

LAPORAN PRAKTIKUM

SENSOR & AKTUATOR



Nama : Ni Putu Devira Ayu Martini
NRP : 1120800012
Kelas : 1 S2 Elektro 2020
Pengampu : Bp. Endra Pitowarno

**POLITEKNIK ELEKTRONIKA NEGERI
SURABAYA**
2020

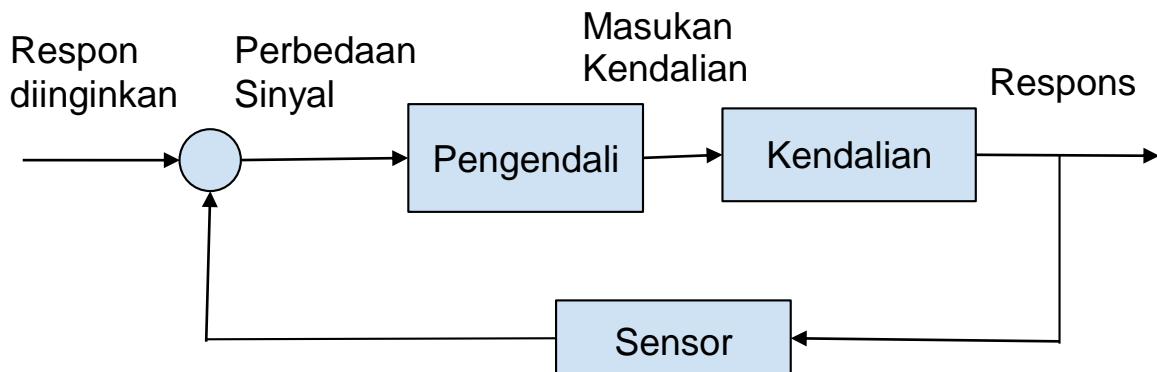
PRAKTIKUM I
PENGUJIAN KONVERSI SINYAL AKSELERASI, KECEPATAN DAN POSISI
MENGGUNAKAN DATA DEFINITIF

I. TUJUAN

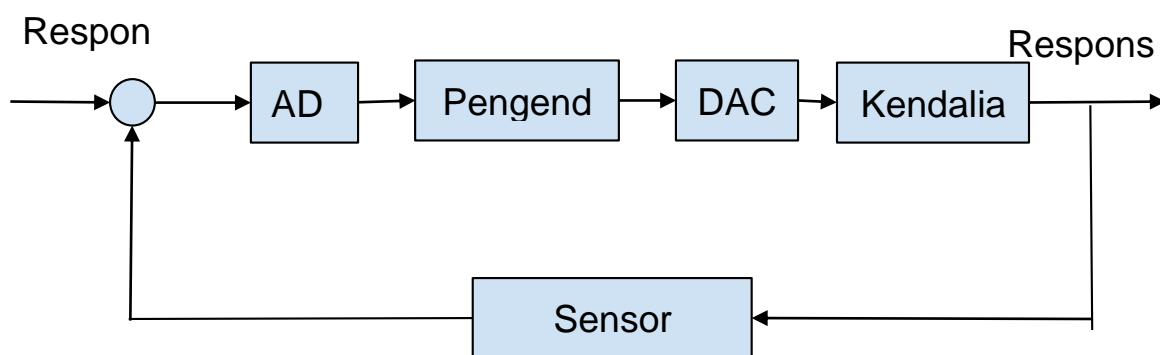
Praktikan memahami dan menguasai cara mengukur dan mengkonversi sinyal data posisi ke kecepatan, data kecepatan ke percepatan dan sebaliknya.

II. TEORI

Di dalam sistem kontrol digital, sebagai pengendali (*controller*) dipergunakan komputer, mikroprosesor, mikrokontroller ataupun rangkaian logika lainnya untuk mengolah dinamika sistem. Dari segi bentuk sinyal yang bekerja didalam sistem, dapat kita bedakan bahwa pada sistem analog, maka sinyal masukan yang diproses oleh pengendali maupun sinyal keluaran adalah berupa sinyal analog, sedangkan sinyal yang diproses oleh pengendali adalah sinyal digital, dan sinyal keluaran umumnya juga berupa sinyal analog. Dari perbedaan sinyal yang bekerja pada sistem analog dan sistem digital, maka pada sistem digital diperlukan komponen yang berfungsi untuk melakukan konversi bentuk sinyal, konverter tersebut dikenal dengan nama ADC (Analog to Digital Converter) dan DAC (Digital to Analog Converter). Perbedaan diantara sistem analog dan sistem digital diperlihatkan dalam berikut.



Gambar 1. Diagram blok Sistem Analog



Gambar 2. Diagram blok Sistem Digital

Sinyal digital diperoleh melalui ADC yang berfungsi untuk melakukan proses pencuplikan (*sampling*) terhadap sinyal analog sehingga menghasilkan sinyal diskrit, selanjutnya sinyal diskrit dilakukan kuantisasi dan seterusnya dikonversi kesinyal digital.

Umumnya suatu pengendali (*plant*) bekerja dengan sinyal analog, oleh karena itu sinyal digital sebagai keluaran dari pengendali harus dikonversi kembali ke sinyal analog melalui DAC. Dalam proses matematikanya, maka setiap blok fungsional diatas perlu dicari model matematisnya. Dalam sistem analog maka transformasi Laplace yang digunakan, sedangkan dalam sistem digital yang digunakan adalah transformasi Z. Di dalam buku ini hanya transformasi Z saja yang diberikan.

1. Pemilihan sensor untuk mengukur sinyal umpan balik
2. Pemilihan aktuator untuk menggerakan kendali

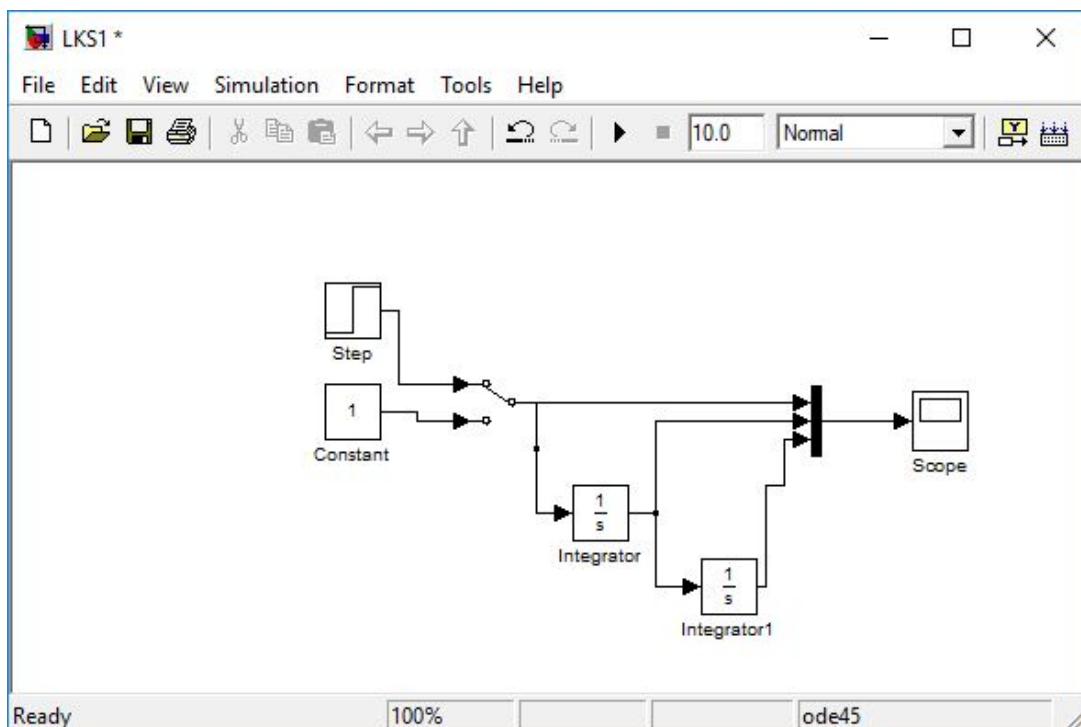
III. PERALATAN

1. PC
2. Aplikasi Matlab Simulink

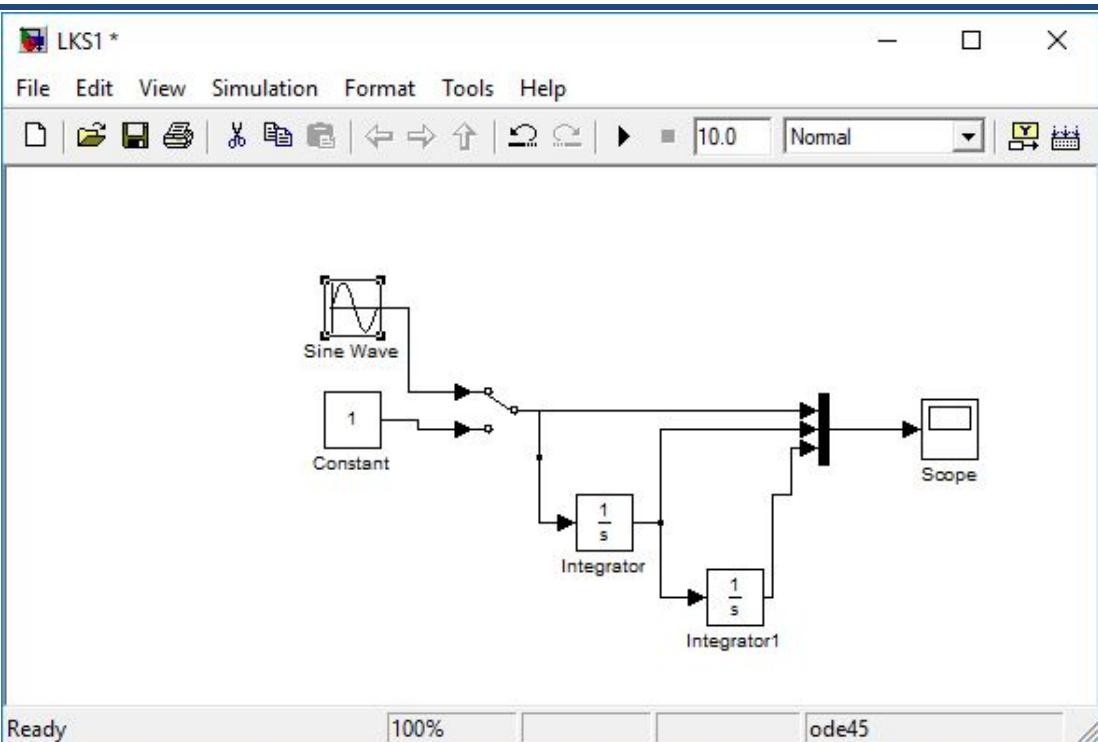
IV. PROSEDUR

IV.1 LKS-1

1. Menyiapkan peralatan yang dibutuhkan
2. Membuka aplikasi Matlab
3. Mengklik simbol Simulink untuk membuat model/skema
4. Membuat fungsi input di Simulink berupa sinyal STEP dengan nilai steady state sembarang (default 1)



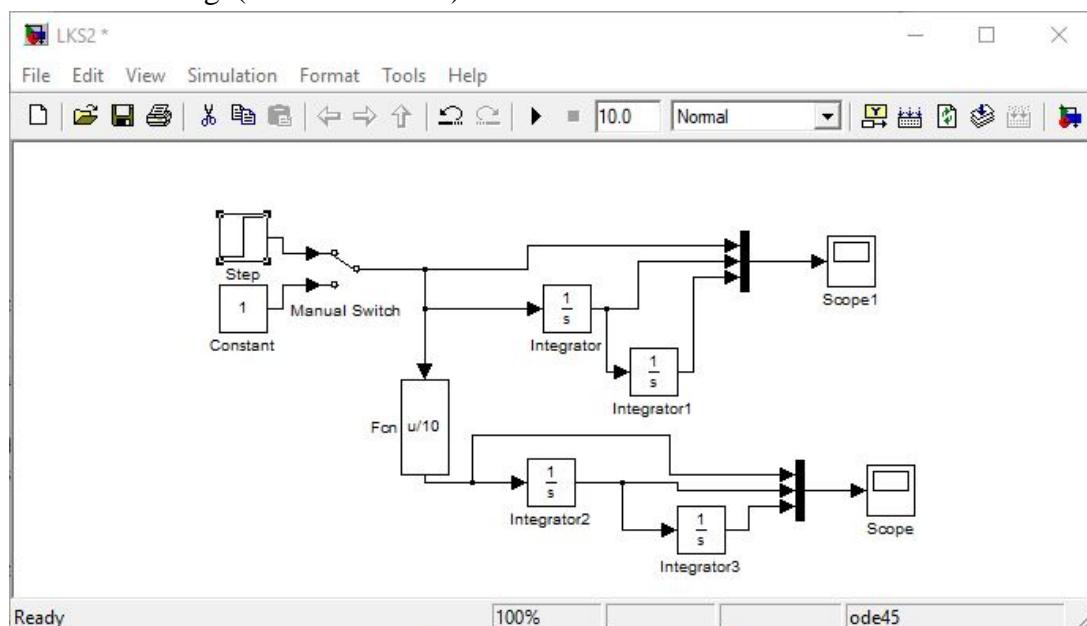
5. Jalankan model Simulink diatas dan lihat grafik yang didapat dari Scope
6. Kemudian membuat fungsi input di Simulink berupa sinyal Sinus dengan nilai tengah (DC Offset) adalah sebesar amplituda sinyal sinus (sepahru sinyal utuh). Frekuensi sinyal sinus tentukan antara 1~5Hz.



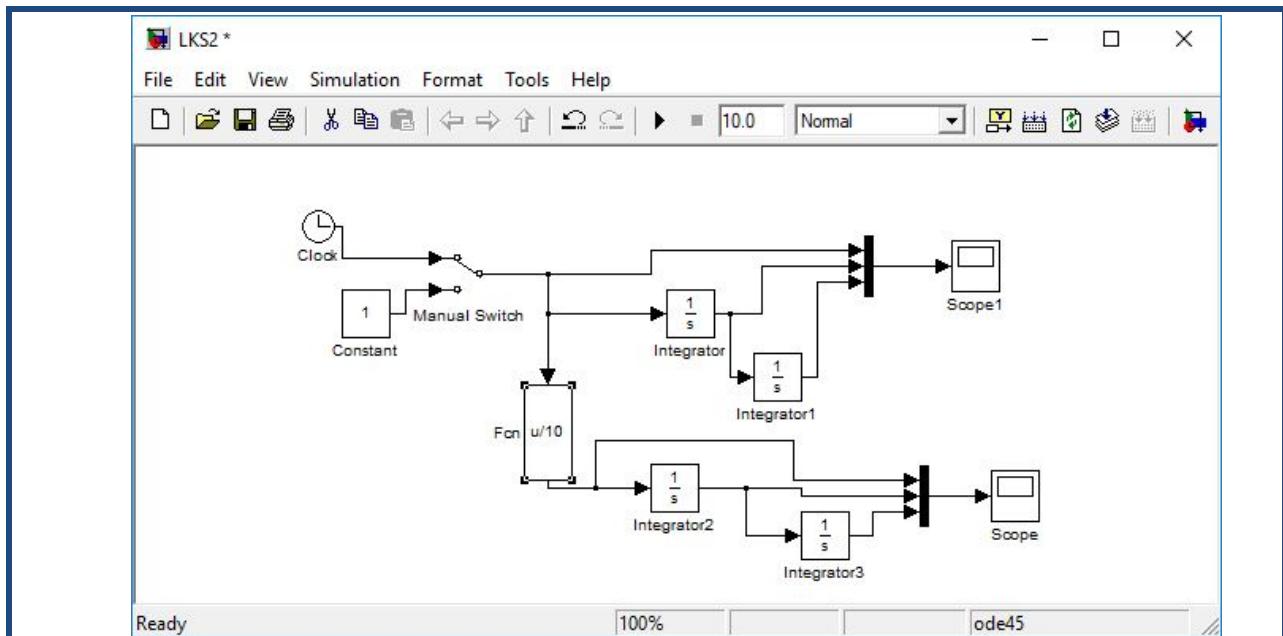
7. Jalankan model Simulink diatas dan lihat grafik yang didapat dari Scope

IV.2. LKS-2

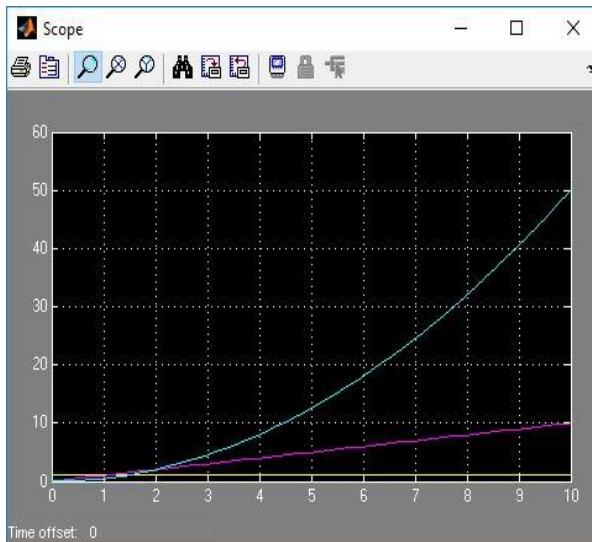
1. Menyiapkan peralatan yang dibutuhkan
2. Membuka aplikasi Matlab
3. Mengklik simbol Simulink untuk membuat model/skema
4. Buat fungsi input di Simulink berupa sinyal STEP dengan nilai steady state sembarang. (default adalah 1)



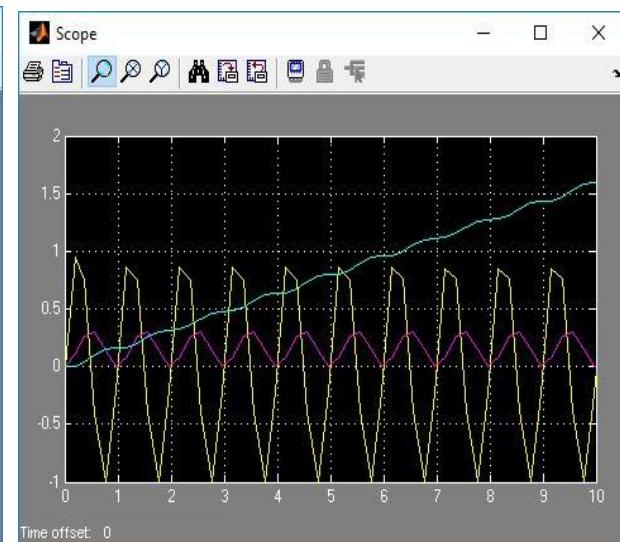
5. Buat fungsi input di Simulink berupa sinyal LINIER dengan nilai gradien 0,1; 0,5; 1; dan 1,5. (dilakukan dengan memindah input Clock dan mengubah denumerator fungsi Fcn masing-masing ke 10, 2, 1 dan 0,5).



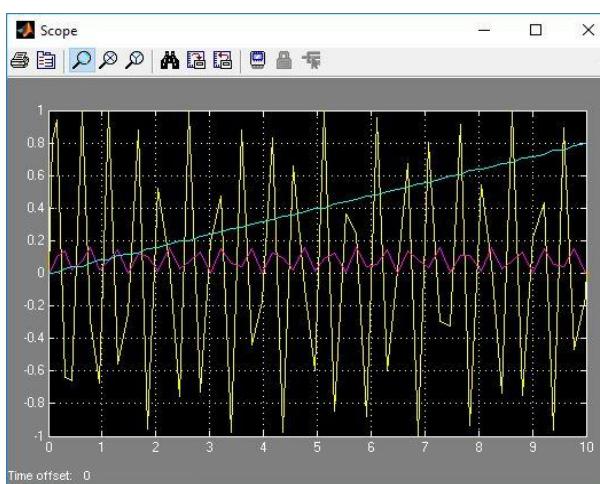
V. HASIL PERCOBAAN



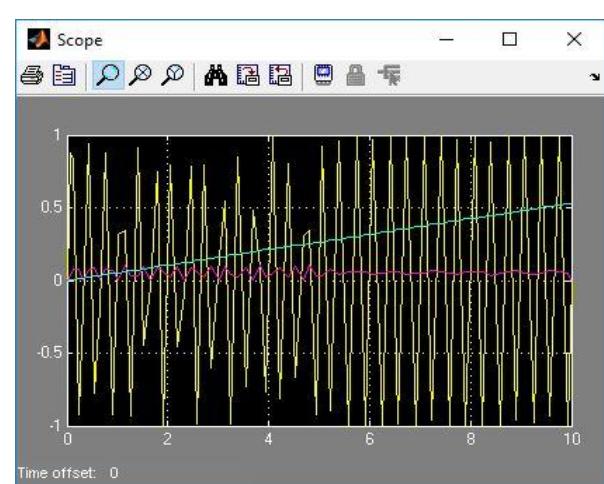
Gambar 1.1 Sinyal STEP nilai Steady State 1.



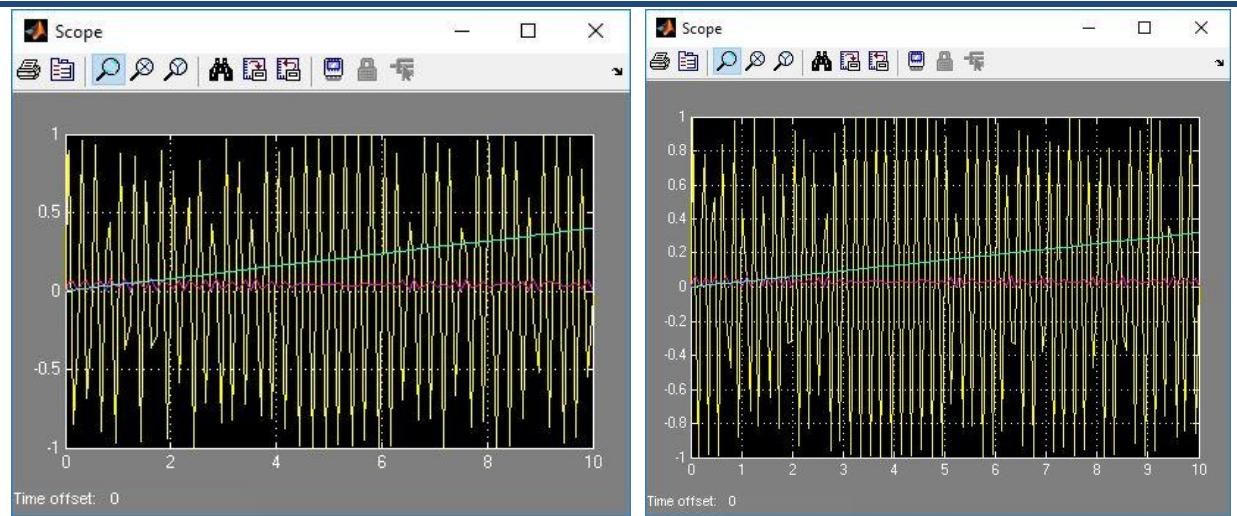
Gambar 1.2. Sinyal Sinus $F=1$ Hz (6.28 rad/s)



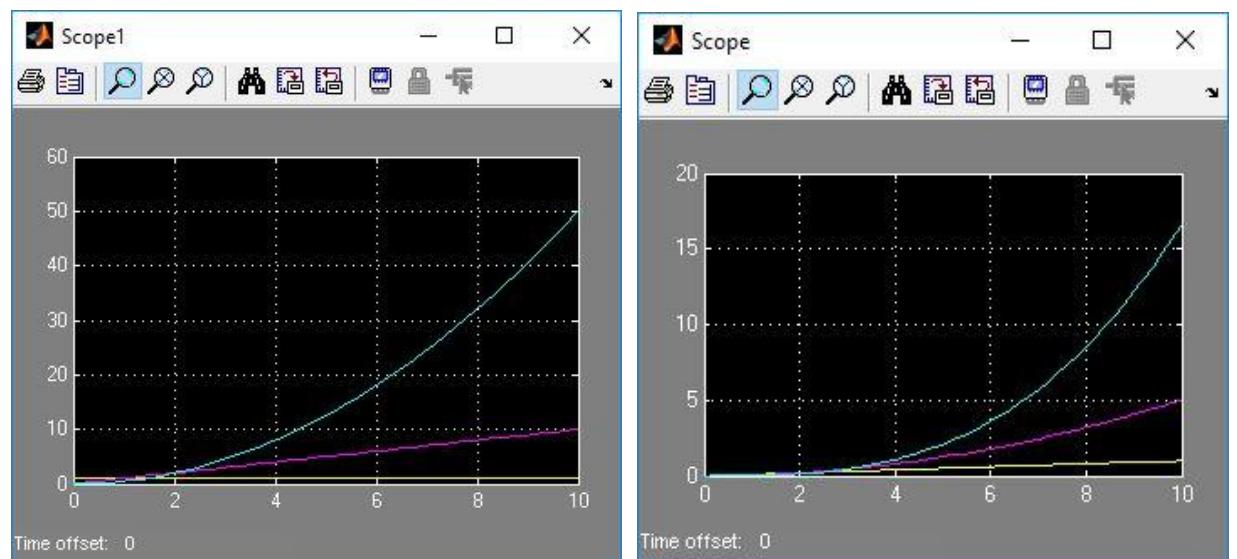
Gambar 1.3 Sinyal Sinus $F= 2$ Hz (12.56 rad/s)



Gambar 1.4 Sinyal Sinus $F = 3$ Hz (18.84 rad/s)

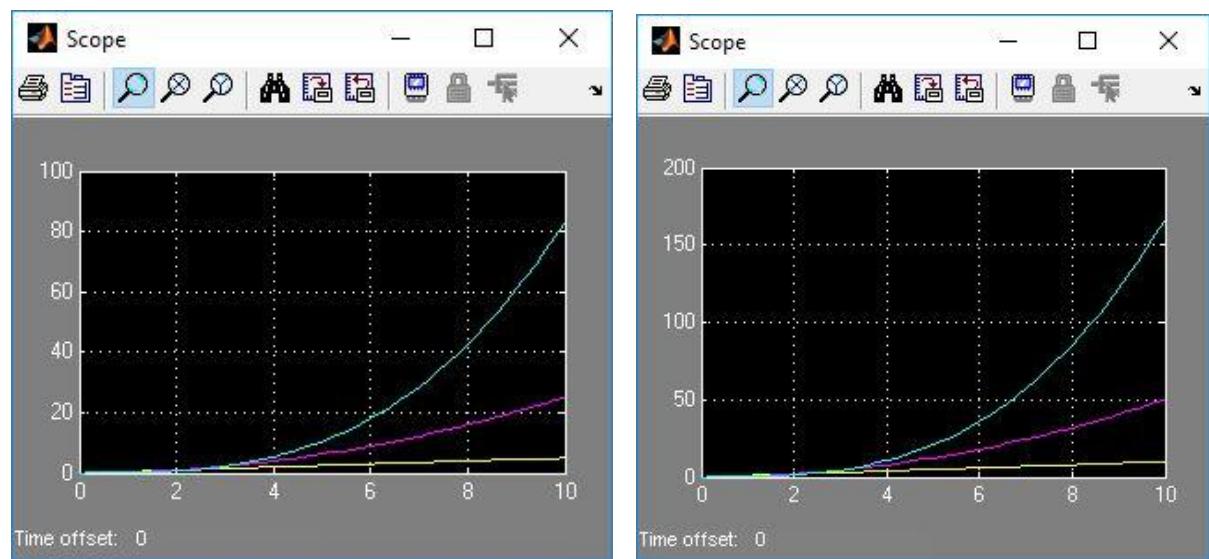


Gambar 1.5 Sinyal Sinus $F = 4$ Hz (25.13 rad/s) Gambar 1.6 Sinyal Sinus $F = 5$ Hz (31.41 rad/s)



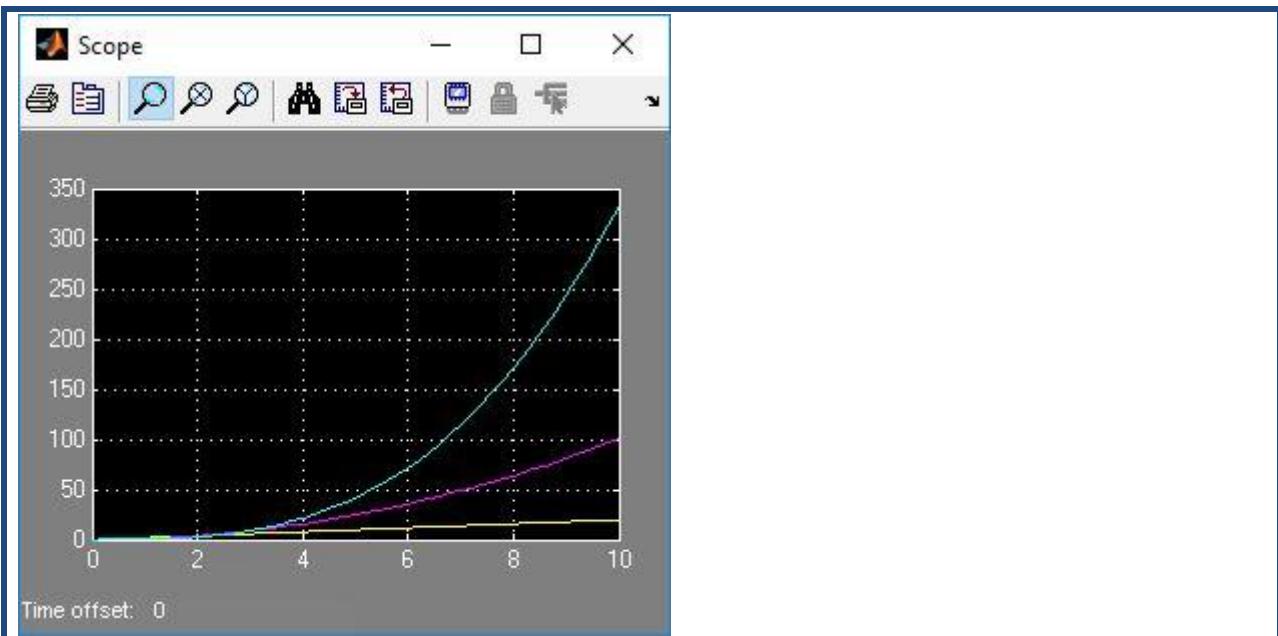
Gambar 2.1 Sinyal STEP Steady State 1

Gambar 2.2. Sinyal Linier $m= 0.1$ Fcn= 10.



Gambar 2.3 Sinyal Linier $m= 0.5$ Fcn = 2.

Gambar 2.4 Sinyal Linier $m= 1$ Fcn=1.



Gambar 2.5 m= 1.5 Fcn= 0.5

ANALISA

Pada praktikum pertama mata kuliah sensor & aktuator ini membahas tentang Pengujian Konversi Sinyal Akselerasi, Kecepatan dan Posisi menggunakan data definitif. Dilakukan pengamatan sinyal Akselerasi, Kecepatan & Posisi berupa sinyal step & sinusoida.

Model Simulink LKS-1 dengan input sinyal STEP berupa sinyal percepatan (Sinyal warna kuning), nilai kecepatan berupa grafik linear (Sinyal warna merah muda), sedangkan nilai posisi berupa grafik eksponensial terhadap waktu (Sinyal warna biru). Hal ini dikarenakan kecepatan merupakan hasil integral dari sinyal percepatan, dan posisi merupakan hasil integral kedua dari sinyal percepatan. Pada praktikum kali ini, nilai percepatan diberikan data konstan sebesar 1, maka:

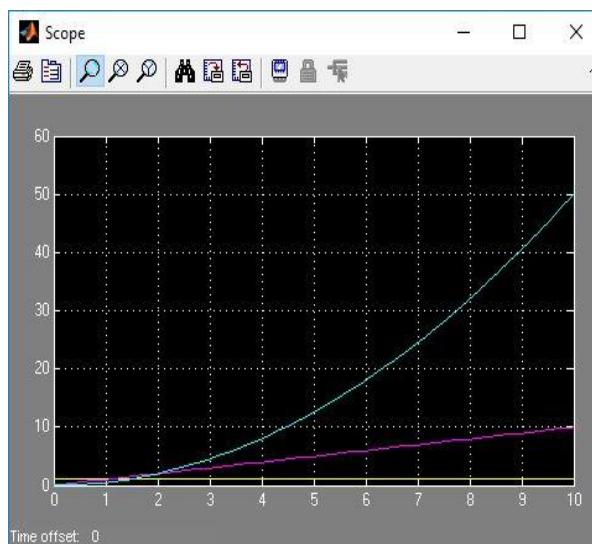
$$v(t) = \int a dt$$

Dimana $a = 1 \text{ ms}^{-2}$

$$v(t) = \int 1 dt$$

$$v(t) = t \text{ dan } s(t) = \int t dt$$

$$s(t) = \frac{1}{2}t^2$$



Sehingga, nilai kecepatan menjadi semakin besar sebanding dengan bertambahnya waktu dan nilai posisi juga semakin besar sesuai dengan grafik eksponensial sebanding dengan bertambahnya waktu.

Percobaan kedua adalah dengan input sinyal sinusoida dengan frekuensi yang bervariasi antara 1~5 Hz. Pada praktikum kali ini diketahui bahwa semakin besar nilai frekuensi yang diberikan, maka sinyal kecepatan yang terlihat akan semakin rapat (tidak bergelombang), begitupula dengan sinyal posisi. Namun, jika frekuensi pada sinyal sinusoida semakin rendah, maka grafik kecepatan dan posisi akan bergelombang. Pengujian dengan input sinusoida berfungsi untuk mengetahui respon sistem terhadap perubahan nilai input dalam satuan waktu.

Pada percobaan ketiga adalah dengan input sinyal linear dengan nilai gradien tertentu. Dari praktikum ini dapat dilihat bahwa grafik kecepatan dan posisi sebanding dengan nilai gradien dari sinyal percepatan. Semakin besar nilai gradien dari sinyal input, maka semakin besar juga nilai kecepatan dan posisinya.

KESIMPULAN

Dari praktikum ini dapat dilihat dan diketahui bahwa:

- ★ Sinyal kecepatan adalah hasil dari nilai integral sinyal percepatan
- ★ Sinyal posisi adalah hasil dari nilai integral kedua sinyal percepatan

- ★ Semakin besar nilai frekuensi yang diberikan pada sinyal percepatan berupa sinyal sinusoida, maka sinyal kecepatan dan posisinya akan semakin rapat atau tidak terjadi osilasi.
- ★ Grafik kecepatan dan posisi sebanding dengan nilai gradien dari sinyal input percepatan berupa sinyal linear dengan nilai gradien tertentu.

PERCOBAAN II
PENGUJIAN KONVERSI SINYAL AKSELERASI, KECEPATAN DAN POSISI
MENGGUNAKAN DATA DEFINITIF

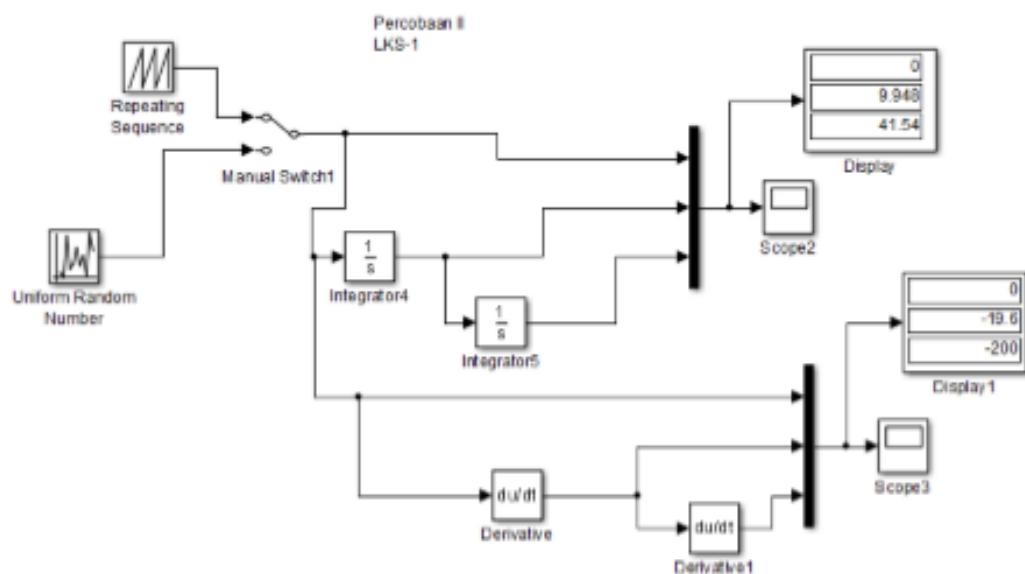
I. DASAR TEORI

II. PERALATAN

1. PC
2. Software Matlab

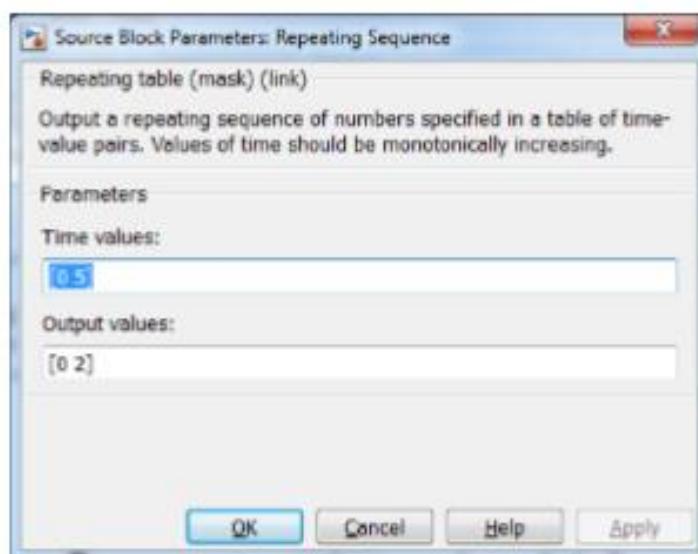
III. PROSEDUR

1. Menyiapkan peralatan yang dibutuhkan
2. Membuat tool di Simulink seperti dibawah ini :



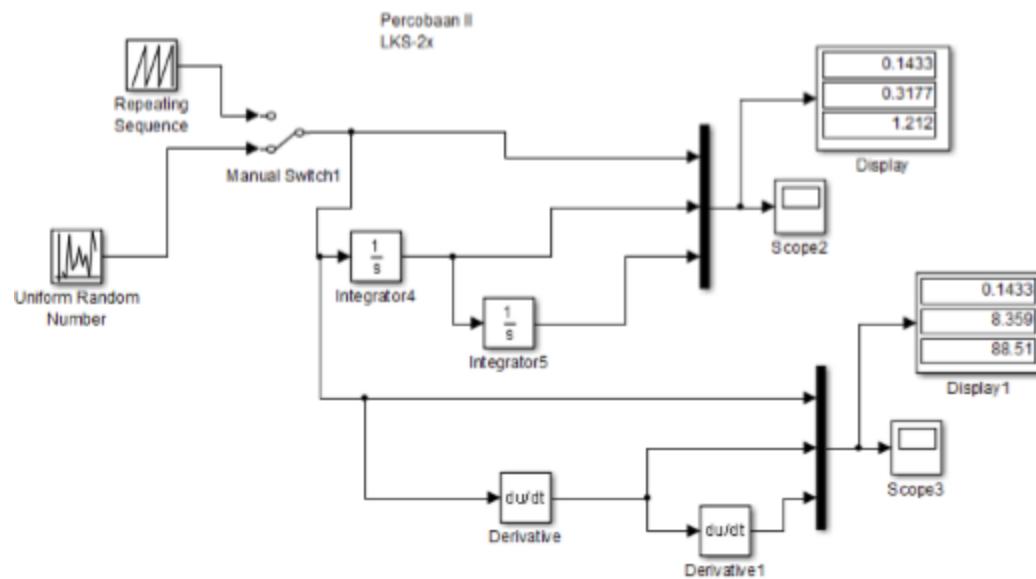
Gambar 1. LKS-1

3. Membuat fungsi input di Simulik berupa sinyal input akselerasi RAMP dengan nilai steady state nol, maksimum/puncak sembarang, dengan periode sembarang.



Gambar 2. Contoh pengaturan sinyal RAMP

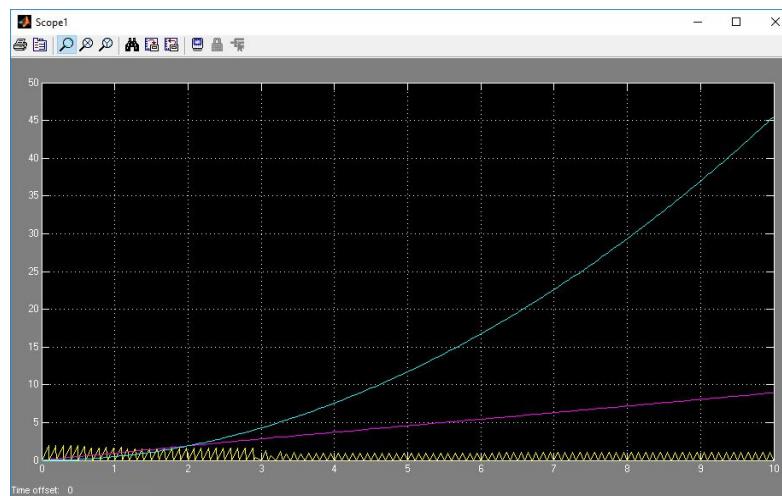
4. Membuat tool simulink seperti pada gambar dibawah ini:



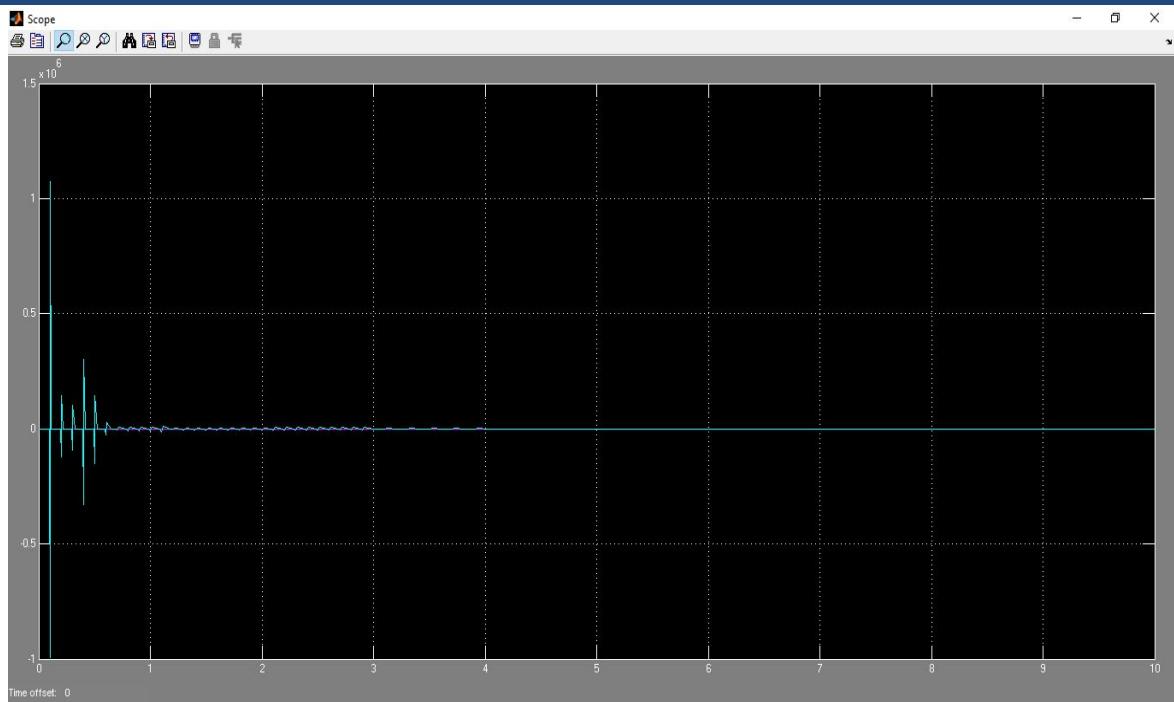
Gambar 3. LKS-2

5. Membuat fungsi input di Simulink berupa sinyal input akselerasi RANDOM dengan frekuensi antara 10Hz hingga 50 Hz

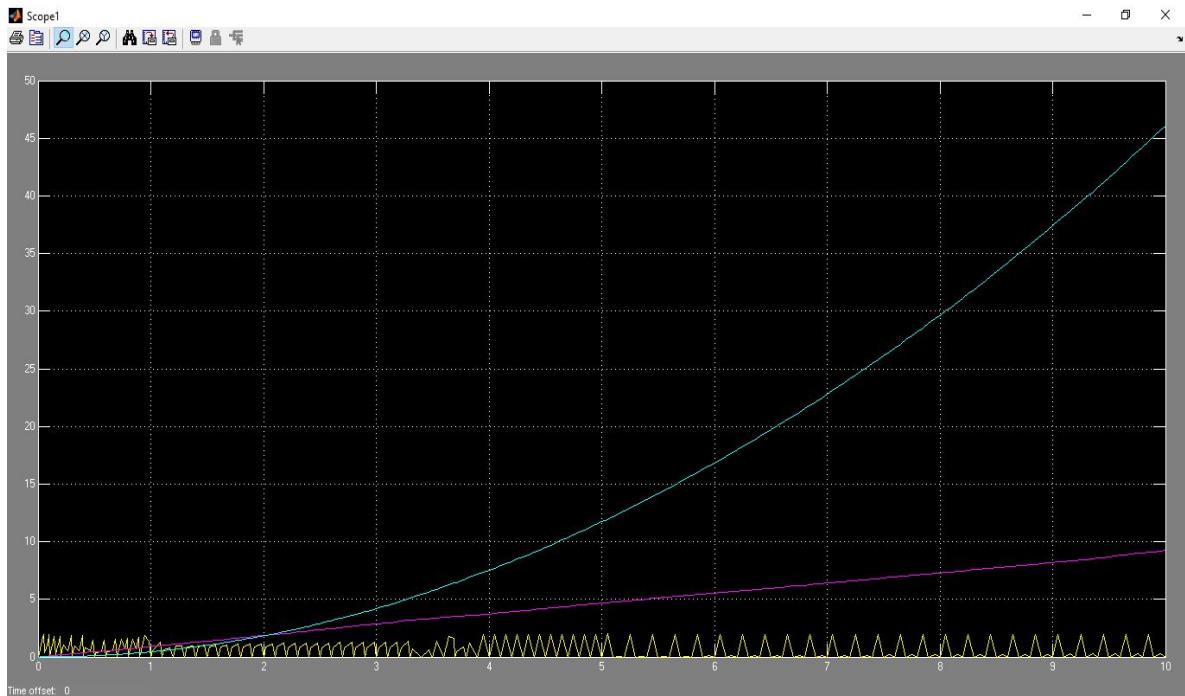
IV. DATA PERCOBAAN



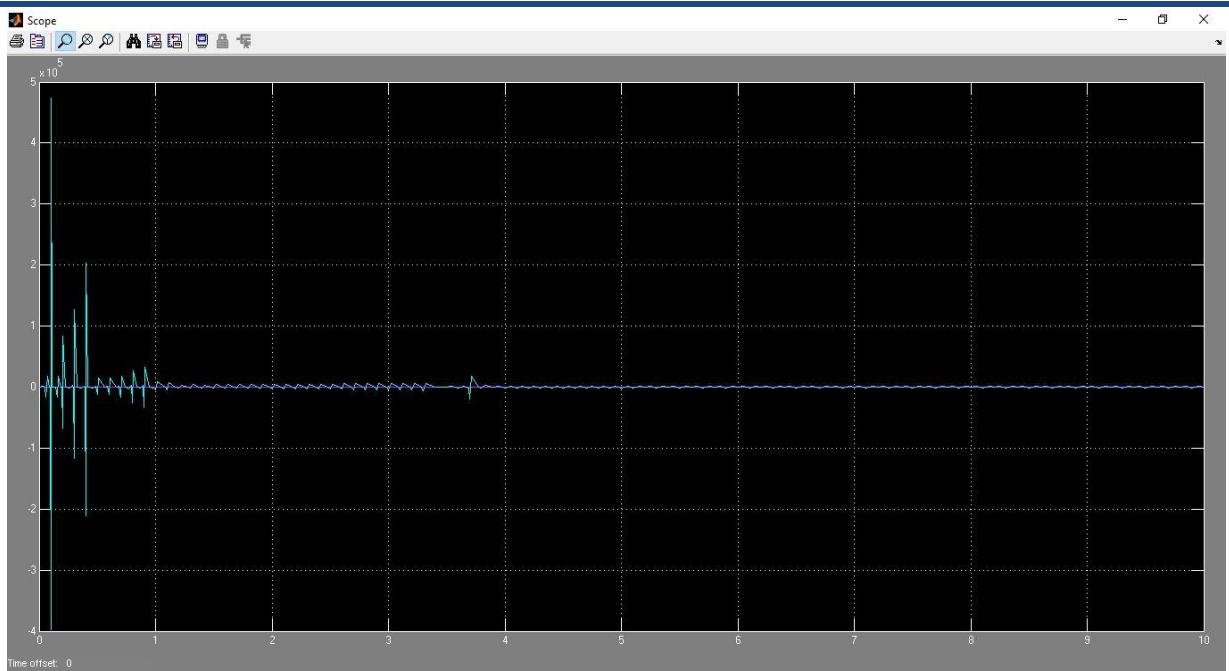
Gambar 1. Time Value = [0 0.1] Integrator



Gambar 2. Time Value = [0 0.1] Derivative



Gambar 3. Time Value = [0 0.05] Integrator



Gambar 4. Time Value = [0 0.05] Derivative

V. ANALISA

Pada praktikum kali ini membahas tentang Pengujian Konversi Sinyal Akselerasi, Kecepatan dan posisi menggunakan data definitif. Yaitu menggunakan blok repeating sequence pada simulink. Blok sequence repeating memungkinkan untuk menentukan sinyal arbitrer yang akan diulang secara periodik dari waktu ke waktu. Ketika simulasi mencapai nilai waktu maksimum dalam vektor nilai waktu, sinyal diulang.

Pada percobaan ini, dicoba blok repeating sequence dengan Time Value 0.1 & 0.5. Didapat saat $t=0.1$ diintegralkan, sinyal percepatan akan berosilasi dengan nilai maksimum 0.1. Kemudian didapatkan sinyal kecepatan akan linear, begitupula dengan sinyal posisi membentuk grafik secara eksponensial tanpa adanya osilasi.

Saat $t=0.1 \& 0.5$ masuk ke blok derivative atau diderivativekan, maka pada awalnya terdapat osilasi disekitar nilai t yang diberikan, dengan nilai maksimum osilasi sebesar $t \times 10$. Misalnya diberikan $t = 0.1$, maka osilasi maksimum sebesar $0.1 \times 10 = 1$ seperti yang dilihat ada grafik hasil percobaan.

VI. KESIMPULAN

- Blok repeating sequence dilakukan dengan bantuan simulink matlab
- Blok repeating sequence saat diintegralkan maka terjadi osilasi dengan maksimum dengan t yang diberikan.
- Blok repeating sequence saat diderivativekan maka terjadi osilasi diawal respons dengan nilai maksimum $t \times 10$.

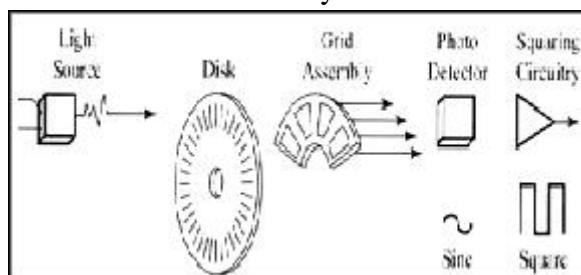
PERCOBAAN III

PENGUJIAN KONVERSI SINYAL AKSELERASI, KECEPATAN DAN POSISI MENGGUNAKAN PEMBACAAN DATA RIIL SENSOR ABSOLUTE ROTARY ENCODER

I. DASAR TEORI

Rotary encoder adalah divais elektromekanik yang dapat memonitor gerakan dan posisi. Rotary encoder umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh rotary encoder untuk diteruskan oleh rangkaian kendali. Rotary encoder umumnya digunakan pada pengendalian robot, motor drive, dsb.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau divais berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Gambar 1 menunjukkan bagan skematik sederhana dari rotary encoder. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi rotary encoder tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi rotary encoder tersebut.



Gambar 1. Blok penyusun rotary encoder

Rangkaian penghasil pulsa (Gambar 2) yang digunakan umumnya memiliki output yang berubah dari +5V menjadi 0.5V ketika cahaya diblok oleh piringan dan ketika diteruskan ke photo-transistor. Karena divais ini umumnya bekerja dekat dengan motor DC maka banyak noise yang timbul sehingga biasanya output akan dimasukkan ke low-pass filter dahulu. Apabila low-pass filter digunakan, frekuensi cut-off yang dipakai umumnya ditentukan oleh jumlah slot yang ada pada piringan dan seberapa cepat piringan tersebut berputar, dinyatakan dengan:

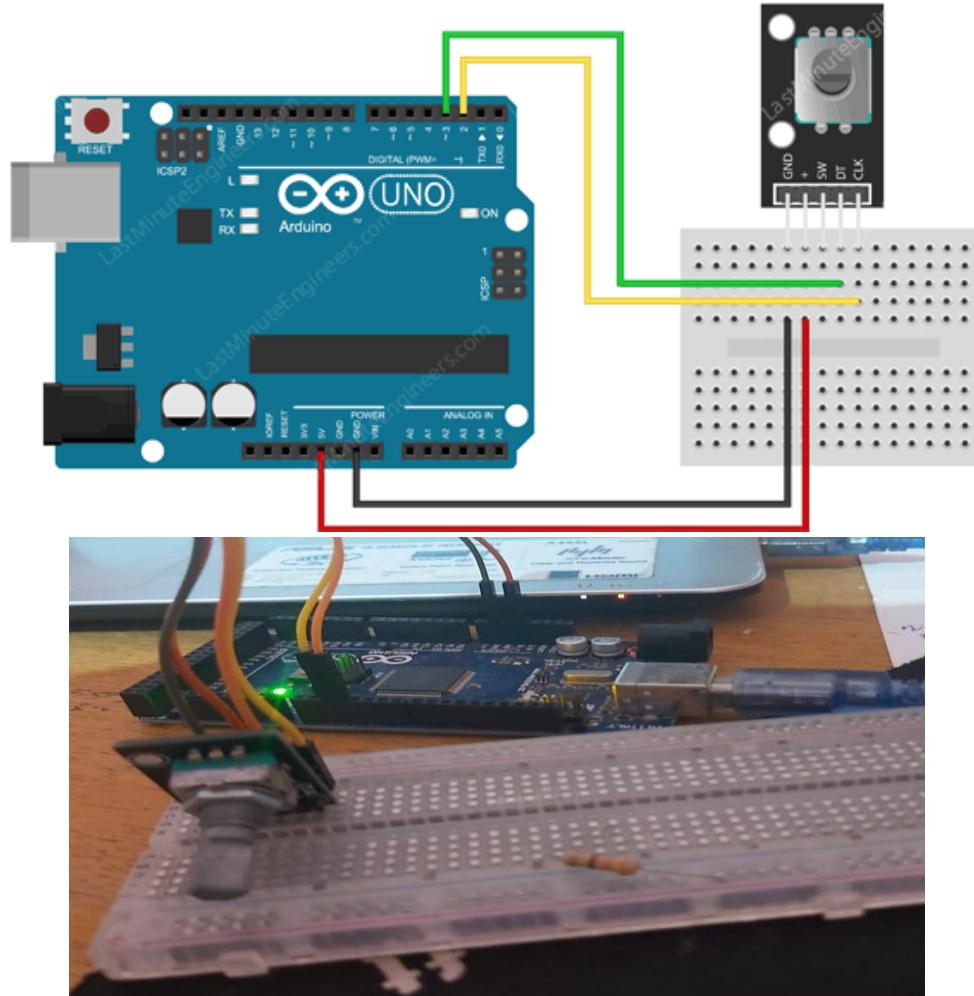
$$f_t = \frac{s_w \pi}{60}$$

II. PERALATAN

1. PC
2. Software Matlab
3. Arduino Mega 2560

4. Rotary Encoder Potensiometer
5. Software Arduino
6. Kabel Jumper
7. Bread Board

III. WIRING HARDWARE



Gambar 2. Wiring Hardware

IV. PROSEDUR

1. Menyiapkan peralatan yang dibutuhkan
2. Membuat program Rotary Encoder di Software Arduino Mega 2560:

```
#define outputA 2
#define outputB 3
int counter = 0;
int aState;
int aLastState;
void setup() {
    pinMode (outputA,INPUT);
    pinMode (outputB,INPUT);

    Serial.begin (9600);
    // Reads the initial state of the outputA
    aLastState = digitalRead(outputA);
}

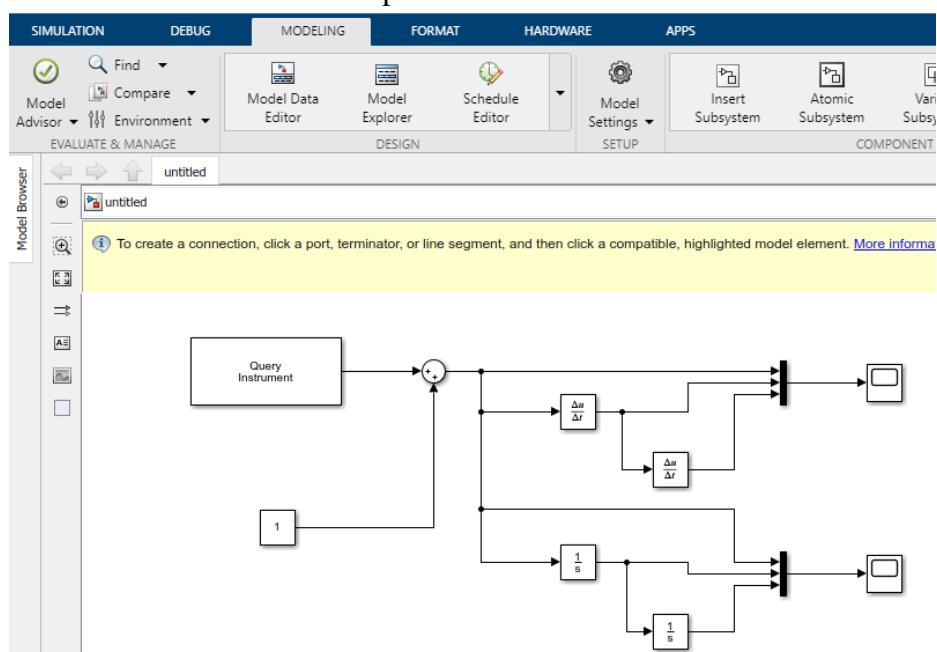
void loop() {
    aState = digitalRead(outputA); // Reads the "current" state of the outputA
    // If the previous and the current state of the outputA are different, that means
    a Pulse has occurred
```

```

if (aState != aLastState){
    // If the outputB state is different to the outputA state, that means the encoder
    // is rotating clockwise
    if (digitalRead(outputB) != aState) {
        counter++;
    } else {
        counter--;
    }
    //Serial.print("Position: ");
    Serial.println(counter);
}
aLastState = aState; // Updates the previous state of the outputA with the
current state
}

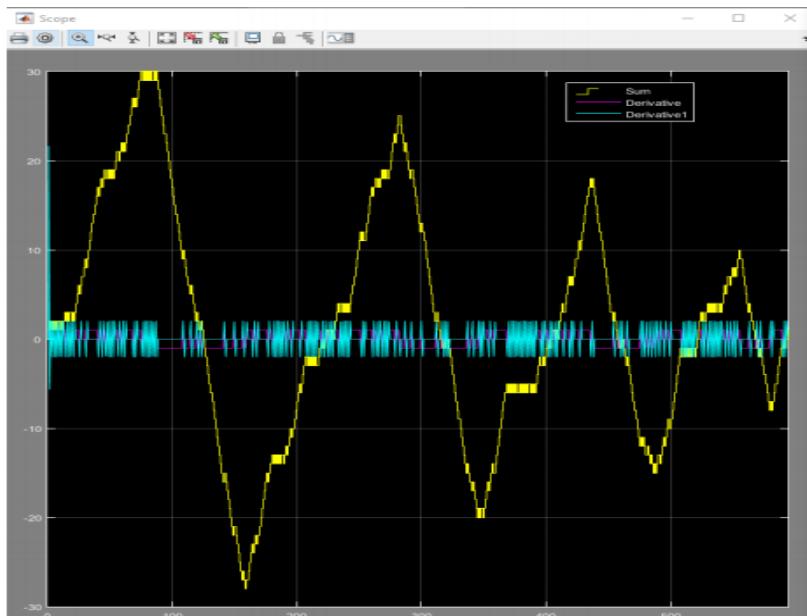
```

3. Membuat tool di Simulink seperti dibawah ini :

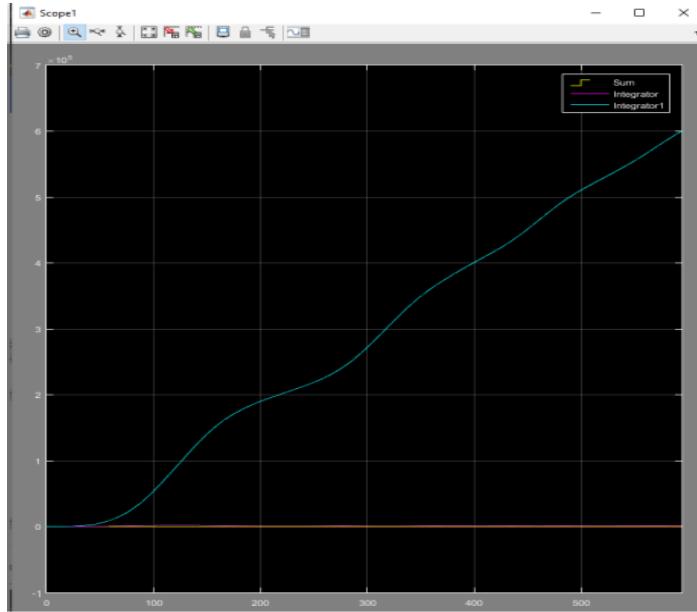


Gambar 1. Simulink Model

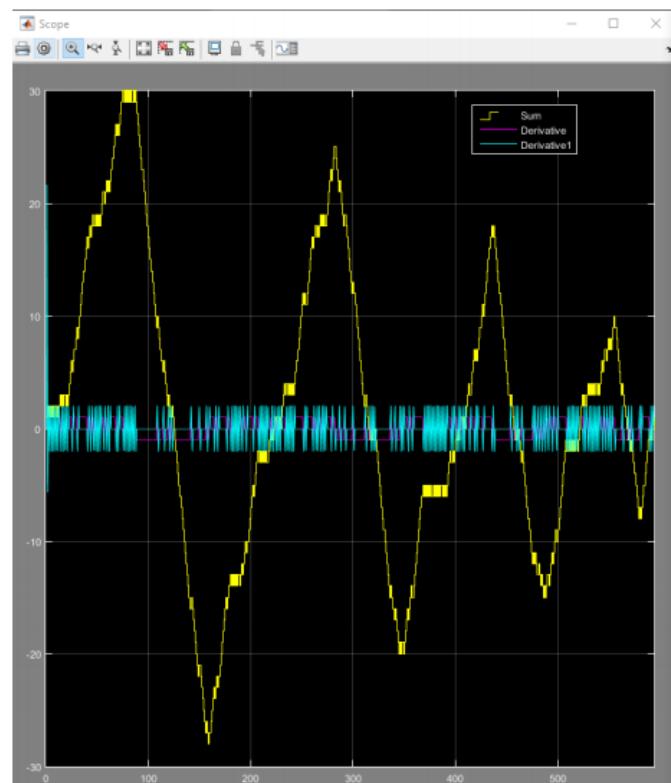
V. DATA PERCOBAAN



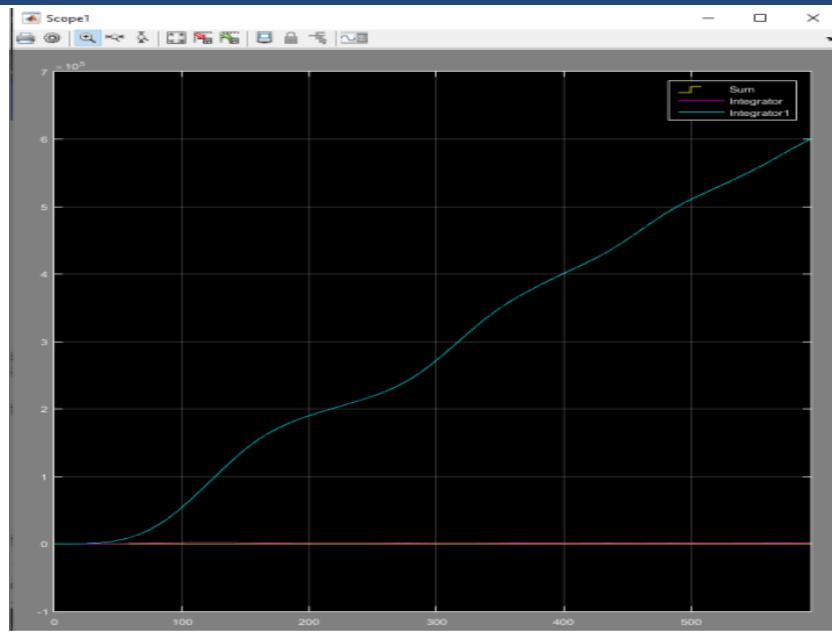
Gambar 1 pos-vel-acc (Simpangan = 40)



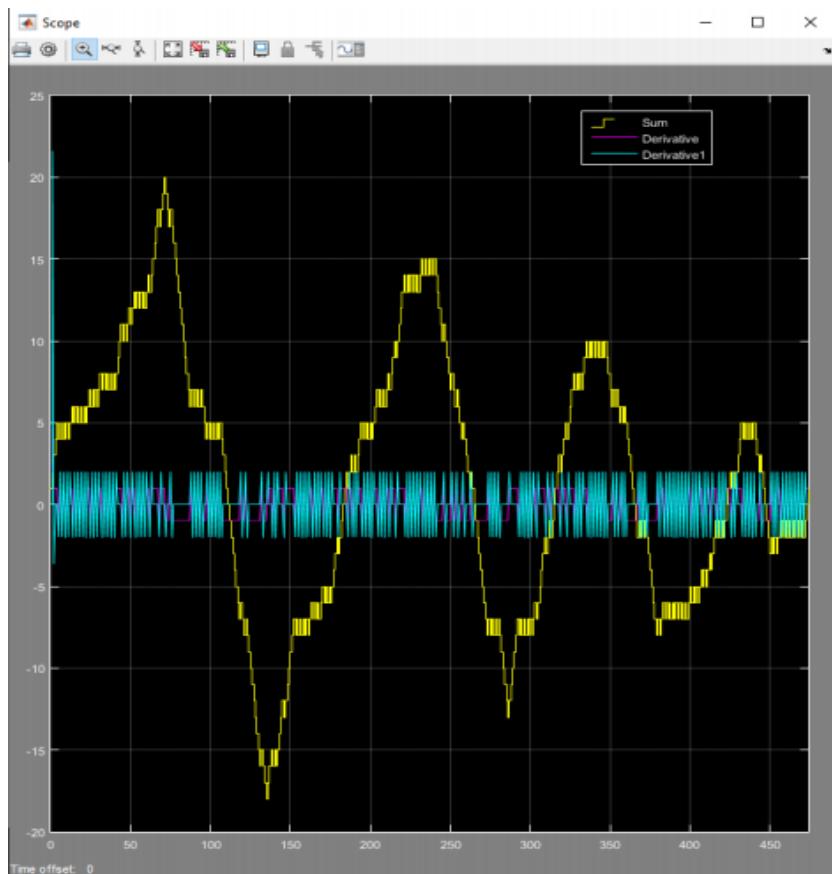
Gambar 2. Acc-vel-pos



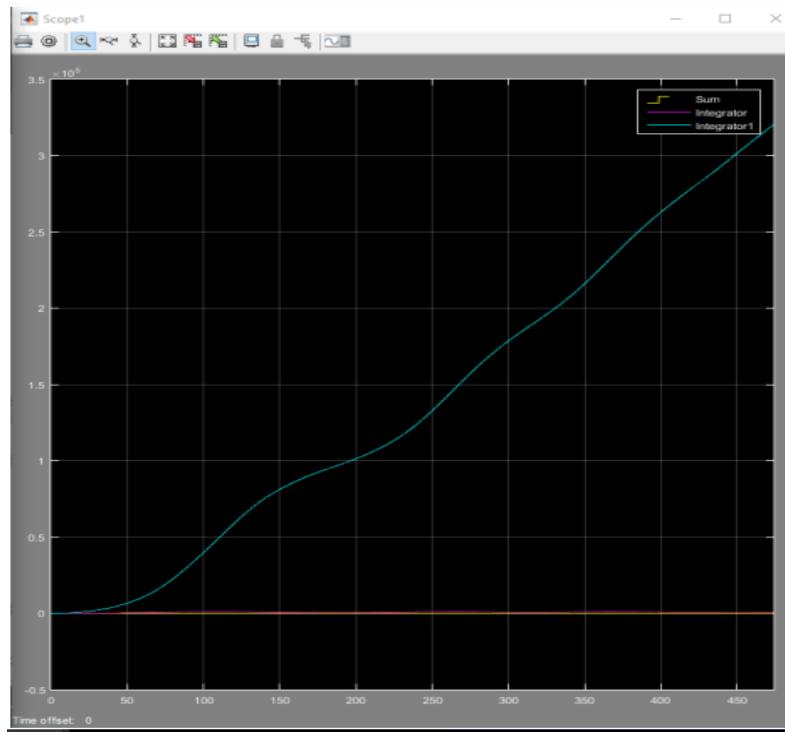
Gambar 3. Pos-vel-acc (Simpangan = 30)



Gambar 4. Acc-vel-pos



Gambar 5. Pos-vel-acc (Simpangan = 20)



Gambar 6. acc-vel-pos

VI. ANALISA

Pada praktikum ketiga mata kuliah Sensor & Aktuator kali ini membahas tentang Pengujian Konversi Sinyal Akselerasi, Kecepatan dan Posisi menggunakan Pembacaan Data Riil Sensor Absolute Rotary Encoder. Praktikum ini hampir sama dengan praktikum pertama, namun kali ini menggunakan hardware yaitu Rotary Encoder berupa potensiometer.

Pada percobaan pertama, yaitu mengetahui dan menganalisa sinyal dengan input nilai posisi berupa sinyal sinusoida. Sinyal kecepatan dari 3 nilai simpangan berosilasi secara periodik. Pada sinyal percepatan, semakin besar nilai posisinya, maka semakin rapat sinyal percepatannya. Begitupula dengan sinyal input riil yaitu sinyal posisi. Semakin besar nilai simpangannya, maka semakin rapat sinyal posisinya disetiap titik tertentu.

Pada percobaan kedua, mengetahui sinyal dengan input nilai percepatan, kemudian masuk kedalam blok derivative, yang nantinya akan menjadi sinyal kecepatan dan posisi. Didapat sinyal posisi mengikuti simpangan yang diberikan. Yaitu semakin besar simpangan yang diberikan, maka semakin besar juga nilai derivative kedua dari percepatannya.

VII. KESIMPULAN

- Simulasi praktikum menggunakan software matlab simulink
- Semakin kecil nilai simpangan yang diberikan, maka semakin rapat grafik nilai pada saat diintegralkan.
- Semakin besar nilai simpangan yang diberikan, maka semakin besar juga nilai percepatan keduanya saat diderivativekan

PRAKTIKUM IV
PENGUJIAN KONVERSI SINYAL AKSELERASI, KECEPATAN DAN POSISI
MENGGUNAKAN SENSOR IMU

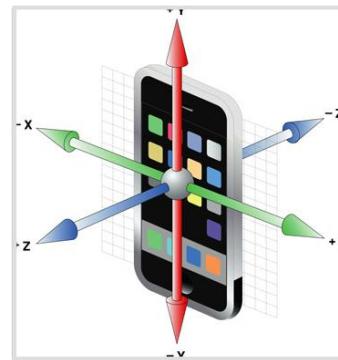
I. TUJUAN

1. Memahami konversi sinyal Akselerasi, Kecepatan dan Posisi pada Sensor IMU
2. Memahami penggunaan komponen pada Aplikasi Matlab sebagai sarana pengerjaan model Simulink.

II. DASAR TEORI

Sebuah **unit pengukuran inersia, inertial measurement unit** atau **IMU**, adalah sebuah perangkat elektronik yang mengukur dan melaporkan kecepatan kerajinan itu, orientasi, dan gaya gravitasi, menggunakan kombinasi akselerometer dan giroskop, kadang-kadang juga magnetometer. IMU biasanya digunakan untuk manuver pesawat, termasuk kendaraan udara tak berawak (UAV), antara lain banyak, dan pesawat ruang angkasa, termasuk satelit dan pendarat. Perkembangan terkini memungkinkan untuk produksi perangkat GPS IMU-enabled. Sebuah IMU memungkinkan penerima GPS untuk bekerja ketika GPS-sinyal tidak tersedia, seperti di terowongan, di dalam bangunan, atau ketika interferensi elektronik hadir. Sebuah IMU nirkabel dikenal sebagai WiMu.

IMU merupakan komponen utama dari sistem navigasi inersia digunakan di pesawat terbang, pesawat ruang angkasa, perahu, dan rudal



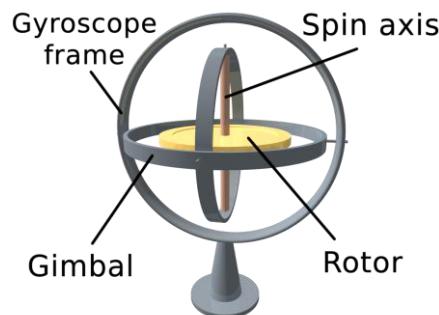
Gambar (a) Akselerometer.

Accelerometer adalah sebuah perangkat yang mengukur percepatan yang tepat . Hal ini tidak selalu sama dengan percepatan koordinat (perubahan kecepatan dari perangkat dalam ruang), tetapi agak jenis percepatan terkait dengan fenomena berat badan yang dialami oleh massa uji yang berada dalam kerangka acuan dari perangkat accelerometer . Untuk contoh di mana jenis percepatan berbeda, accelerometer akan mengukur nilai saat duduk di tanah, karena massa ada bobot, meskipun mereka tidak mengubah kecepatan. Namun, accelerometer di gravitasi jatuh bebas ke arah pusat bumi akan mengukur nilai nol karena, meskipun kecepatan meningkat, berada dalam kerangka acuan di mana ia ringan .

Dengan mengukur berat badan, accelerometer mengukur percepatan jatuh bebas kerangka referensi (kerangka acuan inersial) relatif terhadap dirinya sendiri (accelerometer). Accelerometers kebanyakan tidak menampilkan nilai mereka mengukur, tetapi pasokan ke perangkat lain. Accelerometers nyata juga memiliki keterbatasan praktis dalam seberapa cepat mereka menanggapi perubahan dalam percepatan, dan tidak dapat merespon perubahan atas perubahan frekuensi tertentu.

Model tunggal dan multi-sumbu accelerometer yang tersedia untuk mendeteksi besar dan arah percepatan yang tepat (atau g-force), sebagai vektor kuantitas, dan dapat digunakan untuk orientasi akal (karena arah perubahan berat badan), percepatan koordinat (asalkan menghasilkan g-force atau perubahan g-force), getaran, guncangan, dan jatuh (kasus di mana perubahan percepatan yang tepat, karena cenderung menuju nol). Micromachined accelerometers semakin hadir di perangkat elektronik portabel dan video pengendali permainan, untuk mendeteksi posisi perangkat atau memberikan masukan permainan.

Pasangan accelerometers diperpanjang atas wilayah ruang dapat digunakan untuk mendeteksi perbedaan (gradien) dalam percepatan yang tepat dari frame referensi yang terkait dengan poin. Alat ini disebut gradiometers gravitasi, karena mereka mengukur gradien di medan gravitasi. Pasangan seperti accelerometers dalam teori juga dapat mendeteksi gelombang gravitasi .



Gambar (b) Giroskop

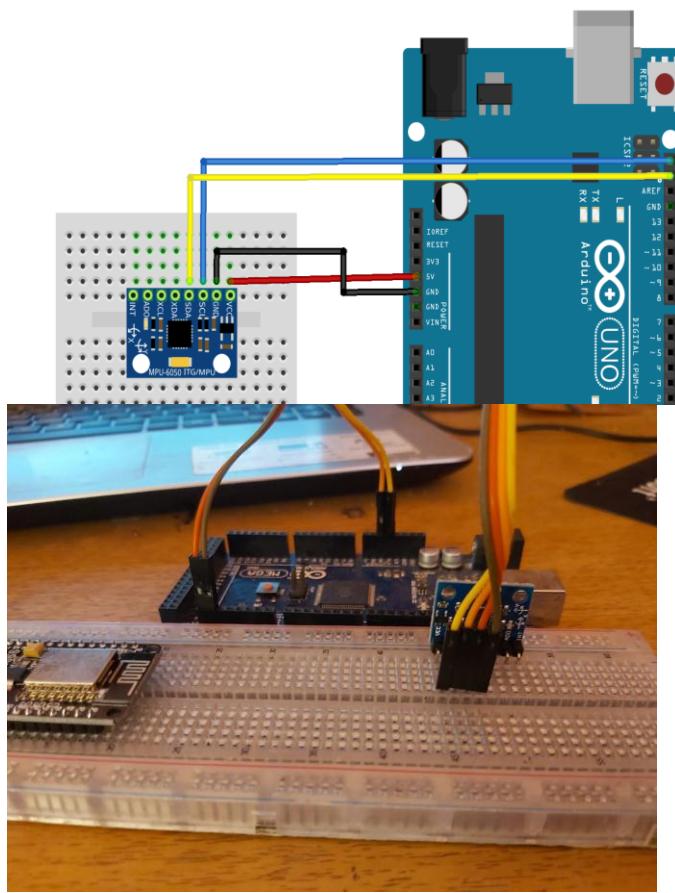
Pada dasarnya, sebuah giroskop mekanik adalah roda berputar atau disk yang berporos bebas untuk mengambil setiap orientasi. Meskipun orientasi ini tidak tetap, perubahan dalam respon terhadap eksternal torsi jauh lebih sedikit dan dalam arah yang berbeda dari itu akan tanpa momentum sudut besar yang terkait dengan tingkat tinggi dari disk berputar dan momen inersia . Karena torsi eksternal diminimalkan dengan mem-mount perangkat di gimbal, orientasi masih hampir tetap, terlepas dari setiap gerak dari platform yang sudah terpasang.

Giroskop berdasarkan prinsip-prinsip operasi lain juga ada, seperti, elektronik microchip-paket MEMS giroskop perangkat yang ditemukan dalam perangkat konsumen elektronik, solid-state laser cincin, giroskop serat optik, dan sangat sensitif giroskop kuantum . Aplikasi giroskop termasuk sistem navigasi inersia mana kompas magnetik tidak akan bekerja (seperti dalam teleskop Hubble) atau tidak akan cukup tepat (seperti dalam ICBM), atau untuk stabilisasi kendaraan terbang seperti radio-dikontrol helikopter atau kendaraan udara tak berawak . Karena presisi tinggi mereka, giroskop juga digunakan untuk menjaga arah dalam terowongan pertambangan.

III. PERALATAN

- Sensor IMU GY-521
- Software Matlab
- PC
- Arduino Mega 2560
- Breadboard
- Kabel Jumper 4 buah

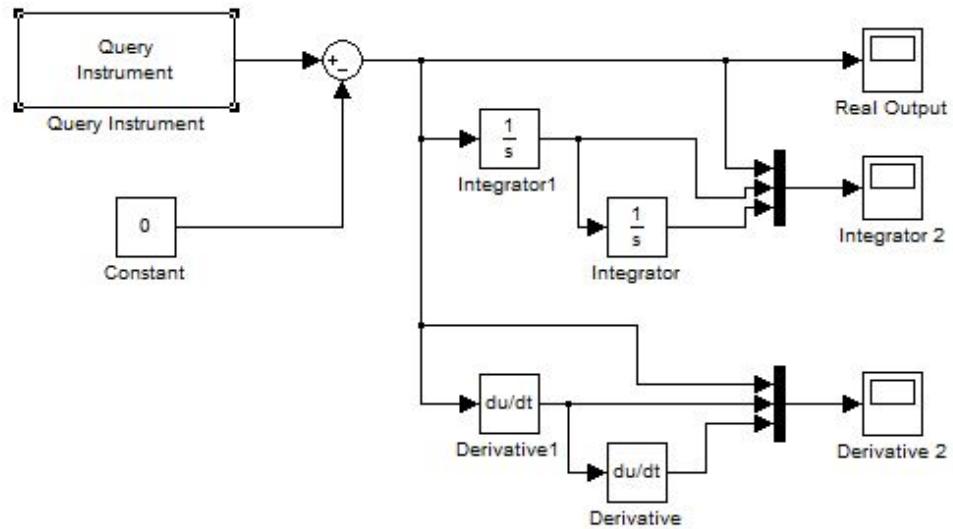
IV. WIRING HARDWARE



Gambar (c) Wiring Hardware Arduino dengan Sensor IMU GY-521

V. HASIL PERCOBAAN

2.1. Model Simulink



```

SENSORIMU | Arduino 1.8.12
File Edit Sketch Tools Help
SENSORIMU §
Wire.beginTransmission(MPU);
Wire.write(0x3B);
Wire.endTransmission(false);
Wire.requestFrom(MPU, 12, true);
AcX=Wire.read()<<8|Wire.read();
AcY=Wire.read()<<8|Wire.read();
AcZ=Wire.read()<<8|Wire.read();
GyX=Wire.read()<<8|Wire.read();
GyY=Wire.read()<<8|Wire.read();
GyZ=Wire.read()<<8|Wire.read();

//Serial.print("Accelerometer: ");
//Serial.print("X = ");
//Serial.println(AcX);
//Serial.print(" | Y = ");
//Serial.println(AcY);
//Serial.print(" | Z = ");
//Serial.println(AcZ);

Invalid library found in C:\Users\user\Documents\Arduino\lib

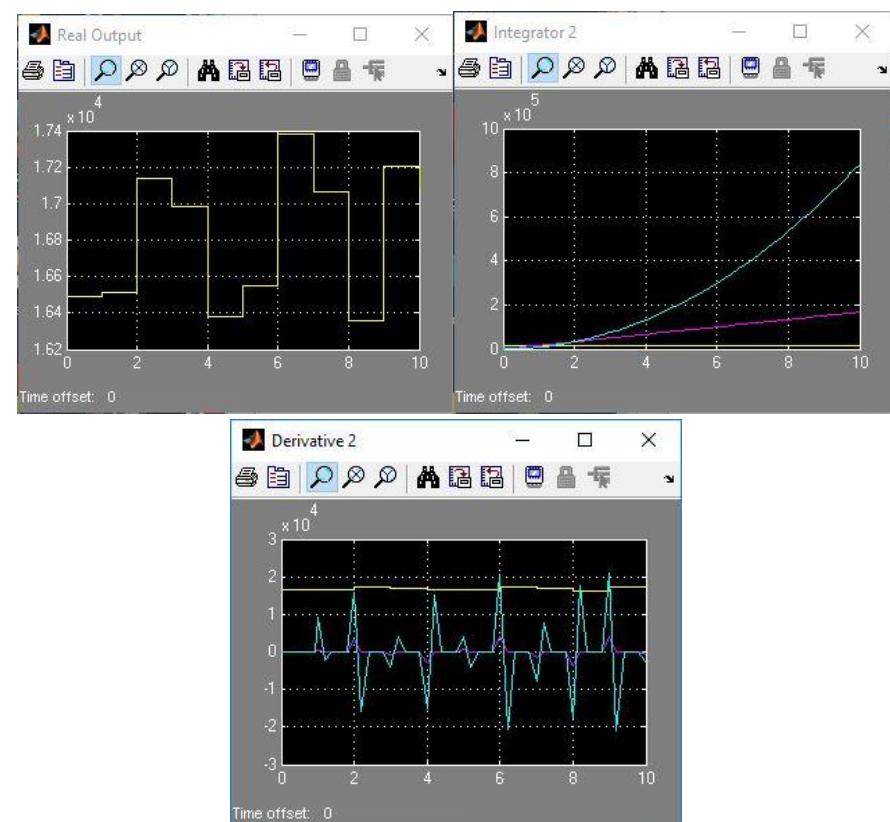
```

Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM2

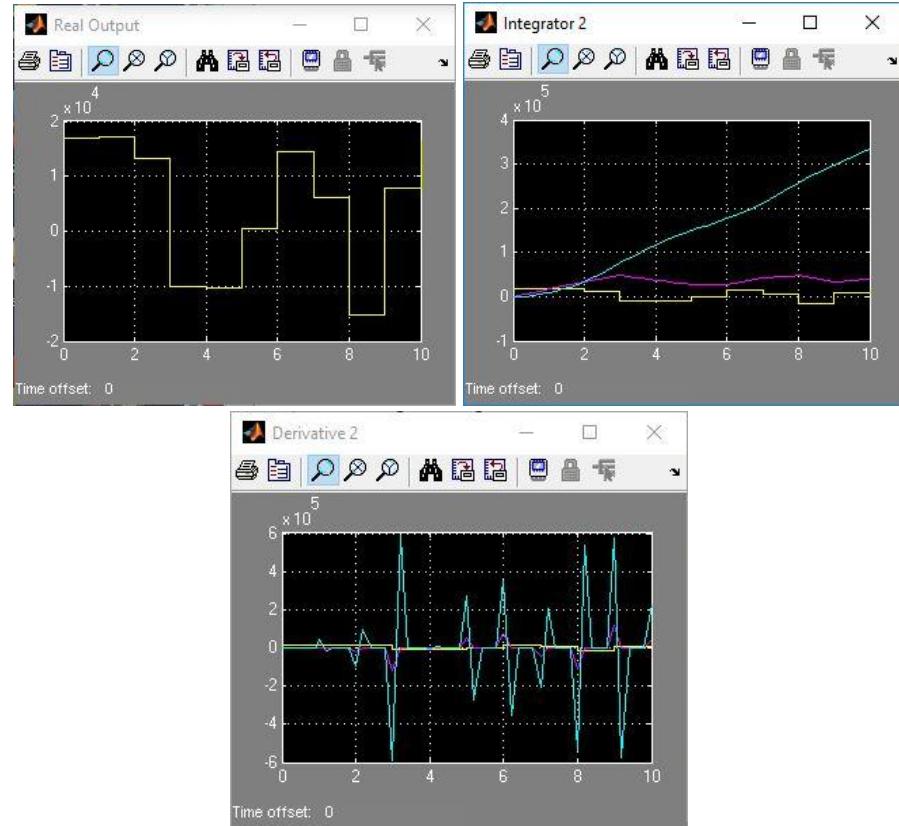
Gambar 1. Model Simulink Sensor 6-DoF & Program Arduino

2.2 Hasil Sensor Accelerometer

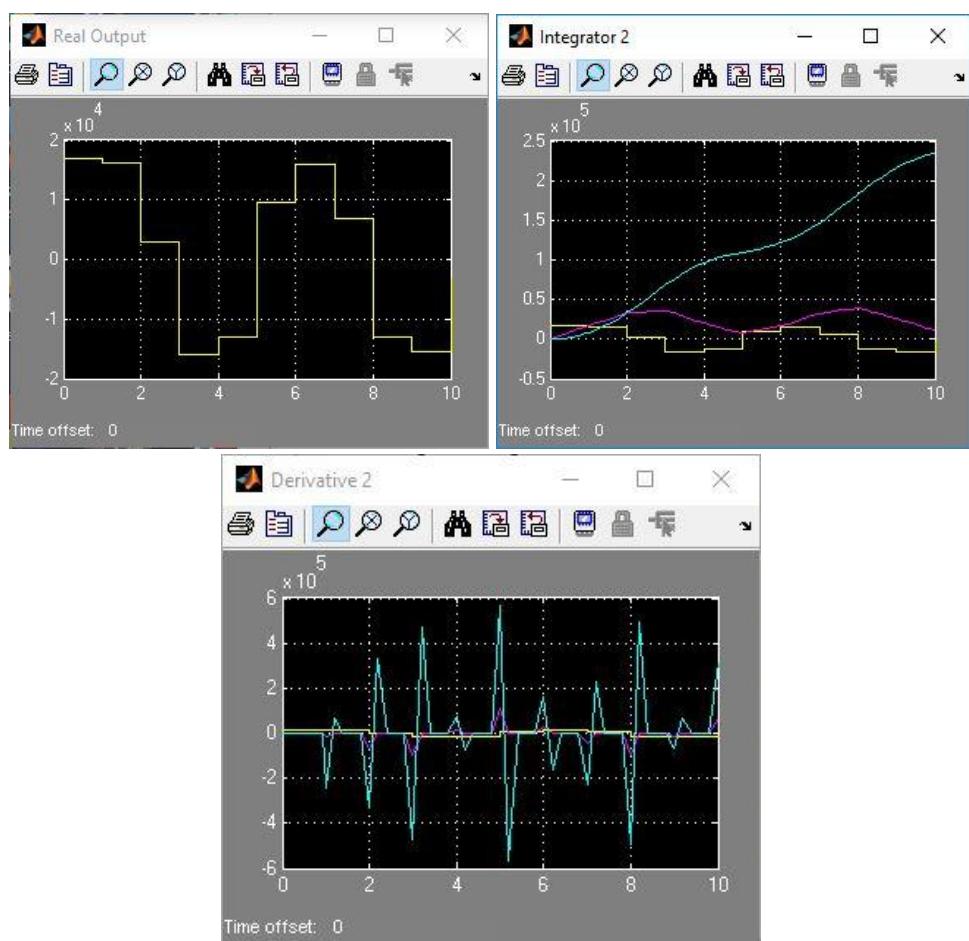
★ AcX



Gambar 2. Grafik sinyal Akselerasi AcX dirotasi pada sumbu X 2x

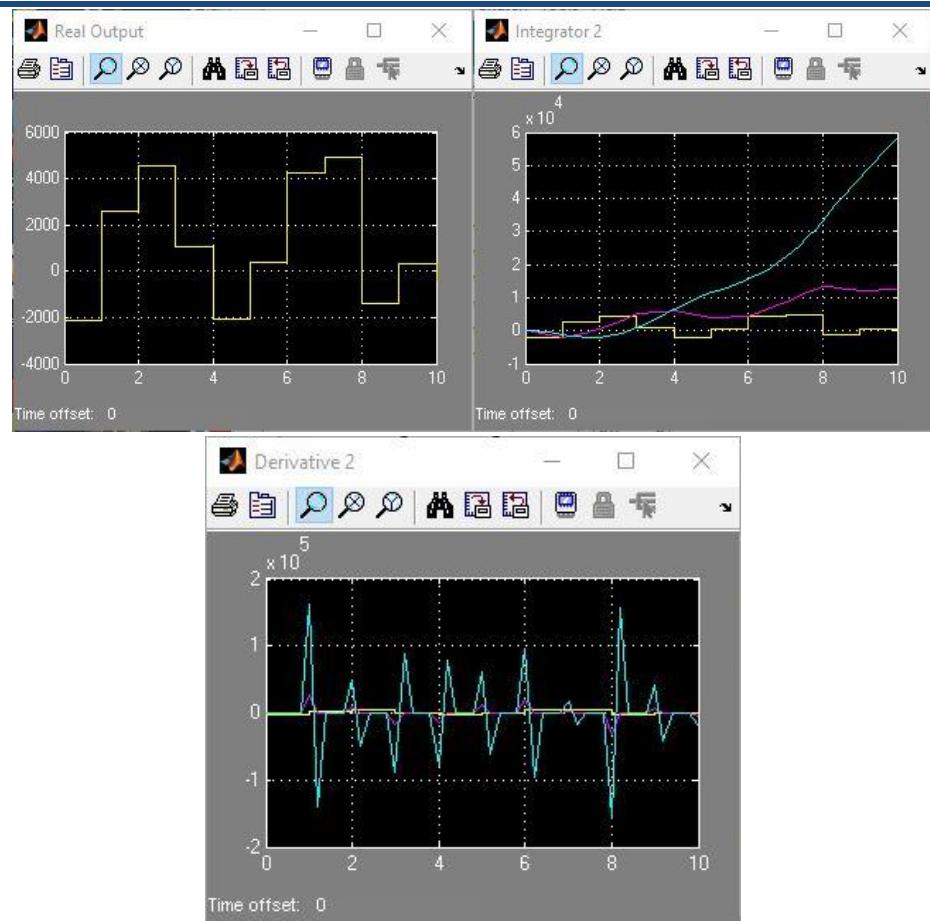


Gambar 3. Grafik sinyal Akselerasi AcX dirotasi pada sumbu Y 2x

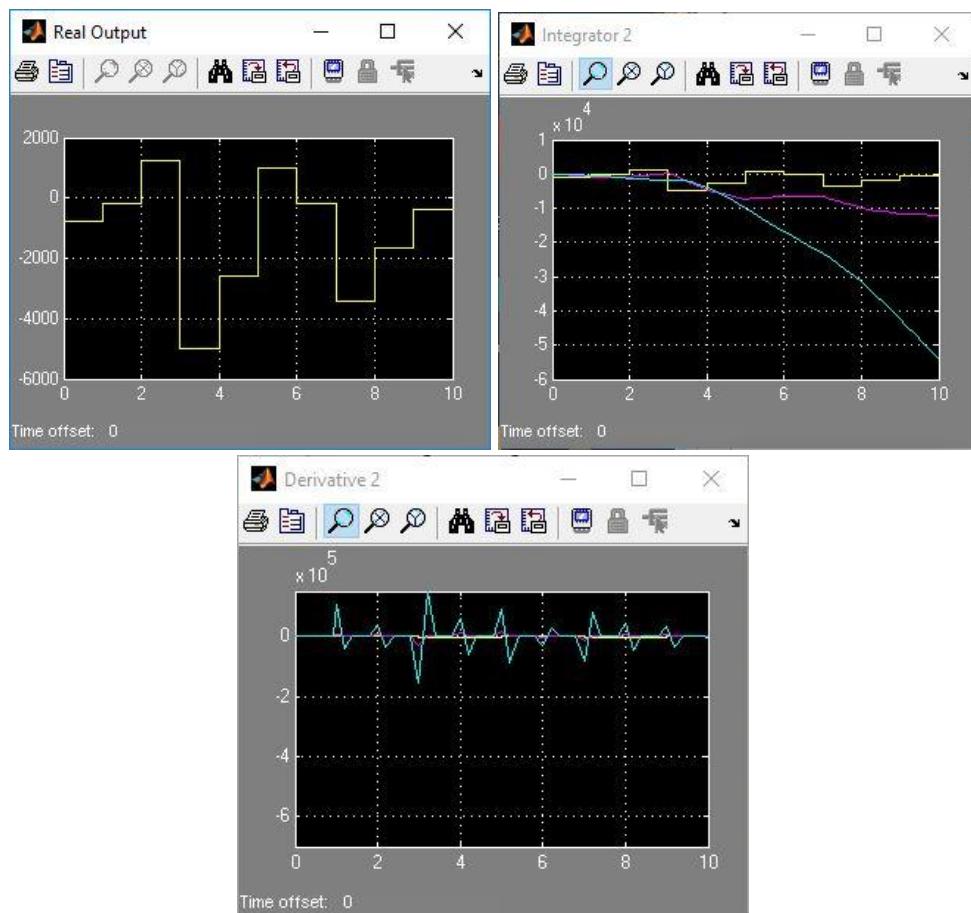


Gambar 4. Grafik sinyal Akselerasi AcX dirotasi pada sumbu Z 2x

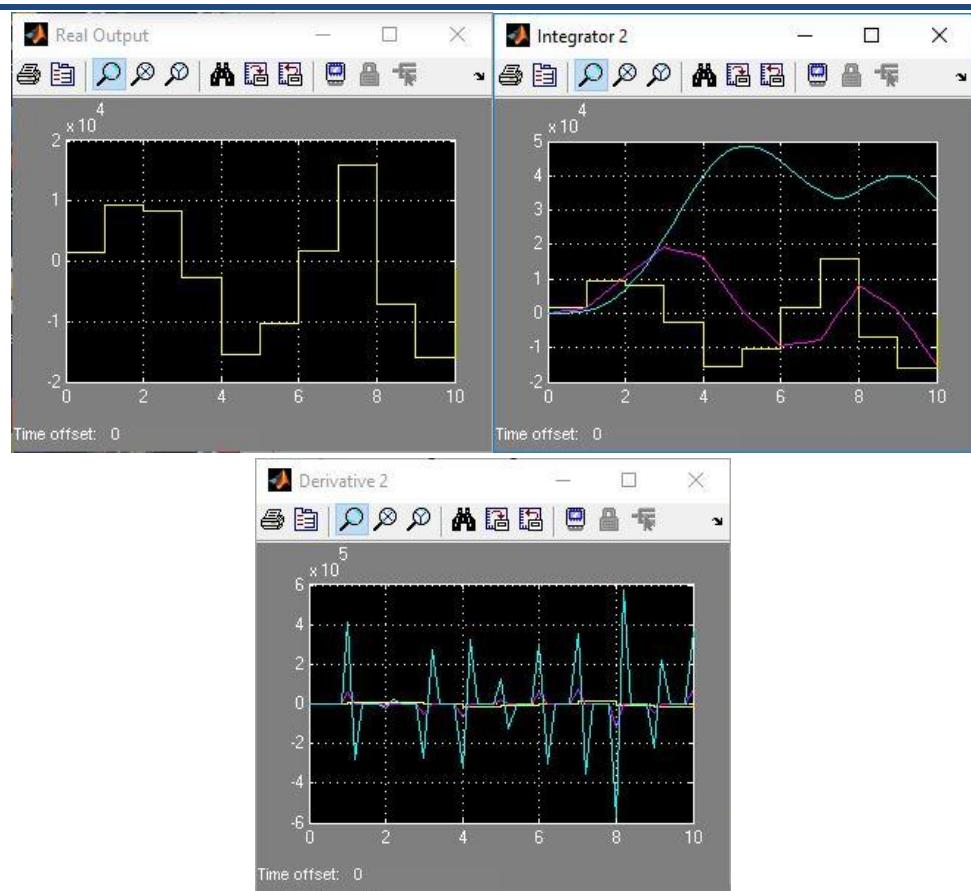
★ AcY



Gambar 5. Grafik sinyal Akselerasi AcY dirotasi pada sumbu X 2x

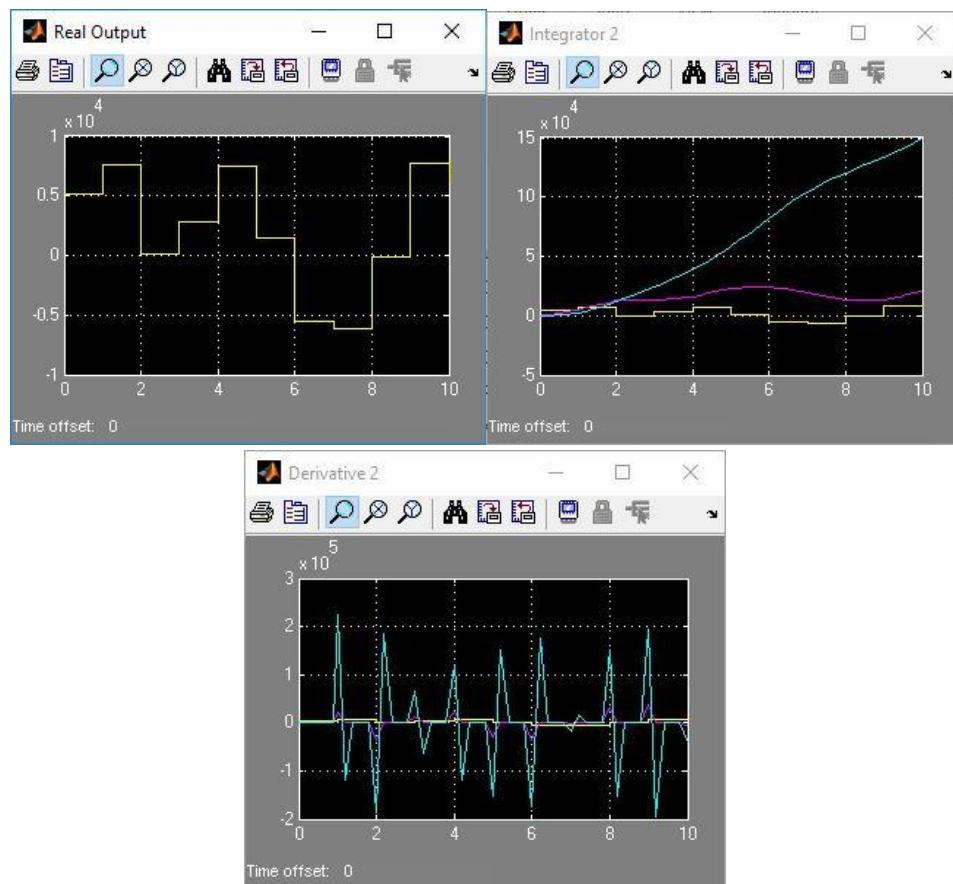


Gambar 6. Grafik sinyal Akselerasi AcY dirotasi pada sumbu Y 2x

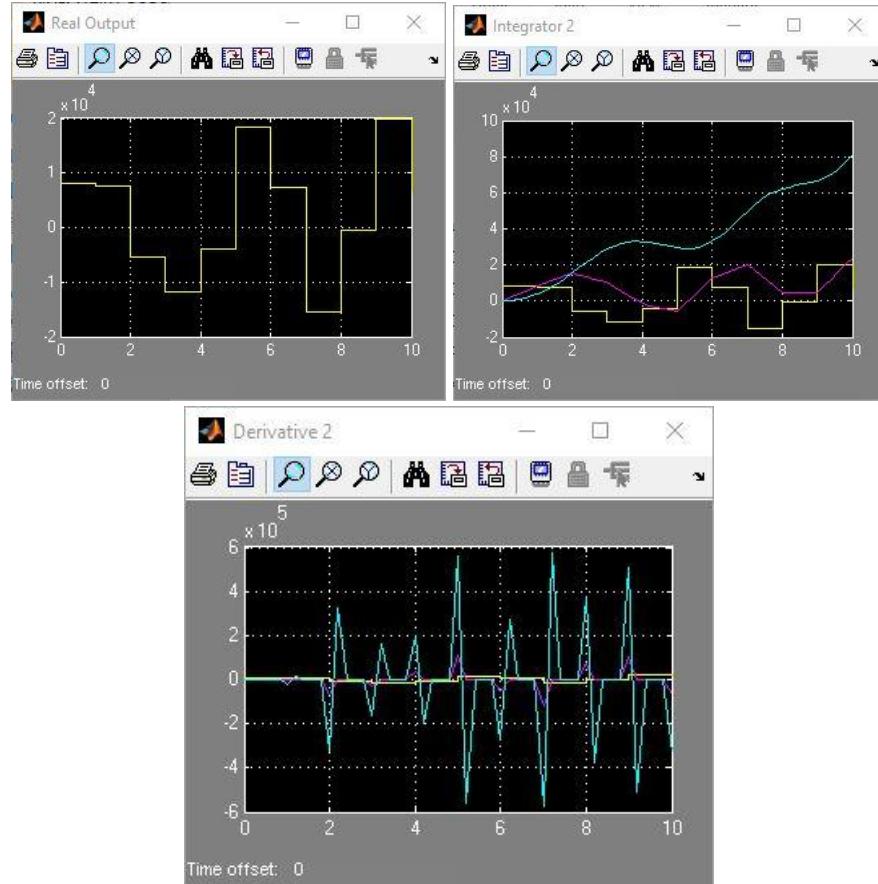


Gambar 7. Grafik sinyal Akselerasi AcY dirotasi pada sumbu Z 2x

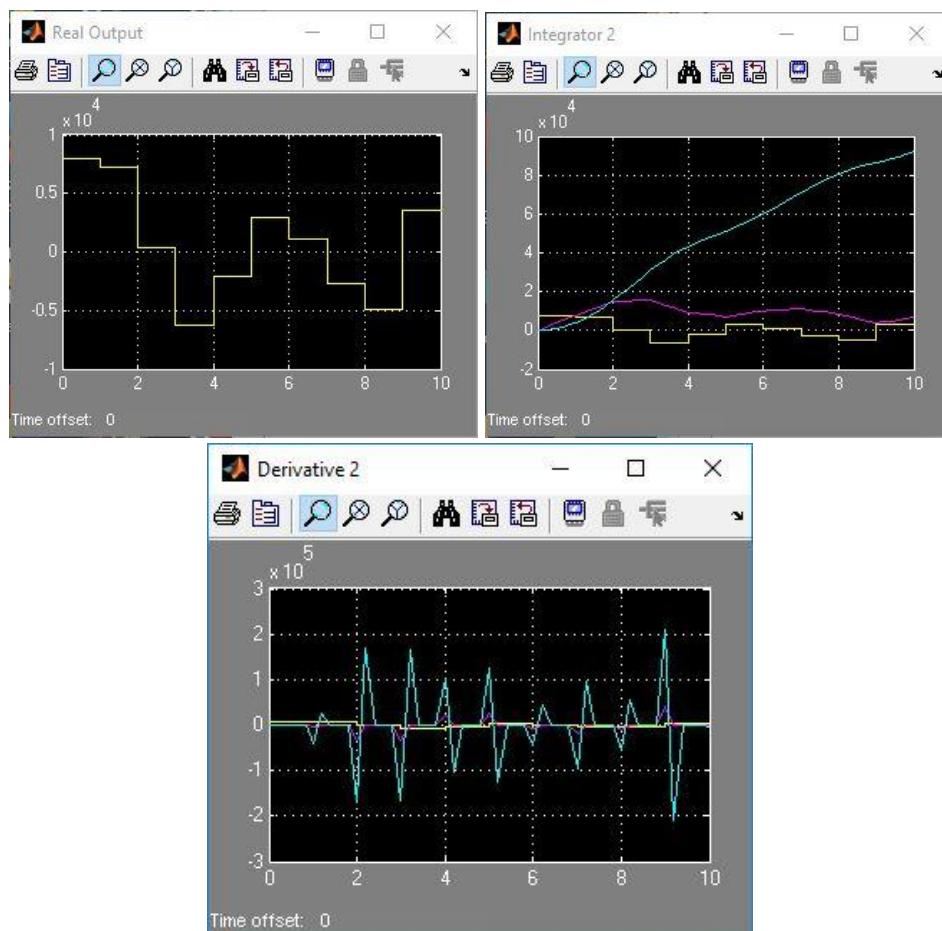
★ AcZ



Gambar 8. Grafik sinyal Akselerasi AcZ dirotasi pada sumbu X 2x

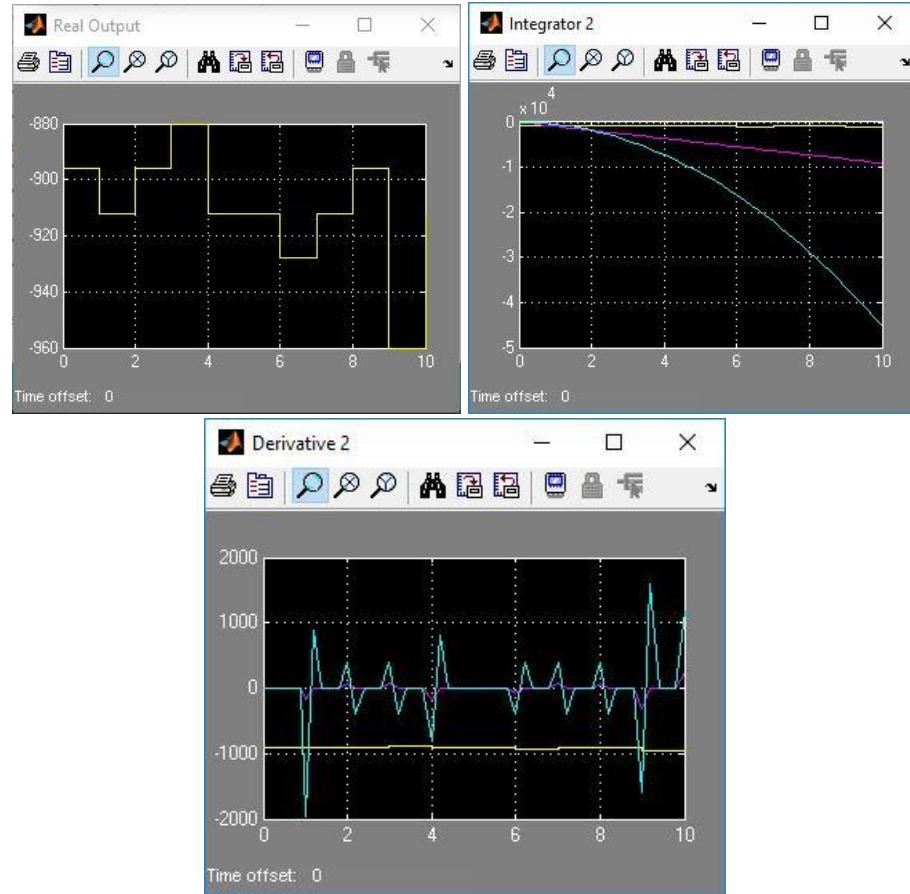


Gambar 9. Grafik sinyal Akselerasi AcZ dirotasi pada sumbu Y 2x



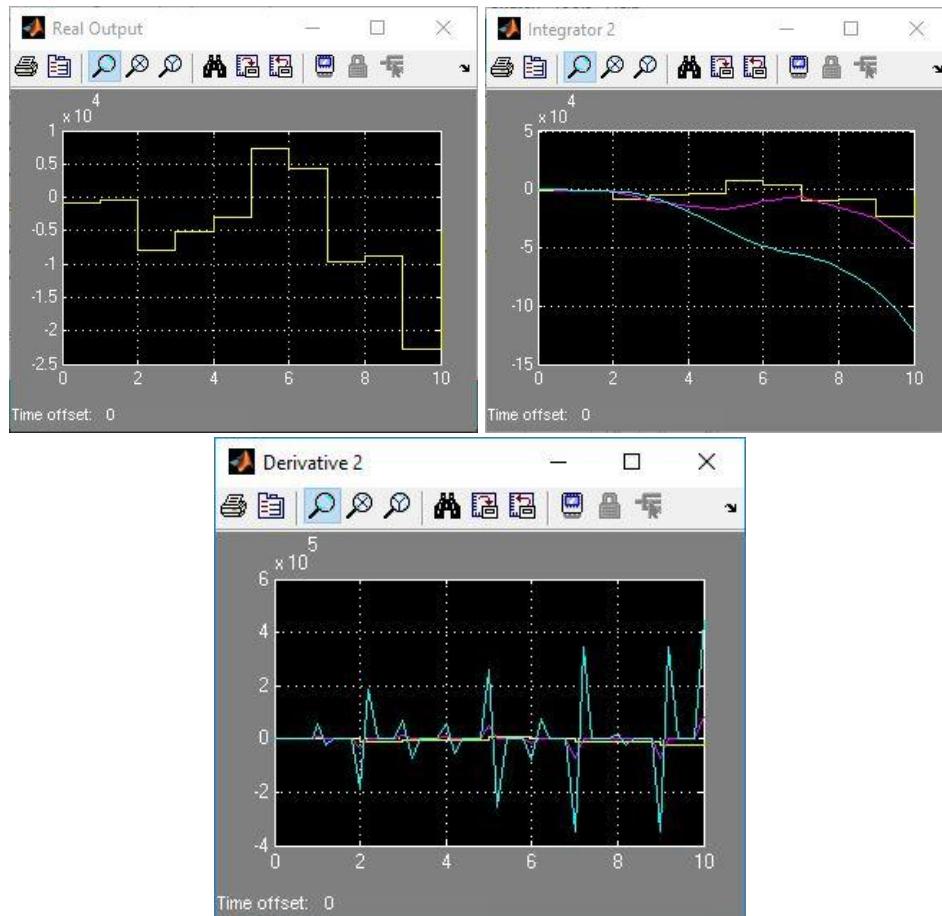
Gambar 10. Grafik sinyal Akselerasi AcZ dirotasi pada sumbu Z 2x

❖ Hasil Sensor Gyrometer
★ GyX



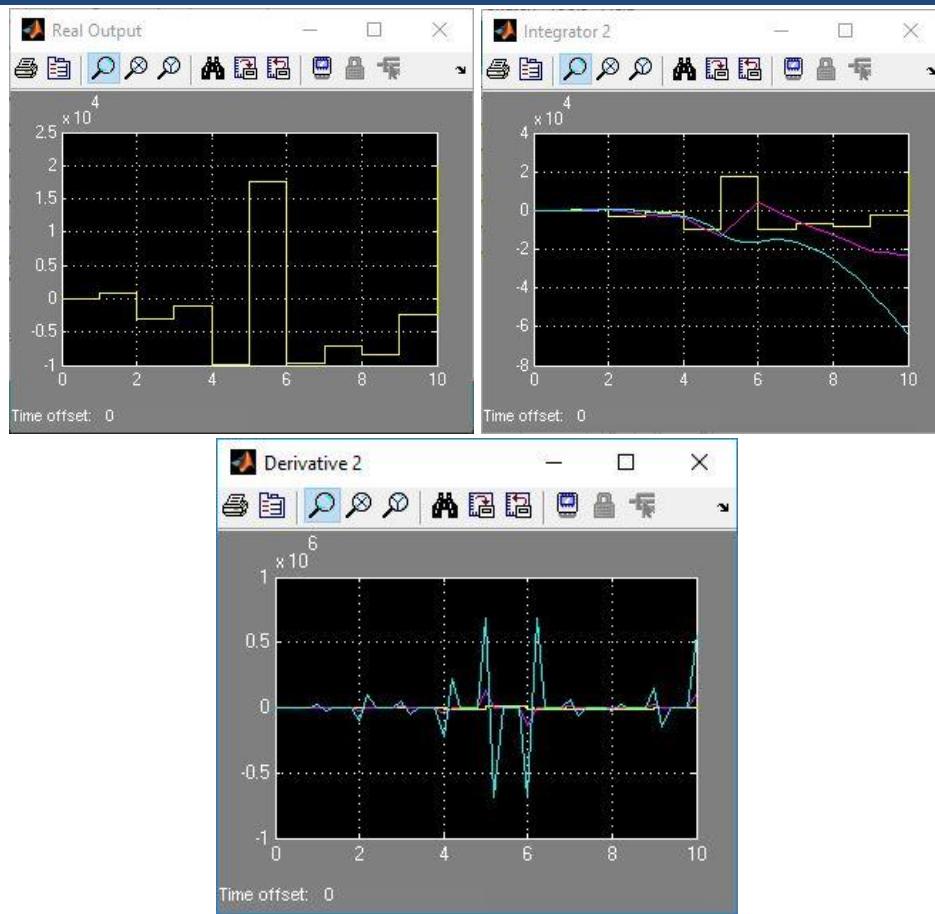
Gambar 10. Grafik sinyal Gyrometer pada sumbu X yang bergerak dengan kecepatan tertentu

★ GyY



Gambar 11. Grafik sinyal Gyrometer pada sumbu Y yang bergerak dengan kecepatan tertentu

★ GyZ

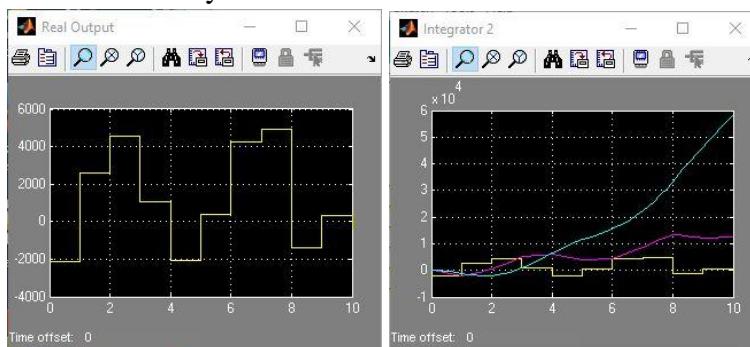


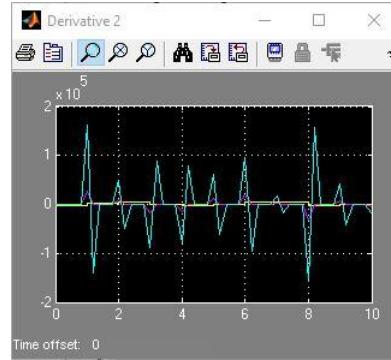
Gambar 12. Grafik sinyal Gyrometer pada sumbu Z yang bergerak dengan kecepatan tertentu

VI. ANALISA

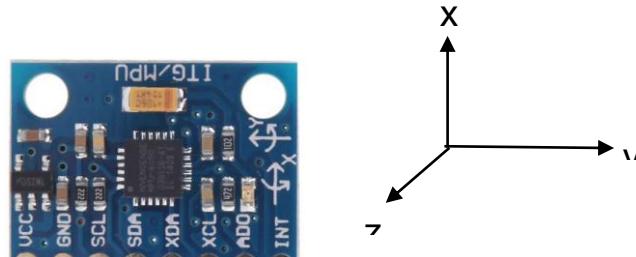
Pada praktikum mata kuliah Sensor & Aktuator kali ini membahas tentang Pengujian Konversi Sinyal Akselerasi, Kecepatan dan Posisi menggunakan Sensor IMU GY-521 6-DoF. Sensor IMU mempertahankan 6 Degree of Freedom (DoF) yang dapat memperkirakan gerakan yaitu posisi (X Y Z) dan orientasi (roll, pitch dan yaw). Untuk mendapatkan hasil pengujian sinyal diperlukan software Matlab, tool yang digunakan adalah Simulink, untuk membuat model pengujian.

Pada percobaan pertama adalah pengujian Sensor Akselerometer ketika dirotasi 2x terhadap setiap sumbu. Dapat dilihat pada contoh pengujian dibawah bahwa grafik sinyal real output dari sensor IMU adalah 2 gelombang yang menandakan bahwa 2x perotasi terhadap sumbu X. Kemudian pada grafik selanjutnya, ketika sinyal real output diberi integrator juga terlihat memiliki 2 gelombang. Namun hal berbeda ketika diberi integrator untuk kedua kalinya.





Pada percobaan kedua adalah pengujian giroskop yaitu pengujian saat sensor IMU digerakkan dengan percepatan tertentu berdasarkan sumbunya.



Pada grafik sinyal pengujian Giroskop terlihat bahwa sinyal Real output ketika diberi Integrator akan turun atau negatif.

VII. KESIMPULAN

- ★ Sinyal ‘Real Output’ adalah representasi dari Sinyal nyata hasil dari pergerakan oleh Sensor IMU
- ★ Ketika sinyal Real Akselerasi diberi integrator 1x, maka sinyal akan tetap mengikuti sinyal sebelumnya (Real) namun berbeda frekuensi
- ★ Ketika sinyal Akselerasi Real diberi integrator 2x, maka sinyal terus naik/turun, dengan sedikit mengikuti sinyal sebelumnya (Real)
- ★ Sinyal Integrator dari Gyroscope akan terus bernilai negatif.

PRAKTIKUM V

PENGUJIAN FLDIGI UNTUK APLIKASI MODULASI-DEMODULASI DATA TEKS DALAM RF-TRANSMISSION

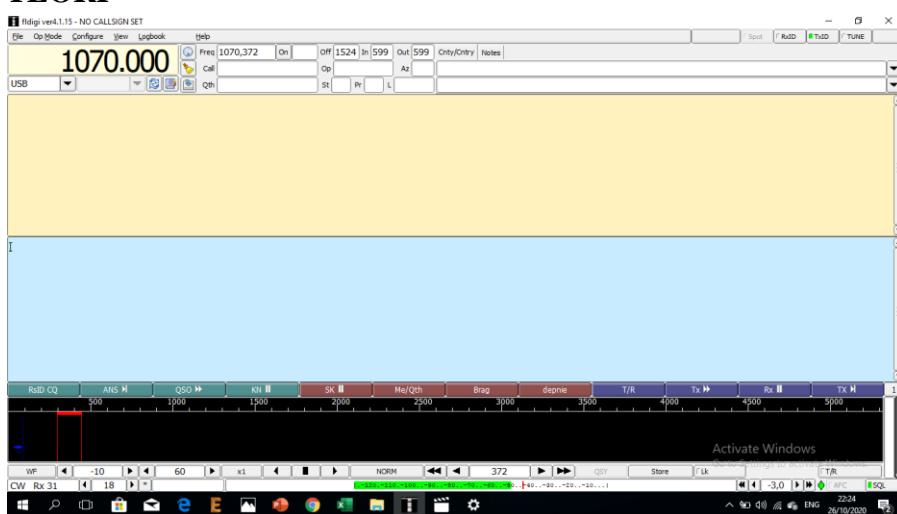
I. TUJUAN

- Memahami cara menggunakan komponen-komponen dalam aplikasi FLDIGI
- Mengamati output berupa data maupun sinyal dengan masukan suara dari Microphone

II. PERALATAN

- PC
- Headphone

III. DASAR TEORI



Gambar 1. Tampilan awal Fldigi

Fldigi (kependekan dari **F**ast **l**ight **d**igital) adalah program gratis dan bersumber terbuka yang memungkinkan kartu suara komputer biasa digunakan sebagai modem data dua arah sederhana. Perangkat lunak ini sebagian besar digunakan oleh operator radio amatir yang menghubungkan mikrofon dan koneksi headphone dari transceiver SSB radio amatir atau radio dua arah FM ke koneksi headphone dan mikrofon komputer.

Interkoneksi ini menciptakan radio yang ditentukan kartu suara yang bandwidth yang tersedia dibatasi oleh frekuensi sampel kartu suara dan bandwidth radio eksternal.

Komunikasi semacam itu biasanya dilakukan pada gelombang radio amatir gelombang pendek dalam mode seperti PSK31 , MFSK , RTTY , Olivia , dan CW (kode morse) . Perangkat lunak ini juga semakin digunakan untuk data pada frekuensi VHF dan UHF menggunakan mode yang lebih cepat seperti 8-PSK .Dengan menggunakan perangkat lunak ini, operator radio amatir dapat berkomunikasi di seluruh dunia dengan hanya menggunakan beberapa watt daya RF.

Perangkat lunak Fldigi juga digunakan untuk komunikasi darurat radio amatir ketika sistem komunikasi lain gagal karena bencana alam atau pemadaman listrik. Transfer file, email, dan formulir FEMA ICS dimungkinkan dengan menggunakan perangkat keras radio yang tidak mahal. Pada aplikasi Fldigi terdapat beberapa mode yaitu :

3.1. CW (Kode Morse)

CW menggunakan Kode Q, singkatan-singkatan dan tanda-baca. Kesemua itu adalah jalan pintas untuk membuat komunikasi lebih cepat dan lebih efisien. • Ham biasanya menggunakan kata CW untuk telegraphy. Istilah CW berasal dari Continous wave walau CW itu alih-alih jauh sebagai Continous wave, tetapi lebih suatu gelombang yang di interupsi secara terus-menerus mengacu pada ritme Kode Morse. Ham menggunakan istilah Morse dan CW saling bertukaran, yang keduanya mempunyai arti yang sama saja. • Bandwidth (kelebaran pita) -6dB dari sinyal yang dibentuk secara patut, kira-kira sama dengan 4 kali kecepatan kirim (sending speed) dalam WPM (Words Per Minute). Contoh: CW pada kecepatan 25 WPM mengambil 100 Hz (pada -6dB). Spektrum yang dibutuhkan untuk mentransmit satu voice sinyal SSB (2,7 kHz) dapat mewadahi lebih dari selusin sinyal CW. • Pada dasarnya bandwidth sempit CW menghasilkan rasio Signal to Noise (S/N) lebih baik dalam kondisi marjinal, dibandingkan dengan sinyal wideband seperti SSB (suatu bandwidth yang lebih lebar mengandung lebih banyak kekuatan noise ketimbang bandwidth yang lebih sempit). Itulah sebabnya kontak DX dalam kondisi marjinal.

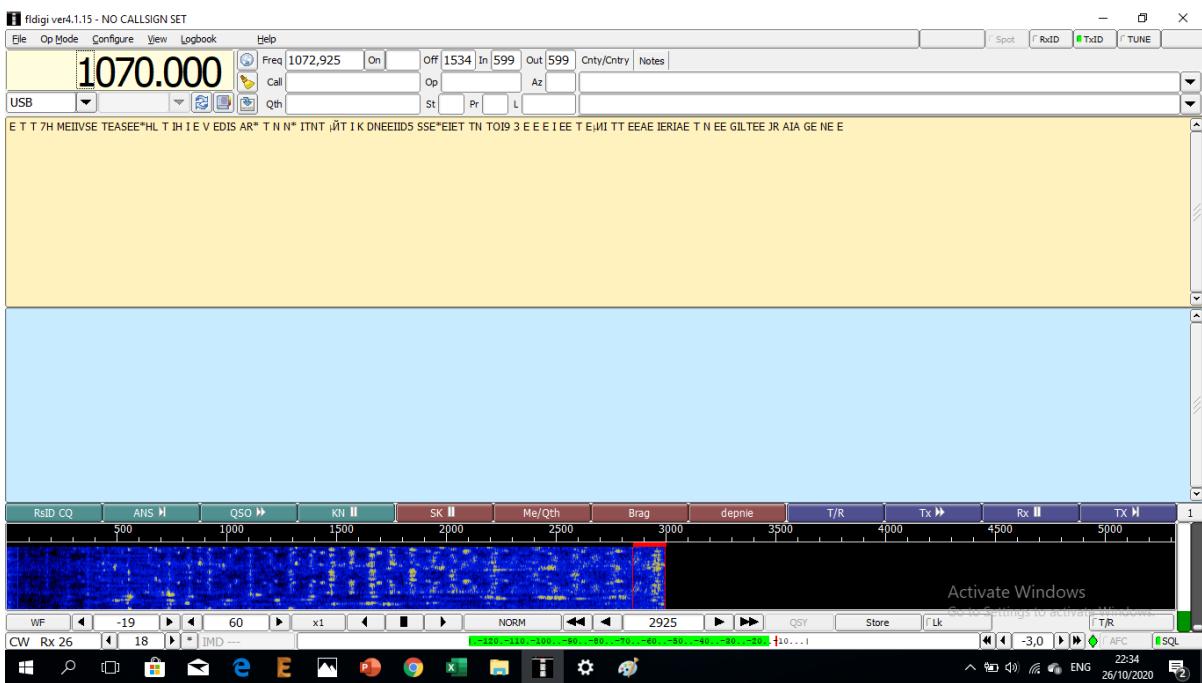
3.2 RTTY

RTTY merupakan moda digital yang paling tua yang digunakan Ham, kecuali CW, yang sesungguhnya adalah juga moda digital. RTTY digunakan untuk mengirim teks. Kode yang digunakan dalam RTTY dibangun dan diurai oleh mesin. Dimasa lalu (semasa mesin telex) mesin mekanikal tersebut menghasilkan dan mengurai kode Baudot; kode aseli teleprinting ini diciptakan tahun 1870 ! Tiap karakter yang diketik pada keyboard mesin, diubah kedalam kode 5 bit, didahului oleh start-bit dan diakhiri dengan stop-bit. Namun dengan 5 bit, hanya dapat dibuat 32 kemungkinan kombinasi ($2^5 = 32$). Karena ada 26 huruf (dalam RTTY hanya huruf besar tersedia) plus 10 angka serta sejumlah tandatanda, kode Baudot ini menghasilkan 2 penandaan yang telah ditentukan untuk tiap kode 5 bit, tergantung pada tata-kedudukan dimana mesin RTTY berada.

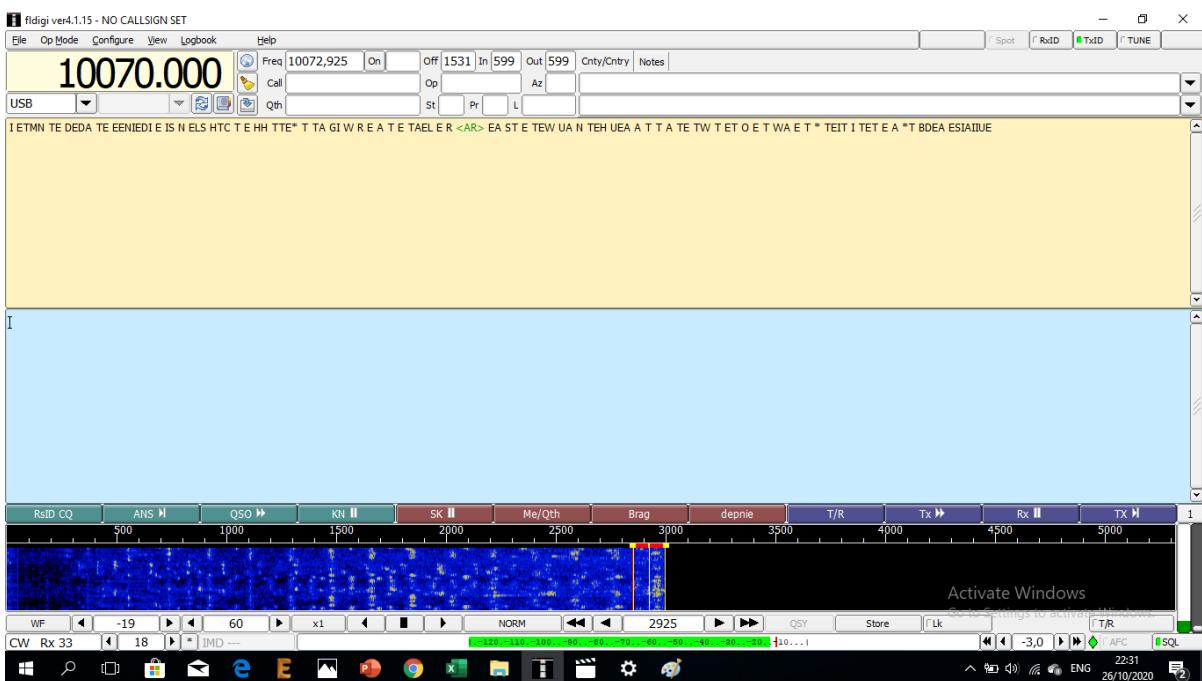
3.3 PSK-31

PSK31 adalah suatu moda digital yang didesain untuk komunikasi keyboard-to-keyboard via radio. Moda ini menggunakan soundcard dalam komputer untuk mengubah pesan-pesan tulisan ketikan menjadi sinyal audio yang dimodulir serta mengubah penerimaan sinyal audio PSK31 menjadi teks. • Sinyal PSK31 beroperasi pada 31,25 baud (cukup besar untuk pesan-pesan dengan ketikan tangan) memiliki secara teori, 31 Hz bandwidth yang sangat sempit pada -6dB (dalam praksis bandwidth ini adalah kirakira 80 Hz). Dalam PSK31 tidak terdapat prosedur komputasi berulang tentang koreksi kekeliruan (error correcting). Tetapi untuk rasio S/N lebih besar dari 10dB, PSK31 sesungguhnya bebas error. Pada rasio S/N rendah, PSK31 kira-kira 5 kali lebih baik dari RTTY. • Tiap karakter dan kode Baudot , yang digunakan dalam RTTY, menggunakan kode binary yang disusun terdiri dari sejumlah tetap 5 bit , yang berarti bahwa panjangnya sama. Namun PSK31 menggunakan varicode yang artinya suatu kode dengan variable length Contoh: suatu huruf ‘q’ di kodekan dengan tidak kurang dari 9 bit (‘110111111 110111111 110111111’), sedangkan huruf ‘e’ terdapat hanya 2 bit (‘11’). Rata-ratanya, satu karakter terdapat 6.15 bit. Pada deretan kotak huruf kecil (lower-case).

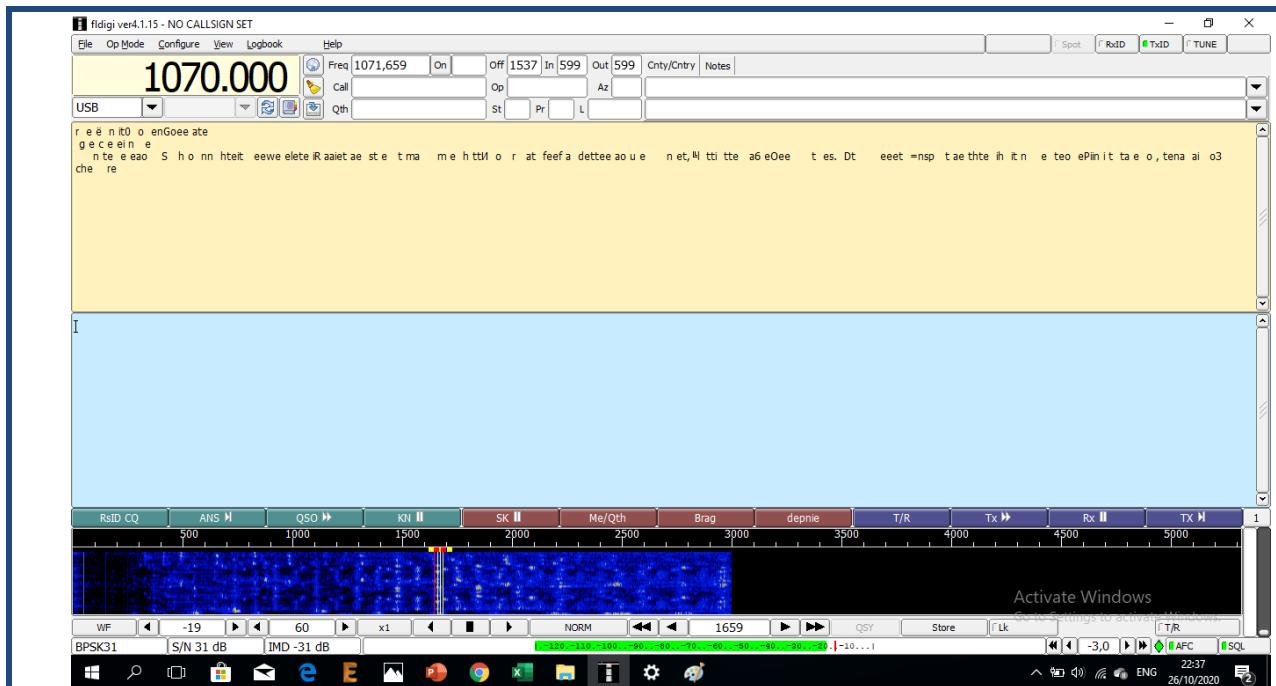
IV. HASIL PERCOBAAN



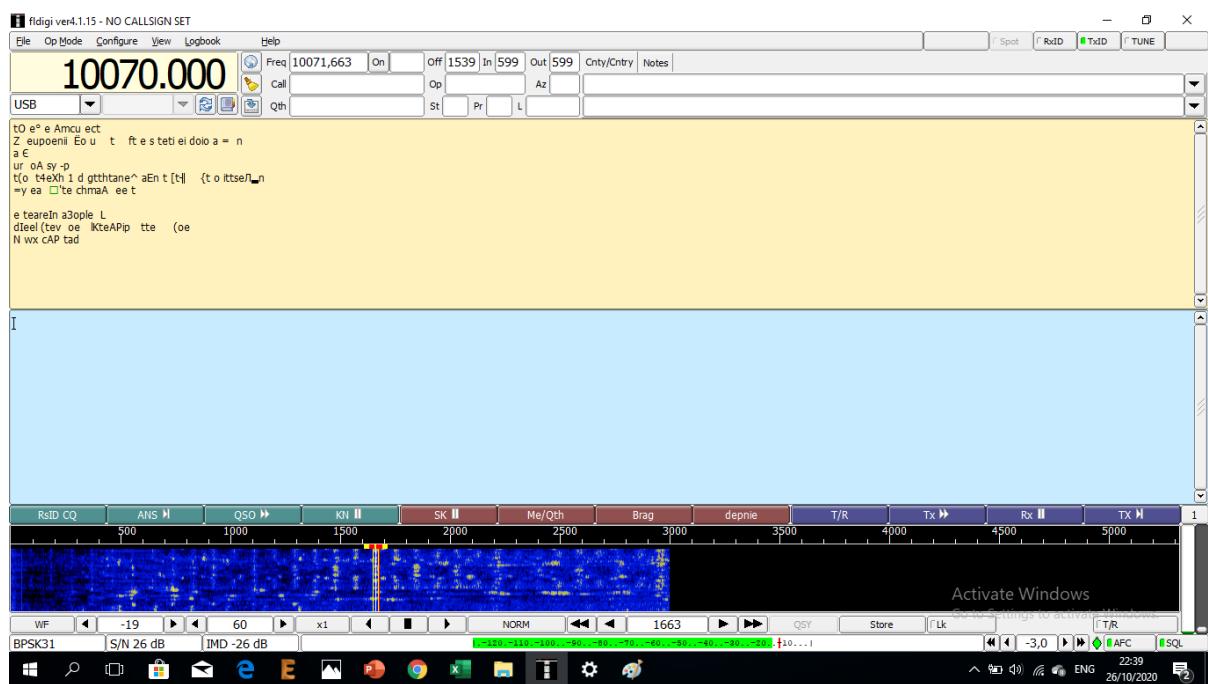
Gambar 1. Mode CW / Kode Morse F= 1070 Hz



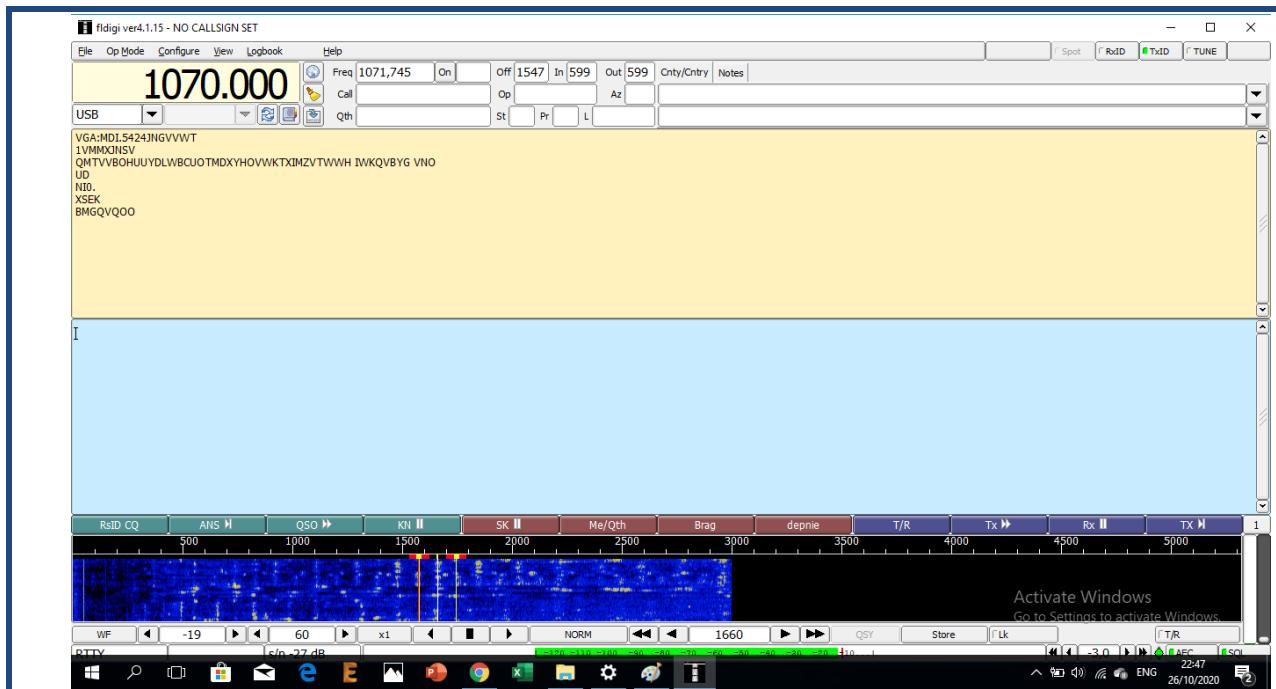
Gambar 2. Mode CW / Kode Morse F= 10070 Hz



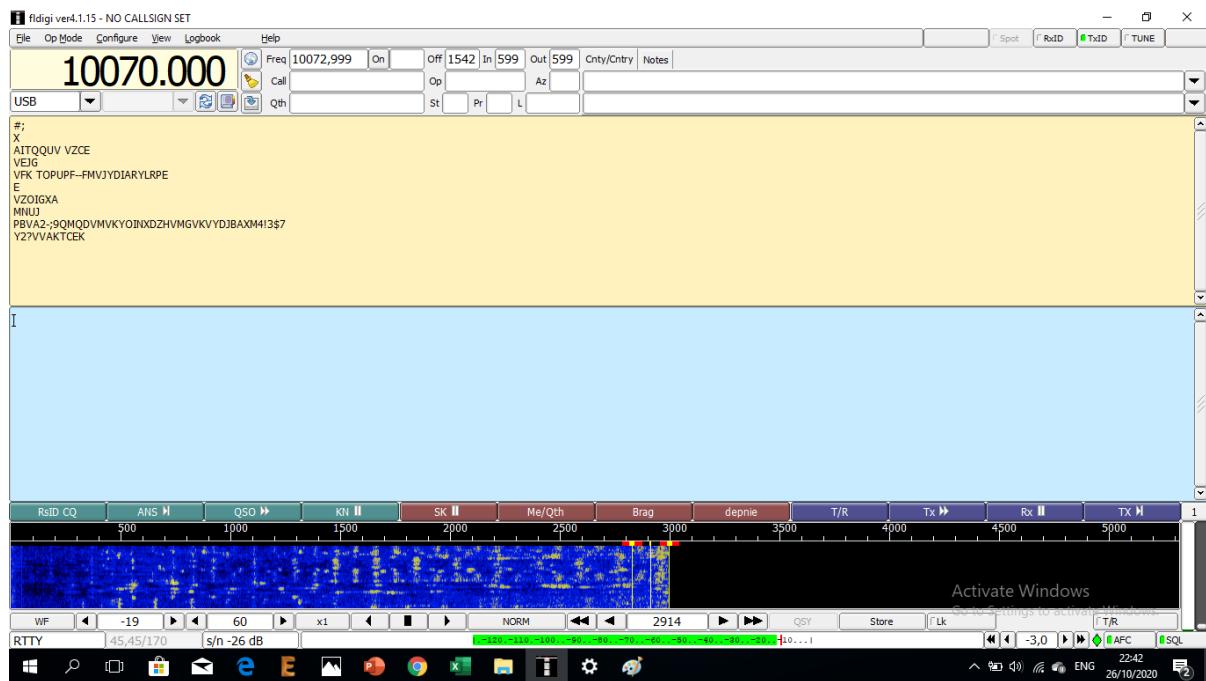
Gambar 3. Mode BPSK-31 F= 1070 Hz



Gambar 4. Mode BPSK-31 F= 10070 Hz



Gambar 5. Mode RTTY F= 1070 Hz

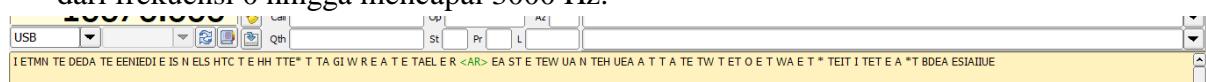


Gambar 6. Mode RTTY F= 10070 Hz

V. ANALISA

Pada praktikum kali ini membahas tentang pengiriman data dengan input berupa audio suara melalui earphone menggunakan aplikasi Fldigi. Pengiriman diatur menggunakan 3 mode yaitu mode CW, RTTY, dan PSK-31.

CW atau Continous Wave adalah gelombang sinyal radio yang secara terus menerus dipancarkan dengan membawa carrier sinyal nada/suara tone dengan sistem pengiriman dan penerimaan mempergunakan kode morse. Input berupa audio lagu rock yang didengarkan pada earphone. Hasil pengiriman data berupa huruf-huruf kapital, terdapat beberapa simbol seperti bintang (*) dan kode <AR>. Dan hasil sinyalnya terukur dari frekuensi 0 hingga mencapai 3000 Hz.

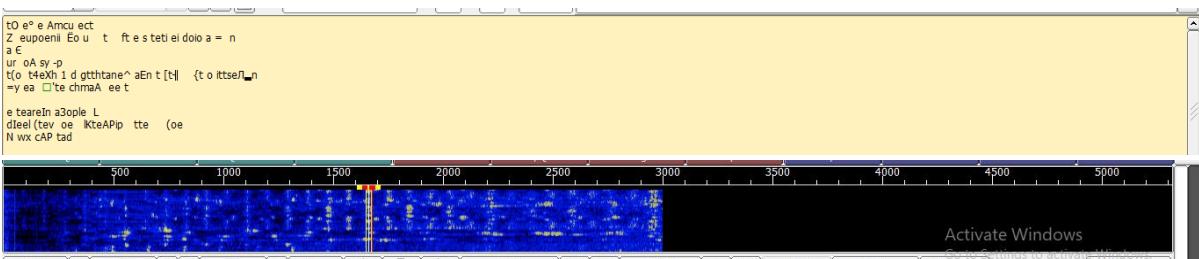




Gambar 5.1 Transimisi Mode CW

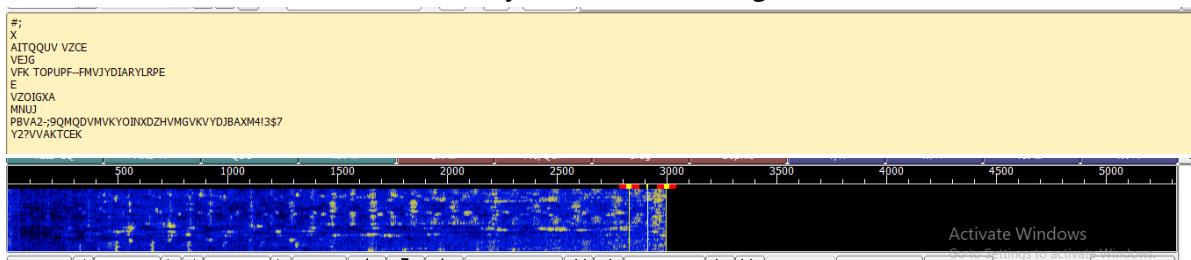
Pada mode CW dilakukan 2 kali percobaan frekuensi yaitu 1070 Hz dan 10070 Hz. Ketika frekuensi 1070 Hz, didapat hasil data berupa huruf kapital saja. Namun berbeda saat 10070 Hz, hasil data berupa huruf kapital, kode dan simbol.

PSK-31 digunakan terutama oleh operator radio amatir untuk melakukan obrolan keyboard-ke-keyboard waktu nyata, paling sering menggunakan frekuensi pada pita radio amatir frekuensi tinggi (gelombang pendek dekat). Input berupa audio lagu rock yang didengarkan menggunakan earphone. Hasil pengiriman data berupa huruf-huruf kecil, kode dengan spesial karakter. Sinyal dari audio terukur memiliki frekuensi paling besar 3000 Hz.



Gambar 5.2 Transimisi Mode PSK-31

RTTY digunakan terutama oleh operator radio amatir untuk melakukan obrolan keyboard-ke-keyboard waktu nyata, paling sering menggunakan frekuensi pada pita radio amatir frekuensi tinggi (gelombang pendek dekat). Suara yang dominan terletak di frekuensi 2900-3000 Hz, terlihat sinyal berwarna kuning.



Gambar 5.3 Transmisi Mode RTTY

VI. KESIMPULAN

- ★ Mode Kode Morse atau CW dapat mengeluarkan 20 huruf dalam 1 menit dan mengeluarkan data dengan alfabet huruf besar dan tidak dapat mengeluarkan tanda.
- ★ Mode PSK-31 mengeluarkan data alfabet kapital
- ★ Mode RTTY mengeluarkan data dengan alfabet kapital, namun dengan penandaan paragraf
- ★ Semakin besar frekuensi yang diatur, semakin banyak jenis data yang dihasilkan.

PRAKTIKUM VI

GRAFIK 3D PADA SENSOR IMU

I. TUJUAN

Pada praktikum kali ini bertujuan untuk :

1. Mengukur dan melakukan proses filter pada data XYZ yang dihasilkan oleh IMU GY-521
2. Melakukan plot 3D pada data pengukuran sebelum di filter dan sesudah di filter.

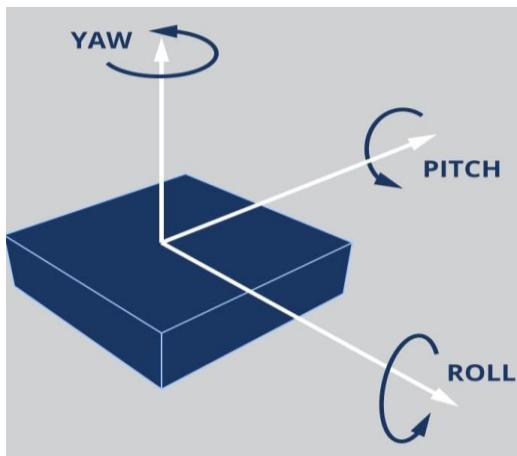
II. PERALATAN

Peralatan yang digunakan adalah :

1. Sensor IMU GY-521
2. PC/Laptop
3. Software Matlab

III. DASAR TEORI

Sebuah unit pengukuran inersia, inertial measurement unit atau IMU, adalah sebuah perangkat elektronik yang mengukur dan melaporkan kecepatan kerajinan itu, orientasi, dan gaya gravitasi, menggunakan kombinasi akselerometer dan giroskop, kadang-kadang juga magnetometer. Imu biasanya digunakan untuk manuver pesawat, termasuk kendaraan udara tak berawak (UAV), antara lain banyak, dan wahana antariksa, termasuk satelit dan pendarat. Perkembangan terkini memungkinkan untuk produksi perangkat GPS IMU-enabled. Sebuah IMU memungkinkan penerima GPS untuk bekerja ketika GPS-sinyal tidak tersedia, seperti di terowongan, di dalam bangunan, atau ketika interferensi elektronik hadir. Sebuah IMU nirkabel dikenal sebagai WiMu.



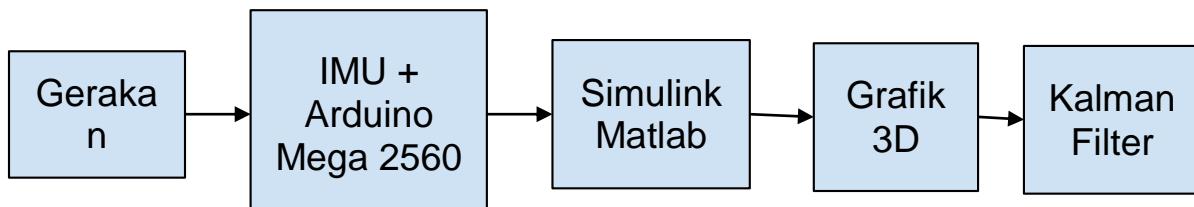
Gambar 1. Inertial Measurement Unit (Yaw-Pitch-Roll)

Akselerometer adalah perangkat yang berfungsi untuk mengukur akselerasi tepat. Akselerasi tepat yang diukur dengan akselerometer belum tentu memiliki ketepatan koordinat (laju perubahan velositas). Sebaliknya, akselerometer melihat akselerasi terkait dengan fenomena berat yang dialami oleh massa uji pada kerangka acuan perangkat akselerometer. Sebagai contoh, akselerometer di permukaan bumi akan mengukur akselerasi $g = 9.81 \text{ m/s}^2$ lurus ke atas karena beratnya. Sebaliknya, akselerometer jatuh bebas atau di luar angkasa akan mengukur nol. Istilah lainnya untuk jenis akselerasi yang bisa diukur oleh akselerometer adalah akselerasi gaya-g. Akselerometer memiliki berbagai aplikasi dalam bidang industri dan sains. Akselerometer yang sangat sensitif digunakan sebagai komponen sistem navigasi inersia pada pesawat tempur dan rudal. Akselerometer juga digunakan untuk mendeteksi dan memonitor getaran pada mesin putar. Selain itu, akselerometer digunakan pada komputer tablet dan kamera digital agar foto di layar selalu ditampilkan tegak.

IV. HASIL DAN ANALISA

Pada praktikum kali ini membahas tentang Grafik 3D pada Sensor IMU. Menggunakan Sensor IMU GY-521 3-Axis untuk melakukan pengukuran akselerasi terhadap sumbu X, Y dan Z ketika diputar sedemikian hingga yang nantinya akan terlihat hasil pergerakannya melalui grafik 3D melalui aplikasi Simulink yang ada pada Matlab.

Berikut adalah proses pengambilan data :



Gambar 1. Proses Pengambilan Data pada ArduSat

Proses yang pertama yaitu perencanaan gerakan. Pada praktikum ini perencanaan gerakan adalah dengan cara diputar .

Selanjutnya percepatan gerakan modul IMU GY-521 akan diukur oleh sensor accelerometer yang terdapat pada modul IMU. Sensor accelerometer pada IMU dapat mengukur dalam 3-axis.

Berikut ini adalah skrip program dari kalman filter

```
function [xhatOut, yhatOut] = KALMAN(u,meas)
% This Embedded MATLAB Function implements a very simple Kalman filter.
%
% It implements a Kalman filter for estimating both the state and output
% of a linear, discrete-time, time-invariant, system given by the following
% state-space equations:
%
% x(k) = 0.914 x(k-1) + 0.25 u(k) + w(k)
% y(k) = 0.344 x(k-1) + v(k)
%
% where w(k) has a variance of 0.01 and v(k) has a variance of 0.1.

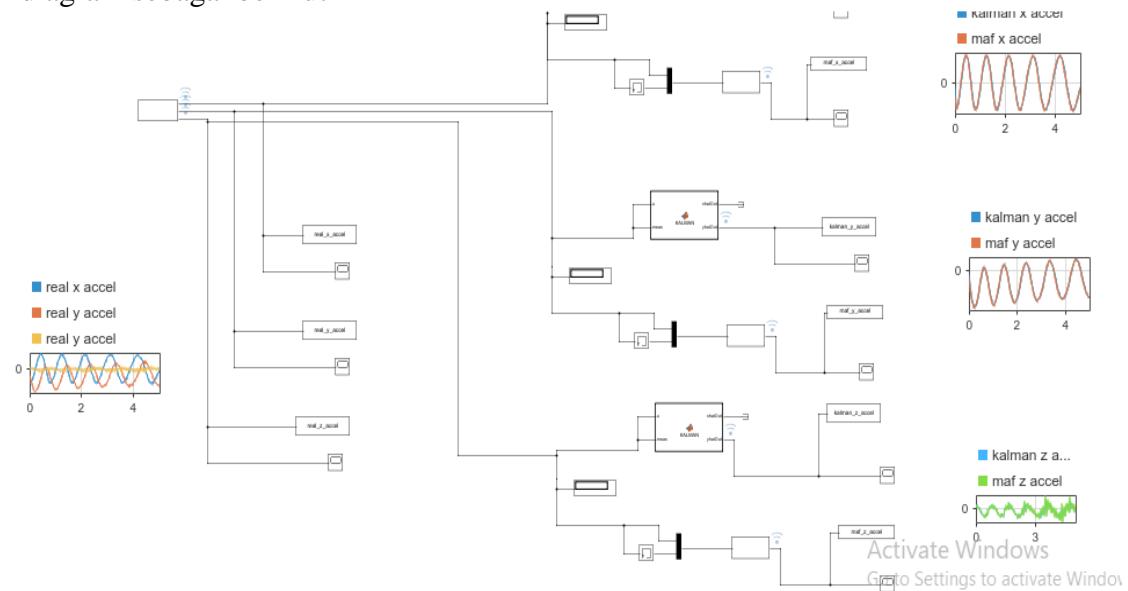
% Author: Phil Goddard (phil@goddardconsulting.ca)
% Date: Q2, 2011.

% Define storage for the variables that need to persist
% between time periods.
persistent P xhat A B C Q R
if isempty(P)
    % First time through the code so do some initialization
    xhat = 0;
    P = 0;
    A = 0.914;
    B = 0.25;
    C = 0.344;
    Q = 0.0015^2;
    R = 0.0015^2;
end
% Propagate the state estimate and covariance matrix:
xhat = A*xhat + B*u;
P = A*P*A' + Q;
```

```
% Calculate the Kalman gain
K = P*C'/(C*P*C' + R);
% Calculate the measurement residual
resid = meas - C*xhat;
% Update the state and error covariance estimate
xhat = xhat + K*resid;
P = (eye(size(K,1))-K*C)*P;
% Post the results
xhatOut = xhat;
yhatOut = C*xhatOut;
```

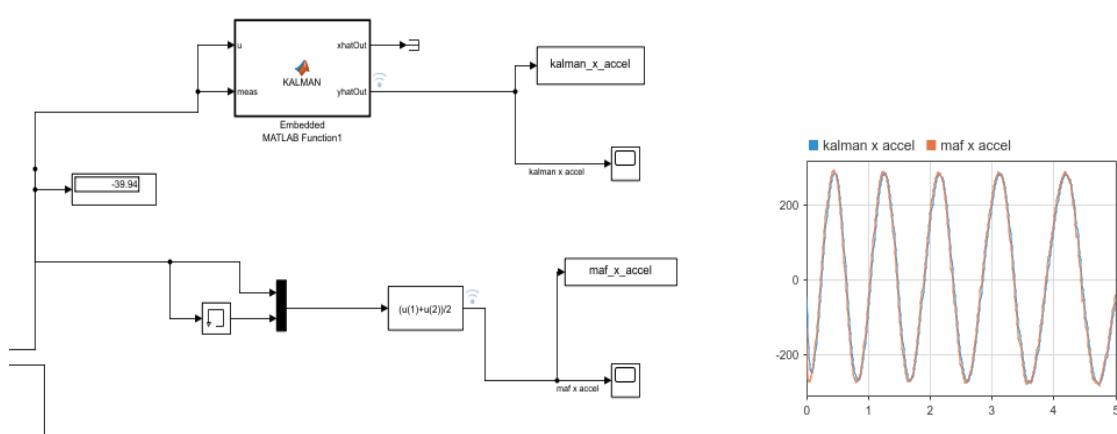
Gambar 1. Skrip program kalman filter

Dari data pengukuran, dilakukan proses filter. Dengan menggunakan Simulink dengan blok diagram sebagai berikut



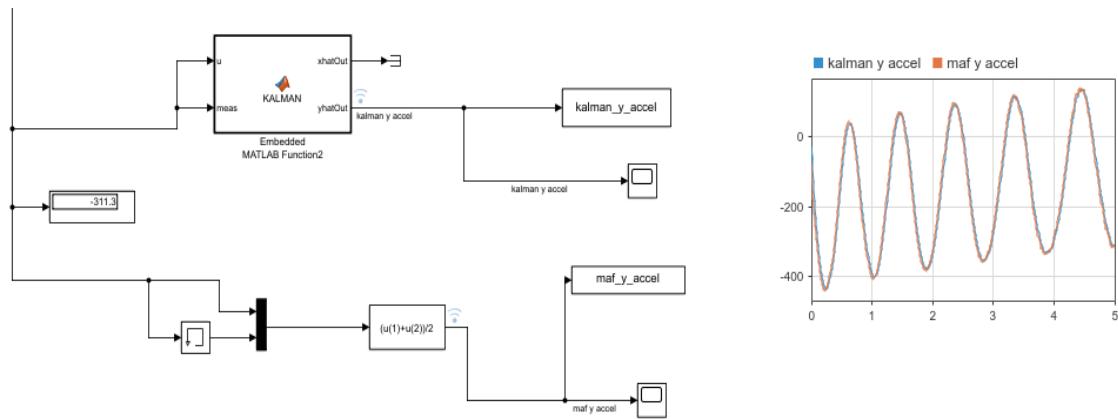
Gambar 2.Blok Diagram Simulink

Berikut ini adalah blog diagram dan grafik accel sumbu x



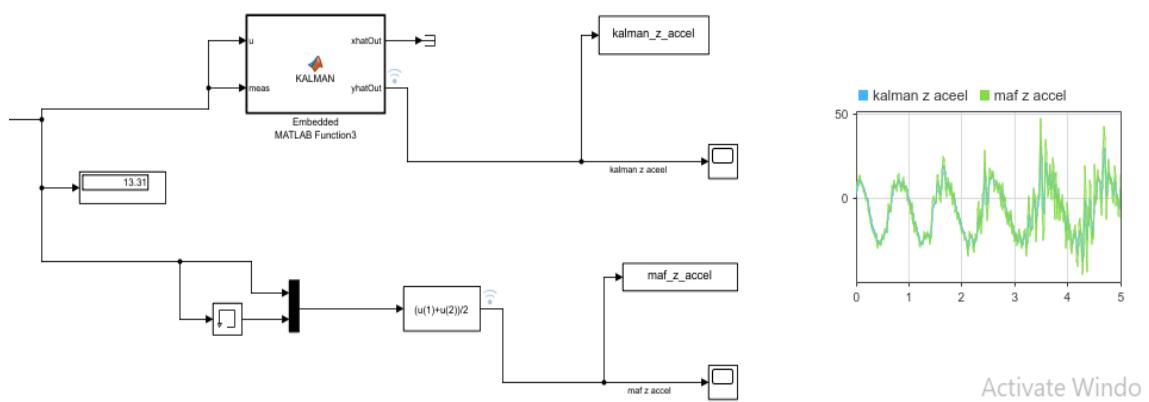
Gamabar 3.Blok Diagram accel sumbu x

Berikut ini adalah blok diagram dan grafik accel sumbu y



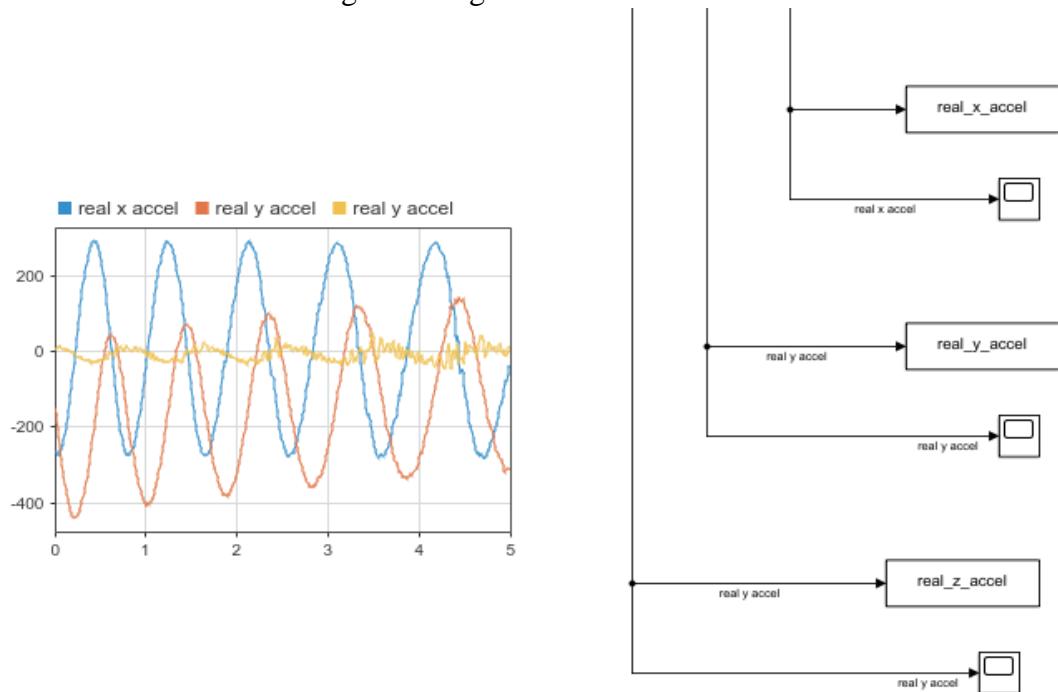
Gambar 4.Blok Diagram accel sumbu Y

Berikut ini adalah blok diagram dan grafik accel sumbu z

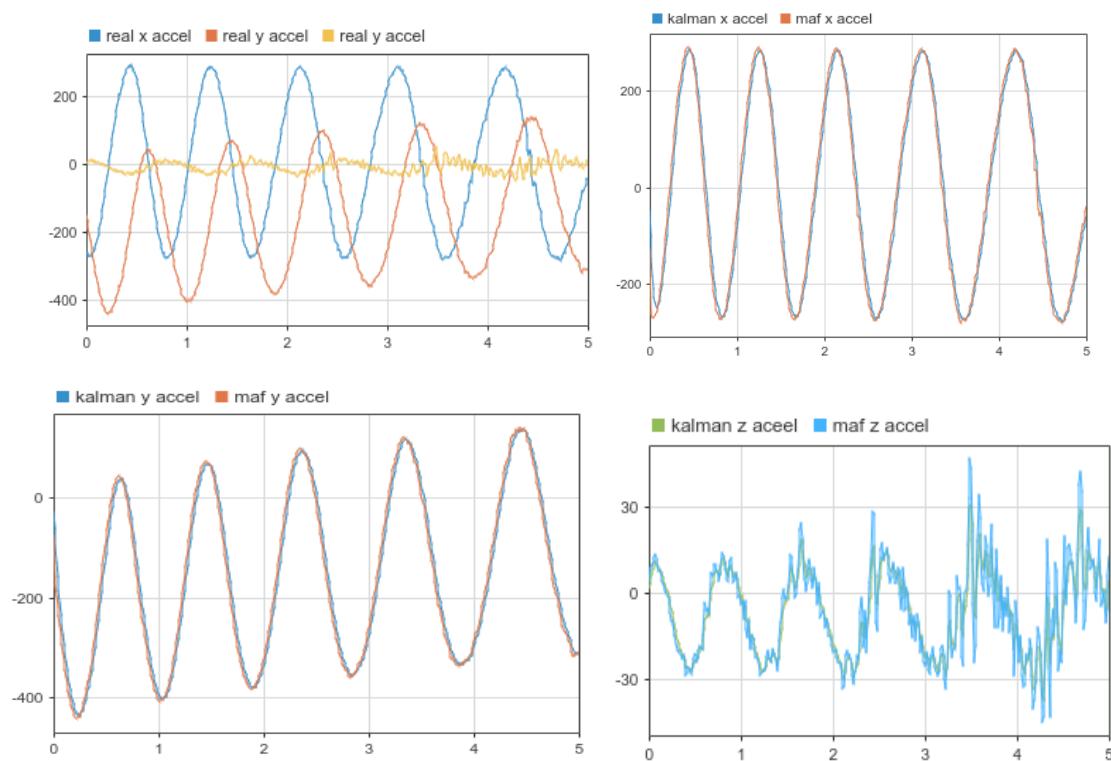


Gambar 5. Blok Diagram accel sumbu Z

Berikut ini adalah blok diagram dan grafik accel data real :



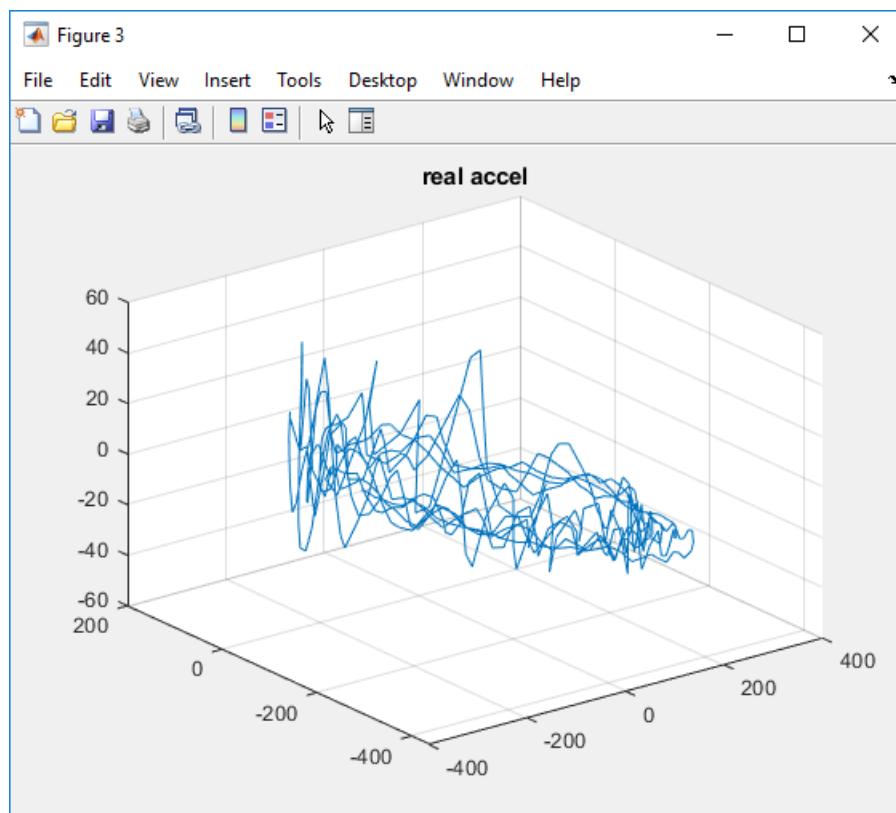
Gambar 6. Blok diagram accel data real



Gambar 7. Hasil Grafik 2D Akselerasi X-Y-Z & Gabungan XYZ

Maka didapatkan hasil sebagai berikut:

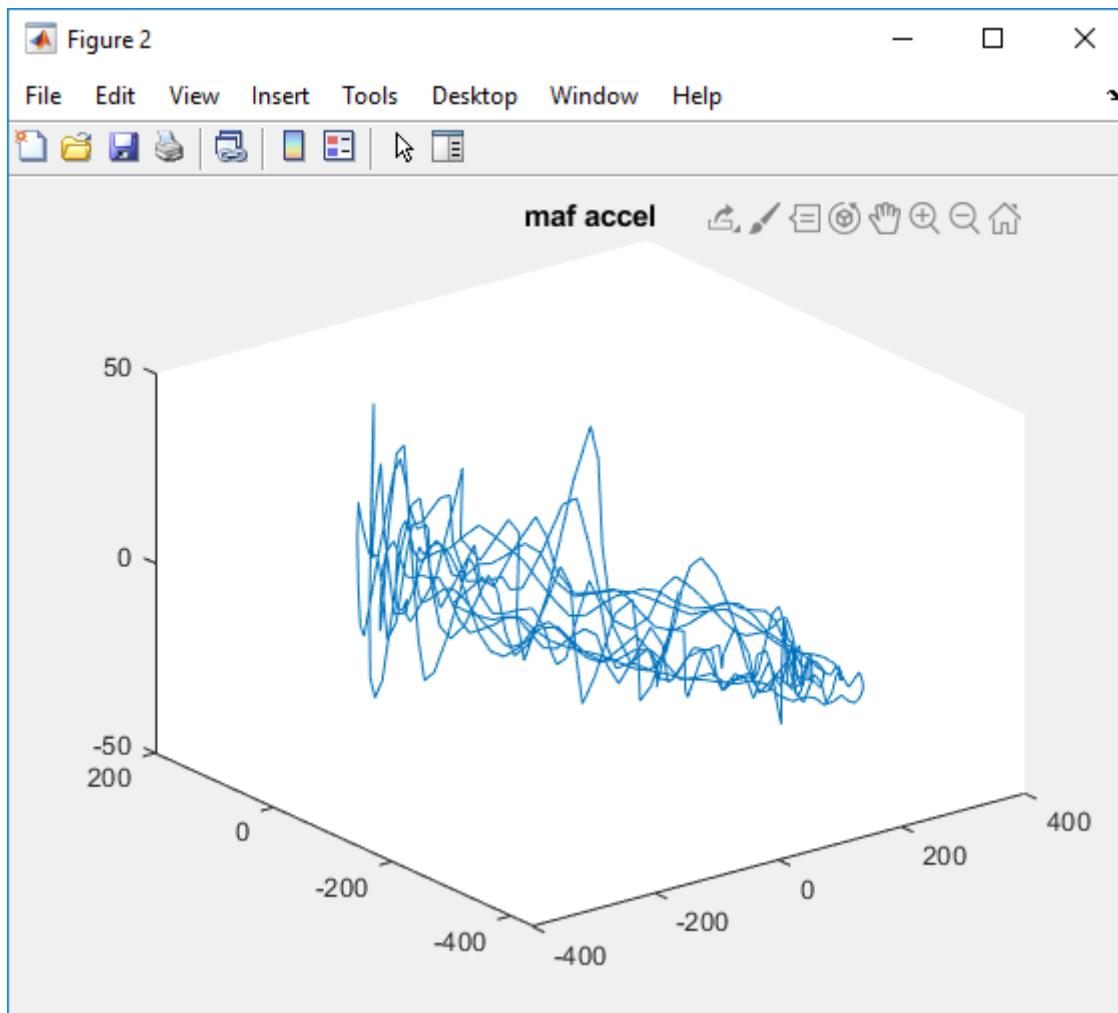
Pada gambar di bawah ini gambar 3D data real dan belum di beri filter



Gambar 8. Hasil plot 3d grafik sumbu x, y, dan z sebelum di filter

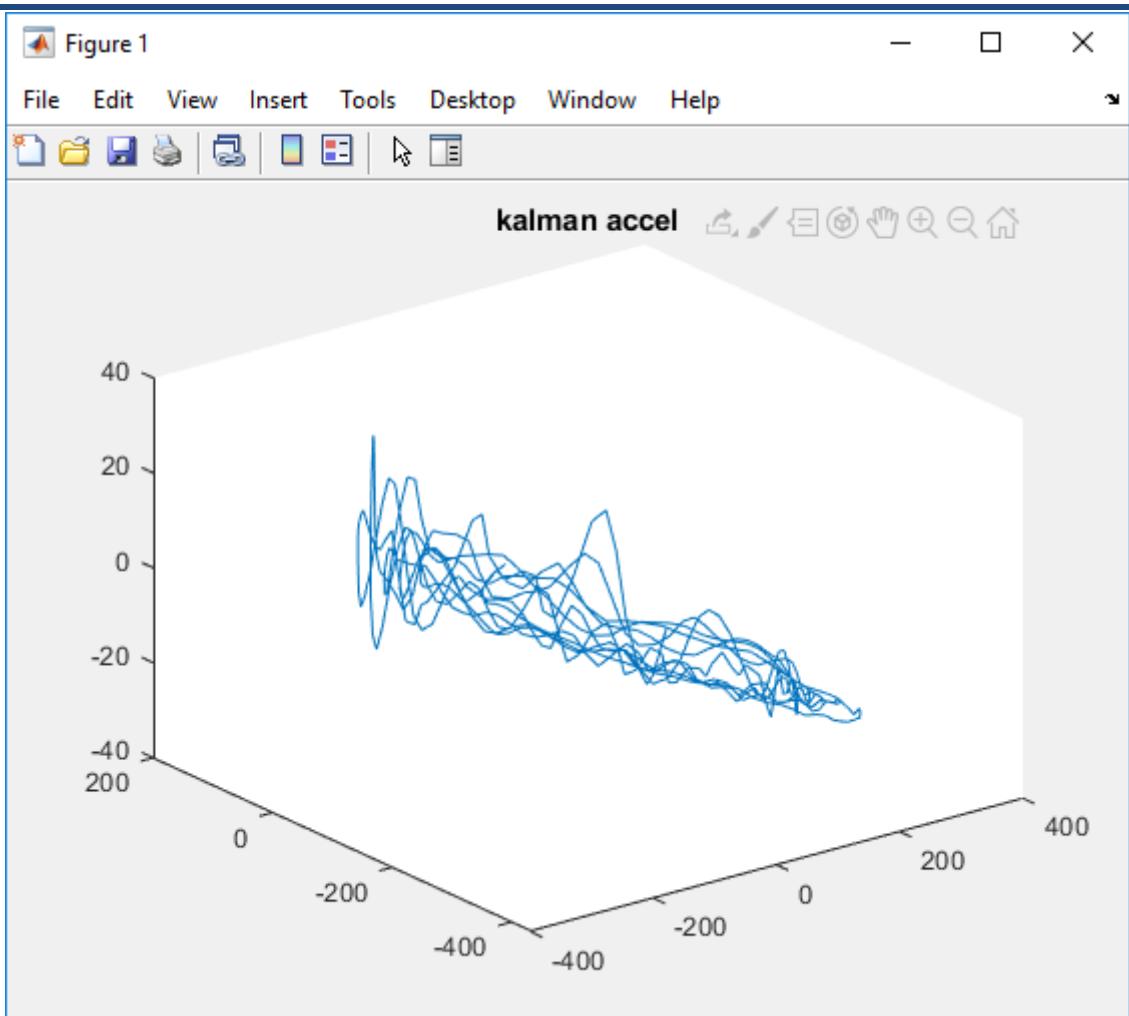
Gambar 8. diatas menunjukkan hasil plot grafik 3D dari pengukuran sumbu x, y, dan z pada pengukuran Accelero IMU. Hasil diatas menunjukkan bahwa sinyal masih acak-acakan dan jelek karena belum dilakukan proses filter. Oleh karena itu untuk memperbaiki sinyal perlu

dilakukan proses filter. Pada percobaan ini dilakukan filter sepuluh tingkat. Dan hasilnya adalah sebagai berikut.



Gambar 9. Hasil plot 3D grafik sumbu x, y, dan z sesudah di filter maf

Dari hasil pada gambar 9. diatas dapat lihat bahwa hasil lebih baik daripada sebelum dilakukan filter, terlihat bahwa hasil lebih halus. Dan noise bisa direduksi lebih banyak dengan melakukan proses filter sepuluh tingkat. Namun pada proses filter ini juga memiliki sisi buruk yaitu terdapat delay. Untuk lebih jelas dapat dilihat pada gambar dibawah ini.



Gambar 10. Hasil plot 3D grafik sumbu x, y, dan z sesudah di filter kalman

Dari hasil pada gambar 10. diatas dapat lihat bahwa filter kalman mampu mereduksi gangguan data dibandingkan metode sebelumnya dan real data. filter kalman memberikan hasil yang lebih halus dan mampu mengikuti trend atau kecenderungan data baik pada saat data menurun maupun menaik.

V. KESIMPULAN

- Untuk merekonstruksi posisi, sensor accelerometer sendiri, tidak dapat diandalkan. sehingga disarankan untuk menggabungkan data accelerometer dengan sensor mems yang lain, seperti gyroscope dan magnetometer.
- Pada percobaan ini dapat disimpulkan bahwa, data pengukuran yang diukur biasanya tercampur dengan noise. Oleh karena itu untuk mengurangi/menghilangkan noise diperlukan proses filter. Filter yang digunakan memiliki tingkat yang mempengaruhi hasil filter, dengan tingkat lebih tinggi akan menghasilkan hasil yang lebih baik, namun delay akan semakin banyak.
- Praktikan telah mampu menampilkan plot perjalanan modul accelerometer yang berputar searah jarum jam pada sumbu Z pada bidang 3 dimensi.

PROJECT I

REACTION BALANCING WHEEL

I. TUJUAN

1. Memahami cara kerja keseimbangan dalam reaction wheel
2. Mendesain mekanik agar reaction wheel dapat seimbang dititik yang diinginkan

II. PERALATAN

1. Mekanik statik: Kotak berukuran 15 x 15 cm
2. Mekanik dinamik: Bundar, diameter berukuran $\pm 10 \text{ cm}$
3. Tatakan Kayu
4. Mur & Baut
5. Arduino Mega 2560
6. Sensor IMU GY-521
7. Motor DC 1000 RPM
8. Driver Motor L-298N
9. Power Supply 12V
10. Kabel Jumper
11. Software Arduino IDE
12. Laptop

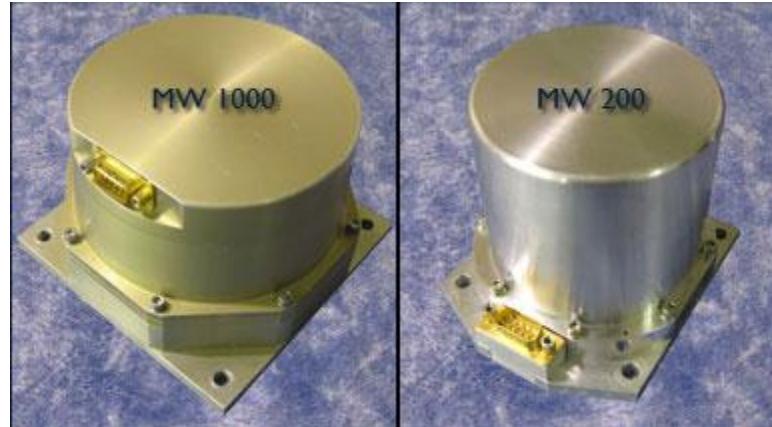
III. DASAR TEORI

Reaction Wheel adalah kelas aktuator listrik untuk satelit yang tidak memerlukan propelan apa pun dan mampu memberikan torsi pada satelit. Konsep dasar dari roda reaksi bergantung pada konsep kekekalan momentum sudut (dijelaskan di atas). Roda reaksi terdiri dari massa yang berputar dengan sejumlah besar inersia. Karena roda reaksi memasukkan massa berputar yang terpisah ke massa satelit (yang juga dapat dianggap sebagai massa berputar), ekspresi momentum sudut harus berisi istilah untuk roda reaksi serta satelit itu sendiri.

Dalam persamaan di atas, menjadi jelas bahwa untuk mempertahankan momentum sudut total yang konstan (roda reaksi h_{total}), membutuhkan pertukaran antara momentum roda reaksi dan momentum satelit. Seseorang juga dapat melihat bagaimana memodifikasi kecepatan atau sumbu roda reaksi dapat memodifikasi kecepatan putaran satelit.

A. Reaction Wheel

Sementara istilah "roda reaksi" secara teknis dapat mencakup semua jenis aktuator massa pemintalan, biasanya digunakan untuk massa pemintalan sumbu tetap seperti MicroWheels MSCI.



Gambar 1. MicroWheel 1000 / MicroWheel 200

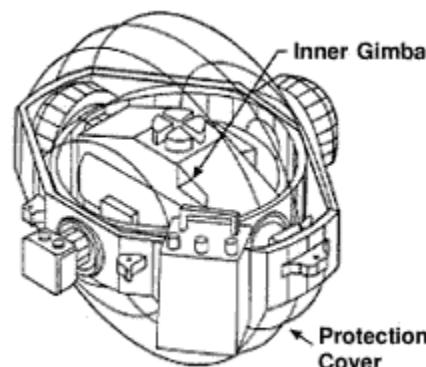
Roda reaksi berisi motor listrik yang mengontrol kecepatan massa yang berputar dengan inersia besar. Momentum roda reaksi dapat diubah dengan mengubah kecepatan massa berputar tetapi tidak dengan sumbu.

B. Roda Momentum

Roda momentum sebenarnya hanyalah kasus khusus dari roda reaksi. Seperti dijelaskan di atas, massa pemintalan yang besar akan menahan torsi eksternal kecil karenakekakuan putarannya. Dengan memutar roda reaksi hingga kecepatan yang relatif tinggi dan meninggalkan roda pada kecepatan tersebut memberikan kekakuan putaran pada sumbu tersebut yang dapat membantu insinyur ACS menjaga agar satelit tetap mengarah ke arah tertentu. Satelit awal adalah "spin-stabilized", yang berarti bahwa seluruh satelit berputar pada sumbu tertentu untuk memberikan kekakuan putaran yang sama. Meskipun ini bekerja dengan baik untuk satelit komunikasi yang mengarahkan antena omnidirectional ke bumi, tetapi tidak akan bekerja dengan baik untuk satelit pencitraan. Roda momentum internal yang memberikan kekakuan gyric yang tinggi merupakan solusi yang sangat baik untuk aplikasi semacam itu.

C. Kontrol Momen Gyro

Pesawat luar angkasa yang sangat besar seperti Stasiun Luar Angkasa Internasional menggunakan Control Moment Gyros (CMG's) untuk mengontrol sikap mereka. CMG adalah roda reaksi pada gimbal dua sumbu. Dengan menggerakkan gimbal dan dengan demikian mengubah sumbu momentum yang disimpan, torsi diterapkan ke satelit. CMG khas ditunjukkan pada gambar di bawah ini:

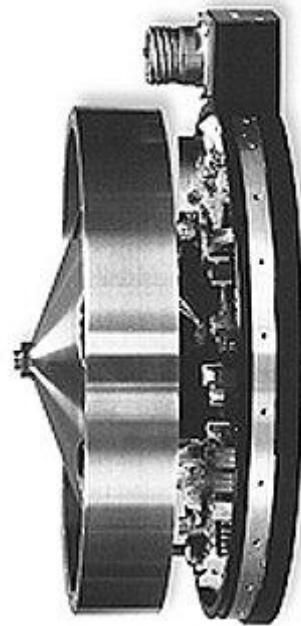


Gambar 2. CMG

Roda reaksi kadang-kadang dioperasikan sebagai (dan disebut sebagai) **roda momentum**, dengan mengoperasikannya pada kecepatan rotasi konstan (atau hampir konstan), untuk mengilhami satelit dengan sejumlah besar momentum sudut yang

tersimpan . Tindakan ini akan mengubah dinamika rotasi pesawat ruang angkasa sehingga torsi gangguan yang tegak lurus dengan satu sumbu satelit (sumbu yang sejajar dengan sumbu putar roda) tidak menghasilkan gerakan sudut pesawat ruang angkasa yang kira-kira sama dengan torsi gangguan; sebaliknya, mereka menghasilkan gerakan sudut (yang umumnya lebih kecil) (presesi) dari sumbu pesawat ruang angkasa itu di sekitar sumbu tegak lurus. Ini memiliki efek cenderung untuk menstabilkan sumbu pesawat ruang angkasa agar mengarah ke arah yang hampir tetap, ^[1] :³⁶² memungkinkan sistem kontrol sikap yang tidak terlalu rumit. Satelit yang menggunakan pendekatan stabilisasi "bias momentum" ini mencakup SCISAT-1 ; dengan mengarahkan sumbu roda momentum agar sejajar dengan vektor orbit-normal, satelit ini berada dalam konfigurasi "bias momentum nada".

Sebuah giroskop kontrol saat (CMG) adalah jenis terkait tetapi berbeda dari sikap aktuator, umumnya terdiri dari roda momentum dipasang pada satu sumbu atau dua-sumbu gimbal. Saat dipasang pada pesawat ruang angkasa yang kaku, menerapkan torsi konstan ke roda menggunakan salah satu motor gimbal menyebabkan pesawat ruang angkasa mengembangkan kecepatan sudut konstan pada sumbu tegak lurus, sehingga memungkinkan kontrol arah penunjuk pesawat ruang angkasa. CMG umumnya mampu menghasilkan torsi berkelanjutan yang lebih besar daripada RW dengan pemanas motor yang lebih sedikit, dan lebih disukai digunakan di pesawat ruang angkasa yang lebih besar dan / atau lebih gesit, termasuk Skylab , Mir , dan Stasiun Luar Angkasa Internasional .



Gambar 3. Reaction wheel

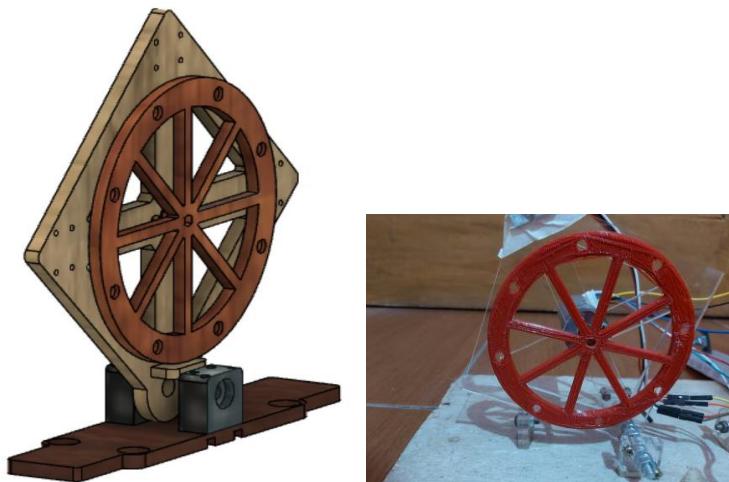
Untuk kontrol tiga sumbu, roda reaksi harus dipasang setidaknya di tiga arah, dengan roda tambahan yang memberikan redundansi ke sistem kontrol sikap. Konfigurasi pemasangan redundan dapat terdiri dari empat roda sepanjang sumbu tetrahedral, ^[3] atau roda cadangan yang dibawa sebagai tambahan pada konfigurasi tiga sumbu. ^[1] :³⁶⁹ Perubahan kecepatan (di kedua arah) dikontrol secara elektronik oleh komputer. Kekuatan bahan yang digunakan dalam roda reaksi menentukan kecepatan lepasnya roda, dan berapa banyak momentum sudut yang dapat disimpannya.

Karena roda reaksi adalah sebagian kecil dari massa total pesawat ruang angkasa, dengan mudah dikendalikan, perubahan kecepatan sementara menghasilkan perubahan kecil pada sudut. Oleh karena itu roda mengizinkan perubahan yang sangat tepat dalam pesawat ruang angkasa sikap . Untuk alasan ini, roda reaksi sering digunakan untuk membidik pesawat ruang angkasa yang membawa kamera atau teleskop.

Seiring waktu, roda reaksi dapat membangun momentum tersimpan yang cukup untuk melebihi kecepatan maksimum roda, yang disebut saturasi, yang perlu dibatalkan. Oleh karena itu, para desainer melengkapi sistem roda reaksi dengan mekanisme kontrol sikap lainnya. Dengan adanya medan magnet (seperti pada orbit rendah Bumi), pesawat ruang angkasa dapat menggunakan magnetorquers (lebih dikenal sebagai batang torsion) untuk mentransfer momentum sudut ke Bumi melalui medan magnet planetnya. ^[1]: 368 Dengan tidak adanya medan magnet, praktik yang paling efisien adalah menggunakan jet sikap berefisiensi tinggi seperti pendorong ion , atau layar surya kecil dan ringan ditempatkan di lokasi yang jauh dari pusat massa pesawat ruang angkasa, seperti pada susunan sel surya atau tiang proyeksi.

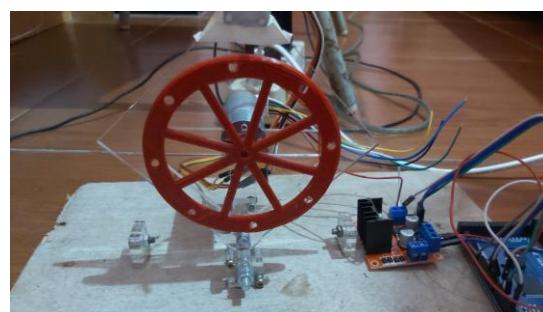
IV. PROSEDUR

1. Mempersiapkan peralatan yang dibutuhkan (File desain mekanik & Source Code bisa dilihat di bit.ly/reactionbalancingwheel)
2. Mempersiapkan desain mekanik statik & dinamik seperti berikut:



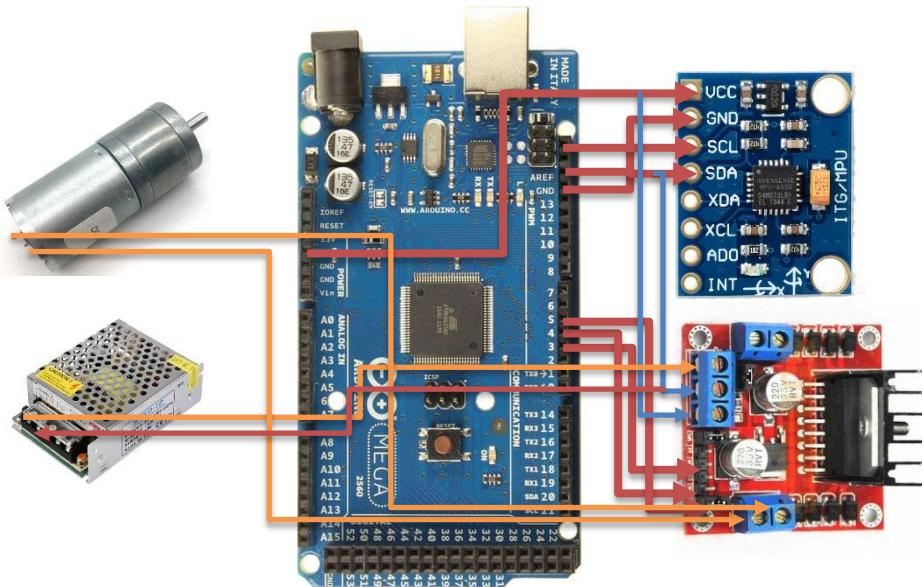
Gambar 4. Desain mekanik

3. Meletakkan setiap komponen seperti pada gambar dibawah ini:



Gambar 5. Peletakkan setiap komponen

4. Menyambungkan setiap hardware dengan kabel jumper/USB seperti wiring hardware hardware dibawah ini:



Gambar 6. Wiring Hardware

5. Membuka software Arduino IDE dilaptop
6. Menjalankan program seperti dibawah ini:

```

7. #include "I2Cdev.h"
8. #include "PID_v1.h" //From https://github.com/br3ttb/Arduino-PID-
   Library/blob/master/PID_v1.h
9. #include "MPU6050_6Axis_MotionApps20.h"
   //https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050
10.
11. MPU6050 mpu;
12.
13. // MPU control/status vars
14. bool dmpReady = false; // set true if DMP init was successful
15. uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
16. uint8_t devStatus; // return status after each device operation (0 =
   success, !0 = error)
17. uint16_t packetSize; // expected DMP packet size (default is 42
   bytes)
18. uint16_t fifoCount; // count of all bytes currently in FIFO
19. uint8_t fifoBuffer[64]; // FIFO storage buffer
20.
21. // orientation/motion vars
22. Quaternion q; // [w, x, y, z] quaternion container
23. VectorFloat gravity; // [x, y, z] gravity vector
24. float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and
   gravity vector
25.
26. double Kp = 20; //Set this first
27. double Kd = 0.2; //Set this secound
28. double Ki = 140; //Finally set this
29. //*****End of values setting*****
30.
31. double input, output;
32. PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);
33.
34. volatile bool mpuInterrupt = false; // indicates whether MPU
   interrupt pin has gone high
35. void dmpDataReady()
36. {

```

```
37.     mpuInterrupt = true;
38. }
39.
40. void setup() {
41.     Serial.begin(115200);
42.
43.     // initialize device
44.     Serial.println(F("Initializing I2C devices..."));
45.     mpu.initialize();
46.
47.     // verify connection
48.     Serial.println(F("Testing device connections..."));
49.     Serial.println(mpu.testConnection() ? F("MPU6050 connection
50.         successful") : F("MPU6050 connection failed"));
51.     // load and configure the DMP
52.     devStatus = mpu.dmpInitialize();
53.
54.
55.     // supply your own gyro offsets here, scaled for min sensitivity
56.     mpu.setXGyroOffset(220);
57.     mpu.setYGyroOffset(76);
58.     mpu.setZGyroOffset(-85);
59.     mpu.setZAccelOffset(1688);
60.
61.     // make sure it worked (returns 0 if so)
62.     if (devStatus == 0)
63.     {
64.         // turn on the DMP, now that it's ready
65.         Serial.println(F("Enabling DMP..."));
66.         mpu.setDMPEnabled(true);
67.
68.         // enable Arduino interrupt detection
69.         Serial.println(F("Enabling interrupt detection (Arduino external
70.             interrupt 0)..."));
71.         attachInterrupt(0, dmpDataReady, RISING);
72.         mpuIntStatus = mpu.getIntStatus();
73.
74.         // set our DMP Ready flag so the main loop() function knows it's
75.         // okay to use it
76.         Serial.println(F("DMP ready! Waiting for first interrupt..."));
77.         dmpReady = true;
78.
79.
80.         //setup PID
81.         pid.SetMode(AUTOMATIC);
82.         pid.SetSampleTime(10);
83.         pid.SetOutputLimits(-255, 255);
84.     }
85.     else
86.     {
87.         // ERROR!
88.         // 1 = initial memory load failed
89.         // 2 = DMP configuration updates failed
90.         // (if it's going to break, usually the code will be 1)
```

```

91.    Serial.print(F("DMP Initialization failed (code "));
92.    Serial.print(devStatus);
93.    Serial.println(F("")));
94. }
95.
96. pinMode (3, OUTPUT);
97. pinMode (5, OUTPUT);
98. pinMode (9, OUTPUT);
99. pinMode (6, OUTPUT);
100.
101. //By default turn off both the motors
102.     digitalWrite(3,LOW);
103.     digitalWrite(5,LOW);
104.     digitalWrite(9,LOW);
105.     digitalWrite(6,LOW);
106. }
107.
108. void loop() {
109.
110.     // if programming failed, don't try to do anything
111.     if (!dmpReady) return;
112.
113.     // wait for MPU interrupt or extra packet(s) available
114.     while (!mpuInterrupt && fifoCount < packetSize)
115.     {
116.         //no mpu data - performing PID calculations and output to
117.         //motors
118.         pid.Compute();
119.         //Print the value of Input and Output on serial monitor to
120.         //check how it is working.
121.         Serial.print(input); Serial.print(" =>");
122.         Serial.println(output);
123.         if (input>174 && input<186)
124.         {
125.             Stop();
126.         }
127.         if (input>169 && input<191)
128.         {
129.             Stop();
130.         }
131.         if (input>159 && input<198) ngadek(input);
132.         else Stop();
133.     }
134.
135.     // reset interrupt flag and get INT_STATUS byte
136.     mpuInterrupt = false;
137.     mpuIntStatus = mpu.getIntStatus();
138.
139.     // get current FIFO count
140.     fifoCount = mpu.getFIFOCount();
141.
142.     // check for overflow (this should never happen unless our
143.     // code is too inefficient)
144.     if ((mpuIntStatus & 0x10) || fifoCount == 1024)

```

```

144.    {
145.        // reset so we can continue cleanly
146.        mpu.resetFIFO();
147.        Serial.println(F("FIFO overflow!"));
148.
149.        // otherwise, check for DMP data ready interrupt (this should
150.        // happen frequently)
151.    }
152.    {
153.        // wait for correct available data length, should be a VERY
154.        // short wait
155.        while (fifoCount < packetSize) fifoCount =
156.            mpu.getFIFOCount();
157.
158.        // read a packet from FIFO
159.        mpu.getFIFOBytes(fifoBuffer, packetSize);
160.
161.        // track FIFO count here in case there is > 1 packet
162.        // available
163.        // (this lets us immediately read more without waiting for
164.        // an interrupt)
165.        fifoCount -= packetSize;
166.
167.        mpu.dmpGetQuaternion(&q, fifoBuffer); //get value for q
168.        mpu.dmpGetGravity(&gravity, &q); //get value for
169.        gravity
170.        mpu.dmpGetYawPitchRoll(ypr, &q, &gravity); //get value
171.        for ypr
172.
173.        void Forward() //Code to rotate the wheel forward
174.        {
175.            analogWrite(3,0);
176.            analogWrite(5,0);
177.            analogWrite(6,0);
178.            analogWrite(9,0);
179.
180.            Serial.print("F"); //Debugging information
181.
182.        void Reverse() //Code to rotate the wheel Backward
183.        {
184.            analogWrite(3,0);
185.            analogWrite(5,0);
186.            analogWrite(9,0);
187.            analogWrite(6,0);
188.
189.            Serial.print("R");
190.        }
191.
192.        void ngadek(int input)
193.        {

```

```
194.     Serial.print("N");
195.     if(input>159 && input <171)
196.     {
197.         analogWrite(6, 0);
198.         analogWrite(9, 255);
199.         delay(3000);
200.
201.         analogWrite(6, 150);
202.         analogWrite(9, 0);
203.         delay(3000);
204.     }
205.     if(input>189 && input <198)
206.     {
207.         analogWrite(6, 255);
208.         analogWrite(9, 0);
209.         delay(3000);
210.
211.         analogWrite(6, 0);
212.         analogWrite(9, 150);
213.         delay(3000);
214.     }
215. }
216. void Stop() //Code to stop both the wheels
217. {
218.     digitalWrite(3,0);
219.     digitalWrite(5,0);
220.     digitalWrite(6,0);
221.     digitalWrite(9,0);
222.     Serial.print("S");
223. }
```

7. Desain mekanik dinamik (bundar) akan berputar sehingga nantinya jika berhasil akan berada dititik tengah
8. Sentuh sedikit demi sedikit desain mekanik statik (kotak)
9. Jika berhasil maka ketika disentuh kekanan atau kekiri, maka desain akan tetap berada pada titik tengah.

V. ANALISA

Pada projek akhir mata kuliah Sensor & Aktuator ini membahas tentang Reaction Balancing Wheel, yang masuk dalam sub-bab pembelajaran ADCS (*Attitude Determination and Control System*). ADCS adalah sebuah sistem kontrol yang bertujuan untuk mengendalikan orientasi sebuah objek. Sistem ini digunakan untuk mengendalikan keadaan objek terhadap kerangka atau sumbu acuan dari bidang tertentu. Tujuannya adalah agar objek yang dikendalikan bisa mempertahankan keadaan posisinya terhadap bidang acuan yang sudah ditentukan, meskipun objek tersebut terus bergerak dan mendapat gangguan dari luar sistem.

Pada percobaan ini, kita mengatur kecepatan motor DC sedemikian rupa sehingga mekanik pada reaction wheel dapat menjaga keseimbangannya pada titik yang telah ditentukan yaitu pada titik tengah. Untuk input disini menggunakan sensor IMU tipe GY-521 yang digunakan untuk menentukan orientasi posisi dari mekanik, maka dari itu sensor tersebut diletakkan diatas desain mekaniknya statiknya. Kemudian aktuator atau penggeraknya menggunakan motor DC yang diletakkan dibelakang desain mekanik. Sedangkan untuk mikrokontroller nya menggunakan Arduino Mega 2560 yang digunakan untuk media kontrol kecepatan maupun metodenya. Untuk mengatur kecepatan motor DC digunakanlah metode PID.

Ketika motor yang terdiri dari rotor dan stator bergerak, maka ada gaya sesuai dengan arah putarannya. Untuk perhitungan momen insersia pada benda ini dapat dihitung menggunakan rumus:

$$I = 0.5ml^2$$

Dimana :

I : Momen Inersia Benda (Kgm^2)

M : massa benda (Kg)

L = Jarak benda terhadap poros atau pusat rotasi (m)

Sehingga:

$$I = 0.5 \times 0.1 \text{ kg} \times 0.05 \text{ m} \times 0.05 \text{ m}$$

$$I = 1.25 \times 10^{-4} \text{ Kgm}^2$$

Tantangan pada alat ini adalah ketika proses disturbing dari environment luar seperti sentuhan tangan. Sehingga motor dan desain mekanik akan jatuh ke kanan atau ke kiri. Alat diharuskan survive ditengah poros. Maka dari itu diperlukan sebuah metode PID untuk mengatur kecepatan dan torsi motor. Namun tidak bisa dipungkiri ketika alat tidak dapat bekerja maksimal atau dalam hal ini, tidak berada pada sumbu porosnya. Hal itu dikarenakan beberapa faktor seperti proses sensing pada input mengalami keerroran sehingga data orientasi posisi akan amburadul, dapat juga karena pemilihan nilai K_p, K_i dan K_d dalam kontrol kecepatannya.

VI. KESIMPULAN

- Reaction Wheel ini adalah perangkat yang terdiri dari roda putar dalam hal ini kita sebut dengan mekanik dinamik (bundar) yang dipasang di motor listrik (DC Brushless) yang kecepatannya dapat dikontrol oleh komputer onboard, dan menghasilkan torsi yang diperlukan untuk kontrol sikap. Alat ini bekerja berdasarkan prinsip kekekalan momentum sudut dalam roda atau piringan.
- ADCS dalam mengendalikan sikap bekerja sesuai dengan input setpoint lalu dihitung dengan kontroller PID untuk mengatur arah dan kecepatan putar reaction wheel. Untuk mendapatkan keluaran PID memerlukan umpan balik dari sensor

MPU6050 (IMU) yang membaca orientasi sudut dari alat. Nilai keluaran sudut ini yang digunakan PID untuk membandingkan setpoint apakah sudut sesuai dengan sudut alat yang dibaca oleh sensor MPU6050.

- Untuk video projek, desain dan program bisa diakses di google drive:
bit.ly/reactionbalancingwheel.