

**IMPLEMENTASI KONTROLER PID KASKADE DAN KALMAN
FILTER PADA SISTEM KESETIMBANGAN ROBOT BERODA DUA
BERBASIS MIKROKONTROLER ARDUINO ARM**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



GABRIEL ANDRIANO BRAMANTYO
NIM. 125060307111002

UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG

2016

LEMBAR PENGESAHAN

**IMPLEMENTASI KONTROLER PID KASKADE DAN KALMAN
FILTER PADA SISTEM KESETIMBANGAN ROBOT BERODA DUA
BERBASIS MIKROKONTROLER ARDUINO ARM**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



GABRIEL ANDRIANO BRAMANTYO

NIM. 125060307111002

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
pada tanggal 28 Oktober 2016

Dosen Pembimbing I

Dosen Pembimbing II

Goegoes Dwi Nusantoro, S.T., M.T.

NIP. 19711013 200604 1 001

Rahmadwati, S.T., M.T., Ph.D.

NIP 19771102 200604 2 003

Mengetahui,

Ketua Jurusan Teknik Elektro

M. Azis Muslim, S.T., M.T., Ph.D.

NIP. 19741203 200012 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas dalam Naskah Skripsi adalah asli dari pemikiran saya, tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik pada perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata dalam naskah Skripsi ini dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai peraturan perundang-undangan yang berlaku (UU No.20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, Oktober 2016

Mahasiswa,

GABRIEL ANDRIANO BRAMANTYO

NIM. 125060307111002



RINGKASAN

Gabriel Andriano Bramantyo, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Oktober 2016, *Implementasi Kontroler PID Kaskade dan Kalman Filter pada Sistem Keseimbangan Robot Beroda Dua Berbasis Mikrokontroler Arduino ARM*, Dosen Pembimbing: Goegoes Dwi Nusantoro dan Rahmadwati.

Konsep sistem keseimbangan robot beroda dua didasarkan pada teori pendulum terbalik. Sebuah sistem kontrol yang sesuai yang dibutuhkan untuk mengontrol sistem sehingga seimbang dan stabil. Robot beroda dua merupakan suatu robot *mobile* yang memiliki sebuah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Menyeimbangkan robot beroda dua memerlukan suatu metode kontrol yang baik dan andal untuk mempertahankan posisi robot dalam keadaan tegak lurus terhadap permukaan bumi tanpa memerlukan pengendali lain dari luar. Pada penelitian skripsi ini bertujuan untuk mendesain dan membangun robot beroda dua yang mampu menyeimbangkan dirinya secara otomatis yang tegak lurus terhadap permukaan bumi di daerah bidang datar. Pada penelitian ini digunakan kontroler Proporsional Integral Derivatif kaskade sebagai pengendali dan kalman filter sebagai pemrosesan data sensor. Hasil parameter kontroler PID kaskade dan kalman filter yang dapat dicapai dari penelitian skripsi ini diperoleh nilai $speedPIDKp = 14$, $speedPIDKi = 15.2$, $speedPIDKd = 4$, $anglePIDAggKp = 50$, $anglePIDAggKi = 0.5$, $anglePIDAggKd = 0.25$, $anglePIDConKp = 20$, $anglePIDConKi = 0.5$, $anglePIDConKd = 0.25$, $angleKalmanFilter = 10$ dan $speedKalmanFilter = 20$.

Kata Kunci : Robot Beroda Dua, PID Kaskade, Kalman Filter.

SUMMARY

Gabriel Andriano Bramantyo, *Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, October 2016, Plant Cascade PID Controller Implementation and Kalman Filter on The System Balances Two Wheeled Robot Microcontroller Arduino Based ARM, Academic Supervisor: Goegoes Dwi Nusantoro and Rahmadwati.*

The concept of the system of the balances two wheeled robot based on the theory of the pendulum upside down. An appropriate control system is needed to control the system so that balanced and stable. Two wheeled robot is a mobile robot which has a wheel on the right and left hand that will not be balanced when without the existence of the controller. Balancing the two wheeled robot requires a good control methods and reliable to maintain the position of a robot in the situation so that it is perpendicular to the surface of the earth without requiring other controllers from the outside. In this research aims to design and build two wheeled robots that are able to balance himself automatically that it is perpendicular to the surface of the earth in the area flat field. In this research used plant cascade Proportional Integral Derivative controller as a controller and Kalman filter as sensor data processing. The results of plant cascade PID controller parameters and Kalman filter that can be achieved from this bachelor theses research obtained the value of speedPIDKp = 14, speedPIDKi = 15.2, speedPIDKd = 4, anglePIDAggKp = 50, anglePIDAggKi = 0.5, anglePIDAggKd = 0.25, anglePIDConKp = 20, anglePIDConKi = 0.5, anglePIDConKd = 0.25, angleKalmanFilter = 10 and speedKalmanFilter = 20.

Keywords : *Two Wheeled Robot, Plant Cascade PID, Kalman Filter.*

PENGANTAR

Segala syukur dan puji hanya bagi Tuhan Yesus Kristus, oleh karena anugerah-Nya yang melimpah, kemurahan dan kasih setia yang besar akhirnya penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Kontroler PID Kaskade dan Kalman Filter pada Sistem Keseimbangan Robot Beroda Dua Berbasis Mikrokontroler Arduino ARM” dengan baik.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

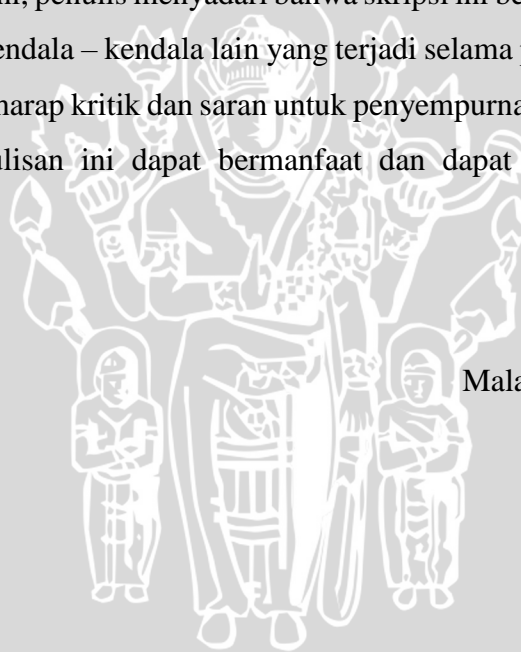
- Tuhan Yesus Kristus yang telah memberikan kelancaran, kemudahan dan hikmat-Nya.
- Keluarga tercinta, kedua orang tua Agustinus Surasno dan Anna Yustina Andriani yang selalu memberikan kasih sayang dan doanya yang tiada akhir. Adik tercinta SONDY dan RAFAEL yang selalu memberikan dukungan.
- Bapak M. Aziz Muslim, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang serta sebagai dosen penasehat akademik.
- Bapak Hadi Suyono S.T., M.T., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
- Bapak Ir. Purwanto, M.T. selaku KKDK Sistem Kontrol.
- Bapak Goegoes Dwi Nusantoro S.T., M.T. selaku dosen pembimbing skripsi atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan.
- Ibu Rahmadwati S.T., M.T., Ph.D. selaku dosen pembimbing skripsi atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan.
- Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
- Rekan seperjuangan dan Sahabat-Sahabat terbaik Hesa, Zakiyah, Suro, Nora, Faizal, Hanif, Tyo, Odi, Regina, Fara, Wuri, dan Dyto, yang telah memberikan semangat, bantuan yang telah diberikan, dukungan, serta pengalaman berharga dan inspirasinya.

- Seluruh Keluarga Besar Tim Robot Teknik Elektro Universitas Brawijaya atas segala bantuan yang telah diberikan.
- Rekan-rekan Tim Robot 2012 Teknik Elektro Universitas Brawijaya, Fikrul, Ronny, Firman, Septian, Sirojul, Reza, Agus, Anggia, dan Wiwin, terima kasih atas kerja sama, niat, semangat, pengorbanan serta pengalaman berharga selama ini, dan juga segala bantuan yang telah diberikan
- Keluarga besar VOLTAGE'12, dan CONTROL ENGINEERING'12 yang memberikan semangat, doa, dan dukungan.
- Seluruh teman-teman serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan satu-persatu, terima kasih telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala – kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang, semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Oktober 2016

Penulis



DAFTAR ISI

RINGKASAN	i
SUMMARY	ii
PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA	4
2.1 Robot Beroda Dua	4
2.2 Konsep Keseimbangan Robot Beroda Dua	6
2.2.1 Mempertahankan Keseimbangan Robot Beroda Dua	7
2.2.2 Membuat Robot Berjalan Maju atau Mundur (Mobilitas)	8
2.3 RoboGames	9
2.4 Kontroler	9
2.4.1 Kontroler Proporsional (P)	10
2.4.2 Kontroler Integral (I)	10
2.4.3 Kontroler Proporsional Integral (D).....	11
2.4.4 Kontroler Proporsional Integral Diferensial (PID)	12
2.5 Kaskade Kontrol.....	13
2.6 Kalman Filter.....	15
2.7 Motor DC	16
2.7.1 Pinsip Kerja Motor DC	17
2.8 <i>Driver</i> Motor DC	18
2.9 <i>Pulse Width Modulation</i>	19
2.10 Mikrokontroler Aduino Due	20
2.11 <i>Gyroscope</i>	21
2.12 <i>Accelerometer</i>	22

BAB III METODE PENELITIAN	23
3.1 Penentuan Spesifikasi Alat	23
3.2 Spesifikasi Desain	24
3.3 Karakterisasi Setiap Blok	25
3.3.1 Karakterisasi <i>Driver Motor (L298N Dual H-Bridge Motor Driver)</i>	25
3.3.2 Karakterisasi Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan Sensor Optical Rotary Encoder.....	26
3.3.3 Karakterisasi sensor <i>Digital Combo Board-6/9 Degrees of Freedom IMU</i> ITG3200/ADXL345.....	29
3.3.3.1 Karakterisasi Sensor <i>Gyroscope (ITG3200) Digital Combo Board-6/9</i> <i>Degrees of Freedom IMU</i>	29
3.3.3.2 Karakterisasi Sensor <i>Accelerometer (ADXL345) Digital Combo</i> <i>Board-6/9 Degrees of Freedom IMU</i>	31
3.3.4 Penentuan Fungsi Alih Motor DC 7,4 V Kinmore (KM-25A370-17.6- 06397) dengan Sensor <i>Optical Rotary Encoder</i>	33
3.4 Perancangan dan Pembuatan Perangkat Keras.....	38
3.4.1 Perancangan Sistem Mekanik Robot Beroda Dua.....	39
3.4.2 Perancangan Sistem Elektrik Robot Beroda Dua	39
3.5 Perancangan Algoritma.....	40
3.5.1 Kalman Filter	40
3.5.2 <i>Tuning PID</i>	41
3.6 Perancangan Blok Diagram Sistem	43
3.7 Perancangan dan Penyusunan Perangkat Lunak.....	44
BAB IV HASIL DAN PEMBAHASAN	46
4.1 Pengujian Pengendali PID Kaskade	46
4.1.1 Pengujian Pengendali Kecepatan dengan Kontroler PID	47
4.1.2 Pengujian Pengendali Sudut dengan Kontroler PID	48
4.1.3 Pengujian Pengendali Kecepatan dan Pengendali Sudut dengan Kontroler PID pada Sistem kesetimbangan Robot Beroda Dua.....	52
4.2 Pengujian Kalman Filter	53
4.3 Pengujian Keseluruhan Sistem.....	54
BAB V PENUTUP	56
5.1 Kesimpulan.....	56
5.2 Saran	57

DAFTAR PUSTAKA	58
LAMPIRAN I	59
LAMPIRAN II	62
LAMPIRAN III	84



DAFTAR TABEL

Tabel 3.1 Hasil Pengujian *Driver* (L298N *Dual H-Bridge Motor Driver*)..... 26

Tabel 3.2 Data Pengukuran Arah dan Kecepatan Motor DC..... 28



DAFTAR GAMBAR

Gambar 2.1	Sketsa Robot Beroda Dua	5
Gambar 2.2	Robot Beroda Dua.....	6
Gambar 2.3	Posisi Seimbang yang Harus Dipertahankan Robot Beroda Dua	7
Gambar 2.4	Ilustrasi Robot Beroda Dua dan Sudut Elevasi.....	7
Gambar 2.5	Arah Putaran Roda untuk Mempertahankan Keseimbangan	7
Gambar 2.6	Arah Putaran Roda pada Fungsi Bobilitas, (a) Robot dalam Kondisi Setimbang, (b) Arah Jatuh Robot, dan (c) Arah Gerak Robot	9
Gambar 2.7	Diagram Blok Kontroler <i>Proportional</i>	10
Gambar 2.8	Diagram Blok Kontroler Integral.....	11
Gambar 2.9	Diagram Blok Kontroler <i>Derivative</i>	11
Gambar 2.10	Diagram Blok Kontroler PID.....	12
Gambar 2.11	<i>Cascade Control System</i>	13
Gambar 2.12	Diagram Blok Kontroler PID Kaskade.....	14
Gambar 2.13	Skema Pengaturan konfigurasi Robot Beroda Dua.....	14
Gambar 2.14	Algoritma Perhitungan Kalman Filter	15
Gambar 2.15	Motor DC Sederhana	16
Gambar 2.16	<i>Prinsip Kerja Motor DC</i>	17
Gambar 2.17	IC Kendali Motor L298N Terdapat pada L298N <i>Dual H-Bridge Motor Driver Board</i>	18
Gambar 2.18	Sinyal PWM secara Umum.....	19
Gambar 2.19	Arduino Due.....	20
Gambar 2.20	Analogi 3-Axis <i>Gyroscope</i>	21
Gambar 2.21	Analogi 3-Axis <i>Accelerometer</i>	22
Gambar 3.1	Perubahan Tegangan <i>Output Driver</i> terhadap <i>Duty Cycle</i>	27
Gambar 3.2	Pengukuran Arah dan Kecepatan Motor DC	28
Gambar 3.3	Data Hasil Pengujian <i>Output Sensor Gyroscope</i>	30
Gambar 3.4	Data Hasil Pengujian <i>Output Sensor Accelerometer</i>	32
Gambar 3.5	<i>Input dan Output Sinyal PRBS</i>	34
Gambar 3.6	<i>System Identification Toolbox</i>	35
Gambar 3.7	Hasil Estimasi Model.....	35
Gambar 3.8	Nilai <i>Time Constant Output Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397)</i> dengan <i>Input Unit Step</i>	36

Gambar 3.9 Nilai <i>Rise Time Output</i> Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan <i>Input Unit Step</i>	37
Gambar 3.10 Nilai <i>Settling Time Output</i> Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan <i>Input Unit Step</i>	37
Gambar 3.11 Desain Mekanik Robot Beroda Dua	38
Gambar 3.12 Perspektif Desain Mekanik Robot Beroda Dua	39
Gambar 3.13 Desain Sistem Elektrik Robot Beroda Dua.....	39
Gambar 3.14 <i>Simulink Tuning</i> PID Manual.....	42
Gambar 3.15 <i>Output</i> Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) Tanpa Kontroler PID	42
Gambar 3.16 <i>Output</i> Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan Kontroler PID	42
Gambar 3.17 Blok Diagram Sistem <i>Loop</i> Tertutup Kesetimbangan Robot Beroda Dua.....	44
Gambar 3.18 Flowchart Program Keseluruhan	45
Gambar 4.1 <i>Output</i> Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan <i>Setpoint</i> 55 Rpm.....	47
Gambar 4.2 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 1 pada Sistem Kesetimbangan Robot Beroda Dua	48
Gambar 4.3 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 2 pada Sistem Kesetimbangan Robot Beroda Dua	49
Gambar 4.4 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 3 pada Sistem Kesetimbangan Robot Beroda Dua	50
Gambar 4.5 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 3 pada Sistem Kesetimbangan Robot Beroda Dua Ketika Diberi Gangguan.....	51
Gambar 4.6 Sistem Kesetimbangan Robot Beroda Dua tanpa Gangguan	52
Gambar 4.7 Sistem Kesetimbangan Robot Beroda Dua dengan Gangguan	52
Gambar 4.8 Data Hasil Pengujian Kalman Filter pada Sensor IMU..	53
Gambar 4.9 Data Hasil Pengujian Kalman Filter pada Sensor <i>Optical Rotary Encoder</i> Kiri	53
Gambar 4.10 Data Hasil Pengujian Kalman Filter pada Sensor <i>Optical Rotary Encoder</i> Kanan	53
Gambar 4.11 Data Hasil Pengujian Keseluruhan Sistem Kesetimbangan Robot Beroda Dua hanya Menggunakan PID <i>Loop</i> Dalam	54

Gambar 4.12 Data Hasil Pengujian Keseluruhan Sistem Keseimbangan Robot Beroda Dua Menggunakan PID Kaskade





BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi robotika telah membuat kualitas kehidupan manusia semakin tinggi. Saat ini perkembangan teknologi robotika telah mampu meningkatkan kualitas maupun kuantitas berbagai industri. Teknologi robotika juga telah menjangkau sisi hiburan dan pendidikan bagi manusia. Salah satu cara menambah tingkat kecerdasan sebuah robot adalah dengan menambah sensor, metode kontrol bahkan memberikan kecerdasan buatan pada robot tersebut. Salah satunya adalah robot beroda dua (R. Muhammad Syafruddi, 2013).

Hal ini tentunya dipicu oleh perkembangan teknologi elektronika, robotika serta IT yang cukup pesat di seluruh dunia. *Self balancing robot* merupakan robot yang memiliki prinsip kerja yang hampir mirip dengan sistem pendulum terbalik (Geoffrey D, Benneth, 2007).

Robot beroda dua merupakan suatu robot *mobile* yang memiliki sebuah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Menyeimbangkan robot beroda dua memerlukan suatu metode kontrol yang baik dan andal untuk mempertahankan posisi robot dalam keadaan tegak lurus terhadap permukaan bumi tanpa memerlukan pengendali lain dari luar. Bahkan sekarang ini konsep robot beroda dua telah diaplikasikan dan digunakan sebagai alat transportasi modern yang bernama *Segway*, *Hovertrax*, dan *Airboard*.

Pada penelitian ini bertujuan untuk mendesain dan membangun robot beroda dua yang mampu menyeimbangkan dirinya secara otomatis yang tegak lurus terhadap permukaan bumi di daerah bidang datar. Pada penelitian ini digunakan kontroler Proporsional Integral Derivatif (PID) sebagai pengendali dan Kalman filter sebagai pemrosesan data sensor. Penggunaan kontroler PID kaskade dan Kalman filter dikarenakan pada sistem kesetimbangan ini dibutuhkan *output* yang cepat dengan akurasi tinggi dan masih terdapat *error* dan *noise* pada sensor *gyroscope* dan sensor *accelerometer* serta *noise* pada motor DC dalam kesetimbangan robot beroda dua. Kontroler PID kaskade digunakan untuk menentukan besarnya arah dan kecepatan putar motor DC sebagai penggerak. Kalman filter digunakan untuk menggabungkan pembacaan sensor *gyroscope* dan *accelerometer* untuk menafsir dan mendapatkan sudut robot, menggabungkan nilai sudut dan menghilangkan derau dari pembacaan sensor *gyroscope* dan sensor *accelerometer*,

mengurangi *error* sensor sudut dan kecepatan roda, menghapus perubahan yang cepat dan membuat robot lebih stabil dengan lebih kecil *overshoot*.

Berdasarkan kemiringan sudut badan robot terhadap permukaan bidang datar, sehingga robot beroda dua ini dapat mempertahankan posisinya tegak lurus dengan seimbang terhadap permukaan bumi pada bidang datar.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut di atas, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana merancang algoritma kontroler PID kaskade dan Kalman filter untuk mempertahankan keseimbangan pada robot beroda dua ?
2. Bagaimana *output* plant robot beroda dua menggunakan kontroler PID kaskade dan Kalman filter pada implementasi ?

1.3 Batasan Masalah

Karena luasnya objek kajian, maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah pada skripsi ini antara lain:

1. Plant yang digunakan adalah robot beroda dua dengan spesifikasi sebagai berikut
 - a. Aktuator yang digunakan adalah motor DC dengan tipe Kinmore (KM-25A370-17.6-06397) dengan sensor *Optical Rotary Encoder*.
 - b. Driver motor yang digunakan adalah L298N *Dual H-Bridge Motor Driver board*.
 - c. Sensor yang digunakan adalah *Digital Combo Board-6/9 Degrees of Freedom* IMU ITG3200/ADXL345.
2. Menggunakan *software* Arduino IDE 1.6.11.
3. Menggunakan *software* Processing 3.2.1.
4. Pusat pengontrolan terdapat pada modul mikrokontroler Arduino Due.
5. Arena yang digunakan memiliki permukaan datar.
6. Robot beroda dua hanya mempertahankan keseimbangan ketika diberi gangguan berupa kemiringan sudut dan dorongan kecil maju dan mundur secara bergantian.

7. Robot beroda dua mampu berdiri dan dapat seimbang antara sudut -10^0 sampai $+10^0$ dari posisi seimbang (sudut tegak lurus terhadap permukaan bumi adalah 0^0).
8. Dalam proses mempertahankan keseimbangan robot beroda dua, robot beroda dua tidak dibatasi dalam melakukan pergerakan dalam upaya mempertahankan keseimbangannya.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian skripsi ini adalah mempertahankan posisi robot beroda dua dalam kondisi seimbang dan tegak lurus terhadap permukaan bumi di bidang datar dengan menggunakan metode kontrol PID kaskade dan Kalman filter.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah dapat dipahaminya sistem pengontrolan kesetimbangan robot beroda dua dengan menggunakan metode kontroler PID kaskade dan Kalman filter.



BAB II

TINJAUAN PUSTAKA

Kendaraan dua roda yang mempunyai keseimbangan ini mempunyai prinsip kerja berdasarkan *inverted* pendulum dan sangat cocok digunakan sebagai sarana transportasi pribadi. Beberapa definisi lain menyatakan bahwa robot merupakan mesin yang terlihat seperti manusia dan melakukan berbagai tindakan yang kompleks dari manusia seperti berjalan atau berbicara, atau suatu peralatan yang bekerja secara otomatis. Definisi lain yang lebih kompleks menyatakan bahwa robot adalah manipulator multi fungsi dan dapat diprogram ulang yang dirancang untuk menggerakkan material, alat, atau perangkat khusus melalui bermacam-macam program untuk menyelesaikan suatu pekerjaan tertentu dan didesain sesuai dengan kebutuhan.

Kontrol pergerakan maju dan mundurnya robot beroda dua ini pada dasarnya seperti pendulum terbalik yang sistemnya tidak stabil sehingga diperlukan sebuah kontroler yang baik dan efisien sehingga mampu membuat robot tersebut stabil

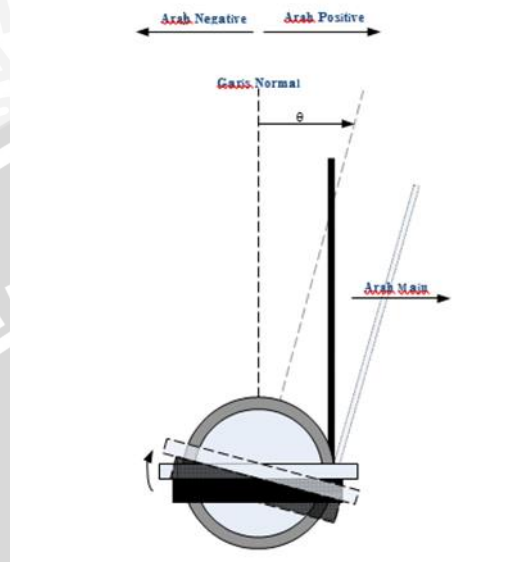
Bab ini menguraikan teori-teori pendukung skripsi, yang terdiri atas :

2.1 Robot Beroda Dua

Robot telah mengalami perkembangan yang sangat pesat. Ketertarikan dunia pada bidang robotika begitu besar , dengan ditandai banyaknya penggunaan robot di berbagai bidang, seperti bidang industri, pertanian, kedokteran, dan juga militer. Tujuan utama pemanfaatan teknologi robotika pada berbagai bidang tersebut tidak lain adalah untuk mempermudah kerja manusia. Kata “robot” pertama kali digunakan oleh penulis Cekoslavia bernama Karel Capek, dalam dramanya yang berjudul “RUR: *Rossum’s Universal Robot*”. Kata “robot” dalam Bahasa Cekoslavia berarti pekerja.

Seperti definisi dari kata robot itu sendiri yang berarti “pekerja”, keberadaan robot untuk membantu pekerjaan manusia memang tidak bisa dipungkiri. Robot didesain untuk membantu manusia dengan beberapa kelebihan dibandingkan pekerja manusia. Pekerjaan yang diselesaikan robot memiliki hasil yang lebih akurat dan yang dibutuhkan lebih cepat, dengan biaya yang lebih murah. Kekurangan yang dimiliki robot adalah tidak fleksibel. Salah satu alasan penggunaan teknologi robot adalah karena robot memiliki kemampuan khusus untuk menyelesaikan suatu pekerjaan tertentu, dengan tingkat akurasi yang tinggi dan mampu mengerjakan sesuatu dalam waktu yang relatif lebih cepat jika dibandingkan dengan manusia.

Robot dapat dikelompokkan menjadi beberapa jenis sesuai dengan spesifikasinya, misalkan fungsi dan alat gerak. Sesuai fungsinya, ada beberapa jenis robot, diantaranya robot penjelajah, robot industri, robot pemadam api, robot medis, robot nano, robot militer, dan robot kompetisi. Jenis robot sesuai alat geraknya alat gerak adalah robot beroda dan robot berkaki. Salah satu kategori dalam robot beroda adalah robot beroda dua.



Gambar 2.1 Sketsa Robot Beroda Dua

Robot beroda dua adalah robot yang mempunyai dua buah penggerak yang dihubungkan ke roda yang terdapat disebelah kanan dan kiri robot. Cara kerja secara general untuk dapat menyeimbangkan robot ini adalah ketika robot akan terjatuh kesisi depan maupun kesisi belakang, maka untuk menstabilkan atau menyeimbangkannya harus mengirimkan sinyal kontrol yang sesuai ke motor DC sehingga motor DC berputar ke arah depan dan mengikuti pergerakan robot dan dapat kembali ke posisi semula, begitu juga sebaliknya jika robot akan terjatuh ke arah belakang untuk menstabilkan atau menyeimbangkannya harus mengirimkan sinyal kontrol yang sesuai ke motor DC sehingga motor DC berputar ke arah belakang dan mampu mengikuti pergerakan robot sehingga dapat kembali ke posisi semula. Robot beroda dua ini pada dasarnya memiliki prinsip kerja seperti pendulum terbalik yang sistemnya tidak stabil, sehingga diperlukan sebuah kontroler yang baik dan efisien sehingga mampu membuat robot tersebut stabil.



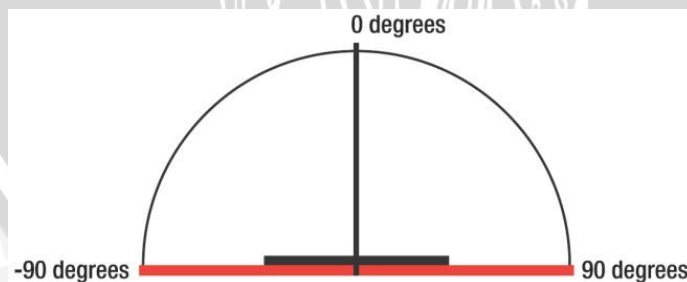
Gambar 2.2 Robot Beroda Dua

Sumber: wolfdenelectronics.com

2.2 Konsep Keseimbangan Robot Beroda Dua

Prinsip kerja *balancing robot* adalah dengan menjaga keseimbangan sistem. Untuk menjaga robot beroda dua ini seimbang, kontroler perlu mengetahui sudut relatif terhadap tanah, sehingga controller dapat memerintahkan motor dengan kecepatan dan arah yang tepat yang dibutuhkan agar tidak terjatuh. Untuk keakuratan pengukuran sudut atau kemiringan dari robot perlu mendeteksi kecepatan rotasi dan gaya gravitasi sumbu X-nya, menggunakan satuan pengukuran inertial atau inertial measurement unit (IMU).

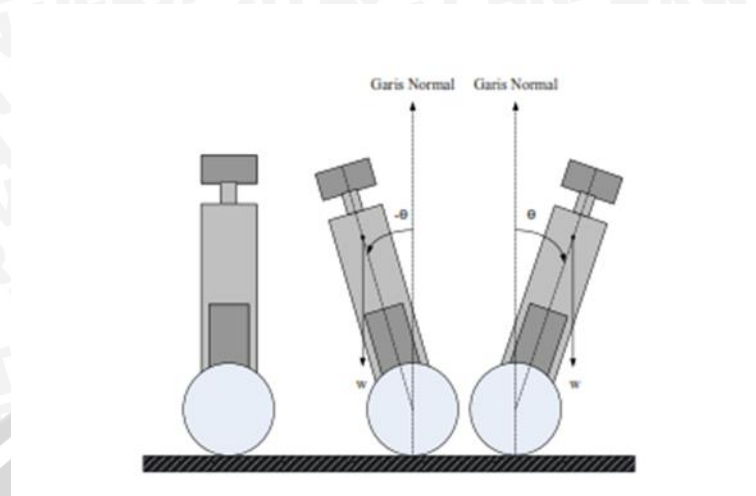
IMU adalah sebuah board PCB kecil yang berisi sensor *gyroscope* dan sensor *accelerometer*. Pengukuran ini memiliki referensi 0 derajat yang menyebabkan motor berhenti. Apabila pengukuran lebih dari 0 derajat maka motor akan berputar proporsional ke depan. Demikian sebaliknya bila kurang dari 0 derajat, motor akan berputar berlawanan arah secara proporsional. Gambar 2.3 menunjukkan posisi tegak terhadap bumi sehingga motor diperintahkan untuk *stop*.



Gambar 2.3 Posisi Seimbang yang Harus Dipertahankan Robot Beroda Dua

(Sumber: J.-D. Warren, J. Adams dan H. Molle, *Arduino Robotics*, New York: Apress, 2011)

Untuk memperjelas gambaran sistem, perhatikan Gambar 2.4:

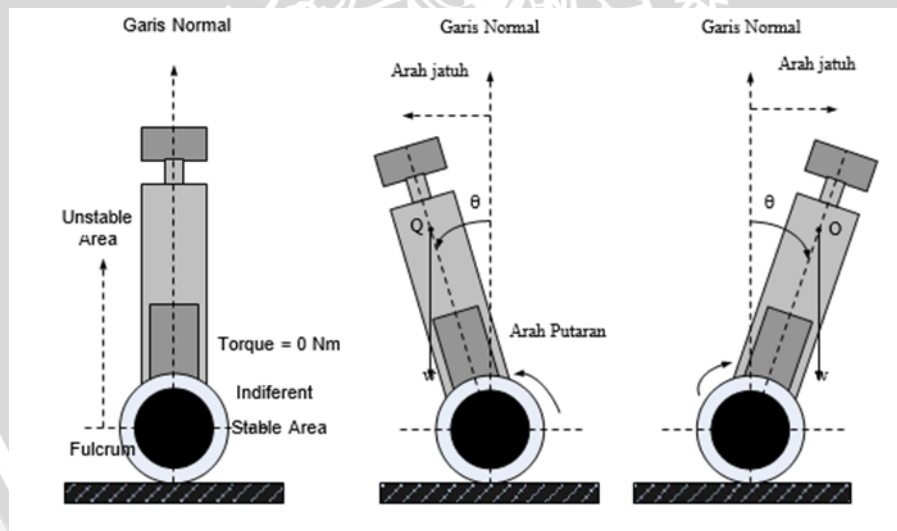


Gambar 2.4 Ilustrasi Robot Beroda Dua dan Sudut Elevasi

(Sumber: J.-D. Warren, J. Adams dan H. Molle, Arduino Robotics, New York: Apress, 2011)

2.2.1 Mempertahankan Kesetimbangan Robot Beroda Dua

Prinsip kerja kesetimbangan robot beroda dua dapat dijelaskan sebagai sebuah *inverted* pendulum yang bertumpu pada kedua rodanya.



Gambar 2.5 Arah Putaran Roda untuk Mempertahankan Kesetimbangan

(Sumber: J.-D. Warren, J. Adams dan H. Molle, Arduino Robotics, New York: Apress, 2011)

Dalam proses menuju kesetimbangan robot memiliki kecenderungan jatuh ke kiri maupun ke kanan tergantung titik berat dari robot tersebut. Apabila dimisalkan robot jatuh ke arah kiri seperti gambar di bawah ini. Bila robot jatuh ke arah kiri sejauh θ maka harus ada putaran roda dengan torque tertentu searah dengan jatuhnya roda. Kekuatan putaran

harus lebih besar dibandingkan dengan besarnya θ agar robot dapat kembali ke posisi semula.

Berikut ini persamaan stationary pivot point dari *balancing robot /inverted* pendulum.

$$\ddot{\theta} = \frac{g}{l} \sin \theta \quad (2-1)$$

Diferensial pangkat dua dari θ adalah fungsi percepatan sudut. *Inverted* pendulum akan bergerak menjauh dari garis normal dengan percepatan tertentu. Nilai percepatan yang akan dihasilkan berbanding terbalik dengan panjang pole *inverted* pendulum atau tinggi robot. Semakin panjang pole maka semakin kecil percepatan *inverted* pendulum. Nilai inilah yang harus dipertimbangkan sistem kontrol untuk diberikan kepada motor agar kesetimbangan dapat terjadi.

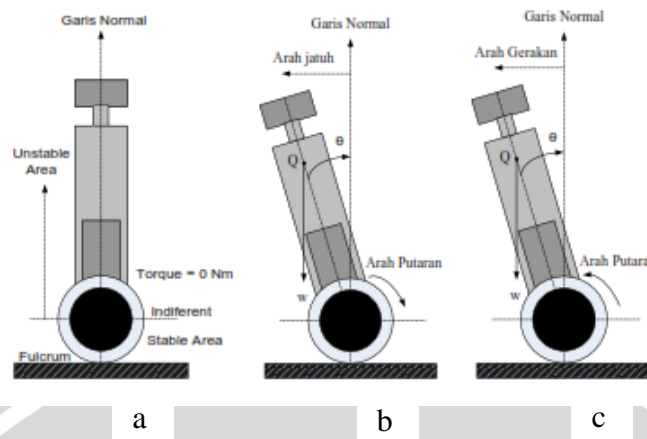
Setelah diberikan *torque* dengan kekuatan tertentu, maka robot akan kembali menuju ke titik kesetimbangan (garis normal). Robot juga memiliki kesempatan untuk jatuh ke arah sebaliknya dengan percepatan tertentu. Arah putaran tersebut tetap searah dengan arah jatuhnya robot. Dapat disimpulkan bahwa robot akan bergerak di antara titik stasionernya sampai mencapai kesetimbangan. Semakin lama sudut θ semakin mengecil. *Rise time* dan nilai *steady state* nya tergantung metode kontrol yang diterapkan pada robot tersebut.

2.2.2 Membuat Robot Berjalan Maju atau Mundur (Mobilitas)

Setelah robot dapat mencapai kesetimbangan pada titik stasionernya, maka saat ini akan diupayakan untuk dapat berjalan maju maupun mundur. Ada 2 tahap yang harus dilakukan robot agar robot dapat berjalan maju atau mundur.

Pada Gambar 2.6 a menjelaskan robot dalam kondisi setimbang. Dalam kondisi setimbang ini akan dibuat melangkah maju. Apabila gerakan diinginkan ke kiri, maka tahap pertama harus dibuat putaran roda ke kanan clockwise (CW). Hal ini menyebabkan badan dari robot jatuh ke arah kiri sejauh θ .

Setelah robot dapat mencapai kesetimbangan pada titik stasionernya, maka saat ini akan diupayakan untuk dapat berjalan maju maupun mundur. Ada 2 tahap yang harus dilakukan robot agar robot dapat berjalan maju atau mundur.



Gambar 2.6 Arah Putaran Roda pada Fungsi Bobilitas, (a) Robot dalam Kondisi Setimbang, (b) Arah Jatuh Robot, dan (c) Arah Gerak Robot

(Sumber: J.-D. Warren, J. Adams dan H. Molle, *Arduino Robotics*, New York: Apress, 2011)

Pada Gambar 2.6 b menjelaskan arah jatuh robot dan Gambar 2.6 c menjelaskan arah gerak robot. Apabila telah terbentuk sudut elevasi tersebut, maka arah putaran dibalik dengan kecepatan dan *torque* tertentu agar robot dapat berjalan sesuai dengan yang diinginkan. Demikian bila robot diinginkan berjalan mundur maka seluruh tahapan tersebut dibalik.

2.3 RoboGames

RoboGames (formerly ROBOlympics) adalah suatu perlombaan robot tingkat internasional yang diselenggarakan di San Mateo Fairgrounds, Amerika Serikat. Perlombaan ini diikuti oleh 37 negara, diantaranya adalah Indonesia dan terdapat 50 kategori perlombaan yang dilombakan, diantaranya adalah *Balancer Race*. *Balancer Race* merupakan kategori perlombaan yang mempunyai tujuan menyelesaikan lintasan perlombaan dalam waktu tercepat dan dalam keadaan seimbang, sebuah robot *balancing* didefinisikan sebagai sebuah robot beroda dua atau satu yang dapat menjaga keseimbangannya tanpa dukungan eksternal. Dalam kategori *Balancer Race* terdapat dua jenis robot yaitu robot otomatis dan robot manual (robot yang dikendalikan oleh *remote controlled*).

2.4 Kontroler

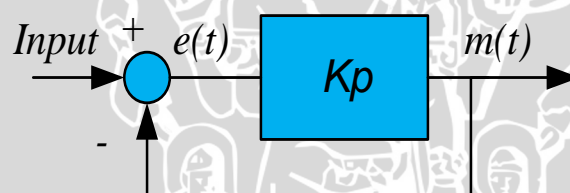
Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem

dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata, K., 1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespon *input* dan *output*.

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *outputnya* beresilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan *output* transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

2.4.1 Kontroler *Proportional* (P)

Kontroler *proportional* memiliki *output* yang besarnya sebanding dengan besarnya sinyal *error*. *Output* kontroler merupakan perkalian antara penguatan proporsional dengan sinyal *error*. Gambar 2.7 menunjukkan diagram blok kontroler *proportional* dan Persamaan 2.2 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2.7 Diagram Blok Kontroler *Proportional*

Sumber: Ogata, K. (1997)

$$m(t) = K_p e(t) \quad (2-2)$$

dimana:

K_p = adalah penguatan proporsional

$e(t)$ = sinyal error

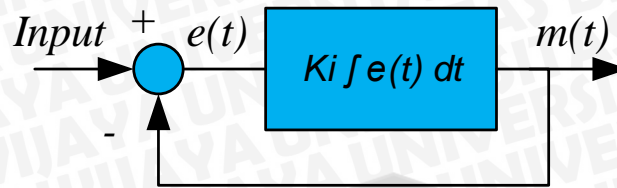
$m(t)$ = output kontroler

Penambahan K_p akan mempercepat kecepatan *output transient* dan mengurangi kesalahan keadaan mantap.

2.4.2 Kontroler *Integral* (I)

Kontroler integral memiliki karakteristik seperti sebuah operasi integral, *output* kontroler dipengaruhi oleh perubahan yang sebanding dengan perubahan nilai sinyal *error*. *Output* kontroler merupakan penjumlahan terus menerus dari perubahan sinyal *error*.

Gambar 2.8 menunjukkan diagram blok kontroler integral dan Persamaan 2.3 dan Persamaan 2.4 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2.8 Diagram Blok Kontroler Integral

Sumber: Ogata, K. (1997)

$$\frac{dm(t)}{dt} = Ki e(t) \quad (2-3)$$

$$m(t) = Ki \int e(t) dt \quad (2-4)$$

dimana:

Ki = adalah penguatan integral

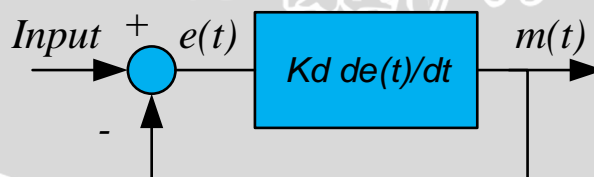
$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

Aksi kontrol integral digunakan untuk menghilangkan sinyal *error* dalam keadaan mantap.

2.4.3 Kontroler *Derivative* (D)

Kontroler *derivative* memiliki sifat seperti suatu operasi differensial. Perubahan yang mendadak pada masukan kontroler mengakibatkan perubahan yang sangat besar dan cepat. Kontroler ini tidak akan menghasilkan *output* saat sinyal *error* konstan sehingga tidak akan mempengaruhi keadaan mantap. Gambar 2.9 menunjukkan diagram blok kontroler *derivative* dan Persamaan 2.5 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2.9 Diagram Blok Kontroler *Derivative*

Sumber: Ogata, K. (1997)

$$m(t) = Kd \frac{de(t)}{dt} \quad (2-5)$$

dimana:

Kd = adalah penguatan *derivative*

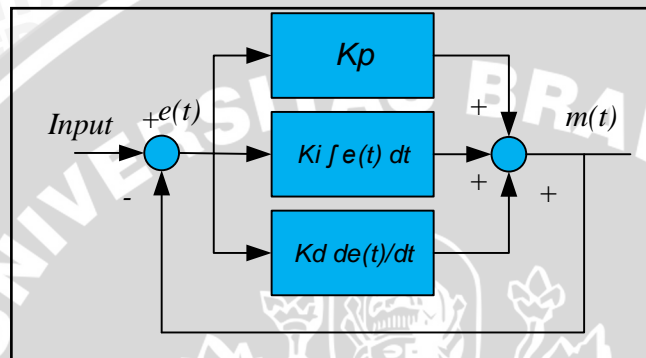
$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

Kontroler ini digunakan untuk memperbaiki atau mempercepat *output transient*.

2.4.4 Kontroler *Proportional Integral Derivative* (PID)

Gabungan aksi kontrol *proportional*, *integral*, dan *derivative* yang terlihat dalam Gambar 2.10 mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut (Persamaan 2.6):



Gambar 2.10 Diagram Blok Kontroler PID

Sumber: Ogata, K. (1997)

$$m(t) = K_p \cdot e(t) + K_i \cdot \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2-6)$$

dimana:

K_p = adalah penguatan *proportional*

K_i = adalah penguatan *integral*

K_d = adalah penguatan *derivative*

$e(t)$ = sinyal *error*

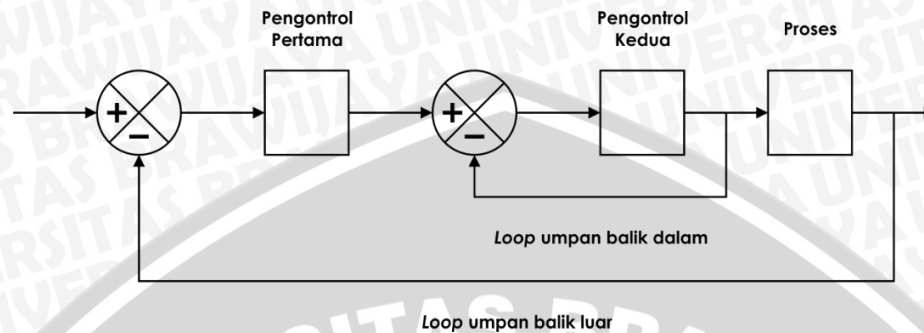
$m(t)$ = *output* kontroler

2.5 Kaskade Kontrol

Kaskade kontrol adalah salah satu strategi kontrol yang dapat meningkatkan kontrol suatu proses jika dibandingkan dengan kontroler PID yang konvensional. Agar suatu *plant* memiliki efisiensi tinggi dan lebih murah, maka struktur suatu *plant* menjadi lebih kompleks. Agar mengatasi masalah *plant* yang kompleks maka digunakan salah satu strategi kontrol yang disebut kaskade kontrol.

Kontrol kaskade adalah kontrol yang melibatkan penggunaan dua buah pengontrol dengan keluaran dari pengontrol pertama merupakan titik pengaturan bagi pengontrol

kedua. *Loop* umpan balik untuk salah satu pengontrol berada di dalam *loop* umpan balik untuk pengontrol yang lain, seperti yang ditampilkan dalam Gambar 2.11. Sistem seperti ini dapat menghasilkan perbaikan karakteristik *output* sistem terhadap gangguan yang muncul (W. Bolton, 2006: 290).



Gambar 2.11 Cascade Control System

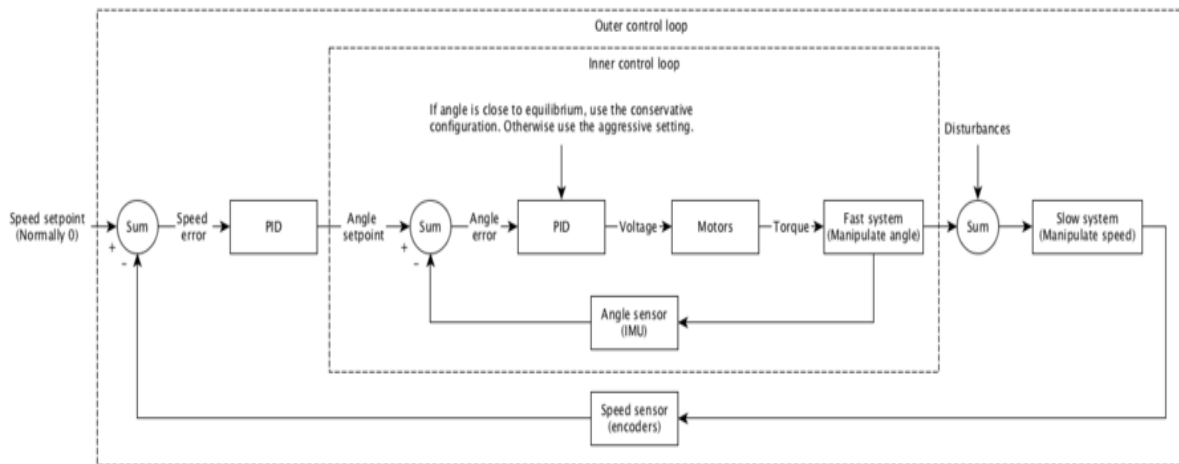
Sumber: K. Ogata, (1985)

Kontrol kaskade meliputi penggabungan dua kontroler menjadi satu yang difungsikan secara bersamaan dan dikontrol secara bersama pula. (Khairul Ramadhan, 2014)

Kekurangan dari kontrol umpan balik konvensional adalah aksi koreksi untuk sebuah gangguan tidak akan dimulai hingga hasil variabel yang dikontrol menyimpang dari set point. Kontrol umpan maju memberikan kemajuan besar daripada kontrol umpan balik untuk proses yang memiliki waktu *delay*. Bagaimanapun, kontrol umpan maju memerlukan gangguan yang dapat diukur dengan jelas dan perlu sebuah model yang harus dapat menghitung keluaran dari kontroler. Pendekatan alternatif yang dapat meningkatkan *output* dinamik adalah dengan menggunakan pengukuran secara sekunder dan kontrol umpan balik sekunder.

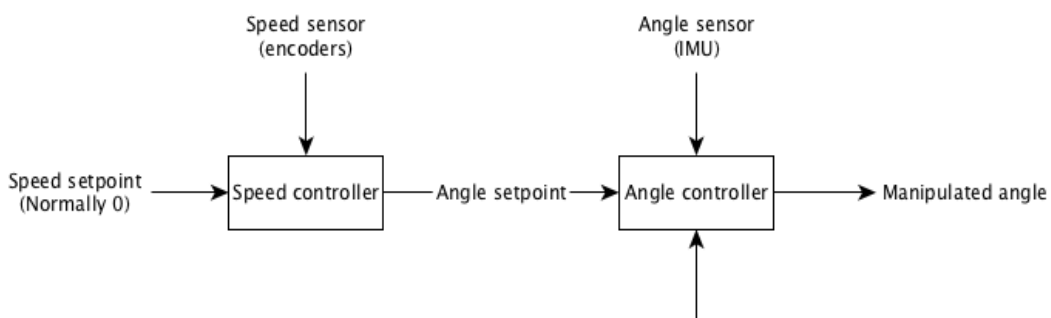
Pengukuran sekunder digunakan agar gangguan dapat dikenali sebelum variabel yang dikontrol, tetapi gangguan tidak perlu diukur. Pendekatan ini menggunakan 2 umpan balik yang disebut dengan kaskade kontrol.

Kontroler PID kaskade yang akan diaplikasikan pada robot beroda dimana kontroler P untuk kecepatan *encoder*, kontroler I untuk posisi, kontroler D untuk percepatan sudut. Kontroler PID kaskade terdapat dua *loop* yaitu *loop* luar (*outer control loop*) untuk mengukur kecepatan robot yang diukur dengan encoder pada motor. *Loop* dalam (*inner control loop*) untuk mengontrol sudut robot yang diukur dengan IMU. Gambar 2.12 menunjukkan diagram blok kontroler PID kaskade Seperti terlihat *loop* luar memiliki dinamika lambat dan *loop* dalam jauh lebih cepat.



Gambar 2.12 Diagram Blok Kontroler PID Kaskade

Secara teori, memprogram robot beroda dua dengan kontroler PID kaskade hanya menggunakan *loop* dalam tetapi perilaku robot beroda dua terkadang beralih perlahan-lahan maju atau mundur di permukaan bidang datar dengan cepat sehingga menyebabkan umpan balik yang hilang dari kecepatan roda. Sehingga *loop* luar juga digunakan untuk mendapatkan kinerja tambahan yang digunakan pada *loop* dalam, yang berarti bahwa parameter pengatur akan berubah tergantung pada kemiringan sudut robot. Gambar 2.13 menunjukkan skema pengaturan konfigurasi robot beroda dua. Pengaturan konfigurasi berdasarkan pada kecepatan dan posisi sudut pada robot beroda dua dimana *speed controller* berfungsi untuk mengatur kecepatan motor DC yang diukur dengan *encoder* sedangkan *angle controller* berfungsi untuk mengatur posisi sudut robot beroda dua.



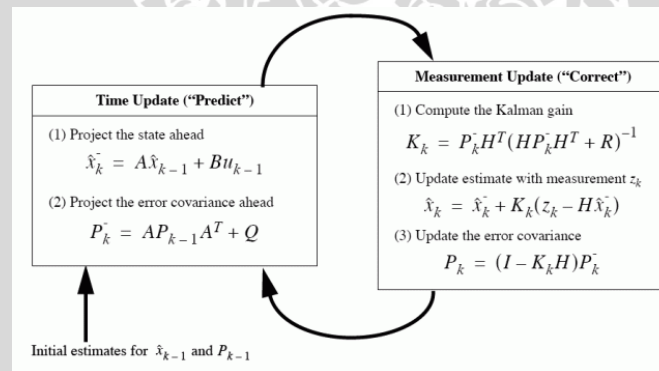
Jika sudut mendekati kesetimbangan, menggunakan konfigurasi konservatif (mempertahankan keadaan) Sebaliknya menggunakan konfigurasi agresif (mendapatkan kembali stabilitas)

Gambar 2.13 Skema Pengaturan konfigurasi Robot Beroda Dua

Ketika sudut mendekati kesetimbangan berarti parameter pengatur dibuat konfigurasi konservatif, hanya penyesuaian kecil yang dibuat untuk menjaga berdiri robot. Jika sudut robot mendekati dengan jatuh, parameter pengatur dibuat konfigurasi agresif, diaktifkan untuk melakukan segala sesuatu untuk mendapatkan kembali stabilitas.

2.6 Kalman Filter

Algoritma filtering yang bagus merupakan algoritma yang dapat menghilangkan *noise* dari sinyal elektromagnetik tetapi tetap mempertahankan informasi yang berguna. Kalman filter merupakan algoritma yang dapat memperkirakan variabel dari berbagai macam proses. Dalam istilah matematika, kita dapat mengatakan Kalman filter memperkirakan kondisi dari sistem linier. Kalman filter tidak hanya bekerja baik dalam praktek, tetapi secara teoritis juga menarik karena dapat ditunjukkan bahwa dari semua filter yang mungkin, Kalman filter merupakan salah satu filter yang meminimalkan varian dari perkiraan *error*. Kalman filter seringkali diimplementasi dalam sistem *embedded control*, karena dalam rangka untuk mengontrol suatu proses, diperlukan perkiraan yang akurat dari variabel proses.



Gambar 2.14 Algoritma Perhitungan Kalman Filter

Sumber: cs.brown.edu

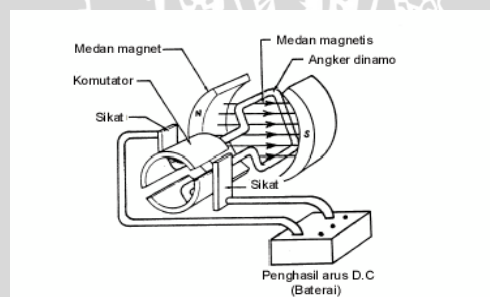
Menjelaskan masalah filter pada proses yang linier, sehingga filter tersebut dikenal dengan nama Kalman filter. Kalman Filter dapat menghilangkan *noise* dari suatu sinyal yang mengandung informasi dan mengambil informasi tersebut untuk diproses lebih lanjut. Suatu proses yang menggunakan Kalman filter untuk memfilter *noise* harus dapat disajikan dalam dua persamaan, yaitu persamaan *state* dan persamaan keluaran. Masing-masing persamaan mempunyai *noise* dan saling bebas, sehingga tidak ada korelasi saling antara kedua *noise*. Hasil perhitungan optimal yang digunakan untuk mendapatkan nilai estimasi pada Kalman filter.

2.7 Motor DC

Motor listrik sangat sering digunakan sebagai elemen kontrol akhir dalam sistem kontrol posisi dan kecepatan. Prinsip kerja dasar dari sebuah motor listrik adalah gaya yang bekerja pada konduktor yang berada di dalam suatu medan magnet ketika ada arus yang melewati konduktor tersebut (W. Bolton, 2004)

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar).

Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen.



Gambar 2.15 Motor DC Sederhana

Sumber: teorick.blogspot.com

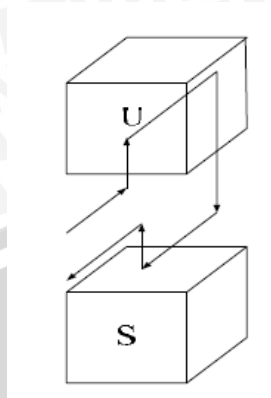
Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

2.7.1 Prinsip Kerja Motor DC

Daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu.

Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus berfungsi sebagai tempat

berlangsungnya proses perubahan energi dan daerah tersebut dapat dilihat pada Gambar 2.16 :



Gambar 2.16 Prinsip Kerja Motor DC

Sumber: biondiocta.wordpress.com

Dengan mengacu pada hukum kekekalan energi :

Proses energi listrik = energi mekanik + energi panas + energi didalam medan magnet. (2-7)

Maka dalam medan magnet akan dihasilkan kumparan medan dengan kerapatan fluks sebesar B dengan arus adalah I serta panjang konduktor sama dengan L maka diperoleh gaya sebesar F , dengan persamaan sebagai berikut :

$$F = B I L \dots (N) \quad (2-8)$$

Arah dari gaya ini ditentukan oleh aturan kaidah tangan kiri, adapun kaidah tangan kiri tersebut adalah sebagai berikut :

Ibu jari sebagai arah gaya (F), telunjuk jari sebagai fluks (B), dan jari tengah sebagai arus (I). Bila motor DC mempunyai jari-jari dengan panjang sebesar (r), maka hubungan persamaan dapat diperoleh :

$$Tr = Fr = B I L r \dots (Nm) \quad (2-9)$$

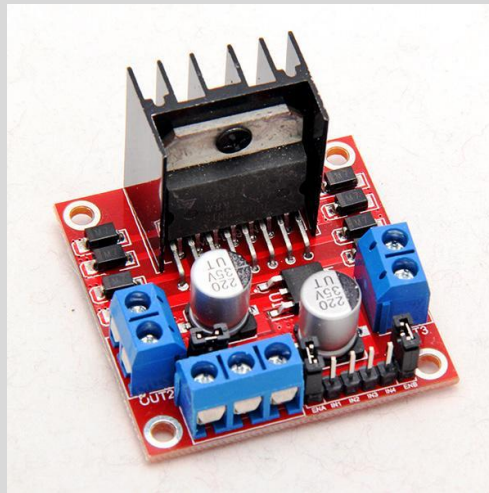
Saat gaya (F) tersebut dibandingkan, konduktor akan bergerak didalam kumparan medan magnet dan menimbulkan gaya gerak listrik yang merupakan reaksi lawan terhadap tegangan sumber.

Agar proses perubahan energi mekanik tersebut dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar daripada tegangan gerak yang disebabkan reaksi lawan.

Dengan memberi arus pada kumparan jangkar yang dilindungi oleh medan maka menimbulkan perputaran pada motor.

2.8 Driver Motor DC

Motor DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada motor DC sedangkan arus keluaran pada mikrokontroler sangat kecil. Kendali motor merupakan pilihan alternatif yang harus digunakan untuk mengendalikan kecepatan motor DC Kinmore (KM-25A370-17.6-06397). Ada beberapa jenis kendali motor yang dapat digunakan untuk mengatur kecepatan motor, yaitu menggunakan rangkaian *H-Bridge* transistor, *H-Bridge* MOSFET, dan IC driver motor. Pada skripsi ini kendali motor menggunakan IC driver L298N yang terdapat pada L298N *Dual H-Bridge Motor Driver board*. (lihat Gambar 2.17) berbasis *H-Bridge*, mampu menangani beban hingga 4A pada tegangan 6V – 46V. Dalam chip terdapat dua rangkaian *H-Bridge*. Selain itu driver ini mampu mengendalikan 2 motor sekaligus dengan arus beban masing-masing 2 A.



Gambar 2.17 IC Kendali Motor L298N Terdapat pada L298N Dual H-Bridge Motor Driver Board.

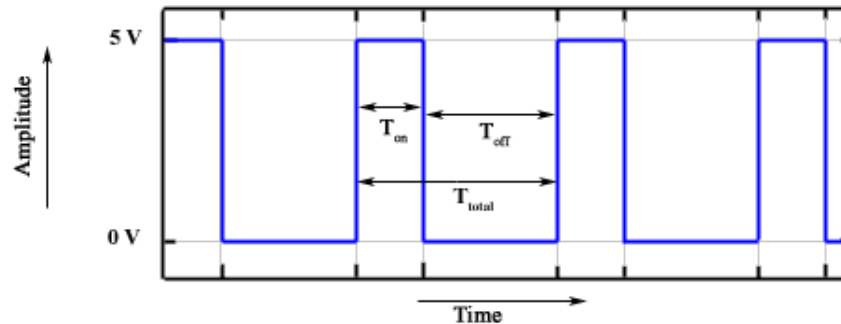
Sumber: www.shopclues.com

Kecepatan motor DC dapat diatur dengan beberapa cara, yaitu dengan mengatur fluks medan, dengan mengatur tahanan jangkar, dan dengan mengatur tegangan sumber. Cara yang ketiga ini merupakan pengaturan yang sering digunakan karena penggunaannya yang relatif mudah (Zuhal, 2004). Pengaturan tegangan sumber biasanya menggunakan metode *Pulse Width Modulation* (PWM).

2.9 Pulse Width Modulation

PWM adalah metode yang dapat digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal pwm, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada motor. Persamaan untuk perhitungan *duty cycle* ditunjukkan pada persamaan 2.10 dengan T_{on} adalah periode logika tinggi, dan T adalah periode keseluruhan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.18.



Gambar 2.18 Sinyal PWM secara Umum

Sumber: hekkilledmywire.wordpress.com

$$\text{Duty Cycle} = \frac{T_{on}}{T} \times 100\% \dots (\%) \quad (2-10)$$

Dengan :

T_{on} = Periode logika tinggi

T = Periode Keseluruhan

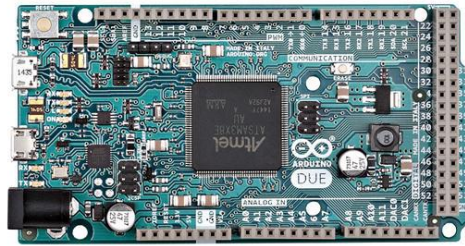
$$V_{dc} = \text{Dutycycle} \times V_{cc} \dots \dots \dots (V)$$

Sedangkan frekuensinya dapat ditentukan dengan rumus sebagai berikut:

$$f_{on} = \frac{f_{clk} / 10}{N \cdot 256} \dots \dots \dots (Hz) \quad (2-11)$$

2.10 Mikrokontroler Arduino Due

Arduino Due adalah papan mikrokontroler berdasarkan Atmel SAM3X8E ARM Cortex-M3 (lihat Gambar 2.19). Board ini memiliki 54 *digital input/output* pin (14 pin dapat digunakan sebagai *output* PWM), 16 *input* analog, 16 MHz osilator kristal, USB koneksi, jack listrik, header ICSP, dan tombol *reset*.



Gambar 2.19 Arduino Due

Sumber: www.arduino.cc

Arduino Due dapat diaktifkan melalui koneksi micro USB atau dengan catu daya eksternal. Eksternal (non-USB) daya dapat berasal baik dari AC ke adaptor DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug jack pusat-positif ukuran 2.1 mm konektor power. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin header dari konektor power. Arduino dapat beroperasi dengan catu daya eksternal 6 sampai 20 volt. Namun jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak papan. Kisaran yang disarankan adalah 7 sampai 12 volt.

Arduino Due memiliki 512 KB flash memory, 96 KB dari SRAM. Masing-masing dari 54 pin digital di Arduino Due dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode ()*, *digitalWrite ()*, dan *digitalRead ()*, beroperasi dengan daya 3,3 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara default terputus) dari 20-50 Kohm. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega16U2 USB-to-TTL.
- Eksternal interrupts: 2 (*interrupt 0*), 3 (*interrupt 1*), 18 (*interrupt 5*), 19 (*interrupt 4*), 20 (*interrupt 3*), dan 2 (*interrupt 2*). Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.
- PWM: 0 - 13. Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite ()*.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan SPI library.

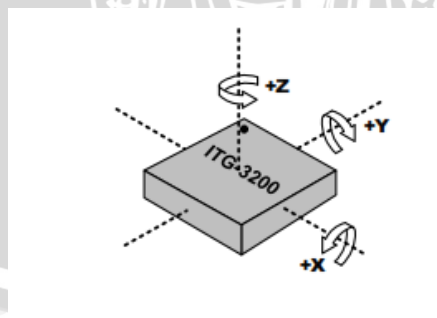
- LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai *high* LED menyala dan ketika pin bernilai *low* LED mati.
- I2C: 20 (SDA) dan 21 (SCL). Dukungan I2C (TWI) komunikasi menggunakan wire.

Arduino Due memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. Arduino Due menyediakan 4 UART TTL (5V) untuk komunikasi serial. Sebuah Arduino Due sebagai saluran komunikasi serial melalui *micro* USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. Namun pada Windows diperlukan sebuah file inf. Perangkat lunak Arduino terdapat *monitor serial* yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari papan Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip *USB-to-serial* dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *Software Serial Library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada board Arduino Due. Arduino Due juga mendukung I2C (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan kawat untuk menyederhanakan penggunaan bus I2C.

2.12 Gyroscope

Gyroscope adalah suatu alat yang berupa sensor *gyro* untuk menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi dengan cepat pada sumbu. Berdasarkan keterangan tersebut, maka pergerakan *gyroscope* digambarkan seperti pada Gambar 2.20.



Gambar 2.20 Analogi 3-Axis Gyroscope

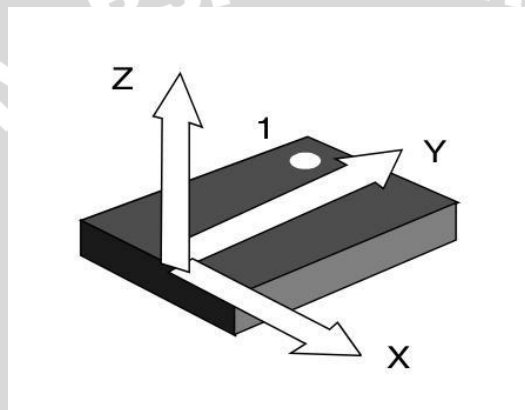
Sumber: datasheet PS-ITG-3200-00-01.4

Gyroscope memiliki *output* yang peka terhadap kecepatan sudut dari arah sumbu x yang nantinya akan menjadi sudut ϕ (*roll*), dari sumbu y nantinya menjadi sudut (*pitch*), dan sumbu z nantinya menjadi sudut ψ (*yaw*).

Penggunaan *gyroscope* dalam kehidupan sehari-hari belum banyak. *Gyroscope* lazimnya digunakan pada pesawat terbang, Kapal, Helikopter, dan lain-lain. Hal tersebut untuk mengurangi getaran yang ditimbulkan mesin agar keseimbangan tidak goyah. *Gyroscope* baru-baru ini juga diterapkan pada mainan Helikopter dengan tujuan menyeimbangkan saat diterbangkan.

2.13 Accelerometer

Accelerometer adalah alat yang digunakan untuk mengukur percepatan mendeteksi dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi. Sensor *accelerometer* mengukur percepatan dari 3 sumbu gerakan akibat gerakan benda yang melekat padanya. Gambar 2.21 menunjukkan daerah kerja *accelerometer*.



Gambar 2.21 Analogi 3-Axis Accelerometer

Sumber: datasheet ADXL345

Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut juga percepatan (*acceleration*). Jika kecepatan semakin berkurang daripada kecepatan sebelumnya, disebut *deceleration*. Percepatan juga bergantung pada arah atau orientasi karena merupakan penurunan kecepatan yang merupakan besaran *vector*. Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula.

BAB III

METODE PENELITIAN

Pada penulisan skripsi ini akan dirancang kontroler Proporsional Integral Derivatif (PID) kaskade yang berfungsi sebagai pengendali robot beroda dua dan Kalman Filter yang berfungsi sebagai pemrosesan data pada sensor *gyroscope*, sensor *accelerometer*, dan sensor *rotary encoder*. Perancangan robot beroda dua ini dilakukan secara bertahap sehingga memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan. Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan metode untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

1. Penentuan spesifikasi alat. .
2. Spesifikasi desain.
3. Karakterisasi setiap blok.
 - a. Karakterisasi *driver motor (L298N Dual H-Bridge Motor Driver)*.
 - b. Karakterisasi motor DC 7,4 V kinmore (KM-25A370-17.6-06397) dengan sensor *optical rotary encoder*.
 - c. Karakterisasi sensor *digital combo board-6/9 degrees of freedom IMU ITG3200/ADXL345*.
 - d. Karakterisasi aktuator (motor DC).
 - e. Penentuan fungsi alih motor DC 7,4 V kinmore (KM-25A370-17.6-06397) dengan sensor *optical rotary encoder*.
4. Perancangan dan pembuatan perangkat keras.

Perancangan dan pembuatan perangkat keras meliputi sistem mekanik, sistem elektrik robot beroda dua dan sistem keseluruhan (*hardware* dan *software*) robot beroda dua.
5. Perancangan algoritma.
6. Perancangan blok diagram sistem.
7. Perancangan dan penyusunan perangkat lunak.

3.1 Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut :

- 1) Robot berbahan dasar FR4 PCB dan mika akrilik.

- 2) Robot beroda dua menggunakan sistem penggerak roda yang terletak disisi kiri dan kanan badan robot serta digerakkan oleh dua buah motor DC.
- 3) Robot beroda dua menggunakan kontroler PID kaskade dan Kalman filter, serta mampu melakukan gerak dasar maju, mundur, dan berotasi terhadap sumbu as roda (hingga 10^0) dan mampu mempertahankan keseimbangan agar robot tidak terjatuh.
- 4) Arena yang digunakan dalam melakukan pengujian memiliki permukaan datar (kemiringan 0^0 terhadap bidang datar).
- 5) Sistem keseimbangan robot beroda dua menggunakan modul sensor yang terdiri dari sensor *gyroscope* dengan tipe ITG-3200 dan sensor *accelerometer* dengan tipe ADXL-345 yang berfungsi sebagai pembacaan kecepatan sudut yang akurat dalam keadaan diam (statis).
- 6) Sistem keseimbangan robot beroda dua menggunakan Arduino Due sebagai *board* utama yang berfungsi untuk memproses *input* dari sensor menuju aktuator.
- 7) Sistem keseimbangan robot beroda dua menggunakan modul yang dirancang oleh Arduino dengan tipe L298N *Dual H-Bridge Motor Driver board* sebagai modul pengendali motor DC yang dapat digunakan mengatur dua buah motor DC secara bersamaan.
- 8) Menggunakan catu daya rangkaian elektronik sebesar 5 V DC dan catu daya motor sebesar 7,4 V.

3.2 Spesifikasi Desain

Desain yang diinginkan pada perancangan pengontrolan kesetimbangan robot beroda dua mempunyai spesifikasi yaitu:

1. Robot beroda dua dapat seimbang antara sudut -10^0 sampai $+10^0$ dari posisi seimbang (sudut tegak lurus terhadap permukaan bumi adalah 0^0).
2. *Error Steady State* $< 5\%$
Error Steady State $< 5\%$, karena sistem yang baik memiliki *output* dengan batas nilai akhir 5% dari *setpoint*.
3. $\xi = 1$
 $\xi = 1$, agar *output* tidak berosilasi dan dapat mencapai *setpoint*

4. *Settling time* < 5 detik

Settling time < 5 detik, karena diharapkan penggunaan *tuning* kontrol PID kaskade mampu mempercepat *settling time* sistem kurang dari 5 detik sehingga saat robot beroda dua diberi gangguan maka robot beroda dua dapat kembali dengan cepat mencapai kesetimbangan.

5. *Output* tanpa *Overshoot*

Output yang memiliki *overshoot* dapat membuat simpangan robot beroda dua semakin besar, semakin besar simpangan robot artinya semakin tidak stabil robot. Penggunaan *tuning* kontroler PID kaskade diharapkan mampu meminimalkan *overshoot* sehingga membuat robot beroda menjadi lebih stabil dalam mempertahankan kesetimbangan.

3.3 Karakterisasi Setiap Blok

Pada bagian ini pengujian dilakukan pada masing-masing blok. Pengujian ini bertujuan untuk mengetahui apakah masing-masing blok dapat bekerja sesuai dengan fungsinya seperti yang telah direncanakan. Pengujian tersebut meliputi :

3.3.1 Karakterisasi *Driver Motor* (*L298N Dual H-Bridge Motor Driver*)

Tujuan dari karakterisasi *driver motor* (*L298N Dual H-Bridge Motor Driver*) adalah mengetahui karakteristik, kinerja dan *output* rangkaian *driver motor* *L298N Dual H-Bridge Motor Driver* dengan membandingkan *output* tegangan *driver* dengan masukan *duty cycle* sinyal PWM yang diberikan oleh Arduino Due.

Peralatan yang digunakan terdiri atas:

1. Catu daya (Baterai LI-PO 2S 7.4 Volt)
2. *Driver Motor* (*L298N Dual H-Bridge Motor Driver*).
3. Arduino Due.
4. Multimeter.
5. Perangkat komputer.
6. Kabel penghubung.

Prosedur pengujian meliputi:

1. Menghubungkan *output* tegangan Catu daya (Baterai LI-PO 2S 7.4 Volt) pada *input* tegangan referensi *driver motor* (*L298N Dual H-Bridge Motor Driver*).

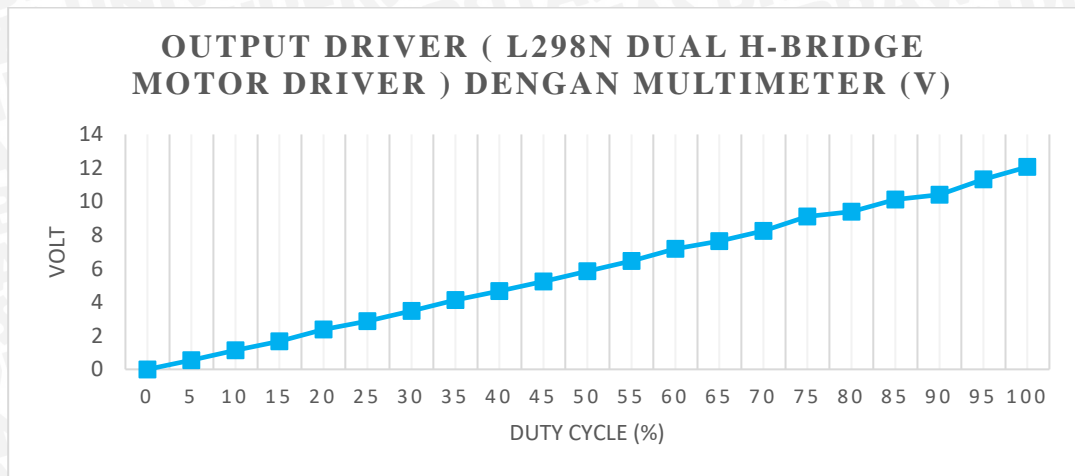
2. Menghubungkan *input* tegangan *driver* motor (L298N *Dual H-Bridge Motor Driver*) dengan pin *output* PWM di Arduino Due.
3. Menghubungkan *output* tegangan *driver* motor (L298N *Dual H-Bridge Motor Driver*) dengan multimeter.
4. Mengatur *duty cycle* sinyal PWM pada Arduino Due dengan nilai 0%-100%.
5. Mengamati dan mencatat hasil pembacaan multimeter di setiap kenaikan 5%.

Data pengujian *driver* motor (L298N *Dual H-Bridge Motor Driver*) ditunjukkan dalam Tabel 3.1. Pada Tabel tersebut terdapat *duty cycle* sebagai *input driver*. *Output driver* berupa tegangan yang diukur menggunakan multimeter.

Duty Cycle (%)	Output driver (L298N Dual H-Bridge Motor Driver) dengan multimeter (V)
0	0
5	0.541
10	1.143
15	1.674
20	2.382
25	2.863
30	3.482
35	4.124
40	4.672
45	5.241
50	5.847
55	6.46
60	7.183
65	7.65
70	8.25
75	9.004
80	9.302
85	10.128
90	10.5
95	11.416
100	12.01

Tabel 3.1 Hasil Pengujian Driver (L298N Dual H-Bridge Motor Driver)

Berdasarkan Tabel 3.1 didapatkan kurva *output* tegangan *driver* (V) terhadap masukan *duty cycle* sinyal PWM (%) seperti ditunjukkan pada Gambar 3.1



Gambar 3.1 Perubahan Tegangan *Output Driver* terhadap *Duty Cycle*

Dari grafik didapatkan *gain driver* motor adalah :

$$m = \frac{y_{60} - y_{50}}{x_{60} - x_{50}} = \frac{7.183 - 5.847}{50 - 40} = 0.133$$

3.3.2 Karakterisasi Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan sensor *Optical Rotary Encoder*

Tujuan karakterisasi motor DC adalah mengetahui karakter atau gain motor DC. Hal tersebut diperoleh dengan mengamati kecepatan motor DC terhadap perubahan tegangan *input* motor DC.

Peralatan yang digunakan terdiri atas:

1. Arduino Due R3,
2. Motor DC Kinmore (KM-25A370-17.6-06397)
3. Tachometer digital,
4. Perangkat komputer, dan
5. Catu daya (Baterai LI-PO 2S 7.4 Volt).

Prosedur pengujian meliputi:

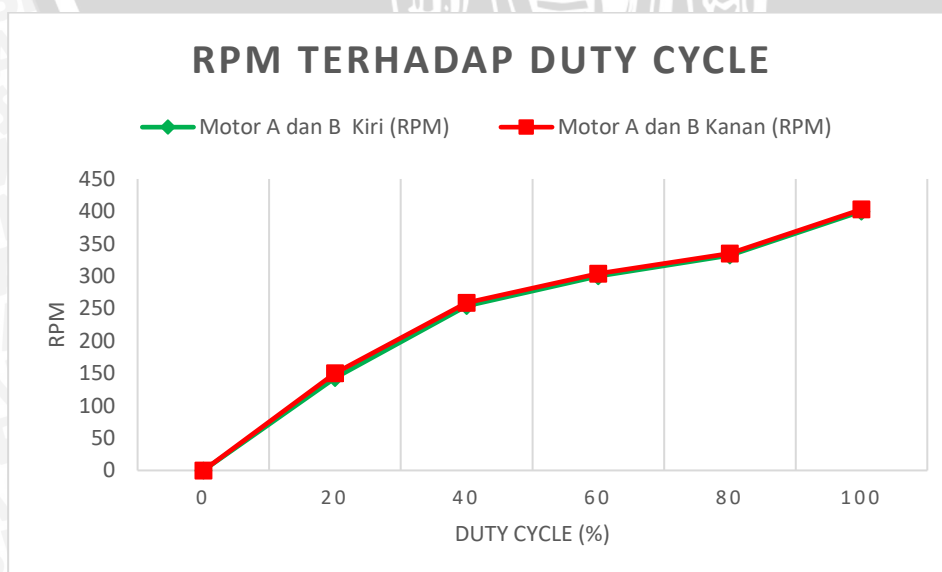
1. Menghubungkan *output* tegangan Catu daya (Baterai LI-PO 2S 7.4 Volt) pada *input* tegangan referensi *driver* motor (L298N Dual H-Bridge Motor Driver).
2. Menghubungkan *input* tegangan *driver* motor (L298N Dual H-Bridge Motor Driver) dengan pin output PWM di Arduino Due

3. Menghubungkan *output* tegangan *driver* motor dengan motor DC 7,4 V
4. Mengatur *duty cycle* sinyal PWM pada Arduino Due dengan nilai 0%-100%.
5. Mengamati dan mencatat hasil pembacaan putaran motor setiap kenaikan 20%.

Data hasil pengujian motor DC 7,4 V kinmore (KM-25A370-17.6-06397) dengan sensor *optical rotary encoder* ditunjukkan dalam Tabel 3.2.

Data Motor DC A dan MOTOR DC B				
NO	Duty Cycle (%)	PWM	RPM	Arah putaran Motor DC
1	0	0	0	Diam
2	20	-51	143	Berputar ke Kiri
3	40	-102	254	Berputar ke Kiri
4	60	-153	300	Berputar ke Kiri
5	80	-204	332	Berputar ke Kiri
6	100	-255	400	Berputar ke Kiri
7	0	0	0	Diam
8	20	51	150	Berputar ke Kanan
9	40	102	259	Berputar ke Kanan
10	60	153	304	Berputar ke Kanan
11	80	204	335	Berputar ke Kanan
12	100	255	403	Berputar ke Kanan

Tabel 3.2 Data Pengukuran Arah dan Kecepatan Motor DC



Gambar 3.2 Pengukuran Arah dan Kecepatan Motor DC

Dari hasil pengujian aktuator yaitu motor DC Kinmore (KM-25A370-17.6-06397) berdasarkan Tabel 4.1 dan gambar 3.2 didapatkan karakteristik dari motor DC Kinmore (KM-25A370-17.6-06397) semakin besar *duty cycle* (%) yang digunakan maka akan semakin besar pula kecepatan putaran yang akan dihasilkan oleh Kinmore (KM-25A370-17.6-06397).

3.3.3 Karakterisasi Sensor *Digital Combo Board-6/9 Degrees of Freedom IMU ITG3200/ADXL345*

Dalam melakukan pengujian sensor, terdapat dua kali pengujian dengan hasil yang diinginkan berbeda satu sama lain yaitu pengujian sensor IMU (sensor *gyroscope* dan sensor *accelerometer*). Namun modul sensor yang digunakan dalam melakukan pengujian ini adalah sama karena modul sensor IMU tersebut merupakan satu modul yang terdiri dari sensor *gyroscope* dengan tipe ITG-3200 dan sensor *accelerometer* dengan tipe ADXL-345.

3.3.3.1 Karakterisasi Sensor *Gyroscope (ITG3200) Digital Combo Board-6/9 Degrees of Freedom IMU*

Tujuan dari karakterisasi sensor *Digital Combo Board-6/9 Degrees of Freedom IMU ITG3200/ADXL345* adalah mengetahui tingkat keakuratan dari sensor *gyroscope* dalam membaca perubahan posisi sudut yang dinamis dari robot beroda dua.

Peralatan yang digunakan terdiri atas:

1. Arduino Due R3,
2. Sensor SparkFun 6 *Degrees of Freedom IMU Digital Combo Board - ITG3200/ADXL345*,
3. Sistem mekanik robot beroda dua yang terhubung dengan sensor *gyroscope* + sensor *accelerometer* dan mikrokontroler Arduino Due R3
4. Perangkat komputer,
5. Catu daya 5 Volt,dan
6. Catu daya 3,3 Volt.

Prosedur pengujian meliputi:

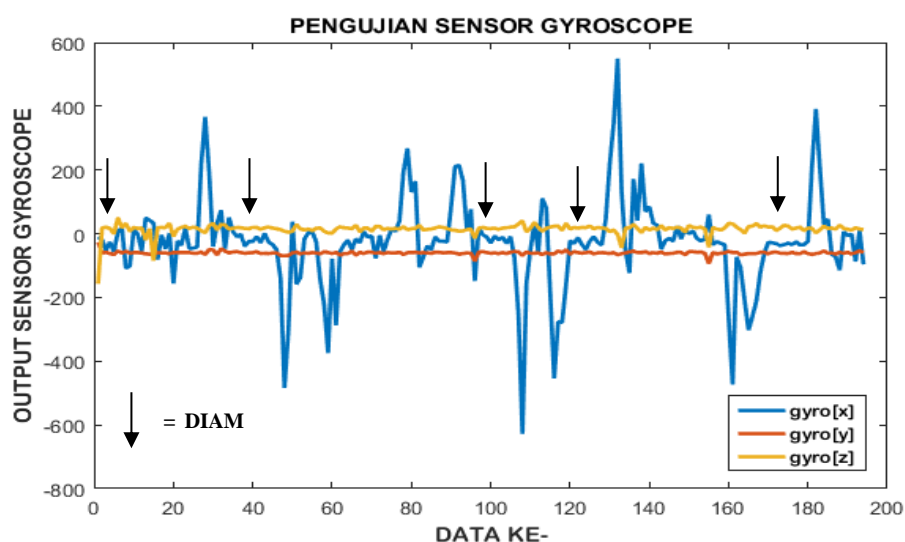
1. Robot dikondisikan dalam posisi tegak, kemudian program pembacaan dan *data logger* sensor *gyroscope* dijalankan. Pada saat program sedang *running*, robot dimiringkan ke depan dan ke belakang secara kontinyu dalam selang waktu tertentu.

2. Setelah selang waktu tersebut, pembacaan dan *data logger output* sensor dihentikan dan kemudian data *output* berupa besaran sudut tersebut dilihat.
3. Menggambarkan data keluaran sensor *gyroscope* pada grafik untuk melihat kelinieran dari data modul sensor tersebut.

Pengujian sensor *gyroscope* terhubung dengan mekanik robot beroda dua dan mikrokontroler Arduino Due. Kemudian data dari sensor *gyroscope* akan dibuat *input* dan keluarannya adalah kecepatan putar motor DC. Kemiringan dari robot beroda dua diubah dalam lang waktu tertentu. Perubahan sudut dari sensor *gyroscope* yang terjadi akan ditampilkan lewat komputer pada *serial monitor* arduino.

Pengujian sensor *gyroscope* yang merupakan sensor kecepatan sudut. Kecepatan sudut merupakan besaran vektor dimana nilainya juga bergantung pada arah pergerakan robot beroda dua. Dimana pengujian dilakukan dengan cara memiringkan robot ke depan dan ke belakang secara kontinyu dalam selang waktu tertentu.

Pada mulanya robot dikondisikan dalam posisi tegak, kemudian program pembacaan dan *data logger* sensor *gyroscope* dijalankan. Pada saat program sedang running, robot dimiringkan ke depan dan ke belakang secara kontinyu dalam selang waktu tertentu. Setelah selang waktu tersebut, pembacaan dan *data logger output* sensor dihentikan dan kemudian data *output* tersebut dilihat. Gambar 3.3 menunjukkan data *output* dari sensor *gyroscope* hasil dari pengujian.



Gambar 3.3 Data Hasil Pengujian *Output* Sensor *Gyroscope*

Berdasarkan data yang terdapat pada Gambar 3.3 di atas, ketika sensor *gyroscope* diam, nilai *output* dari sensor adalah mendekati nol, artinya kecepatan sudut dari robot

mendekati nilai nol. Ketika robot digerakkan dengan arah pergerakan yang pertama, robot digerak-miringkan ke depan, nilai *output* sensor *gyroscope* adalah sekitar $600^{\circ}/s$. Nilai kecepatan sudut yang dihasilkan tersebut bernilai positif. Dan ketika robot digerakkan dengan arah pergerakan yang kedua, robot digerak-miringkan ke belakang, nilai *output* adalah sekitar $-600^{\circ}/s$. Nilai kecepatan sudut yang dihasilkan adalah bernilai negatif.

Berdasarkan Gambar 3.3 di atas dan penjabaran data pada paragraf di atas, sensor *gyroscope* mendeteksi kemiringan melalui arah pergerakan robot. Dimana, ketika robot digerakmiringkan ke depan, nilai *output* sensor *gyroscope* bernilai positif dan bernilai negatif ketika digerak-miringkan ke belakang. Nilai *output* sensor *gyroscope* selalu mendekati nilai nol ketika robot dalam keadaan diam walaupun terjadi kemiringan pada robot.

3.3.3.2 Karakterisasi Sensor *Accelerometer* (ADXL345) *Digital Combo Board-6/9*

Degrees of Freedom IMU

Tujuan dari karakterisasi sensor *accelerometer* (ADXL345) *Digital Combo Board-6/9 Degrees of Freedom IMU* ITG3200/ADXL345 adalah mengetahui tingkat keakuratan dari *accelerometer* dalam membaca perubahan sudut dari robot beroda dua.

Peralatan yang digunakan terdiri atas:

1. Arduino Due R3,
2. Sensor SparkFun 6 *Degrees of Freedom IMU Digital Combo Board - ITG3200/ADXL345*,
3. Sistem mekanik robot beroda dua yang terhubung dengan sensor *gyroscope* + sensor *accelerometer* dan mikrokontroler Arduino Due R3
4. Perangkat komputer,
5. Catu daya 5 Volt,dan
6. Catu daya 3,3 Volt.

Prosedur pengujian meliputi:

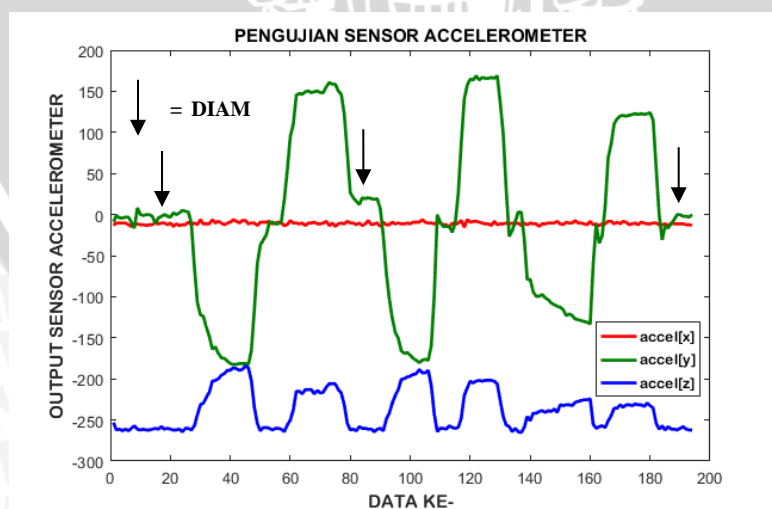
1. Robot dikondisikan dalam posisi tegak, kemudian program pembacaan dan *data logger* sensor *gyroscope* dijalankan.Pada saat program sedang *running*, robot dimiringkan ke depan dan ke belakang secara kontinu dalam selang waktu tertentu.

2. Setelah selang waktu tersebut, pembacaan dan *data logger output* sensor dihentikan dan kemudian data *output* berupa besaran sudut tersebut dilihat .
3. Menggambarkan data keluaran sensor *accelerometer* pada grafik untuk melihat kelinieran dari data modul sensor tersebut.

Pengujian sensor *accelerometer* terhubung dengan mekanik robot beroda dua dan mikrokontroler Arduino Due. Kemudian data dari sensor *accelerometer* akan dibuat *input* dan keluarannya adalah kecepatan putar motor DC. Kemiringan dari robot beroda dua diubah-ubah dalam lang waktu tertentu. Perubahan sudut dari sensor *accelerometer* yang terjadi akan ditampilkan lewat komputer pada *serial monitor* Arduino.

Sensor *accelerometer* adalah salah satu sensor yang akan digunakan untuk mendeteksi kemiringan atau posisi robot beroda dua. Untuk itu, hal yang pertama diuji adalah *output* dari sensor *accelerometer*. Sensor *accelerometer* mengukur kemiringan robot beroda dua dalam 3 sumbu, yaitu x, y dan z. Pengujian dilakukan dengan cara memiringkan robot ke depan dan ke belakang secara kontinyu dalam selang waktu tertentu.

Pada mulanya, robot dikondisikan dalam posisi tegak, kemudian program pembacaan dan *data logger* sensor *accelerometer* dijalankan. Pada saat program sedang *running*, robot dimiringkan ke depan dan ke belakang secara kontinyu dalam selang waktu tertentu. Setelah selang waktu tersebut, pembacaan dan *data logger output* sensor dihentikan dan kemudian data *output* tersebut dilihat. Gambar 3.4 adalah gambar yang menunjukkan data *output* dari sensor *accelerometer* hasil dari pengujian.



Gambar 3.4 Data Hasil Pengujian *Output* Sensor *Accelerometer*

Berdasarkan Gambar 3.4 di atas, *range* nilai *output* sensor *accelerometer* adalah berkisar antara +300 hingga -300. Ketika robot berada pada kemiringan awal, sedikit

miring ke belakang, x bernilai sekitar -10, y bernilai sekitar 0, z bernilai sekitar -261. Kemudian robot lebih dimiringkan ke belakang, x berubah nilai menjadi sekitar -13, y bernilai sekitar -168, dan z bernilai sekitar -188. Pada kemiringan robot yang ketiga, yaitu robot dimiringkan ke depan, x bernilai sekitar -10, y bernilai sekitar 148 dan z bernilai sekitar 212. Nilai x yang bertanda negatif mengindikasikan bahwa arah kemiringan robot mempengaruhi tanda nilai dari x . Ketika robot berada posisi kemiringan yang keempat, robot dimiringkan ke belakang, x bernilai sekitar -9, y bernilai sekitar -175 dan z bernilai sekitar -196. Dan pada kemiringan robot yang kelima, robot dimiringkan ke depan, x masih bernilai sekitar -9, y masih bernilai sekitar 163 dan z bernilai -202.

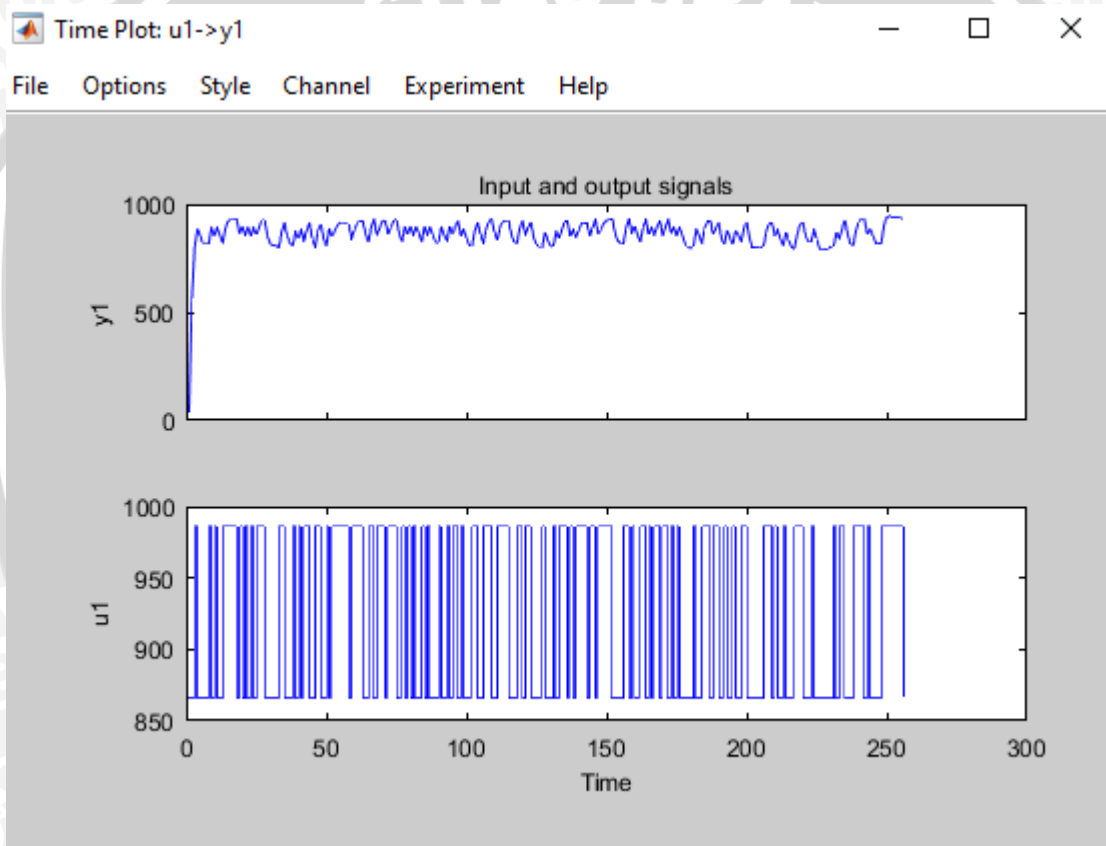
Berdasarkan Gambar 3.4 dan penjabaran data pada paragraf di atas, nilai x dari hasil pengujian mempresentasikan nilai dari kemiringan robot. Hal itu dapat dilihat ketika robot berdiri tegak, nilai y mendekati nol, bernilai makin positif ketika semakin dimiringkan ke depan dan bernilai makin negatif ketika robot semakin dimiringkan ke belakang. Untuk nilai x dan z , kedua *output* tersebut tidak mempresentasikan nilai dari kemiringan robot. Nilai x selalu berada pada nilai sekitar lebih kecil dari nol, seperti tidak merespon terjadinya kemiringan robot. Begitu juga nilai dari z , tidak mempresentasikan nilai dari kemiringan robot. Dapat disimpulkan bahwa sensor *accelerometer* sudah bekerja dengan baik dan *output* sensor *accelerometer* yang digunakan untuk mendeteksi kemiringan robot adalah nilai dari y .

3.3.4 Penentuan Fungsi Alih Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan Sensor *Optical Rotary Encoder*

Pengontrolan kecepatan motor DC 7,4 V kinmore (KM-25A370-17.6-06397) dengan sensor *optical rotary encoder* menggunakan Arduino Due sebagai pengolah dan memberikan data berupa *Pulse Width Modulation* (PWM) agar motor bergerak. Motor DC yang digunakan pada perancangan ini tidak diketahui karakteristiknya, sehingga yang perlu dilakukan adalah melakukan pengujian dengan menggunakan sensor *optical rotary encoder*. Karakteristik motor DC 7,4 V kinmore (KM-25A370-17.6-06397) dengan sensor *optical rotary encoder* pada perancangan ini didapatkan dengan cara memberikan *input unit step*.

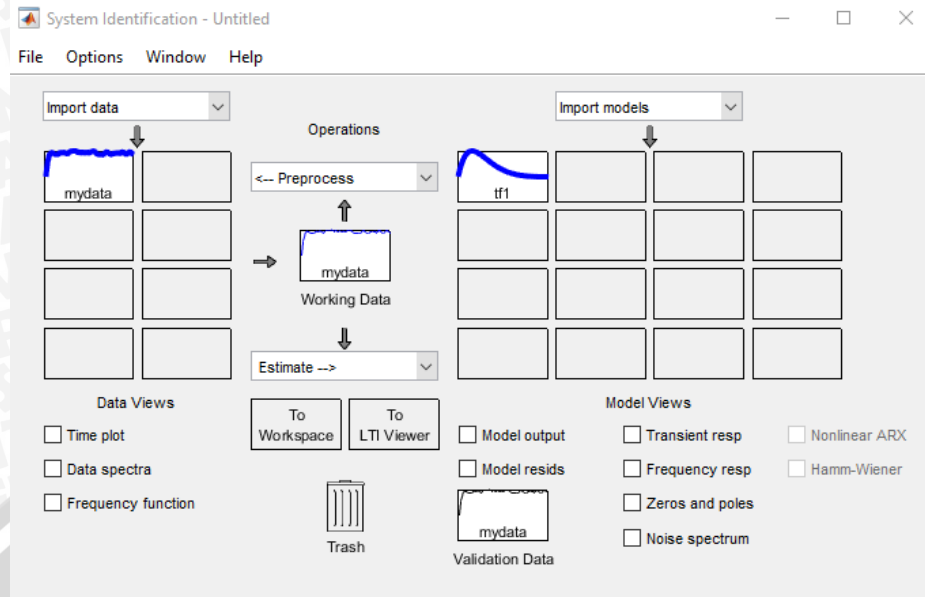
Fungsi alih motor DC didapatkan dari pemodelan dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS). Langkah yang dilakukan untuk membangkitkan sinyal PRBS adalah sebagai berikut:

1. Mencari nilai yang linear dari hasil tegangan *output* terhadap *duty cycle* PWM.
2. Memasukkan nilai batas atas dan bawah berdasarkan nilai yang linier untuk membangkitkan sinyal PRBS.
3. Sinyal PRBS yang telah dibangkitkan kemudian digunakan sebagai *input* motor DC.
4. Setelah didapatkan data sinyal PRBS dan data kecepatan *output* motor DC 7,4 V kinmore (KM-25A370-17.6-06397) dengan sensor *optical rotary encoder* (lihat Gambar 3.5), selanjutnya adalah melakukan identifikasi dengan menggunakan *software* Matlab.



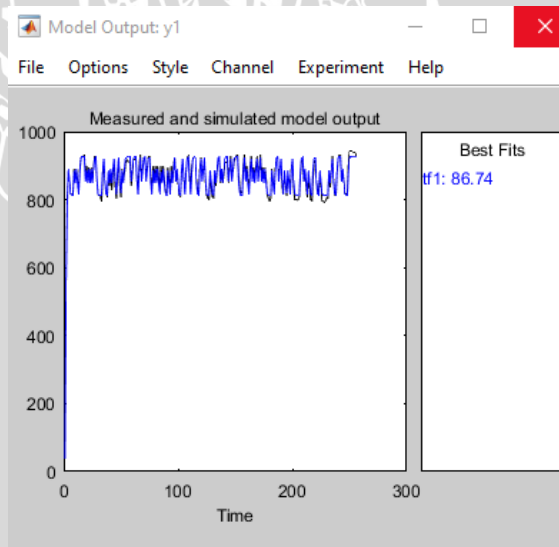
Gambar 3.5 *Input dan Output Sinyal PRBS*

5. Dengan menggunakan sintaks *ident* pada *command window* pada Matlab, data sinyal PRBS dan data kecepatan motor yang telah disimpan kemudian di *import* pada blok *System Identification Toolbox* (lihat Gambar 3.6).



Gambar 3.6 System Identification Toolbox

6. Setelah melakukan beberapa estimasi model berdasarkan data yang telah di import didapatkan fungsi alih motor DC dengan *best fits* sebesar 86.74 seperti dalam Gambar 3.7.



Gambar 3.7 Hasil Estimasi Model

7. Dari hasil identifikasi, fungsi alih motor DC yang didapat adalah:

$$\frac{Y(s)}{U(s)} = \frac{3.535}{s^2 + 3.308s + 3.765} \quad (3-1)$$

$$F(s) = \frac{3.535}{s^2 + 3.308s + 3.765} \quad (3-2)$$

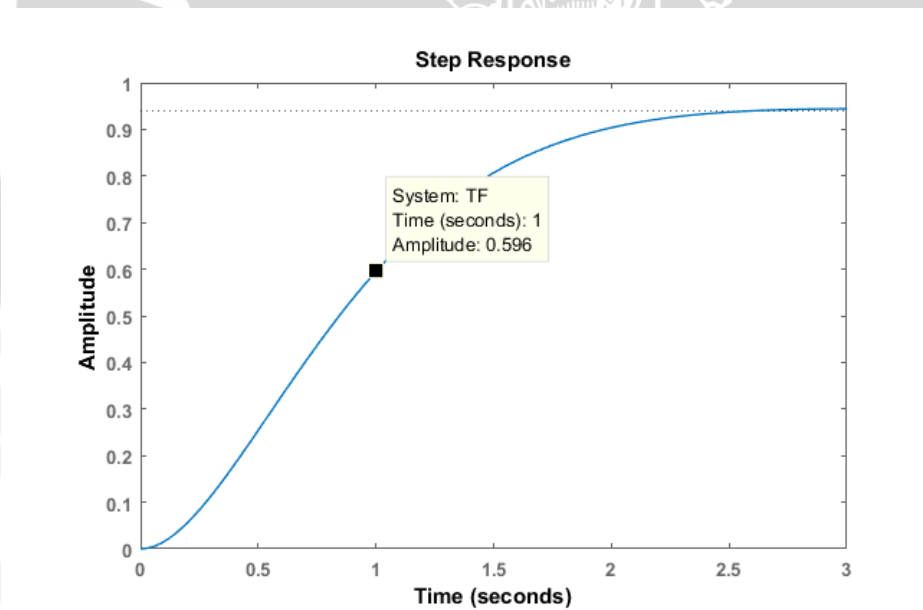
$$F(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3-2)$$

$$2\xi\omega_n = 3.308 \quad (3-3)$$

$$\omega_n = \sqrt{3.765} = 1.940 \quad (3-4)$$

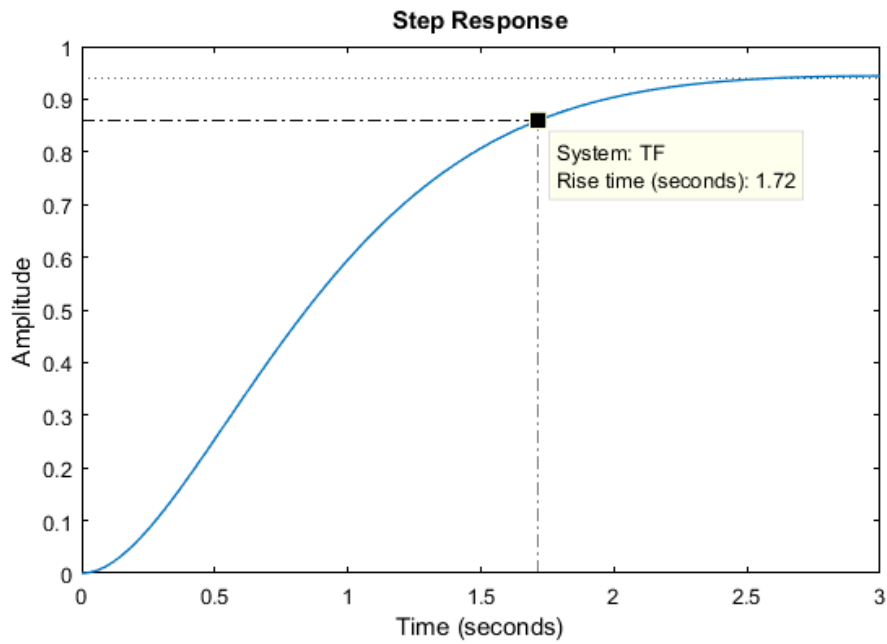
$$\xi = \frac{3.535}{2 \times 1.940} = 0.911 \quad (3-5)$$

Dengan memberikan *input unit step* pada program Matlab didapatkan *output* dengan nilai *time constant* yang merupakan waktu yang dibutuhkan untuk mencapai 63,2% dari nilai *steady state*, yaitu 0.596 detik seperti dalam Gambar 3.8.



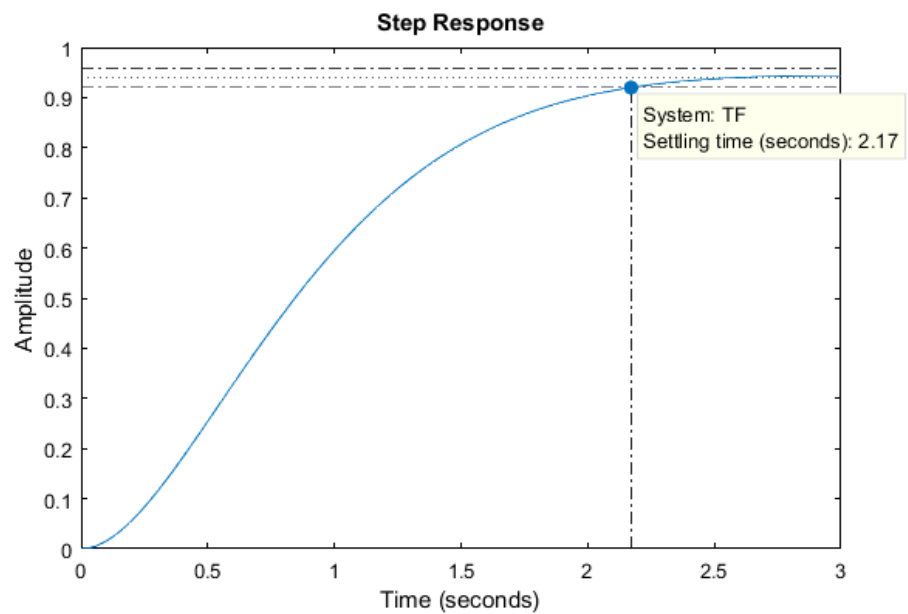
Gambar 3.8 Nilai *Time Constant Output* Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan *Input Unit Step*

Dengan memberikan *input unit step* pada program Matlab didapatkan nilai *rise time* adalah 1.726 detik seperti dalam Gambar 3.9.



Gambar 3.9 Nilai *Rise Time* Output Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan Input Unit Step

Dengan memberikan *input unit step* pada program Matlab didapatkan nilai *settling time* adalah 2.171 detik seperti dalam Gambar 3.10.



Gambar 3.10 Nilai *Settling Time* Output Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397 dengan Input Unit Step

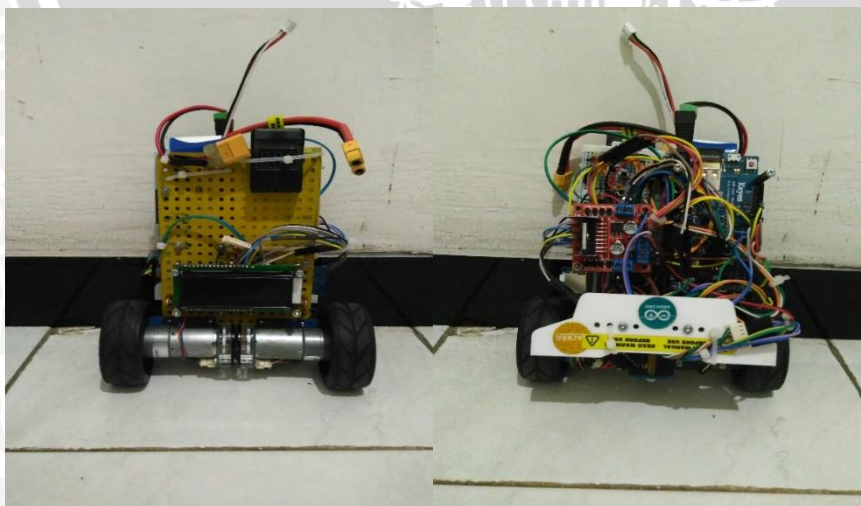
3.4 Perancangan dan Pembuatan Perangkat Keras

Perancangan dan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya. Hal ini dimaksudkan agar robot beroda dua dengan menggunakan metode PID kaskade dan Kalman filter dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan.

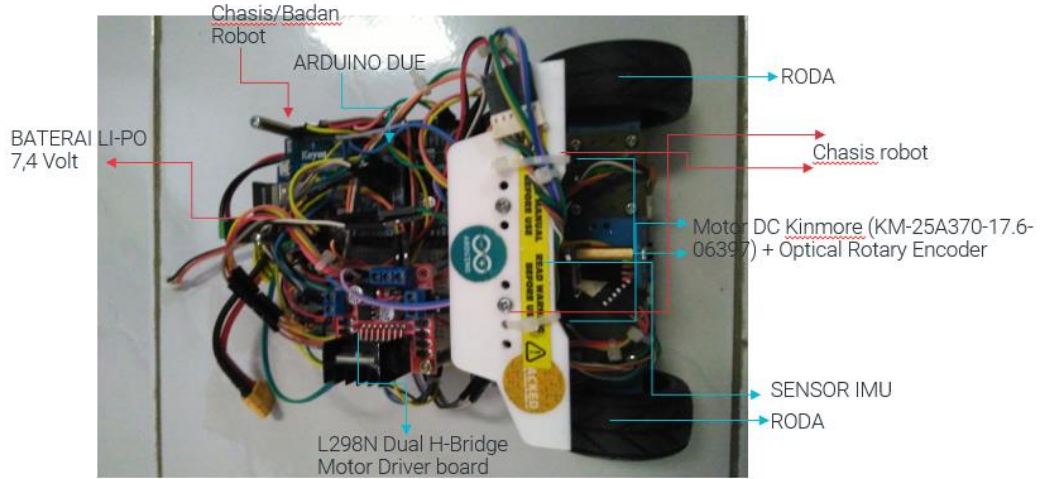
3.4.1 Perancangan Sistem Mekanik Robot Beroda Dua

Sistem mekanik yang baik, mendukung pergerakan robot menjadi lebih baik, oleh karena itu perancangan mekanik robot beroda dua dalam hal ini bodi dan rangka robot harus proporsional dengan panjang dan lebar serta tinggi dari robot. Chasis pada robot beroda dua merupakan komponen mekanis yang menjadi penopang dari komponen-komponen yang lain seperti rangkaian kontroler utama, modul sensor, modul aktuator, baterai (LI-PO), dan lain-lain. Badan robot/chasis robot terbuat dari bahan fiber (FR4) yang biasanya digunakan sebagai bahan PCB dengan ketebalan 1,6 mm. Bahan ini dipilih karena relatif ringan, kuat dan tidak mudah bengkok. Lubang koneksi pada chasis dibuat dalam bentuk array, untuk memberikan fleksibilitas bagi pengguna untuk memasang komponen pada chasis tersebut. Komponen bisa dihubungkan sesuai desain robot yang ingin dibuat dalam beberapa bentuk dan posisi.

Gambar perspektif dan rancangan ukuran robot beroda dua ditunjukkan dalam Gambar 3.11, Gambar 3.12, dan Gambar 3.13.



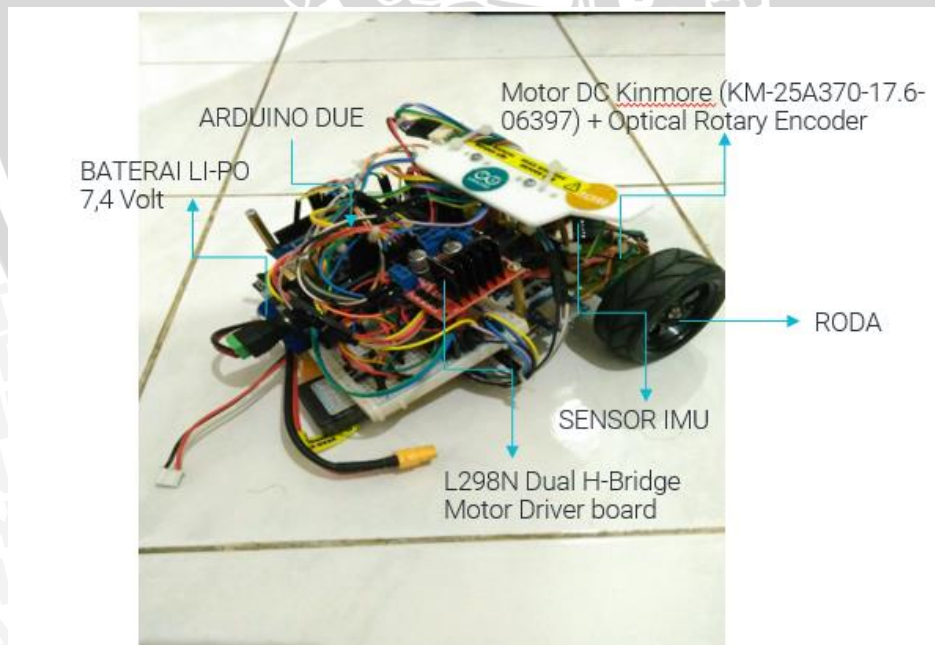
Gambar 3.11 Desain Mekanik Robot Beroda Dua



Gambar 3.12 Perspektif Desain Mekanik Robot Beroda Dua

3.4.2 Perancangan Sistem Elektrik Robot Beroda Dua

Sistem elektrik, terdiri dari bagian catu daya (baterai LI-PO), bagian masukan, bagian kendali, dan bagian keluaran. Pada bagian masukan berupa sensor IMU (sensor *gyroscope* + sensor *accelerometer*) dan sensor *optical rotary encoder*. Pada bagian kendali menggunakan mikrokontroler Arduino Due R3. Pada bagian keluaran berupa *driver* motor sebagai penggerak aktuator robot. Gambar 3.13 menunjukkan penempatan dari rangkaian elektrik pada robot beroda dua.



Gambar 3.13 Desain Sistem Elektrik Robot Beroda Dua

3.5 Perancangan Algoritma

Dalam perancangan perangkat lunak, hal pertama yang dilakukan adalah mengetahui karakteristik motor DC 7,4 V kinmore (KM-25A370-17.6-06397) dengan sensor *optical rotary encoder*. Setelah itu untuk mencari parameter kontroler PID dengan metode *trial and error*. Setelah didapatkan parameter kontroler kemudian didiskritisasi agar dapat diolah oleh mikrokontroler.

3.5.1 Kalman Filter

Sebelum Kalman filter dapat digunakan dan dapat diimplementasikan pada robot beroda dua dilakukan perancangan algoritma Kalman filter dalam bentuk sistem matematis. Secara umum algoritma Kalman filter akan ditemukan banyak matriks yang rumit dan vektor matematis. Pada penelitian ini menggunakan *simple one dimensional* Kalman filter agar lebih mudah dalam perancangan algoritma Kalman filter ke dalam bahasa pemrograman bahasa c pada software Arduino IDE. Algoritma *simple one dimensional Kalman filter* seperti pada persamaan berikut :

$$\begin{aligned} x &= x \\ p &= p + q \end{aligned} \quad (3-6)$$

$$\begin{aligned} k &= p / (p + r) \\ x &= x + k * (\text{measurement} - x) \\ p &= (1 - k) * p \end{aligned} \quad (3-7)$$

Pada persamaan 3.6 yang pertama dua rumus ini mewakili dari prediksi Kalman filter sedangkan persamaan 3.7 tiga kedua rumus tersebut menghitung pembaharuan pengukuran.

Variabel x untuk nilai filter, q untuk proses bising, r untuk derau sensor, p untuk kesalahan perkiraan dan k untuk penguatan Kalman. Keadaan yang ditentukan oleh filter nilai-nilai variabel-variabel ini. Filter yang diterapkan dengan setiap pengukuran dan diinisialisasi dengan derau proses q, derau sensor r, perkiraan awal kesalahan saat p dan nilai awal x. Nilai awal untuk p tidak sangat penting karena disesuaikan selama proses. Itu harus cukup tinggi untuk mempersempit. Nilai awal untuk pembacaan juga tidak sangat penting, karena telah diperbarui selama proses. Tetapi nilai-nilai untuk proses derau dan derau sensor sangat penting untuk mendapatkan pembacaan sudut yang akurat.

3.5.2 Tuning PID

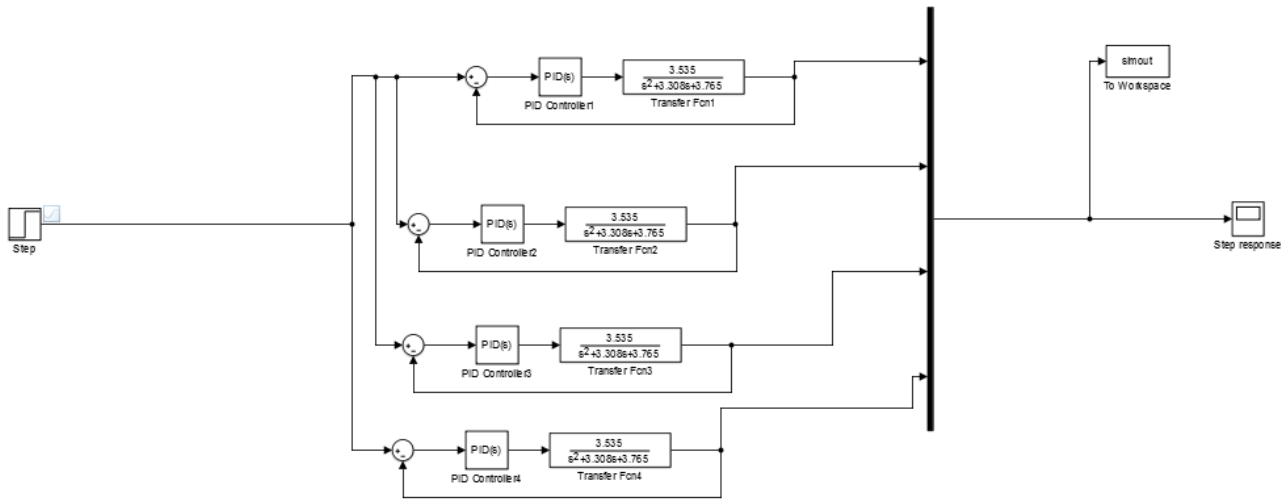
Tuning PID dalam penelitian ini dilakukan secara manual atau dengan metoda *trial and error*. Metoda *trial and error*. Metoda ini umumnya cocok digunakan untuk beberapa sistem kontrol, salah satu diantaranya adalah sistem kontrol PID. Dalam merancang sistem kontrol PID *output* yang diharapkan dari metoda *trial and error* harus memenuhi beberapa kriteria seperti:

1. Memiliki *rise time* dan *settling time* yang cepat.
2. Tidak memiliki *steady state error*.
3. *Overshoot* sekecil mungkin.

Berikut adalah langkah-langkah yang dilakukan dalam *tuning* PID:

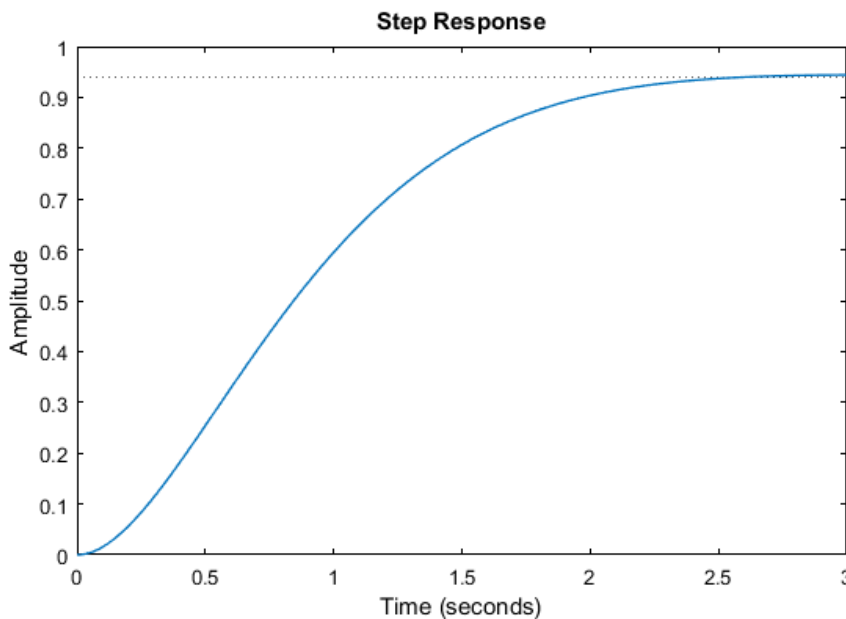
1. Membuat robot beresilasi dengan cara menentukan mengubah parameter *gain* proporsional (K_p), mengabaikan parameter derivatif (K_d) dan integral (K_i) terlebih dahulu, dimana *gain* proporsional (K_p) menguatkan *output* agar *output* mendekati *input* (*setpoint*) dan memelihara sistem untuk selalu dalam keadaan stabil .
2. Setelah diperoleh nilai parameter K_p yang sesuai, langkah berikutnya adalah menentukan nilai parameter K_i dan mengabaikan nilai K_d . Nilai K_i dituning agar didapatkan keluaran keadaan tunak lebih mendekati *setpoint*. Dengan penambahan integral maka sistem akan tetap dipaksa supaya tetap pada keadaan *steady* dan juga akan mengurangi *error*.
3. Langkah selanjutnya adalah menentukan nilai parameter K_d ., maka nilai parameter K_d . Penambahan fungsi derivatif dimaksudkan agar sistem lebih peka terhadap *error* dan mengurangi osilasi, karena dengan penambahan nilai parameter derivatif berarti menambahkan nilai *zero* sehingga parameter ini mempercepat *output*.

Setelah didapatkan nilai K_p K_i dan K_d dari hasil metode *tuning* pid manual, kemudian dilakukan pengujian terhadap sistem, dan parameter yang sesuai dengan sistem. dengan menggunakan MATLAB *SIMULINK* 2016 ditunjukkan pada Gambar 3.14 berikut ini:

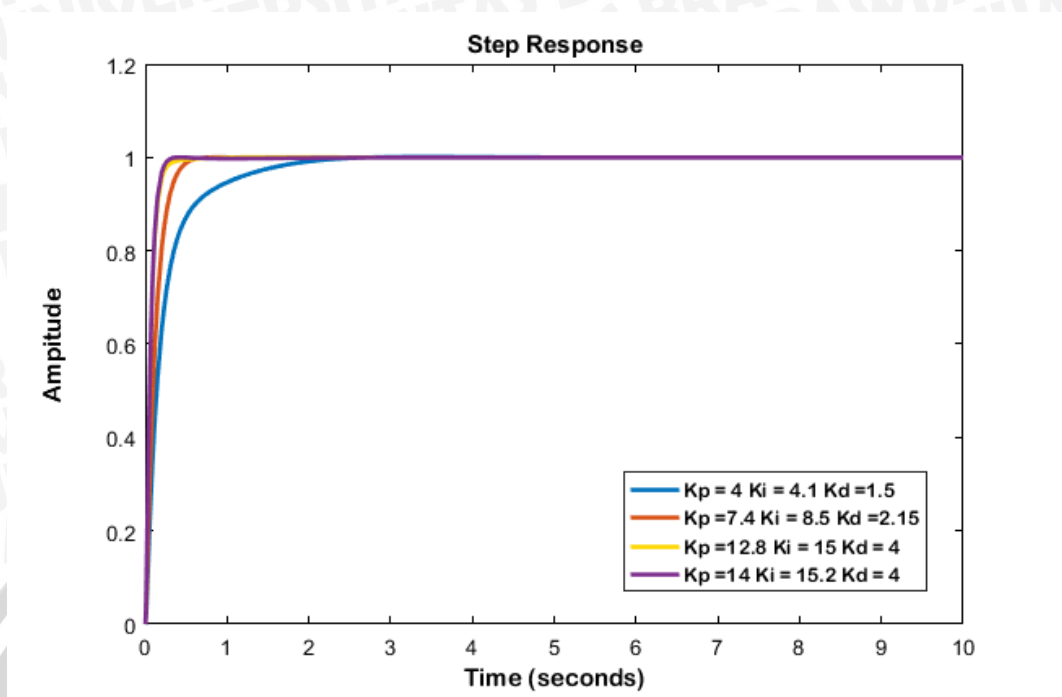


Gambar 3.14 Simulink Tuning PID Manual

Dari Gambar 3.15 dapat diketahui bahwa *output* sistem tanpa menggunakan PID lebih lama menuju *steady state*. Sedangkan pada Gambar 3.16 dengan digunakannya parameter PID hasil *tuning* manual didapatkan *output* yang lebih cepat daripada *output* tanpa menggunakan PID. Dari 4 jenis parameter PID yang didapat dipilih nilai PID yang memiliki *output* terbaik yaitu: $K_p = 14$, $K_i = 15.2$, and $K_d = 4$.



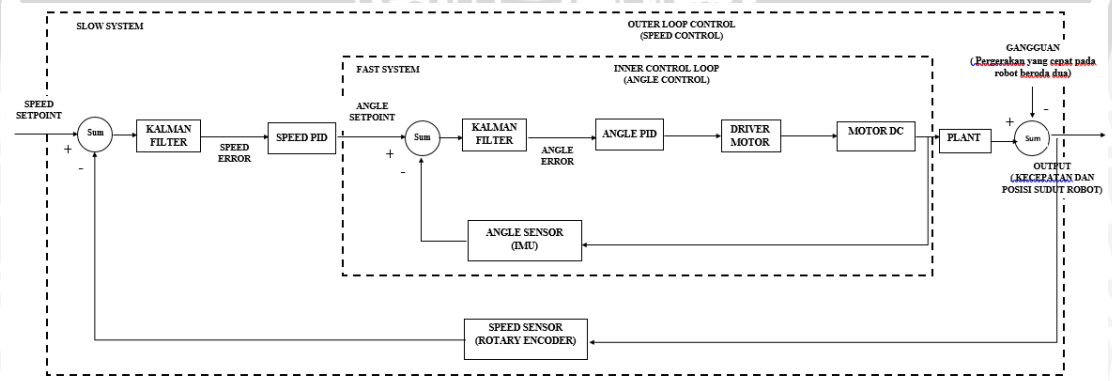
Gambar 3.15 Output Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) tanpa Kontroler PID



Gambar 3.16 Output Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397) dengan Kontroler PID

3.6 Perancangan Blok Diagram Sistem

Pada perancangan alat diperlukan perancangan diagram blok sistem yang dapat menjelaskan sistem secara garis besar yang sangat diperlukan pada perancangan alat dan diharapkan alat dapat bekerja sesuai dengan yang diharapkan dan direncanakan didesain dapat dilihat pada Gambar 3.17.



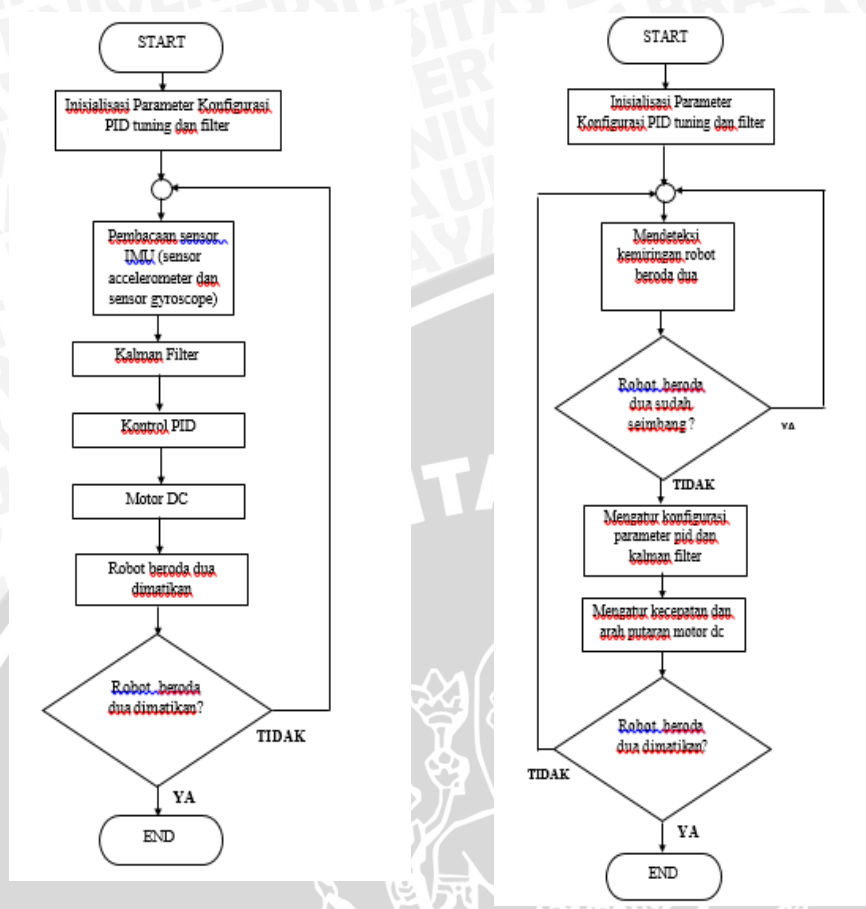
Gambar 3.17 Blok Diagram Sistem Loop Tertutup Keseimbangan Robot Beroda Dua

Keterangan dari diagram blok pada Gambar 3.17 sebagai berikut:

1. *Loop* luar (*inner control loop*) *input* berupa kecepatan. Sementara *loop* dalam (*outer control loop*) berupa sudut, apabila pertama kali robot beroda dua dinyalakan maka kecepatan dan sudut merupakan *setpoint*.
2. Kontroler yang digunakan adalah kontroler Proporsional Integral Derivatif (PID) menggunakan perangkat keras Arduino Due.
3. *Plant* yang digunakan adalah robot beroda dua.
4. *Driver* motor menggunakan L298N *Dual H-Bridge Motor Driver board*.
5. Aktuator yang digunakan adalah motor DC Kinmore (KM-25A370-17.6-06397) dengan 7,4 Volt.
6. Sensor *optical rotary encoder* sebagai pembaca putaran motor digunakan untuk *feedback* sistem.
7. Sensor IMU sebagai pembaca sudut digunakan untuk *feedback* sistem.
8. Gangguan pada robot berupa perubahan pergerakan yang cepat pada robot beroda dua dengan memberi dorongan kecil pada *body* depan dan belakang robot beroda dua.

3.7 Perancangan dan Penyusunan Perangkat Lunak

Flowchart/algoritma program dalam perancangan dan penyusunan perangkat lunak merupakan gambaran alur proses program yang dilakukan oleh kontroler pada saat implementasi pada sistem robot beroda dua. *Flowchart* program dalam skripsi ini dapat dilihat dalam Gambar 3.18.



Gambar 3.18 Flowchart Program Keseluruhan

Flowchart program pada Gambar 3.18 menunjukkan urutan kerja dari perangkat lunak yang dirancang. Yang pertama dilakukan adalah inisialisasi parameter konfigurasi PID dan Kalman filter. Setelah mendapat nilai pembacaan dari sensor IMU (sensor *accelerometer* dan sensor *gyroscope*) akan dibaca menggunakan metode Kalman filter untuk mendapat kemiringan robot yang mana nilai tersebut akan di proses dalam kontrol PID dan mengeluarkan *output* berupa PWM untuk menggerakkan motor dengan kecepatan dan arah putaran yang sudah ditentukan.

Flowchart program pada Gambar 3.18 yang kedua merupakan urutan kerja sistem keseimbangan robot beroda dua yang dibuat. Yang pertama dilakukan adalah menentukan keseimbangan dari robot. Selanjutnya dari yang sudah dirancang maka dengan otomatis program akan memilih kecepatan dan arah putar motor DC.

BAB IV

HASIL DAN PEMBAHASAN

Hasil dan pembahasan dilakukan dengan melakukan pengujian sistem. Pengujian sistem dilakukan untuk mengetahui apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan skema pembuatan. Pengujian pada sistem meliputi pengujian setiap blok maupun pengujian sistem secara keseluruhan. Pengujian setiap blok dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan skema pembuatan. Pengujian pada sistem ini dibagi menjadi tiga bagian, yaitu:

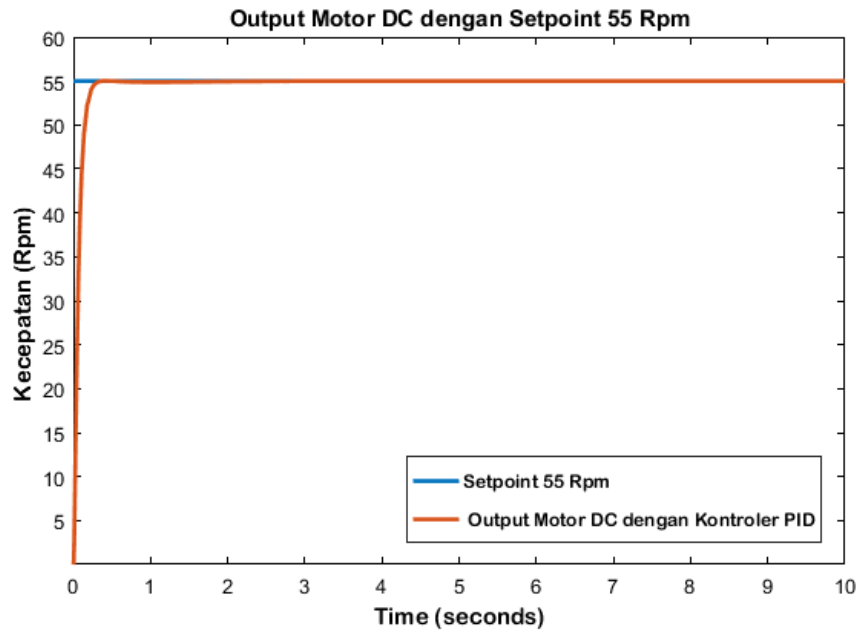
1. Pengujian sinyal kendali PID.
2. Pengujian Kalman filter.
3. Pengujian keseluruhan sistem.

4.1 Pengujian Pengendali PID Kaskade

Pengujian ini dilakukan untuk mendapatkan nilai yang tepat dari masing-masing konstanta parameter PID dan untuk mengetahui *output* pergerakan robot beroda dua sehingga bisa diketahui kesalahan ketika robot beroda dua tidak dapat seimbang. Pengujian pengendali PID kaskade dibagi menjadi beberapa bagian yaitu :

4.1.1 Pengujian Pengendali Kecepatan dengan Kontroler PID

Pengujian ini dilakukan untuk mengetahui apakah nilai parameter yang didapatkan sudah sesuai dengan setpoint. Pada pengujian ini setpoint yang diberikan sebesar 55 rpm (rotasi per menit). Hasil dari pengujian sistem ditunjukkan dalam Gambar 4.1



Gambar 4.1 Output Motor DC 7,4 V Kinmore (KM-25A370-17.6-06397 dengan Setpoint 55 Rpm

Dari *output* yang ditunjukkan dalam Gambar 4.1 dapat disimpulkan bahwa sistem tidak terdapat *error steady state*, tidak ada *overshoot*, dan *settling time* sebesar 0.2 detik

4.1.2 Pengujian Pengendali Sudut dengan Kontroler PID

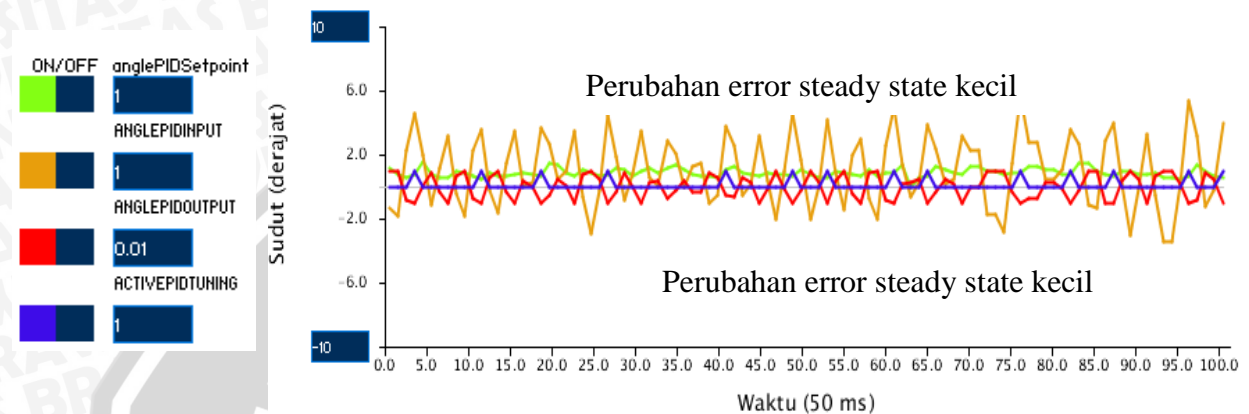
Pengujian ini dilakukan untuk mencari nilai optimal (*tuning manual*) parameter pengendali sudut dengan kontroler PID terhadap *output* sistem kesetimbangan robot beroda dua. Mula-mula robot beroda dua diletakkan dengan posisi berdiri tegak lurus terhadap permukaan bidang datar (0^0) kemudian dilepaskan untuk melihat kemampuan robot beroda dua dalam menjaga keseimbangan terhadap parameter pengendali sudut dengan kontroler PID yang sudah ditentukan. Pengujian juga dilakukan dengan memberikan gangguan kepada robot beroda dua. Data hasil pengujian pengendali sudut dengan kontroler PID digambarkan dalam bentuk grafik terdapat pada Gambar 4.2, Gambar 4.3, dan Gambar 4.4.

a. Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 1:

Memberi nilai parameter K_p konfigurasi pengaturan agresif ($anglePIDAggKp$) lebih besar dari nilai parameter K_p konfigurasi pengaturan konservatif ($anglePIDConKp$) selanjutnya nilai parameter K_i konfigurasi pengaturan agresif ($anglePIDAggKi$) diatur lebih kecil dari nilai parameter K_i konfigurasi pengaturan konservatif ($anglePIDConKi$) dan nilai parameter K_d

konfigurasi pengaturan agresif (anglePIDAggKd) diatur lebih kecil dari nilai parameter K_d konfigurasi pengaturan konservatif (anglePIDConKd).

$\text{anglePIDAggKp} = 50$, $\text{anglePIDAggKi} = 2$, $\text{anglePIDAggKd} = 0.8$, $\text{anglePIDConKp} = 20$, $\text{anglePIDConKi} = 3$, dan $\text{anglePIDConKd} = 1.2$



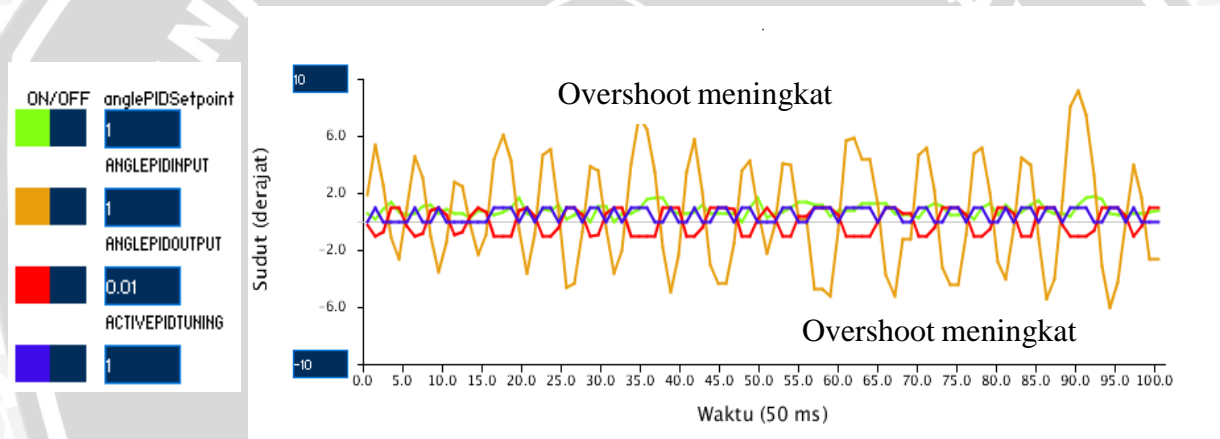
Gambar 4.2 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 1 pada Sistem Kesetimbangan Robot Beroda Dua

Dari *output* yang ditunjukkan dalam Gambar 4.2 memperlihatkan robot beroda dua ketika diberi parameter pengendali sudut dengan kontroler PID konfigurasi 1. Dari Gambar 4.2 dapat dilihat bahwa *overshoot* masih ada dan *error steady state* lebih besar jika dibanding dengan Gambar 4.4 (parameter pengendali sudut dengan kontroler PID konfigurasi 3) dan *overshoot* yang terjadi kecil dan dan *error steady state* lebih kecil jika dibanding dengan Gambar 4.3 (parameter pengendali sudut dengan kontroler PID konfigurasi 2). Dengan menggunakan parameter pengendali sudut dengan kontroler PID konfigurasi 1 ini pada awalnya robot beroda dua mampu berdiri dan kemudian bergerak maju terus setelah itu saat kembali ke posisi setimbang robot beroda dua terjatuh sehingga pergerakan robot dalam mencapai kesetimbangan masih belum halus. Oleh karena itu, dapat dikatakan bahwa kesetimbangan robot yang stabil masih belum dicapai dengan menggunakan parameter pengendali sudut dengan kontroler PID konfigurasi 1.

b. Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 2:

Memberi nilai parameter Kp konfigurasi pengaturan agresif ($anglePIDAggKp$) lebih besar dari nilai parameter Kp konfigurasi pengaturan konservatif ($anglePIDConKp$) selanjutnya nilai parameter Ki konfigurasi pengaturan agresif ($anglePIDAggKi$) diatur lebih besar dari nilai parameter Ki konfigurasi pengaturan konservatif ($anglePIDConKi$) dan nilai parameter Kd konfigurasi pengaturan agresif ($anglePIDAggKd$) diatur lebih kecil dari nilai parameter Kd konfigurasi pengaturan konservatif ($anglePIDConKd$).

$$anglePIDAggKp = 50, anglePIDAggKi = 0.6, anglePIDAggKd = 0.1, anglePIDConKp = 40, anglePIDConKi = 0.2, dan anglePIDConKd = 0.2.$$



Gambar 4.3 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 2 pada Sistem Kesetimbangan Robot Beroda Dua

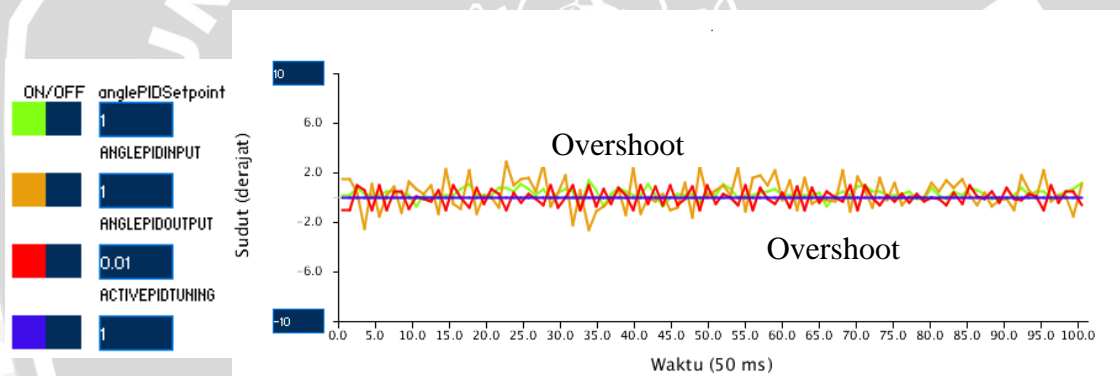
Dari *output* yang ditunjukkan dalam Gambar 4.3 memperlihatkan robot beroda dua ketika diberi parameter pengendali sudut dengan kontroler PID konfigurasi 2. Dari Gambar 4.3 dapat dilihat bahwa *overshoot* yang terjadi meningkat jika dibandingkan dengan Gambar 4.2 (parameter pengendali sudut dengan kontroler PID konfigurasi 1) dan Gambar 4.4 (parameter pengendali sudut dengan kontroler PID konfigurasi 3). Dengan menggunakan parameter pengendali sudut dengan kontroler PID konfigurasi 2 ini pada awalnya robot beroda dua mampu berdiri dan kemudian bergerak mundur terus hingga akhirnya jatuh sehingga robot beroda dua dalam mencapai kesetimbangan tidak mampu mempertahankan kesetimbangannya. Oleh karena itu, dapat dikatakan bahwa kesetimbangan robot yang stabil

tidak dapat dicapai dengan menggunakan parameter pengendali sudut dengan kontroler PID konfigurasi 2.

c. Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 3:

Memberi nilai parameter K_p konfigurasi pengaturan agresif ($anglePIDAggKp$) lebih besar nilai parameter K_p konfigurasi pengaturan konservatif ($anglePIDConKp$) selanjutnya nilai parameter K_i konfigurasi pengaturan agresif ($anglePIDAggKi$) diatur sama dengan nilai parameter K_i konfigurasi pengaturan konservatif ($anglePIDConKi$) dan nilai parameter K_d konfigurasi pengaturan agresif ($anglePIDAggKd$) diatur sama dengan nilai parameter K_d konfigurasi pengaturan konservatif ($anglePIDConKd$).

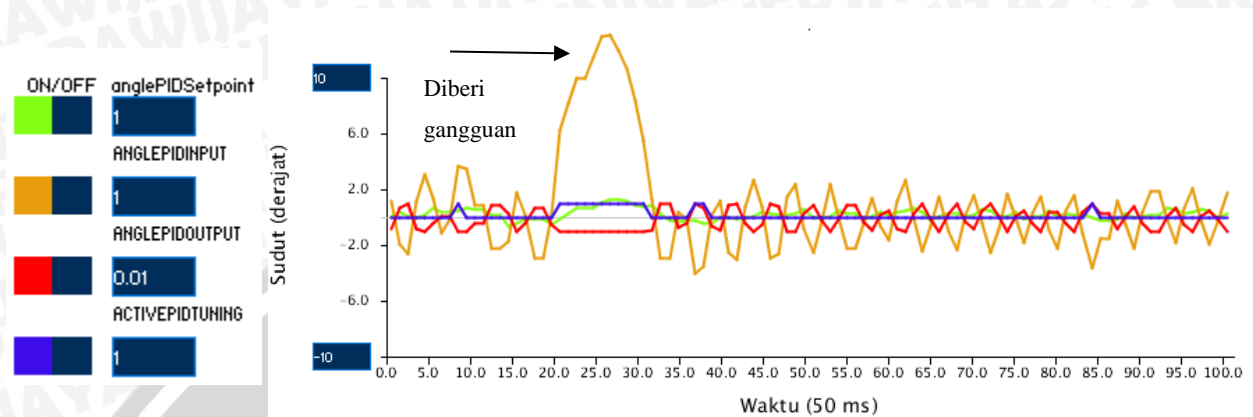
$anglePIDAggKp = 50$, $anglePIDAggKi = 0.5$, $anglePIDAggKd = 0.25$, $anglePIDConKp = 20$, $anglePIDConKi = 0.5$, dan $anglePIDConKd = 0.25$.



Gambar 4.4 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 3 pada Sistem Keseimbangan Robot Beroda Dua

Dari *output* yang ditunjukkan dalam Gambar 4.4 memperlihatkan robot beroda dua ketika diberi parameter pengendali sudut dengan kontroler PID konfigurasi 3. Dari Gambar 4.4 dapat dilihat bahwa *overshoot* yang terjadi kecil dan *error* mendekati nol jika dibandingkan dengan Gambar 4.2 (parameter pengendali sudut dengan kontroler PID konfigurasi 1) dan Gambar 4.3 (parameter pengendali sudut dengan kontroler PID konfigurasi 2). sehingga didapatkan nilai $anglePIDAggKp$, $anglePIDAggKi$, $anglePIDAggKd$, $anglePIDConKp$, $anglePIDConKi$, dan $anglePIDConKd$ dengan nilai paling ideal. Hasil parameter pengendali sudut dengan kontroler PID konfigurasi 3 ini nilai *overshoot* lebih kecil dan pergerakan robot dalam mencapai kesetimbangan lebih halus. Oleh karena itu, dapat dikatakan bahwa

kesetimbangan robot yang stabil dapat dicapai dengan menggunakan parameter pengendali sudut dengan kontroler PID konfigurasi 3.

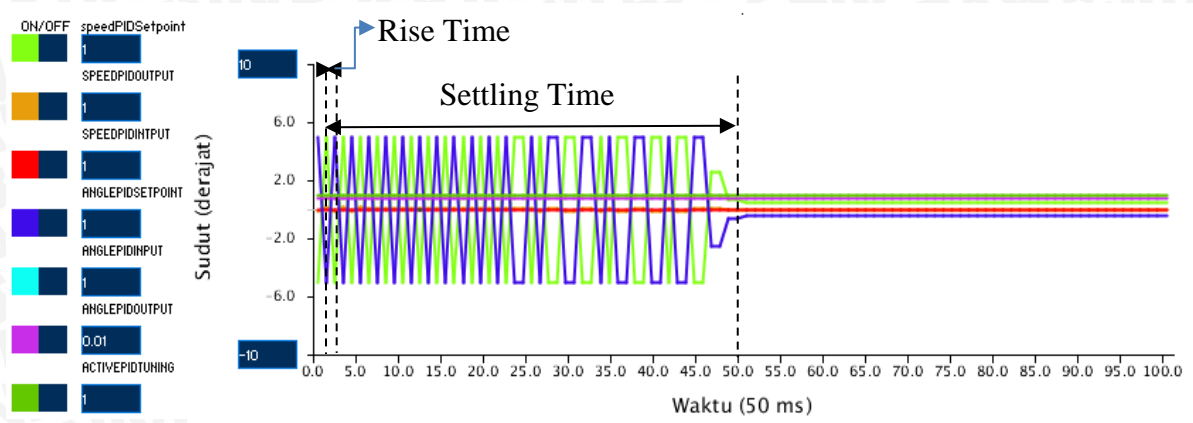


Gambar 4.5 Pengaruh Parameter Pengendali Sudut dengan Kontroler PID Konfigurasi 3 pada Sistem Keseimbangan Robot Beroda Dua Ketika Diberi Gangguan

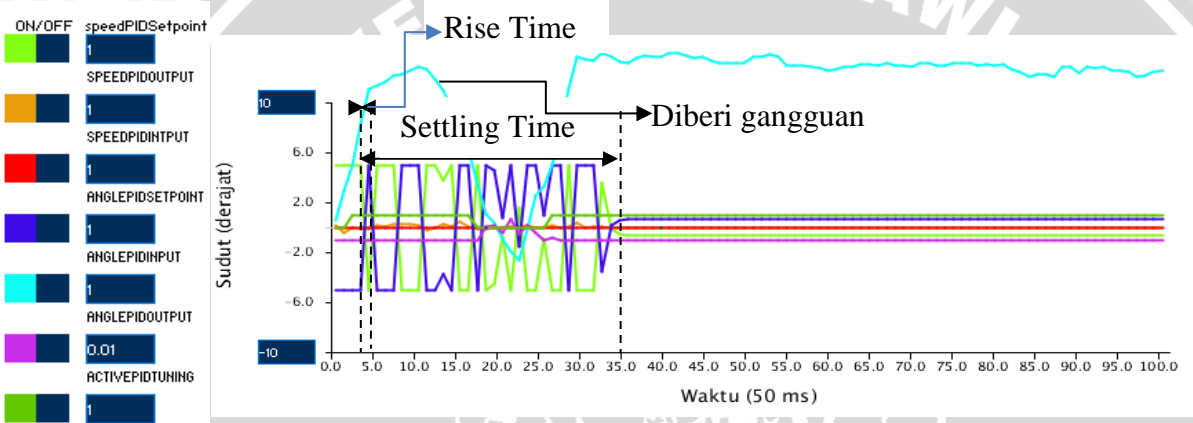
Dari *output* yang ditunjukkan dalam Gambar 4.5 memperlihatkan *output* dari robot beroda dua ketika diberi parameter pengendali sudut dengan kontroler PID konfigurasi 3 dan diberi gangguan. Pengujian ini dilakukan dengan memberikan gangguan ketika nilai parameter pengendali sudut dengan kontroler PID mencapai ideal. Gambar 4.5 dapat dilihat bahwa ketika diberikan gangguan yaitu pada detik ke-20 maka robot beresilasi pada detik ke-20 hingga detik ke-30. Namun setelah itu robot kembali mencapai kesetimbangan.

4.1.3 Pengujian Pengendali Kecepatan dan Pengendali Sudut dengan Kontroler PID pada Sistem Keseimbangan Robot Beroda Dua

Pengujian dilakukan untuk mengetahui seberapa andal robot beroda dua dalam melakukan proses kesetimbangan. Mula-mula robot beroda dua diberdirikan posisi tegak lurus terhadap permukaan bidang datar (0^0) kemudian dilepaskan untuk melihat kemampuan robot beroda dua dalam menjaga keseimbangan terhadap parameter konfigurasi kecepatan dan parameter konfigurasi sudut dengan kontroler PID yang optimal. Data hasil pengujian keseluruhan sistem kesetimbangan robot beroda dua digambarkan dalam bentuk grafik pada Gambar 4.6 dan Gambar 4.7.



Gambar 4.6 Sistem Kesetimbangan Robot Beroda Dua tanpa Gangguan



Gambar 4.7 Sistem Kesetimbangan Robot Beroda Dua dengan Gangguan

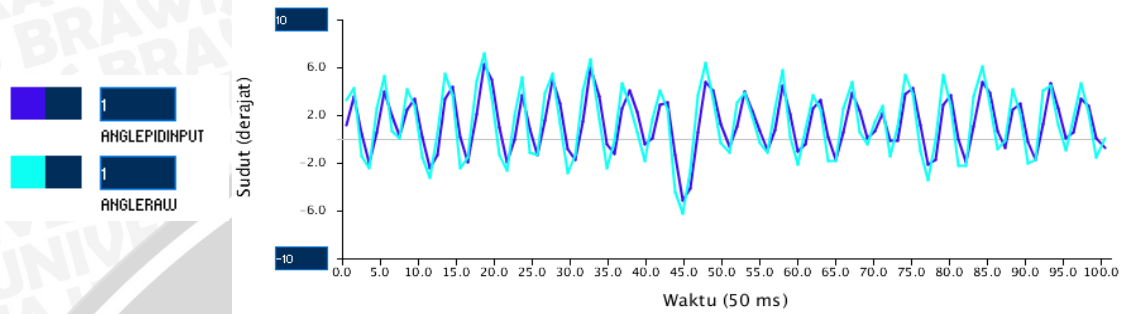
Dari Gambar 4.6 dan Gambar 4.7 secara keseluruhan dari pengujian ini dapat dilihat dapat disimpulkan bahwa sistem kesetimbangan robot beroda dua tanpa gangguan didapatkan settling time sebesar 2.5 detik dan sistem kesetimbangan robot beroda dua ketika diberi gangguan didapatkan settling time sebesar 1.75 detik.

4.2 Pengujian Kalman Filter

Seperti yang sudah dijelaskan, tujuan utama dari Kalman filter adalah menghilangkan *noise* pengukuran pada sensor *accelerometer* serta mengurangi *drift* dari sensor *gyroscope*. Kalman filter memberikan prediksi sudut yang lebih baik dari data asli dari sensor *gyroscope* dan sensor *accelerometer*. Pengujian ini dilakukan untuk mengetahui hasil *output* sensor mendekati dengan hasil *output* sensor yang sudah diberikan metode Kalman filter.

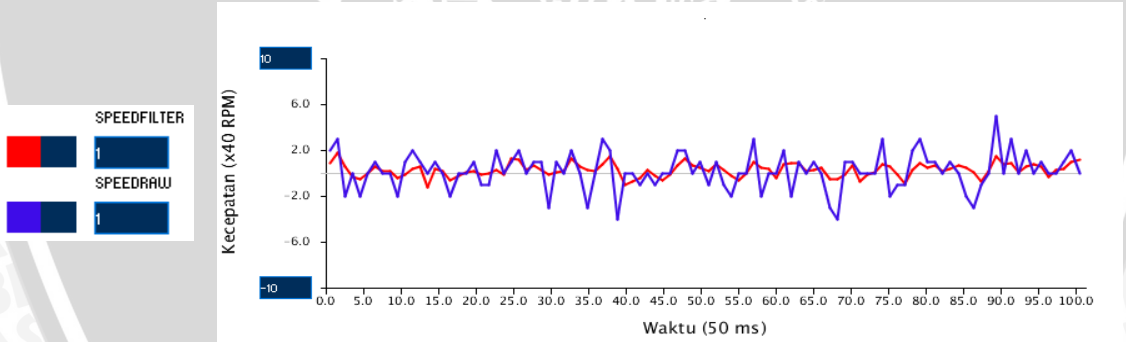
Pengujian Kalman filter pada sistem kesetimbangan robot beroda dua. Mula-mula robot beroda dua diberdirikan posisi tegak lurus terhadap permukaan bidang datar (0^0) kemudian

dilepaskan untuk melihat kemampuan robot beroda dua dalam menjaga keseimbangan hingga *collapse* atau terjatuh. Data hasil pengujian Kalman filter digambarkan dalam bentuk grafik.

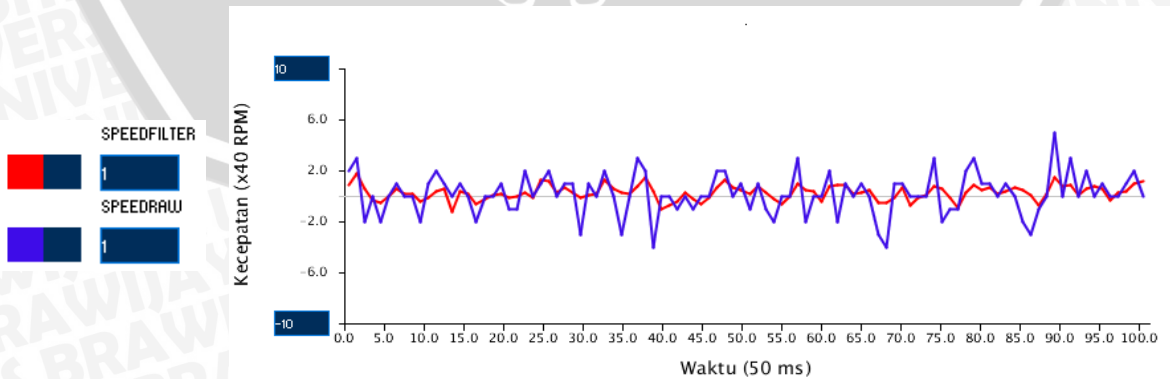


Gambar 4.8 Data Hasil Pengujian Kalman Filter pada Sensor IMU

Dari Gambar 4.8 terlihat bahwa pengujian Kalman filter pada sensor IMU (sensor *gyroscope* + sensor *accelerometer*) sudah akurat, terlihat dari keluaran sensor IMU (*angleraw*) berupa besaran sudut mendekati dengan hasil keluaran sensor IMU berupa besaran sudut yang sudah di Kalman filter (*anglepidinput*). Dapat disimpulkan bahwa pengujian Kalman filter pada sensor IMU sudah bekerja dengan baik.



Gambar 4.9 Data Hasil Pengujian Kalman Filter pada Sensor *Optical Rotary Encoder* Kiri



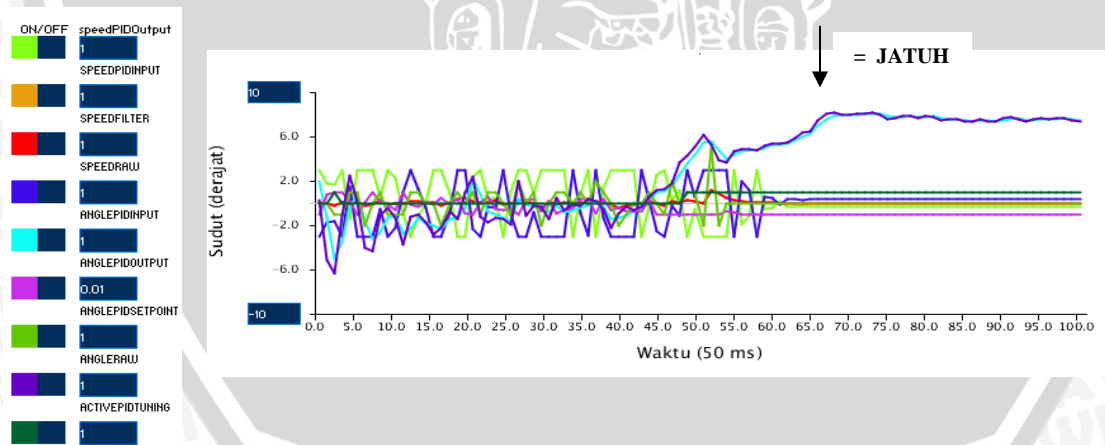
Gambar 4.10 Data Hasil Pengujian Kalman Filter pada Sensor *Optical Rotary Encoder* Kanan

Dari Gambar 4.9 dan Gambar 4.10 terlihat bahwa pengujian Kalman filter pada sensor *optical rotary encoder* kiri dan sensor *optical rotary encoder* kanan sudah akurat, terlihat dari keluaran sensor *optical rotary encoder (speedraw)* berupa kecepatan motor DC mendekati dengan hasil keluaran sensor *optical rotary encoder* berupa kecepatan motor DC Kalman filter (*speedfilter*). Dapat disimpulkan bahwa pengujian Kalman filter pada sensor *optical rotary encoder* sudah bekerja dengan baik.

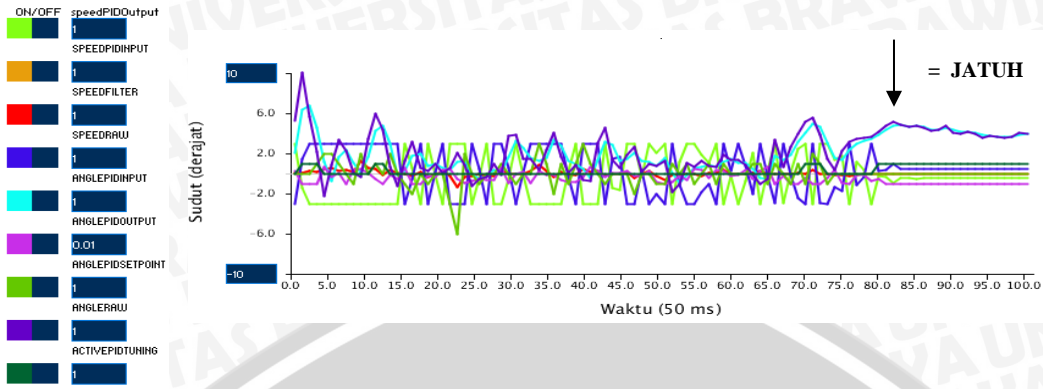
4.3 Pengujian Keseluruhan Sistem

Pengujian ini dilakukan untuk mengetahui kerja dari perangkat keras (*hardware*) dan perangkat lunak (*software*) setelah diintegrasikan dalam sebuah sistem yaitu sistem kesetimbangan robot beroda dua.

Pengujian keseluruhan sistem kesetimbangan robot beroda dua. Mula-mula robot beroda dua diberdirikan posisi tegak lurus terhadap permukaan bidang datar (0^0) kemudian dilepaskan untuk melihat kemampuan robot beroda dua dalam menjaga keseimbangan hingga *collapse* atau terjatuh. Parameter yang diamati pada skripsi ini adalah lama waktu sistem dapat bertahan pada keadaan setimbang sebelum *collapse* atau terjatuh. Data hasil pengujian keseluruhan sistem kesetimbangan robot beroda dua digambarkan dalam bentuk grafik pada Gambar 4.11 dan Gambar 4.12.



Gambar 4.11 Data Hasil Pengujian Keseluruhan Sistem Kesetimbangan Robot Beroda Dua hanya Menggunakan PID Kaskade Loop Dalam



Gambar 4.12 Data Hasil Pengujian Keseluruhan Sistem Keseimbangan Robot Beroda Dua Menggunakan PID Kaskade

Dari Gambar 4.11 dan Gambar 4.12 secara keseluruhan dari pengujian ini dapat dilihat bahwa sistem keseimbangan robot beroda dua dapat bertahan lebih lama pada keadaan setimbang sebelum *collapse* atau terjatuh dengan menggunakan PID kaskade daripada hanya menggunakan PID kaskade loop dalam. Dapat disimpulkan bahwa sistem robot beroda dua yang dirancang dan dibuat telah berjalan dengan baik. Proses pengontrolan PID kaskade dan Kalman filter , robot beroda dua berhasil setimbang dengan nilai Parameter Proporsional Integral Derivatif (PID) kaskade yang paling optimal adalah $speedPIDKp = 14$, $speedPIDKi = 15.2$, $speedPIDKd = 4$, $anglePIDAggKp = 50$, $anglePIDaggKi = 0.5$, $anglePIDKd = 0.25$, $anglePIDConKp = 20$, $anglePIDConKi = 0.5$, dan $anglePIDConKd = 0.25$. dan nilai parameter Kalman filter yang paling optimal $angleKalmanFilter = 10$ dan $speedKalmanFilter = 20$.

BAB V PENUTUP

5.1. Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada robot beroda dua maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan hasil identifikasi menggunakan sinyal uji PRBS dan sintaks *ident* pada *software* Matlab didapatkan fungsi alih motor DC adalah

$$\frac{Y(s)}{U(s)} = \frac{3.535}{s^2 + 3.308s + 3.765} .$$
 Dari fungsi alih tersebut didapatkan parameter

Kontroler PID menggunakan metode *trial and error* didapatkan nilai $speedK_p=14$, $speedK_i=15.2$ dan $speedK_d=4$.

2. Berdasarkan hasil pengujian didapatkan *output* plant robot beroda dua sebagai berikut :

- a. Setelah melakukan pengendali kecepatan dengan kontroler PID pada *setpoint* 55 rpm dan menganalisis pengujiannya ,tidak terdapat *error steady state*, tidak ada *overshoot*, dan *settling time* sebesar 0.2 detik.

- b. Setelah melakukan pengendali sudut dengan kontroler PID dan menganalisis pengujiannya tanpa gangguan maupun dengan gangguan dari luar berupa dorongan kecil pada body bagian depan dan belakang robot beroda dua, maka dapat disimpulkan kesetimbangan robot beroda dua tercapai dengan nilai parameter sudut dengan kontroler PID yang optimal adalah $anglePIDAggK_p = 50$, $anglePIDAggK_i = 0.5$, $anglePIDAggK_d = 0.25$, $anglePIDConK_p = 20$, $anglePIDConK_i = 0.5$, dan $anglePIDConK_d = 0.25$. Semakin kecil simpangan robot artinya semakin stabil robot.

- c. Setelah melakukan pengendali kecepatan dan pengendali sudut dengan kontroler PID pada sistem kesetimbangan beroda dua dan menganalisis pengujiannya sistem tanpa gangguan didapatkan *settling time* sebesar 2.5 detik dan ketika sistem diberi gangguan berupa dorongan kecil maju atau mundur didapatkan *settling time* sebesar 1.75 detik.

- d. Setelah melakukan pengujian kalman filter dan menganalisis pengujiannya, maka dapat disimpulkan kesetimbangan robot beroda dua

tercapai dengan nilai parameter kalman filter yang paling optimal adalah $angleKalmanFilter = 10$ dan $speedKalmanFilter = 20$. Terlihat keluaran sensor IMU (*angleraw*) berupa besaran sudut mendekati dengan hasil keluaran sensor IMU. Begitu juga dengan keluaran sensor *optical rotary encoder* (*speedraw*) berupa kecepatan motor DC mendekati dengan hasil keluaran sensor *optical rotary encoder* berupa kecepatan motor DC kalman filter (*speedfilter*). Semakin kecil nilai koefisien pada Kalman filter *noise* yang dihasilkan artinya semakin tidak stabil robot.

5.2. Saran

Didalam perancangan dan pembuatan alat yang dilakukan oleh penelitian ini masih banyak terdapat kelemahan dan kekurangan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut, Saran yang dapat diberikan untuk penelitian selanjutnya sebagai berikut :

1. Dilakukan metode pengontrolan selain menggunakan kontroler PID kaskade dan metode Kalman filter agar robot beroda dua dapat seimbang dengan baik tanpa terjatuh.
2. Dilakukan uji kestabilan sistem menggunakan metode kriteria *Routh-Hurwitz* sehingga dapat lebih mudah menemukan ketidakstabilan sistem kesetimbangan pada robot beroda dua dan memastikan kestabilan sistem kesetimbangan robot beroda dua.
3. Pengembangan balancing robot beroda dua dapat digunakan untuk aplikasi *line follower* (penjejak garis) atau *wall follower* (penjejak dinding).
4. Dalam penelitian ini robot beroda dua masih belum bisa seimbang dengan baik dan masih terjatuh. Diperlukan penyempurnaan konstruksi mekanik pada badan robot beroda dua agar dapat lebih mudah menentukan titik setimbang dan robot beroda dua dapat seimbang tanpa terjatuh.
5. Metode *tuning* konstanta kendali PID dilakukan dengan metode *auto tuning* agar hasil yang diperoleh lebih optimal.
6. Robot kesetimbangan beroda dua ini dapat dikembangkan sehingga dapat melakukan *tracking* (berjalan) otomatis dengan *image processing* (pengolahan gambar atau pemrosesan dengan input berupa gambar).

DAFTAR PUSTAKA

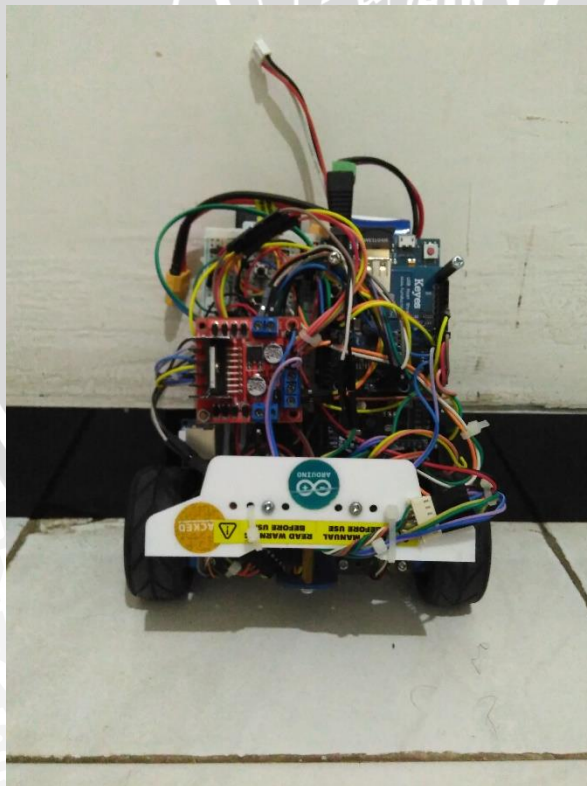
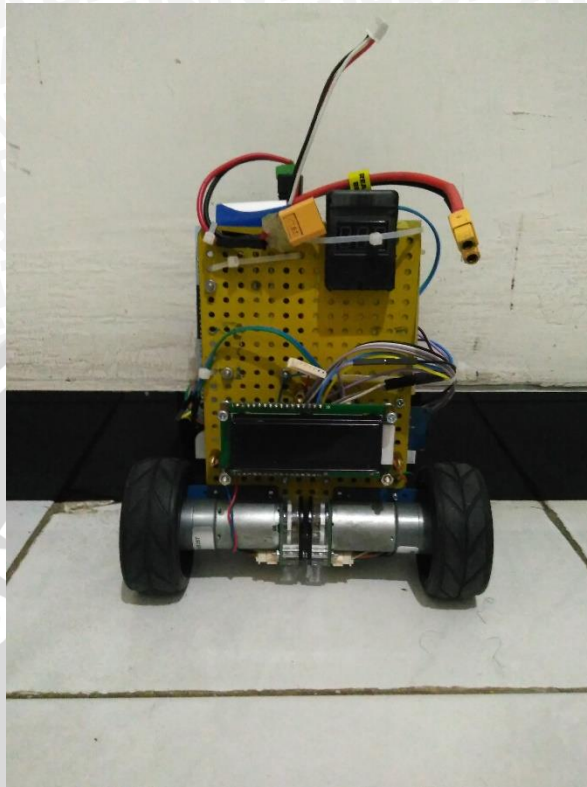
- Miller, P., 2008, *Building a Two Wheeled Balancing Robot*, Disertasi, Faculty of Engineering and Surveying, University of Southern Queensland, Queensland.
- Stang, J., 2005, *The Inverted Pendulum*, Tesis, Teknik, Cornell University, Cornell.
- Krakow, K.I., 2005, *System-specific PI Control Theory for Fluid and Motion Systems*, Universal Publishers, United States..
- Colton, S., 2007. *The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing*, <http://web.mit.edu/scolton/www/filter.pdf>, diakses pada tanggal 1 Desember 2015.
- Wicaksono, E.G., 2012, *Balancing Robot Beroda Dua Menggunakan Metode Kendali Proporsional Derivatif*, Skripsi, Teknik, Universitas Diponegoro, Semarang.
- Ogata, K. 1997. *Teknik Kontrol Automatik – terjemahan: Ir. Edi Laksono*. Erlangga: Jakarta.
- Dorf, R. C., dan H. Bishop, R., 2010, *Modern Control System Twelfth Edition*, Prentice Hall International, United Kingdom.
- K. J. Astrom & T. Haggund, 1995: *PID Controllers: Theory, Design, and Tuning*.
- An, Wei, and Yangmin Li. "Simulation and Control of a Two-wheeled Self-balancing Robot." Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on. IEEE, 2013.
- Wen, Yan-Hou, Yu-Sheng Lin, and Yih-Guang Leu. "Design and implementation of the balance of two-wheeled robots." Advanced Robotics and Intelligent Systems (ARIS), 2013 International Conference on. IEEE, 2013.

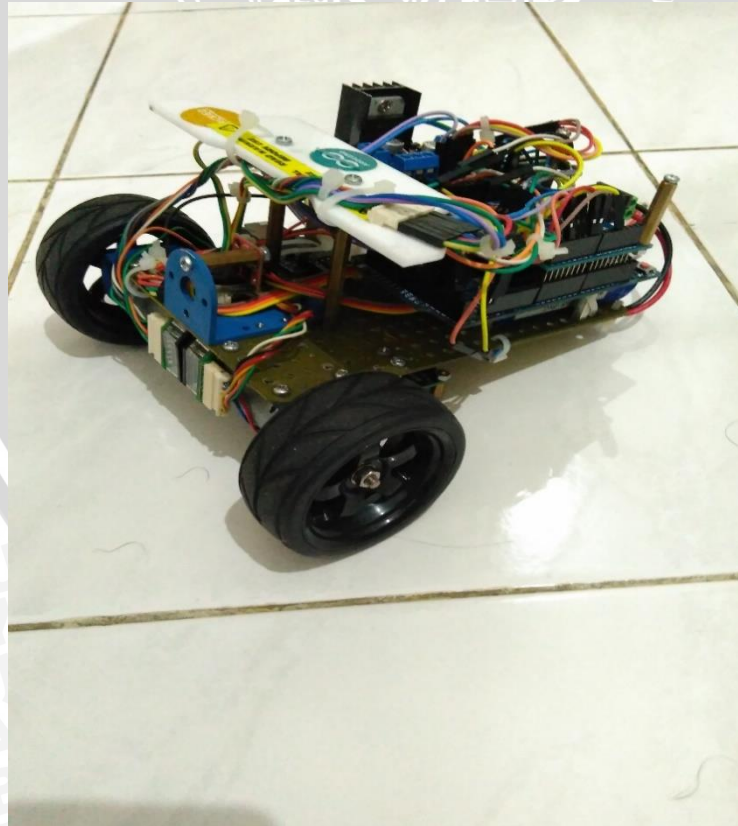
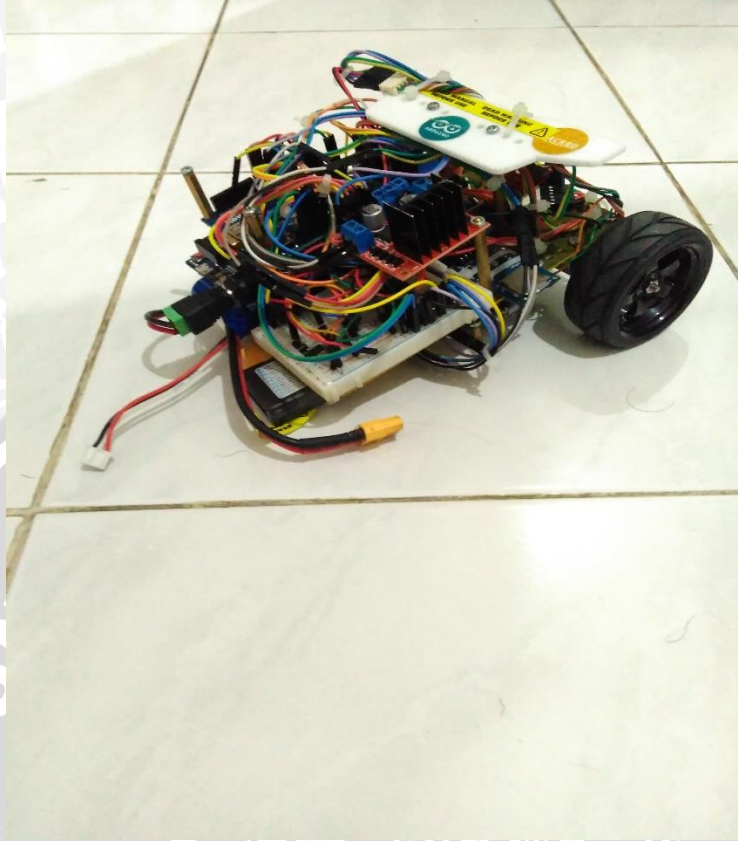
LAMPIRAN I

FOTO ALAT



FOTO ALAT





LAMPIRAN II

LISTING PROGRAM



PROGRAM UTAMA

```
#include <Wire.h>

#include <stdlib.h>

#include <L29x.h>

// This optional setting causes Encoder to use more optimized code,
// It must be defined before Encoder.h is included.

#define ENCODER_OPTIMIZE_INTERRUPTS

#include <Encoder.h>

#include <PID_v1.h>

#include <Button.h>

#include <SerialCommand.h>

#include <MovingAverageFilter.h>

#include <FIR_v2.h>

#include <KalmanFilter.h>

#include <FreeRTOS_ARM.h>

#include <DueFlashStorage.h>

#include <SPI.h>

#include <SD.h>

#include <PS3BT.h>

#include <usbhub.h>

#include "steer.h"
```

```
#include "Motor_Claire.h"
```

```
DueFlashStorage dueFlashStorage;
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(44, 45, 32, 33, 34, 35);
```

```
/**/ PS3 controller
```

```
#ifdef dobogusinclude
```

```
#include <spi4teensy3.h>
```

```
#endif
```

```
/**/
```

```
USB Usb;
```

```
BTD Btd(&Usb); // You have to create the Bluetooth Dongle instance like so
```

```
/* You can create the instance of the class in two ways */
```

```
PS3BT PS3(&Btd); // This will just create the instance
```

```
// turning
```

```
#define TURN_RIGHT -1
```

```
#define TURN_NONE 0
```

```
#define TURN_LEFT 1
```

```
##define FORWARDS 2
```

```
//#define BACKWARDS 3
```

```
const int JOYSTICK_DEADBAND = 28;
```

```
int turningDirection = TURN_NONE;
```

```
float turningOffset = 0.0;
```

```
int lastDirection;
```

```
String PIDConfigType="";
```

```
// Redefine AVR Flash string macro as nop for ARM
```

```
#undef F
```

```
#define F(str) str
```

```
#define TO_RAD(x) (x * 0.01745329252) // *pi/180
```

```
#define TO_DEG(x) (x * 57.2957795131) // *180/pi
```

```
const uint8_t LED_PIN = 13;
```

```
// motor PWM remote
```

```
const int L_MOTOR_PWM = 5;
```



UNIVERSITAS BRAWIJAYA


```
const int R_MOTOR_PWM = 3;

// change direction of motors

const int L_MOTOR_IN_A = 7;

const int L_MOTOR_IN_B = 6;

const int R_MOTOR_IN_A = 2;

const int R_MOTOR_IN_B = 11;

Motor_Claire motor_R(R_MOTOR_PWM, R_MOTOR_IN_A, R_MOTOR_IN_B);

Motor_Claire motor_L(L_MOTOR_PWM, L_MOTOR_IN_A, L_MOTOR_IN_B);

// motor config

L29x motorLeft(5, 7, 6); // enable (PWM), motor pin 1, motor pin 2

L29x motorRight(3, 2, 11); // enable (PWM), motor pin 1, motor pin 2

/* -----
 * ----- ENCODERS -----
 * ----- */

//// encoder Arduino ports

const int L_ENCODER_A = 24;
```

67

```
const int L_ENCODER_B = 25;
```

```
const int R_ENCODER_A = 52;
```

```
const int R_ENCODER_B = 53;
```

```
// Change these pin numbers to the pins connected to your encoder.
```

```
// Best Performance: both pins have interrupt capability
```

```
// Good Performance: only the first pin has interrupt capability
```

```
// Low Performance: neither pin has interrupt capability
```

```
Encoder MotorLeft(L_ENCODER_A, L_ENCODER_B);
```

```
Encoder MotorRight(R_ENCODER_A, R_ENCODER_B);
```

```
// button declarations
```

```
Button startBtn(30, false, false, 20);
```

```
Button stopBtn(31, false, false, 20);
```

```
Button calibrateBtn(42, false, false, 20);
```

```
// start and stop
```

```
boolean started = true;
```

```
SerialCommand sCmd; // The demo SerialCommand object
```

```
// PID variables

double anglePIDSetpoint, anglePIDInput, anglePIDOutput;

double speedPIDInput, speedPIDOutput, speedPIDSetpoint;

double anglePIDInputFiltered;

// filters

MovingAverageFilter speedMovingAverageFilter(14);//fix

MovingAverageFilter angleMovingAverageFilter(4);

//MovingAverageFilter speedMovingAverageFilter(40);//fix1

//MovingAverageFilter angleMovingAverageFilter(40);

KalmanFilter speedKalmanFilter;

KalmanFilter angleKalmanFilter;

FIR_v2 speedMovingAverageFilter2;

// The cascading PIDs. The tunings are updated from the code

PID anglePID(&anglePIDInput, &anglePIDOutput, &anglePIDSetpoint, 0, 0, 0, DIRECT);

PID speedPID(&speedPIDInput, &speedPIDOutput, &speedPIDSetpoint, 0, 0, 0, DIRECT);

float motorPWMCommand_R, motorPWMCommand_L, motorPWMCommand_RL;
```

69

```
float motorPWMCommand_Right, motorPWMCommand_Left=0;
```

```
// Sensor variables
```

```
float accel[3]; // Actually stores the NEGATED acceleration (equals gravity, if board not moving).
```

```
float accel_min[3];
```

```
float accel_max[3];
```

```
float magnetom[3];
```

```
float magnetom_min[3];
```

```
float magnetom_max[3];
```

```
float magnetom_tmp[3];
```

```
float gyro[3];
```

```
float gyro_average[3];
```

```
int gyro_num_samples = 0;
```

```
//v_read_imu_task
```

```
// Euler angles
```

```
float yaw;
```

```
float pitch;
```

```
float roll;
```



```
float angleRaw;  
float roll_filtered;  
  
// DCM timing in the main loop  
unsigned long timestamp;  
unsigned long timestamp_old;  
  
float G_Dt; // Integration time for DCM algorithm  
  
long leftMotorPosition = 0;  
long rightMotorPosition = 0;  
long lastLeftMotorPosition = 0;  
long lastRightMotorPosition = 0;  
float leftMotorSpeed = 0;  
float rightMotorSpeed = 0;  
float speedKalmanFiltered = 0;  
float speedFIRFiltered = 0;  
  
const int batteryPin = A7; // battery voltage is on pin A7  
const int chipSelect = 4; // SPI chip select for SD card  
const int cardDetect = 22; // pin that detects whether the card is there  
const int writeLed = 12; // LED indicator for writing to card
```

71

```
const int errorLed = 13;      // LED indicator for error

long lastWriteTime = 0;      // timestamp for last write attempt

long interval = 500;         // time between readings

char fileName[] = "adroit8.csv"; // filename to save on SD card

struct Configuration {

double speedPIDKp;

double speedPIDKi;

double speedPIDKd;

double speedPIDOutputLowerLimit;

double speedPIDOutputHigherLimit;

double speedPIDMoveFactor;

double anglePIDAggKp;

double anglePIDAggKi;

double anglePIDAggKd;

double anglePIDConKp;

double anglePIDConKi;

double anglePIDConKd;

double anglePIDLowerLimit;

double TriggerAngleAggressive;

double calibratedZeroAngle;
```



```
uint8_t anglePIDSampling;  
double angleKalmanFilterR;  
uint8_t speedPIDSampling;  
uint8_t angleSensorSampling;  
uint8_t motorSpeedSensorSampling;  
double speedKalmanFilterR;  
double steerGain;  
double Maxthrottle;  
int motorsON;  
uint8_t motorLeftMinimumSpeed;  
uint8_t motorRightMinimumSpeed;  
int8_t debugLevel;  
uint8_t debugSampleRate;  
uint8_t speedPIDOutputDebug;  
uint8_t speedPIDInputDebug;  
uint8_t speedPIDSetpointDebug;  
uint8_t speedKalmanFilterDebug;  
uint8_t speedRawDebug;  
uint8_t speedMovingAverageFilter2Debug;  
uint8_t anglePIDSetpointDebug;  
uint8_t anglePIDInputDebug;
```



73

```
uint8_t anglePIDOutputDebug;
```

```
uint8_t angleRawDebug;
```

```
uint8_t activePIDTuningDebug;
```

```
};
```

```
Configuration configuration;
```

```
byte b[sizeof(Configuration)];
```

```
double UserControl[1]; //Steer, Throttle
```

```
void setConfiguration() {
```

```
/* Flash is erased every time new code is uploaded. Write the default configuration to flash if first time */
```

```
// running for the first time?
```

```
uint8_t codeRunningForTheFirstTime = dueFlashStorage.read(0); // flash bytes will be 255 at first run
```

```
if (codeRunningForTheFirstTime) {
```

```
Serial.println("codeRunningForTheFirstTime");
```

```
/* OK first time running, set defaults */
```

```
*/
```

```
/*fix
```

```
configuration.speedPIDKp = 14.00;
```

```
configuration.speedPIDKi = 15.20;
```

```
configuration.speedPIDKd = 4.00;
```



```
configuration.speedPIDOutputLowerLimit = -10.00;
configuration.speedPIDOutputHigherLimit = 10.00;
configuration.anglePIDAggKp = 50.00;
configuration.anglePIDAggKi = 2.00;
configuration.anglePIDAggKd = 0.80;
configuration.anglePIDConKp = 20.00;
configuration.anglePIDConKi = 3.00;
configuration.anglePIDConKd = 1.20;
configuration.anglePIDLowerLimit = 5.00;
configuration.calibratedZeroAngle = -14.18;
configuration.anglePIDSampling = 10;
configuration.speedPIDSampling = 10;
configuration.angleKalmanFilterR = 10.00;
configuration.angleSensorSampling = 5;
configuration.motorSpeedSensorSampling = 5;
configuration.speedKalmanFilterR = 20.00;
configuration.TriggerAngleAggressive = 3.50;
configuration.steerGain = 1;
configuration.speedPIDMoveFactor = 0.7;
configuration.Maxthrottle = 6; //Max speed expressed in inclination degrees. Up to the
remote to provide the right scale.
```

75

```
configuration.motorLeftMinimumSpeed = 55;//40;
```

```
configuration.motorRightMinimumSpeed = 55;//40;
```

```
*/
```

```
/*fix1
```

```
configuration.speedPIDKp = 14.00;
```

```
configuration.speedPIDKi = 15.20;
```

```
configuration.speedPIDKd = 4.00;
```

```
configuration.speedPIDOutputLowerLimit = -10.00;
```

```
configuration.speedPIDOutputHigherLimit = 10.00;
```

```
configuration.anglePIDAggKp = 50.00;
```

```
configuration.anglePIDAggKi = 0.60;
```

```
configuration.anglePIDAggKd = 0.10;
```

```
configuration.anglePIDConKp = 40.00;
```

```
configuration.anglePIDConKi = 0.20;
```

```
configuration.anglePIDConKd = 0.20;
```

```
configuration.anglePIDLowerLimit = 5.00;
```

```
configuration.calibratedZeroAngle = -14.18;
```

```
configuration.anglePIDSampling = 5;
```

```
configuration.speedPIDSampling = 5;
```

```
configuration.angleSensorSampling = 10;
```

```
configuration.motorSpeedSensorSampling = 10;
configuration.angleKalmanFilterR = 10.00;
configuration.speedKalmanFilterR = 20.00;
configuration.TriggerAngleAggressive = 3.50;
configuration.speedPIDMoveFactor = 0.7;
configuration.Maxthrottle = 6; //Max speed expressed in inclination degrees. Up to the
remote to provide the right scale.
configuration.motorLeftMinimumSpeed = 55;//40;
configuration.motorRightMinimumSpeed = 55;//40;
*/
//*/Fix new
configuration.speedPIDKp = 14.00;
configuration.speedPIDKi = 15.20;
configuration.speedPIDKd = 4.00;
configuration.speedPIDOutputLowerLimit = -3.00;
configuration.speedPIDOutputHigherLimit = 3.00;
configuration.anglePIDAggKp = 50.00;
configuration.anglePIDAggKi = 0.50;
configuration.anglePIDAggKd = 0.25;
configuration.anglePIDConKp = 20.00;
configuration.anglePIDConKi = 0.50;
```

```
configuration.anglePIDConKd = 0.25;
configuration.anglePIDLowerLimit = 5.00;
configuration.calibratedZeroAngle = -14.18;
configuration.anglePIDSampling = 10;
configuration.speedPIDSampling = 5;
configuration.angleKalmanFilterR = 10.00;
configuration.angleSensorSampling = 15;
configuration.motorSpeedSensorSampling = 10;
configuration.speedKalmanFilterR = 20.00;
configuration.TriggerAngleAggressive = 3.50;
configuration.steerGain = 1;
configuration.speedPIDMoveFactor = 0.7;
configuration.Maxthrottle = 6; //Max speed expressed in inclination degrees. Up to the
remote to provide the right scale.
configuration.motorLeftMinimumSpeed = 55;//40;
configuration.motorRightMinimumSpeed = 55;//40;
///  

configuration.debugLevel = 1;
configuration.debugSampleRate = 50;
configuration.speedPIDSetpointDebug = 0;
configuration.speedPIDOutputDebug = 1;
```

```
configuration.speedPIDInputDebug = 1;

configuration.speedKalmanFilterDebug = 1;

configuration.speedRawDebug = 1;

configuration.speedMovingAverageFilter2Debug = 0;

configuration.anglePIDSetpointDebug = 1;

configuration.anglePIDInputDebug = 1;

configuration.anglePIDOutputDebug = 1;

configuration.angleRawDebug = 1;

configuration.activePIDTuningDebug = 1;

// write configuration struct to flash at address 4
memcpy(b, &configuration, sizeof(Configuration)); // copy the struct to the byte array
dueFlashStorage.write(4, b, sizeof(Configuration)); // write byte array to flash

// write 0 to address 0 to indicate that it is not the first time running anymore
dueFlashStorage.write(0, 0);

}

else {

Serial.println("read configuration struct from flash");

/* read configuration struct from flash */

byte* b2 = (dueFlashStorage.readAddress(4)); // byte array which is read from flash at
address 4
```

```
memcpy(&configuration, b2, sizeof(Configuration)); // copy byte array to temporary struct
```

```
}
```

```
}
```

```
void setup() {
```

```
Serial.begin(115200);
```

```
if (Usb.Init() == -1)
```

```
lcd.begin(16, 2); // start the library
```

```
while(!Serial) {
```

```
}
```

```
Wire.begin();
```

```
*/
```

```
/*
```

```
// initialize LED and cardDetect pins:
```

```
pinMode(writeLed, OUTPUT);
```

```
pinMode(errorLed, OUTPUT);
```

```
pinMode(cardDetect, INPUT_PULLUP);
```

```
// startSDCard() blocks everything until the card is present
```

```
// and writable:
```

UNIVERSITAS BRAWIJAYA



```
if (startSDCard() == true) {  
    Serial.println("card initialized.");  
  
    delay(100);  
  
    // open the log file:  
  
    File logFile = SD.open(fileName, FILE_WRITE);  
  
    // write header columns to file:  
  
    if (logFile) {  
        //logFile.println("TO_DEG(yaw),TO_DEG(pitch),angleRaw,TO_DEG(roll)");  
  
        //logFile.println("anglePIDSetpoint,anglePIDInput,anglePIDOutput");  
  
        logFile.println("accel[x],accel[y],accel[z],gyro[x],gyro[y],gyro[z]");  
  
        logFile.close();  
  
    }  
  
    }  
  
*/  
  
// filters  
  
speedKalmanFilter.setState(0);  
  
/**  
  
float coefficients[] = {  
1,1,1,1,1,1,1,1,1,1    };  
  
speedMovingAverageFilter2.setCoefficients(coefficients);
```

81

```
speedMovingAverageFilter2.setNumberOfTaps(9);

speedMovingAverageFilter2.setGain((float) (1/9));

/**/

setConfiguration();

// create control task

xTaskCreate(vControlTask, NULL, configMINIMAL_STACK_SIZE + 100, NULL, 1,
NULL)

// create button task

xTaskCreate(vInterfaceAndDebugTask, NULL, configMINIMAL_STACK_SIZE+100,
NULL, 1, NULL);

// start FreeRTOS

vTaskStartScheduler();

// should never return

Serial.println(F("Die"));

while(1);

// Serial.print(F("\r\nPS3 Bluetooth Library Started"));

}

// WARNING idle loop has a very small stack (configMINIMAL_STACK_SIZE)

// loop must never block

volatile uint32_t count = 0;
```



```
void loop() {  
  while(1) {  
    // must insure increment is atomic  
    // in case of context switch for print  
    noInterrupts();  
    count++;  
    interrupts();  
  }  
}  
  
void newConfig() {  
  // init speed PID  
  speedPIDSetpoint = 0;  
  speedPID.SetOutputLimits(configuration.speedPIDOutputLowerLimit,  
    configuration.speedPIDOutputHigherLimit);  
  //speedPID.SetOutputLimits(-300, 300); // -3 and +3 degrees, respectively  
  //anglePID.SetMode(AUTOMATIC);  
  speedPID.SetSampleTime(configuration.speedPIDSampling);  
  speedPID.SetTunings(configuration.speedPIDKp, configuration.speedPIDKi,  
    configuration.speedPIDKd);  
  
  //init angle PID  
  anglePIDSetpoint = 0;
```

```
anglePID.SetOutputLimits(-100, 100);  
  
//anglePID.SetMode(AUTOMATIC);  
  
anglePID.SetSampleTime(configuration.anglePIDSampling);  
  
anglePID.SetTunings(configuration.anglePIDConKp, configuration.anglePIDConKi,  
configuration.anglePIDConKd);  
  
angleKalmanFilter.setR(configuration.angleKalmanFilterR);  
  
speedKalmanFilter.setR(configuration.speedKalmanFilterR);  
  
// init motors  
  
motorLeft.setMinimumSpeed(configuration.motorLeftMinimumSpeed);  
motorRight.setMinimumSpeed(configuration.motorRightMinimumSpeed);  
  
// write configuration struct to flash at address 4  
  
//byte b2[sizeof(Configuration)]; // create byte array to store the struct  
  
memcpy(b, &configuration, sizeof(Configuration)); // copy the struct to the byte array  
  
dueFlashStorage.write(4, b, sizeof(Configuration)); // write byte array to flash  
  
}
```

LAMPIRAN III

DATASHEET





Dia 25mm DC Gear Motor Model No.:KM-25A370

- 1 Typical Applications:**
- Vending Machine
 - Garden Tool
 - Electric Lock
 - Actuator

Weight: 95g(approx.)

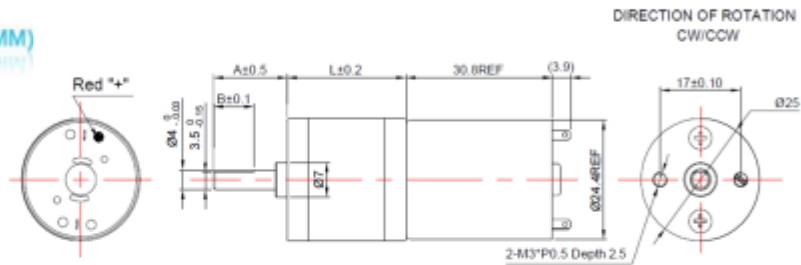
2 KEY FEATURES

- Reversible
- Metal or Carbon Brush DC Motor
- Spur Gears
- Zinc Die Cast Gear Case
- Sleeve Bronze Bushing
- RoHS Compliant

3 DIMENSIONS(MM)

Customization:

- a.Length A, B
- b.Technical Data
- c.Shaft Shape



Gear box length L (mm)	Number Of Stage	General Reduction Ratios						
		1/9.3	1/11	1/24	1/26	-	-	-
19	4	1/9.3	1/11	1/24	1/26	-	-	-
21	5	1/29	1/34	1/45	1/61	-	-	-
23	6	1/75	1/98	1/101	-	-	-	-
25	7	1/125	1/165	1/217	1/248	-	-	-
27	8	1/362	1/450	1/481	1/601	1/810	-	-

4 TECHNICAL DATA

Ratio	Model No.	Voltage		No Load		On Load		
		Operating Range	Nominal Voltage	Current	Speed	Current	Torque	Speed
		V	V	A	r/min	A	kg.cm	r/min
1/9.3	KM-25A370-9.3-09664	6.0-9.0	9.0	0.06	664	0.25	0.2	571
1/29	KM-25A370-29-03360	2.0-3.0	3.0	0.27	360	1.5	1.0	285
1/109	KM-25A370-109-0652	3.0-6.0	6.0	0.06	52	0.4	2.0	40
1/248	KM-25A370-248-0622	3.0-6.0	6.0	0.05	22	0.22	2.0	17
1/810	KM-25A370-810-07.203	3.0-9.0	7.2	0.01	2.8	0.03	2.0	2.6

Unit Conversion: 1g.cm≈0.098mN.m≈0.014oz.in 1mm=0.039in 1g=0.035oz

Remarks

1. All values approximate.
2. Customized electric performance on request by using suitable dc motor windings and ratio.
3. Available with required shaft length, D-shape size and length, grease type etc.



Optical Encoder Pair Kit for Micro Metal Gearmotors, 5V

Overview

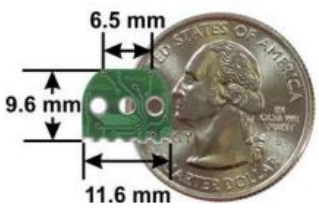
This set includes sensor boards and reflective wheels to add quadrature encoding to two micro metal gearmotors with extended back shafts. 3-tooth and 5-tooth encoder wheels are included to provide options of 12 counts per revolution and 20 counts per revolution of the motor shaft (to compute the encoder counts per revolution of the gearbox output, multiply by the gear ratio). Because the encoder board outputs are direct phototransistor outputs, some signal conditioning is often necessary between the sensor and a digital system processing the signals.

Note: This sensor system is intended for relatively advanced users comfortable with the physical encoder installation and with using the resulting signals. It only works with micro metal gearmotors that have extended back shafts.



Pinout and Installation

The encoder board is designed to be soldered directly to the back of the motor, with the back shaft of the motor protruding through the hole in the middle of the circuit board. The better aligned the board, the better the output signal quality will be. One way to achieve good alignment is to tack down the board to one motor pin and to solder the other pin only when the board is well aligned. Be careful to avoid prolonged heating of the motor pins, which could deform the plastic end cap of the motor or the motor brushes. Once the board is soldered down to the two terminals, the motor leads are connected to the M1 and M2 pads along the edge of the board, along with the power for the sensors and the two quadrature outputs:

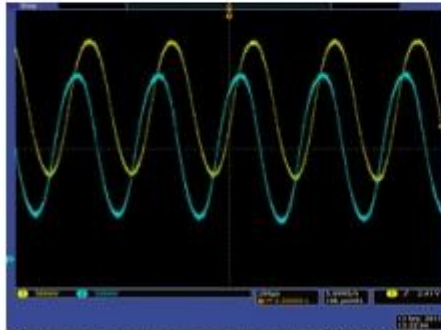


The edge connections are on a 2 mm pitch. The board can be soldered perpendicularly to another PCB, to a 2 mm connector, or to individual wires as show here:

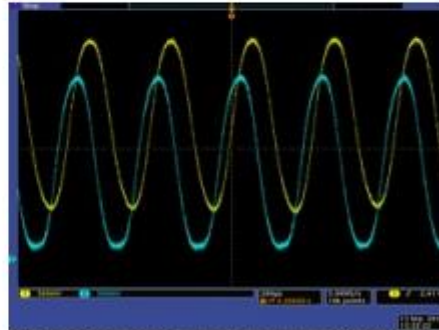


Example of an installed micro metal gearmotor reflective optical encoder.

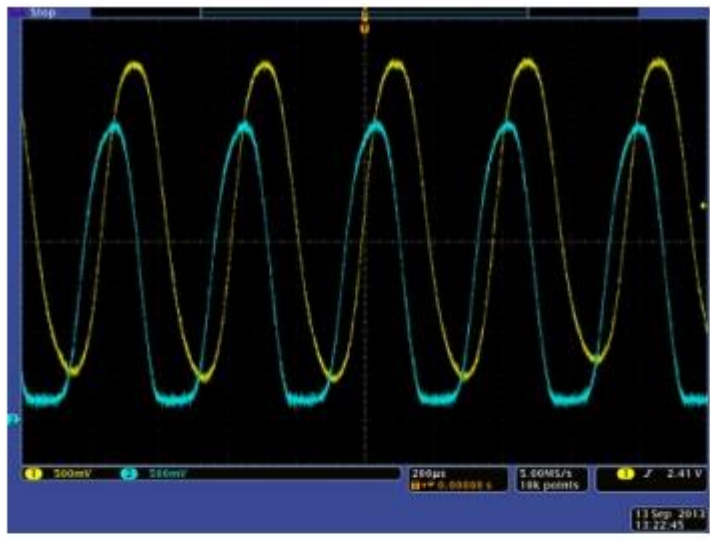
Once the board is soldered to the motor, the plastic encoder wheel can be pushed onto the motor shaft. The encoder wheel should be pushed on far enough that the gap between the wheel and sensors is approximately 0.5 mm. One way to get consistent gaps is to use a few sheets of paper or a business card as a shim when installing the encoder wheel. The best way to confirm optimal placement of the encoder wheel is to look at the signal with an oscilloscope. The following sequence of screen captures show the output for a 5-tooth wheel spinning at approximately 30,000 RPM as the wheel is pushed closer and closer to the sensors.



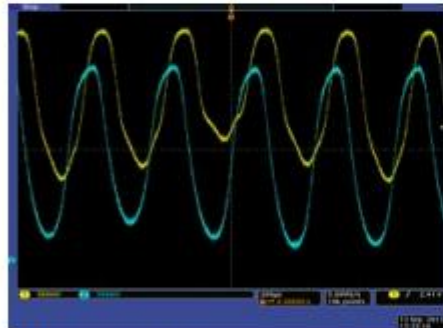
5V encoder version, motor approx. 30k RPM: 5-tooth wheel way too far from sensors.



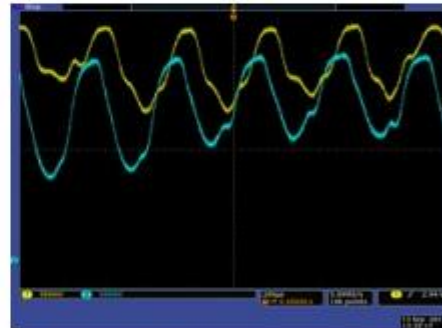
5V encoder version, motor approx. 30k RPM: 5-tooth wheel slightly too far from sensors.



5V encoder version, motor approx. 30k RPM: 5-tooth wheel at optimal distance from sensors.



5V encoder version, motor approx. 30k RPM: 5-tooth wheel slightly too close to sensors.



5V encoder version, motor approx. 30k RPM: 5-tooth wheel way too close to sensors.

The center oscilloscope capture above shows what the signal outputs look like with the wheel mounted at an optimal distance from the sensors. Note that even in this optimal case, the signals from the two channels are different due to inherent variations in the two reflectance sensors, but both signals are 90° out of phase and both span a large voltage range.

With the appropriate signal conditioning (e.g. comparators with hysteresis), all of the outputs, except perhaps for the last case, would still yield reliable position counts. The outputs also tend to improve at lower speeds and with the 3-tooth wheel.

5V and 3.3V versions

The encoder boards are available in two versions that are optimized for operation at 5.0 V and 3.3 V. The only difference is in the electrical configuration of the IR LEDs in the reflectance sensors:

5V version



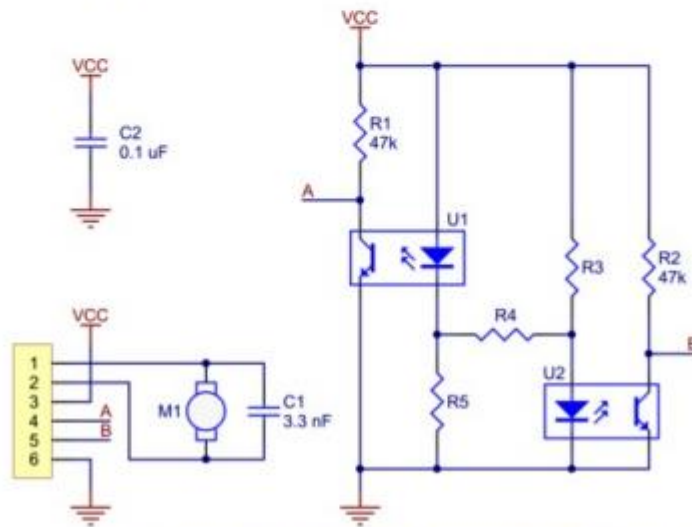
one current-limiting resistor (IR LEDs in series)

3.3V version



two current-limiting resistors (IR LEDs in parallel)

The 5 V version with LEDs in series draws approximately 12 mA; the 3.3 V version with LEDs in parallel draws approximately 24 mA.



5 V version: R4 = 220 Ω, R3 and R5 not populated (emitters in series)

3.3 V version: R4 not populated, R3 and R5 = 180 Ω (emitters in parallel)

Schematic diagram for the micro metal gearmotor reflective optical encoder.



InvenSense Inc.
1197 Borregas Ave, Sunnyvale, CA 94089 U.S.A.
Tel: +1 (408) 988-7339 Fax: +1 (408) 988-8104
Website: www.invensense.com

Document Number: PS-ITG-3200A-00-01.4
Revision: 1.4
Release Date: 03/30/2010

ITG-3200 Product Specification Revision 1.4




	<p align="center">ITG-3200 Product Specification</p>	<p>Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010</p>
---	---	---

1 Document Information

1.1 Revision History

Revision Date	Revision	Description
10/23/09	1.0	Initial Release
10/28/09	1.1	Edits for readability
02/12/2010	1.2	<ul style="list-style-type: none"> • Changed full-scale range and sensitivity scale factor (Sections 2, 3.1, 5.3, and 8.3) • Changed sensitivity scale factor variation over temperature (Section 3.1) • Changed total RMS noise spec (Section 3.1) • Added range for temperature sensor (Section 3.1) • Updated VDD Power-Supply Ramp Rate specification (Sections 3.2 and 4.4) • Added VLOGIC Voltage Range condition (Section 3.2) • Added VLOGIC Reference Voltage Ramp Rate specification (Sections 3.2 and 4.4) • Updated Start-Up Time for Register Read/Write specification (Section 3.2) • Updated Input logic levels for AD0 and CLKIN (Section 3.2) • Updated Level I_{OL} specifications for the I²C interface (Section 3.3) • Updated Frequency Variation Over Temperature specification for internal clock source (Section 3.4) • Updated VLOGIC conditions for I²C Characterization (Section 3.5) • Updated ESD specification (Section 3.6) • Added termination requirements for CLKIN if unused (Section 4.1) • Added recommended power-on procedure diagram (Section 4.4) • Changed DLPF_CFG setting 7 to reserved (Section 8.3) • Changed Reflow Specification description (Section 9.12) • Removed errata specifications
03/05/2010	1.3	<ul style="list-style-type: none"> • Updated temperature sensor linearity spec (Section 3.1) • Updated VDD Power-Supply Ramp Rate timing figure (Sections 3.2 and 4.4) • Updated VLOGIC Reference Voltage timing figure (Section 4.4) • Added default values to registers (all of Section 8) • Updated FS_SEL description (Section 8.3) • Updated package outline drawing and dimensions (Section 9.2) • Updated Reliability (Section 10.1 and 10.2) • Removed Environmental Compliance (Section 11)
03/30/2010	1.4	<ul style="list-style-type: none"> • Removed confidentiality mark

	ITG-3200 Product Specification	Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010
---	---------------------------------------	--

1.2 Purpose and Scope

This document is a preliminary product specification, providing a description, specifications, and design related information for the ITG-3200™. Electrical characteristics are based upon simulation results and limited characterization data of advanced samples only. Specifications are subject to change without notice. Final specifications will be updated based upon characterization of final silicon.

1.3 Product Overview

The ITG-3200 is the world's first single-chip, digital-output, 3-axis MEMS gyro IC optimized for gaming, 3D mice, and 3D remote control applications. The part features enhanced bias and sensitivity temperature stability, reducing the need for user calibration. Low frequency noise is lower than previous generation devices, simplifying application development and making for more-responsive remote controls.

The ITG-3200 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyro outputs, a user-selectable internal low-pass filter bandwidth, and a Fast-Mode I²C (400kHz) interface. Additional features include an embedded temperature sensor and a 2% accurate internal oscillator. This breakthrough in gyroscope technology provides a dramatic 67% package size reduction, delivers a 50% power reduction, and has inherent cost advantages compared to competing multi-chip gyro solutions.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, InvenSense has driven the ITG-3200 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging required for handheld consumer electronic devices. The part features a robust 10,000g shock tolerance, as required by portable consumer equipment.

For power supply flexibility, the ITG-3200 has a separate VLOGIC reference pin, in addition to its analog supply pin, VDD, which sets the logic levels of its I²C interface. The VLOGIC voltage may be anywhere from 1.71V min to VDD max.

1.4 Applications

- Motion-enabled game controllers
- Motion-based portable gaming
- Motion-based 3D mice and 3D remote controls
- "No Touch" UI
- Health and sports monitoring

**ITG-3200 Product Specification**

Document Number: PS-ITG-3200A-00-01.4
Revision: 1.4
Release Date: 03/30/2010

2 Features

The ITG-3200 triple-axis MEMS gyroscope includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyros) on one integrated circuit with a sensitivity of 14.375 LSBs per °/sec and a full-scale range of $\pm 2000^\circ/\text{sec}$
- Three integrated 16-bit ADCs provide simultaneous sampling of gyros while requiring no external multiplexer
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Low frequency noise lower than previous generation devices, simplifying application development and making for more-responsive motion processing
- Digitally-programmable low-pass filter
- Low 6.5mA operating current consumption for long battery life
- Wide VDD supply voltage range of 2.1V to 3.6V
- Flexible VLOGIC reference voltage allows for I²C interface voltages from 1.71V to VDD
- Standby current: 5 μ A
- Smallest and thinnest package for portable devices (4x4x0.9mm QFN)
- No high pass filter needed
- Turn on time: 50ms
- Digital-output temperature sensor
- Factory calibrated scale factor
- 10,000 g shock tolerant
- Fast Mode I²C (400kHz) serial interface
- On-chip timing generator clock frequency is accurate to +/-2% over full temperature range
- Optional external clock inputs of 32.768kHz or 19.2MHz to synchronize with system clock
- MEMS structure hermetically sealed and bonded at wafer level
- RoHS and Green compliant



	ITG-3200 Product Specification	Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010
---	---------------------------------------	--

3 Electrical Characteristics


3.1 Sensor Specifications

Typical Operating Circuit of Section 4.2, VDD = 2.5V, VLOGIC = 1.71V to VDD, T_A = 25°C.

Parameter	Conditions	Min	Typical	Max	Unit	Note
GYRO SENSITIVITY						
Full-Scale Range	FS_SEL=3		±2000		%s	4
Gyro ADC Word Length			16		Bits	3
Sensitivity Scale Factor	FS_SEL=3		14.375		LSB/(%s)	3
Sensitivity Scale Factor Tolerance	25°C	-6		+6	%	1
Sensitivity Scale Factor Variation Over Temperature			±10		%	2
Nonlinearity	Best fit straight line; 25°C		0.2		%	6
Cross-Axis Sensitivity			2		%	6
GYRO ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance			±40		%s	1
ZRO Variation Over Temperature	-40°C to +85°C		±40		%s	2
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.2V		0.2		%s	5
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.2V		0.2		%s	5
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.2V		4		%s	5
Linear Acceleration Sensitivity	Static		0.1		%s/g	6
GYRO NOISE PERFORMANCE						
Total RMS noise	FS_SEL=3 100Hz LPF (DLPFCFG=2)		0.38		%s-rms	1
Rate Noise Spectral Density	At 10Hz		0.03		%s/√Hz	2
GYRO MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	1
Y-Axis		27	30	33	kHz	1
Z-Axis		24	27	30	kHz	1
Frequency Separation	Between any two axes	1.7			kHz	1
GYRO START-UP TIME						
ZRO Settling	DLPFCFG=0 to ±1% of Final		50		ms	6
TEMPERATURE SENSOR						
Range			-30 to +85		°C	2
Sensitivity			280		LSB/°C	2
Temperature Offset	35°C		-13,200		LSB	1
Initial Accuracy	35°C		TBD		°C	
Linearity	Best fit straight line (-30°C to +85°C)		±1		°C	2, 5
TEMPERATURE RANGE						
Specified Temperature Range		-40		85	°C	

Notes:

1. Tested in production
2. Based on characterization of 30 pieces over temperature on evaluation board or in socket
3. Based on design, through modeling and simulation across PVT
4. Typical. Randomly selected part measured at room temperature on evaluation board or in socket
5. Based on characterization of 5 pieces over temperature
6. Tested on 5 parts at room temperature

	ITG-3200 Product Specification	Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010
---	---------------------------------------	--

3.2 Electrical Specifications

Typical Operating Circuit of Section 4.2, VDD = 2.5V, VLOGIC = 1.71V to VDD, T_A=25°C.

Parameters	Conditions	Min	Typical	Max	Units	Notes
VDD POWER SUPPLY						
Operating Voltage Range		2.1		3.6	V	2
Power-Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value (see Figure in Section 4.4)	0		5	ms	2
Normal Operating Current			6.5		mA	1
Sleep Mode Current			5		µA	5
VLOGIC REFERENCE VOLTAGE						
Voltage Range	VLOGIC must be ≤VDD at all times	1.71		VDD	V	
VLOGIC Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value (see Figure in Section 4.4)			1	ms	6
Normal Operating Current			100		µA	
START-UP TIME FOR REGISTER READ/WRITE			20		ms	5
I²C ADDRESS						
	AD0 = 0		1101000			6
	AD0 = 1		1101001			6
DIGITAL INPUTS (AD0, CLKIN)						
V _{BI} , High Level Input Voltage		0.9*VLOGIC			V	5
V _{IL} , Low Level Input Voltage				0.1*VLOGIC	V	5
C _I , Input Capacitance				5	pF	7
DIGITAL OUTPUT (INT)						
V _{OH} , High Level Output Voltage	OPEN=0, Rload=1MΩ	0.9*VLOGIC			V	2
V _{OL} , Low Level Output Voltage	OPEN=0, Rload=1MΩ			0.1*VLOGIC	V	2
V _{OLINT} , INT Low-Level Output Voltage	OPEN=1, 0.3mA sink current			0.1	V	2
Output Leakage Current	OPEN=1		100		nA	4
t _{INT} , INT Pulse Width	LATCH_INT_EN=0		50		µs	4

Notes:

1. Tested in production
2. Based on characterization of 30 pieces over temperature on evaluation board or in socket
4. Typical. Randomly selected part measured at room temperature on evaluation board or in socket
5. Based on characterization of 5 pieces over temperature
6. Guaranteed by design

	<p align="center">ITG-3200 Product Specification</p>	<p>Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010</p>
---	---	---

3.3 Electrical Specifications, continued

Typical Operating Circuit of Section 4.2, VDD = 2.5V, VLOGIC = 1.71V to VDD, T_A=25°C.

Parameters	Conditions	Typical	Units	Notes
I ² C I/O (SCL, SDA)				
V _{IL} , LOW-Level Input Voltage		-0.5 to 0.3*VLOGIC	V	2
V _{IH} , HIGH-Level Input Voltage		0.7*VLOGIC to VLOGIC + 0.5V	V	2
V _{hys} , Hysteresis		0.1*VLOGIC	V	2
V _{OL1} , LOW-Level Output Voltage	3mA sink current	0 to 0.4	V	2
I _{OL} , LOW-Level Output Current	V _{OL} = 0.4V	3	mA	2
	V _{OL} = 0.6V	6	mA	2
Output Leakage Current		100	nA	4
t _{df} , Output Fall Time from V _{IHmax} to V _{ILmax}	C _b bus cap. in pF	20+0.1C _b to 250	ns	2
C _i , Capacitance for Each I/O pin		10	pF	5

Notes:

2. Based on characterization of 5 pieces over temperature.
4. Typical. Randomly selected part measured at room temperature on evaluation board or in socket
5. Guaranteed by design



	ITG-3200 Product Specification	Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010
---	---------------------------------------	--


3.4 Electrical Specifications, continued

Typical Operating Circuit of Section 4.2, VDD = 2.5V, VLOGIC = 1.71V to VDD, TA=25°C.

Parameters	Conditions	Min	Typical	Max	Units	Notes
INTERNAL CLOCK SOURCE	CLKSEL=0, 1, 2, or 3					
Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	4
Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	4
Clock Frequency Initial Tolerance	CLKSEL=0, 25°C	-2		+2	%	1
Frequency Variation over Temperature	CLKSEL=1,2,3; 25°C	-1		+1	%	1
	CLKSEL=0		-15 to +10		%	2
	CLKSEL=1,2,3		+/-1		%	2
PLL Settling Time	CLKSEL=1,2,3		1		ms	3
EXTERNAL 32.768kHz CLOCK	CLKSEL=4					
External Clock Frequency			32.768		kHz	3
External Clock Jitter	Cycle-to-cycle rms		1 to 2		µs	3
Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8.192		kHz	3
Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1.024		kHz	3
PLL Settling Time			1		ms	3
EXTERNAL 19.2MHz CLOCK	CLKSEL=5					
External Clock Frequency			19.2		MHz	3
Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	3
Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	3
PLL Settling Time			1		ms	3
Charge Pump Clock Frequency						
Frequency	1 st Stage, 25°C		8.5		MHz	5
	2 nd Stage, 25°C		68		MHz	5
	Over temperature		+/-15		%	5

Notes:

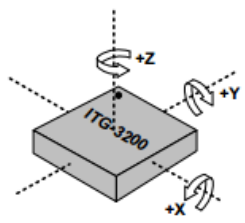
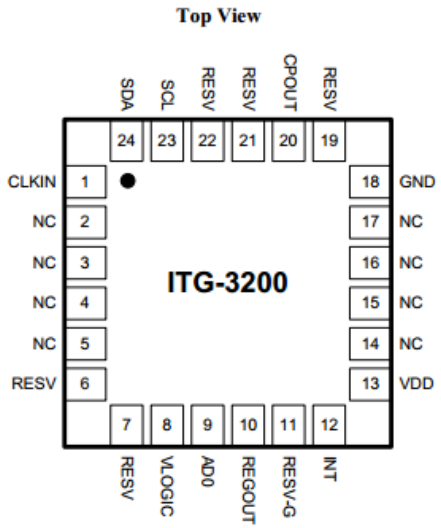
1. Tested in production
2. Based on characterization of 30 pieces over temperature on evaluation board or in socket
3. Based on design, through modeling and simulation across PVT
4. Typical. Randomly selected part measured at room temperature on evaluation board or in socket
5. Based on characterization of 5 pieces over temperature.

	ITG-3200 Product Specification	Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010
---	---------------------------------------	--

4 Applications Information

4.1 Pin Out and Signal Description

Number	Pin	Pin Description
1	CLKIN	Optional external reference clock input. Connect to GND if unused.
8	VLOGIC	Digital IO supply voltage. VLOGIC must be \leq VDD at all times.
9	AD0	I ² C Slave Address LSB
10	REGOUT	Regulator filter capacitor connection
12	INT	Interrupt digital output (totem pole or open-drain)
13	VDD	Power supply voltage
18	GND	Power supply ground
11	RESV-G	Reserved - Connect to ground.
6, 7, 19, 21, 22	RESV	Reserved. Do not connect.
20	CPOUT	Charge pump capacitor connection
23	SCL	I ² C serial clock
24	SDA	I ² C serial data
2, 3, 4, 5, 14, 15, 16, 17	NC	Not internally connected. May be used for PCB trace routing.

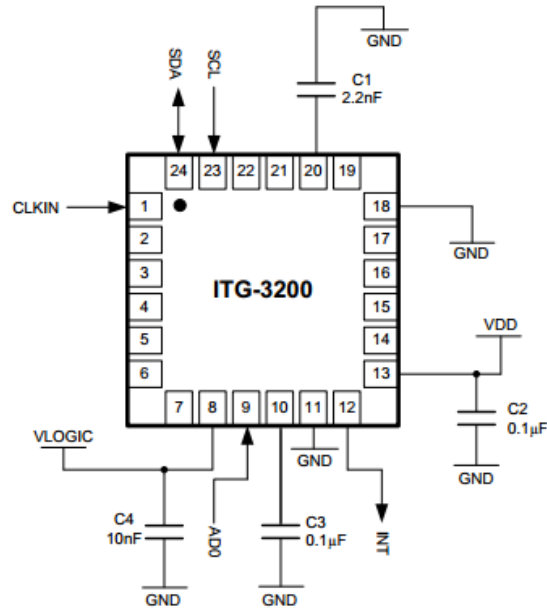


QFN Package
24-pin, 4mm x 4mm x 0.9mm

Orientation of Axes of Sensitivity and Polarity of Rotation

	<p align="center">ITG-3200 Product Specification</p>	<p>Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010</p>
---	---	---

4.2 Typical Operating Circuit



Typical Operating Circuit

4.3 Bill of Materials for External Components

Component	Label	Specification	Quantity
Charge Pump Capacitor	C1	Ceramic, X7R, 2.2nF ±10%, 50V	1
VDD Bypass Capacitor	C2	Ceramic, X7R, 0.1µF ±10%, 4V	1
Regulator Filter Capacitor	C3	Ceramic, X7R, 0.1µF ±10%, 2V	1
VLOGIC Bypass Capacitor	C4	Ceramic, X7R, 10nF ±10%, 4V	1

	<p align="center">ITG-3200 Product Specification</p>	<p>Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010</p>
---	---	---

This information furnished by InvenSense is believed to be accurate and reliable. However, no responsibility is assumed by InvenSense for its use, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. InvenSense reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice. InvenSense makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. InvenSense assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by implication or otherwise under any patent or patent rights of InvenSense. This publication supersedes and replaces all information previously supplied. Trademarks that are registered trademarks are the property of their respective companies. InvenSense sensors should not be used or sold in the development, storage, production or utilization of any conventional or mass-destructive weapons or for any other weapons or life threatening applications, as well as in any other life critical applications such as medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, disaster prevention and crime prevention equipment.

InvenSense, InvenSense logo, ITG, and ITG-3200 are trademarks of InvenSense, Inc.

©2009 InvenSense, Inc. All rights reserved.





3-Axis, $\pm 2\text{ g}/\pm 4\text{ g}/\pm 8\text{ g}/\pm 16\text{ g}$ Digital Accelerometer

ADXL345

FEATURES

- Ultralow power:** as low as 40 μA in measurement mode and 0.1 μA in standby mode at $V_s = 2.5\text{ V}$ (typical)
- Power consumption scales automatically with bandwidth**
- User-selectable resolution**
- Fixed 10-bit resolution**
- Full resolution, where resolution increases with g range, up to 13-bit resolution at $\pm 16\text{ g}$ (maintaining 4 mg/LSB scale factor in all g ranges)**
- Embedded, patent pending FIFO technology minimizes host processor load**
- Tap/double tap detection**
- Activity/inactivity monitoring**
- Free-fall detection**
- Supply voltage range:** 2.0 V to 3.6 V
- I/O voltage range:** 1.7 V to V_s
- SPI (3- and 4-wire) and I²C digital interfaces**
- Flexible interrupt modes mappable to either interrupt pin**
- Measurement ranges selectable via serial command**
- Bandwidth selectable via serial command**
- Wide temperature range** (-40°C to $+85^\circ\text{C}$)
- 10,000 g shock survival**
- Pb free/RoHS compliant**
- Small and thin:** 3 mm \times 5 mm \times 1 mm LGA package

APPLICATIONS

- Handsets
- Medical instrumentation
- Gaming and pointing devices
- Industrial instrumentation
- Personal navigation devices
- Hard disk drive (HDD) protection
- Fitness equipment

GENERAL DESCRIPTION

The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to $\pm 16\text{ g}$. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I²C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/LSB) enables measurement of inclination changes less than 1.0° .

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-fall sensing detects if the device is falling. These functions can be mapped to one of two interrupt output pins. An integrated, patent pending 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor intervention.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, 3 mm \times 5 mm \times 1 mm, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

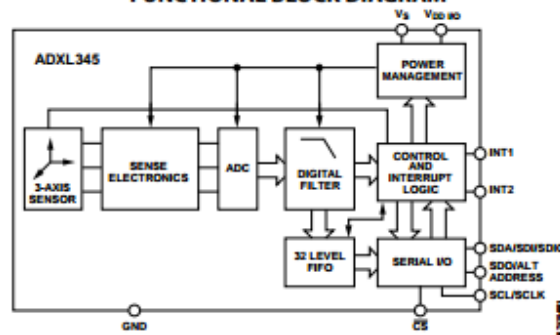


Figure 1.

Rev. 0
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners. See the last page for disclaimers.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 ©2009 Analog Devices, Inc. All rights reserved.

ADXL345

SPECIFICATIONS

T_A = 25°C, V_S = 2.5 V, V_{DD_IO} = 1.8 V, acceleration = 0 g, C_S = 1 μF tantalum, C_D = 0.1 μF, unless otherwise noted.

Table 1. Specifications¹

Parameter	Test Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	User selectable		±2, ±4, ±8, ±16		g
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity ²			±1		%
OUTPUT RESOLUTION					
All g Ranges	10-bit resolution		10		Bits
±2 g Range	Full resolution		10		Bits
±4 g Range	Full resolution		11		Bits
±8 g Range	Full resolution		12		Bits
±16 g Range	Full resolution		13		Bits
SENSITIVITY					
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±2 g, 10-bit or full resolution	232	256	286	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±2 g, 10-bit or full resolution	3.5	3.9	4.3	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±4 g, 10-bit resolution	116	128	143	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±4 g, 10-bit resolution	7.0	7.8	8.6	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±8 g, 10-bit resolution	58	64	71	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±8 g, 10-bit resolution	14.0	15.6	17.2	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±16 g, 10-bit resolution	29	32	36	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±16 g, 10-bit resolution	28.1	31.2	34.3	mg/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C
0 g BIAS LEVEL					
0 g Output for X _{OUT} , Y _{OUT}	Each axis	-150	±40	+150	mg
0 g Output for Z _{OUT}		-250	±80	+250	mg
0 g Offset vs. Temperature for x-, y-Axes			±0.8		mg/°C
0 g Offset vs. Temperature for z-Axis			±4.5		mg/°C
NOISE PERFORMANCE					
Noise (x-, y-Axes)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.0		LSB rms
Noise (z-Axis)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.5		LSB rms
OUTPUT DATA RATE AND BANDWIDTH					
Measurement Rate ³	User selectable	6.25		3200	Hz
SELF-TEST⁴					
Output Change in x-Axis	Data rate ≥ 100 Hz, 2.0 V ≤ V _S ≤ 3.6 V	0.20		2.10	g
Output Change in y-Axis		-2.10		-0.20	g
Output Change in z-Axis		0.30		3.40	g
POWER SUPPLY					
Operating Voltage Range (V _S)		2.0	2.5	3.6	V
Interface Voltage Range (V _{IO})	V _S ≤ 2.5 V	1.7	1.8	V _S	V
	V _S ≥ 2.5 V	2.0	2.5	V _S	V
Supply Current	Data rate > 100 Hz		145		μA
	Data rate < 10 Hz		40		μA
Standby Mode Leakage Current			0.1	2	μA
Turn-On Time ⁵	Data rate = 3200 Hz		1.4		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	°C
WEIGHT					
Device Weight			20		mg

¹ All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

² Cross-axis sensitivity is defined as coupling between any two axes.

³ Bandwidth is half the output data rate.

⁴ Self-test change is defined as the output (g) when the SELF_TEST bit = 1 (in the DATA_FORMAT register) minus the output (g) when the SELF_TEST bit = 0 (in the DATA_FORMAT register). Due to device filtering, the output reaches its final value after 4 × τ when enabling or disabling self-test, where τ = 1/(data rate).

⁵ Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately τ + 1.1 in milliseconds, where τ = 1/(data rate).

ADXL345

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration	
Any Axis, Unpowered	10,000 g
Any Axis, Powered	10,000 g
V _S	−0.3 V to +3.6 V
V _{DDIO}	−0.3 V to +3.6 V
Digital Pins	−0.3 V to V _{DDIO} + 0.3 V or 3.6 V, whichever is less
All Other Pins	−0.3 V to +3.6 V
Output Short-Circuit Duration (Any Pin to Ground)	Indefinite
Temperature Range	
Powered	−40°C to +105°C
Storage	−40°C to +105°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL RESISTANCE

Table 3. Package Characteristics

Package Type	θ _{JA}	θ _{JC}	Device Weight
14-Terminal LGA	150°C/W	85°C/W	20 mg

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

ADXL345

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

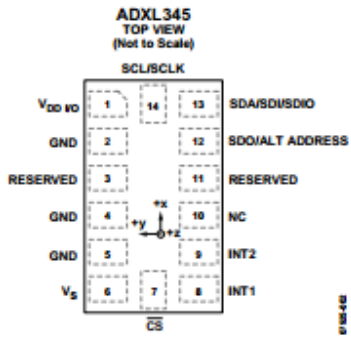


Table 4. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	V _{DDIO}	Digital Interface Supply Voltage.
2	GND	Must be connected to ground.
3	Reserved	Reserved. This pin must be connected to V _S or left open.
4	GND	Must be connected to ground.
5	GND	Must be connected to ground.
6	V _S	Supply Voltage.
7	\overline{CS}	Chip Select.
8	INT1	Interrupt 1 Output.
9	INT2	Interrupt 2 Output.
10	NC	Not Internally Connected.
11	Reserved	Reserved. This pin must be connected to ground or left open.
12	SDO/ALT ADDRESS	Serial Data Output/Alternate I ² C Address Select.
13	SDA/SDI/SDIO	Serial Data (I ² C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).
14	SCL/SCLK	Serial Communications Clock.

ADXL345

AXES OF ACCELERATION SENSITIVITY

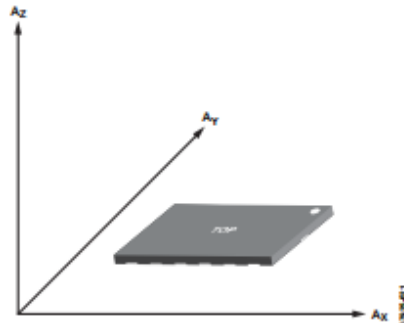


Figure 16. Axes of Acceleration Sensitivity (Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis)

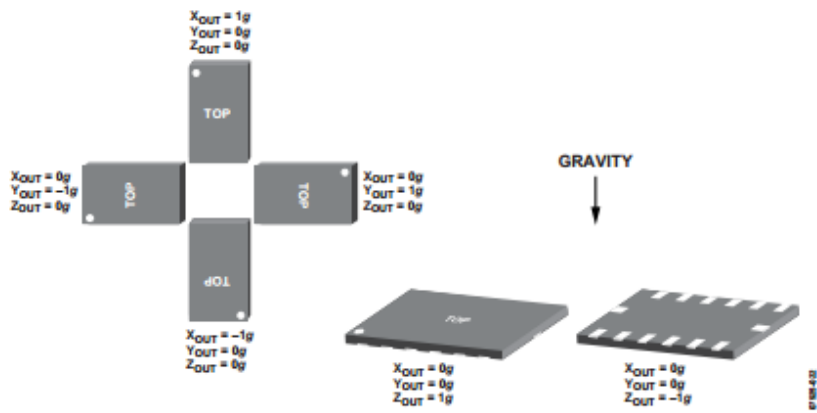


Figure 17. Output Response vs. Orientation to Gravity

ADXL345

LAYOUT AND DESIGN RECOMMENDATIONS

Figure 18 shows the recommended printed wiring board land pattern. Figure 19 and Table 21 provide details about the recommended soldering profile.

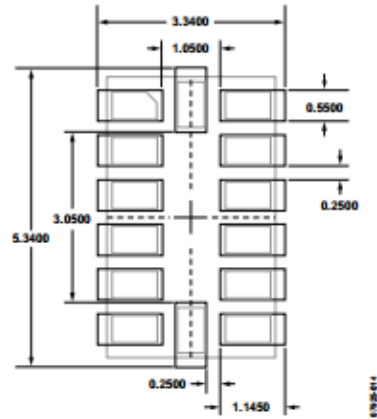


Figure 18. Recommended Printed Wiring Board Land Pattern (Dimensions shown in millimeters)

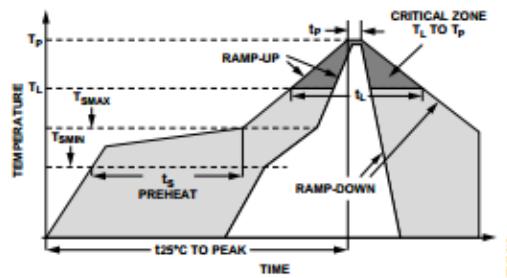


Figure 19. Recommended Soldering Profile

Table 21. Recommended Soldering Profile^{1, 2}

Profile Feature	Condition	
	Sn63/Pb37	Pb-Free
Average Ramp Rate from Liquid Temperature (T_L) to Peak Temperature (T_P)	3°C/sec max	3°C/sec max
Preheat		
Minimum Temperature (T_{SMN})	100°C	150°C
Maximum Temperature (T_{SMAX})	150°C	200°C
Time from T_{SMN} to T_{SMAX} (t_s)	60 sec to 120 sec	60 sec to 180 sec
T_{SMAX} to T_L Ramp-Up Rate	3°C/sec max	3°C/sec max
Liquid Temperature (T_L)	183°C	217°C
Time Maintained Above T_L (t_L)	60 sec to 150 sec	60 sec to 150 sec
Peak Temperature (T_P)	240 + 0/-5°C	260 + 0/-5°C
Time of Actual $T_P - 5^\circ\text{C}$ (t_p)	10 sec to 30 sec	20 sec to 40 sec
Ramp-Down Rate	6°C/sec max	6°C/sec max
Time 25°C to Peak Temperature	6 minutes max	8 minutes max

¹ Based on JEDEC Standard J-STD-020D.1.

² For best results, the soldering profile should be in accordance with the recommendations of the manufacturer of the solder paste used.

ADXL345

OUTLINE DIMENSIONS

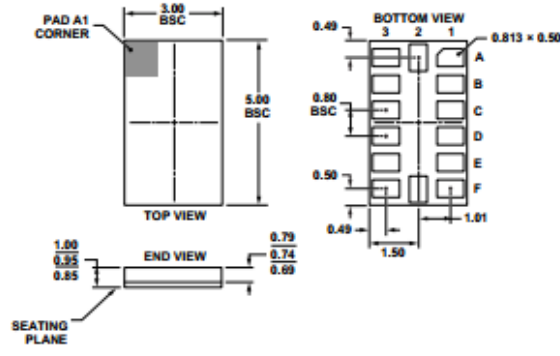


Figure 20. 14-Terminal Land Grid Array [LGA] (CC-14-1)
Solder Terminations Finish Is Au over Ni
(Dimensions shown in millimeters)

ORDERING GUIDE

Model	Measurement Range (g)	Specified Voltage (V)	Temperature Range	Package Description	Package Option
ADXL345BCCZ ¹	±2, ±4, ±8, ±16	2.5	-40°C to +85°C	14-Terminal Land Grid Array [LGA]	CC-14-1
ADXL345BCCZ-RL ¹	±2, ±4, ±8, ±16	2.5	-40°C to +85°C	14-Terminal Land Grid Array [LGA]	CC-14-1
ADXL345BCCZ-RL7 ¹	±2, ±4, ±8, ±16	2.5	-40°C to +85°C	14-Terminal Land Grid Array [LGA]	CC-14-1
EVAL-ADXL345Z ¹				Evaluation Board	
EVAL-ADXL345Z-M ¹				Analog Devices Inertial Sensor Evaluation System, Includes ADXL345 Satellite	
EVAL-ADXL345Z-S ¹				ADXL345 Satellite, Standalone	

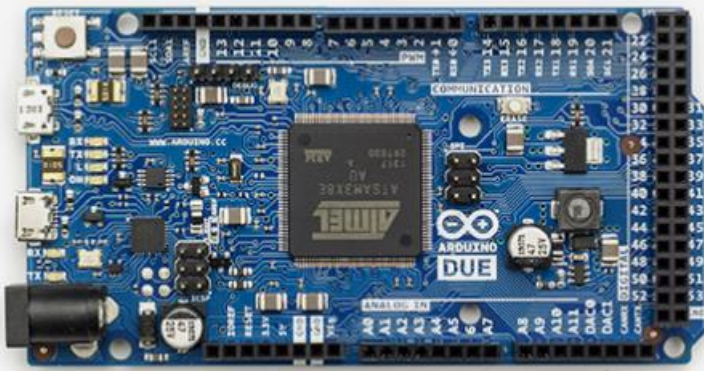
¹ Z = RoHS Compliant Part.

Analog Devices offers specific products designated for automotive applications; please consult your local Analog Devices sales representative for details. Standard products sold by Analog Devices are not designed, intended, or approved for use in life support, implantable medical devices, transportation, nuclear, safety, or other equipment where malfunction of the product can reasonably be expected to result in personal injury, death, severe property damage, or severe environmental harm. Buyer uses or sells standard products for use in the above critical applications at Buyer's own risk and Buyer agrees to defend, indemnify, and hold harmless Analog Devices from any and all damages, claims, suits, or expenses resulting from such unintended use.

©2009 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
D07925-0-5/09(0)



www.analog.com



Arduino Due

The Arduino Due is the first Arduino board based on a 32-bit ARM core microcontroller. With 54 digital input/output pins, 12 analog inputs, it is the perfect board for powerful larger scale Arduino projects.

Overview

- 3.3V
- 32-bit
- 84 MHz
- ARM Core

Technical specs

AVR Arduino microcontroller

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g