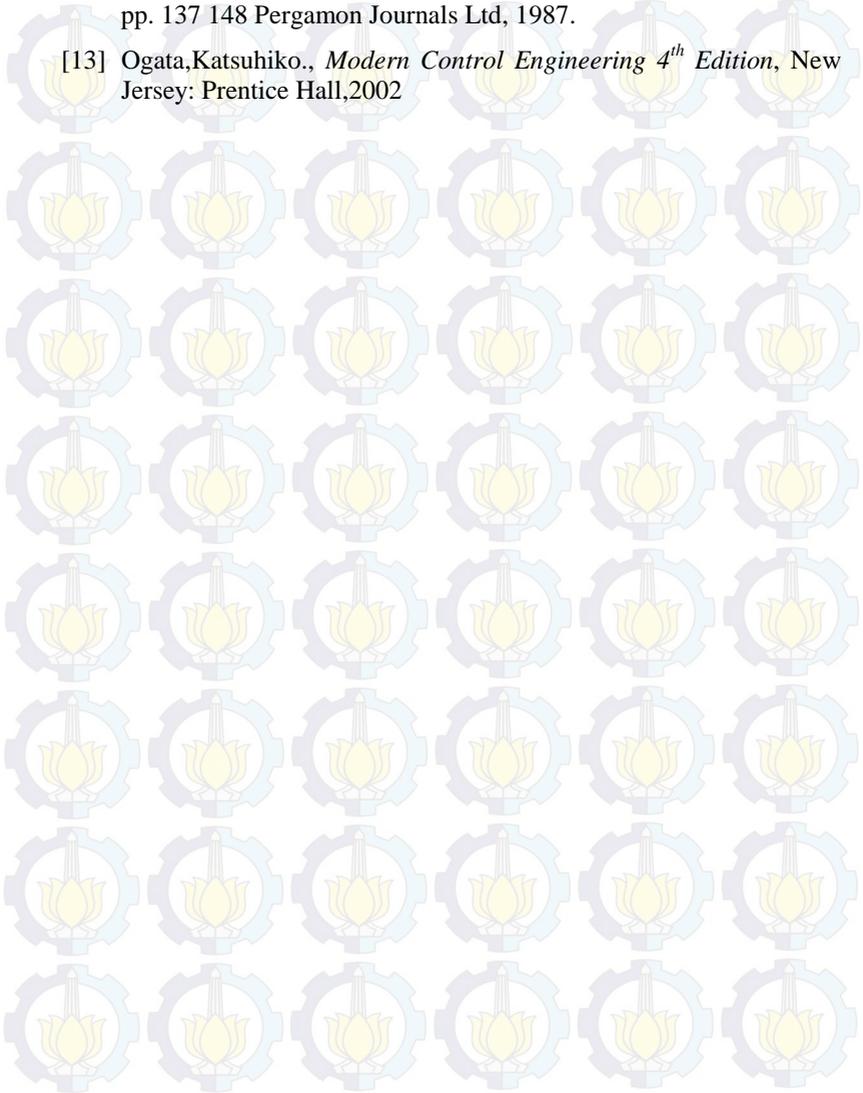
*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] Sriwidodo, Wahyu. “Perancangan dan Implementasi Pengendalian Traksi Berbasis Neuro Fuzzy untuk Simulator Kendaraan Hybrid pada Mode Akselerasi”, *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
- [2] Satrio, Galih. “Desain dan Implementasi Kontroler Optimal Berbasis Neuro Fuzzy untuk Pengendalian Simulator Hybrid Electric Vehicle”, *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
- [3] Momoh, Omonowo D., Omoigui, Michael O. “An Overview of Hybrid Electric Vehicle Technology”, *IEEE*, 978-1-4244-2601, September, 2009.
- [4] Salazar, Fernando. *Internal Combustion Engine*, Departement of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN, 1998.
- [5] Massimo, Ghoini. *DC Motors and Generator*, Polytecnic University of Milan, Italy, 2000
- [6] Maurya, Virendra Kumar, dkk. “Eddy Current Breaking Emnbedded System”, *International Journal of Applied Engineering and Technology ISSN:2277-212X*. Januari, 2011
- [7] \_\_\_\_\_, *Index of Arduino Boards* <URL: <http://www.arduino.cc/en/Main/arduinoBoardUno>> , Desember, 2013
- [8] \_\_\_\_\_, *Help of MATLAB* <URL: <http://www.mathworks.com/help/matlab/>> , Desember, 2013
- [9] Söderström, Torsten., Stoica, Petre., *System Identification*, UK: Prentice Hall, 2001
- [10] Ing. Pavel Jakoubek, “Experimental Identification of Stabile Nonoscillatory Systems from Step-Responses by Selected Methods”, *Konference Studenske Tvuurci-Cinnosti*, 2009.
- [11] Haber, Robert., Bars, Ruth., Schmitz, Ulrich., *Predictive Control in Process Engineering*, Weinheim, Germany: WILEY-VCH Verlag GmbH & Co. KgaA, 2011

[12] Clarke, D.W., Mohtadi, C., Tuffs, P.S., “Generalized Predictive Control-Part I. The Basic Algorithm ” *Automatica*, VoL 23, No. 2, pp. 137 148 Pergamon Journals Ltd, 1987.

[13] Ogata,Katsuhiko., *Modern Control Engineering 4<sup>th</sup> Edition*, New Jersey: Prentice Hall,2002



## LAMPIRAN

### A. Program Arduino

```
char buff;  
char string[4];  
int value,t;  
int stat=0;  
int data_analog=0;  
int data_analog1=0;  
char disp[1];  
  
void setPwmFrequency(int pin, int divisor) {  
    byte mode;  
    if(pin == 5||pin == 6||pin == 9||pin == 10){  
        switch(divisor) {  
            case 1: mode = 0x01; break;  
            case 8: mode = 0x02; break;  
            case 64: mode = 0x03; break;  
            case 256: mode = 0x04; break;  
            case 1024: mode = 0x05; break;  
            default: return;  
        }  
        if(pin == 5 || pin == 6) {  
            TCCR0B = TCCR0B & 0b11111000 | mode;  
        } else {  
            TCCR1B = TCCR1B & 0b11111000 | mode;  
        }  
    } else if(pin == 3 || pin == 11) {  
        switch(divisor) {  
            case 1: mode = 0x01; break;  
            case 8: mode = 0x02; break;  
            case 32: mode = 0x03; break;  
            case 64: mode = 0x04; break;  
            case 128: mode = 0x05; break;  
            case 256: mode = 0x06; break;  
            case 1024: mode = 0x07; break;  
            default: return;  
        }  
        TCCR2B = TCCR2B & 0b11111000 | mode;  
    }  
}
```

```

}

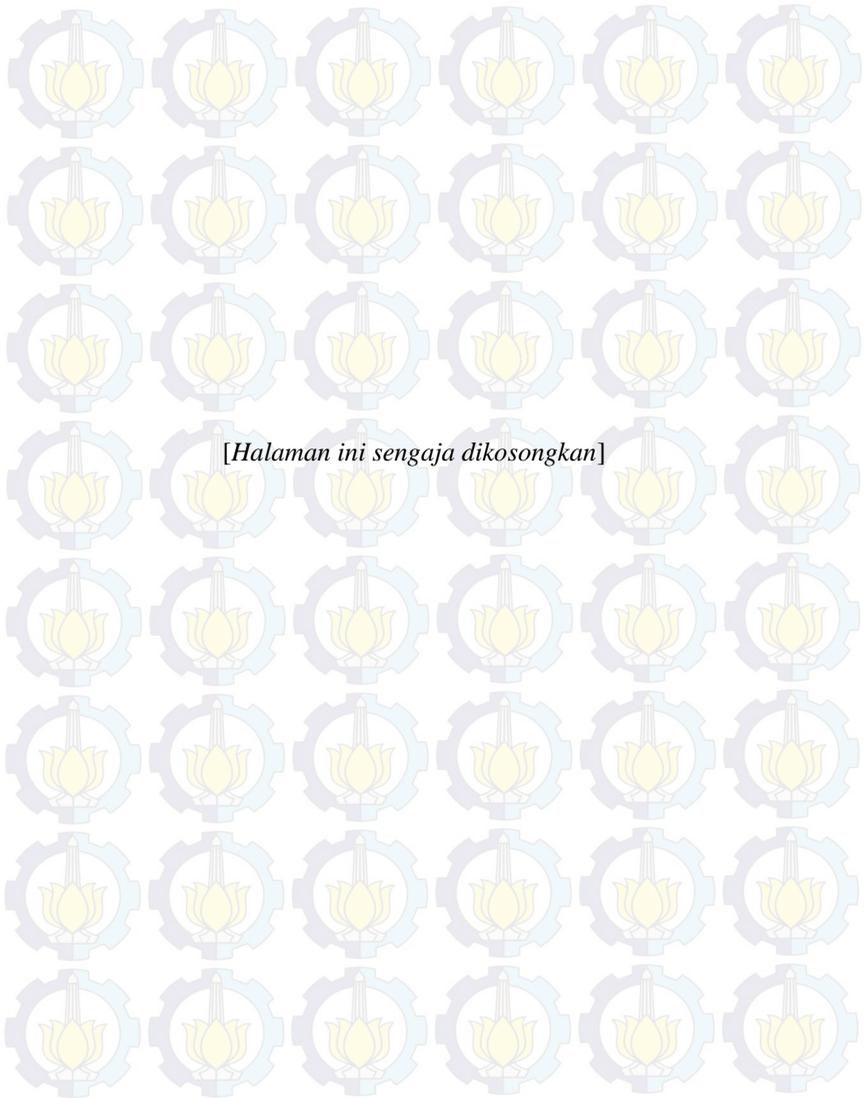
void setup()
{
  Serial.begin(56000);
  setPwmFrequency(5,1024);
  pinMode(8, OUTPUT);
}

void baca_serial()
{
  while (Serial.available())
  {
    buff = Serial.read();
    if (buff == '*')
    {
      stat = 1;
    }
    else if (stat==1)
    {
      string[0] = buff;
      stat=2;
    }
    else if (stat==2)
    {
      string[1] = buff;
      stat=3;
    }
    else if (stat==3)
    {
      string[2] = buff;
      stat=4;
    }
    else if (stat==4 || buff=='#')
    {
      stat=0;
      value=atoi(string);
      analogWrite(5,value);
      Serial.write(value);
    }
    //
  }
}

void loop()
{

```

```
    baca_serial();  
    // if (t <= 10) digitalWrite(8,LOW);  
    // if (t > 10)  
    // {  
    //     digitalWrite(8,HIGH);  
    //     if (t>=20) t=0;  
    // }  
  
    data_analog = analogRead(0)/4;  
    delay(1);  
    data_analog1 = analogRead(2)/4;  
  
    sprintf(displ,"%3d#%3d",data_analog,data_analog1);  
    Serial.print(displ);  
    // t = t + 1;  
    delay(5);  
}
```



## RIWAYAT PENULIS



Ilmiyah Elrosa Citra Resmi lahir di Gresik tanggal 16 September 1992, merupakan anak kedua dari Ilmu Yaqien dan Mulia Iriani. Setelah lulus dari SMA Negeri 1 Gresik tahun 2010, melanjutkan studi di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember (ITS) Surabaya pada tahun yang sama.

Selama studi penulis aktif di Departemen Riset dan Teknologi Himpunan Mahasiswa Teknik Elektro dan pernah menjadi asisten praktikum sistem pengaturan analog dan praktikum otomasi sistem. Pada bulan Januari 2014, penulis mengikuti seminar dan ujian Tugas Akhir di Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, ITS Surabaya sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro.

