



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**PERANCANGAN ALGORITMA INTEGRASI SISTEM
NAVIGASI INERSIA DENGAN GPS BERBASIS
*UNSCENTED KALMAN FILTER***

Muhamad Rizal Ridlo Prakoso
NRP 0711154000016

Dosen Pembimbing

Ir. Rusdhianto Effendi A.K., M.T.
Mochammad Sahal, S.T.,M.Sc

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - EE 184801

**PERANCANGAN ALGORITMA INTEGRASI SISTEM
NAVIGASI INERSIA DENGAN GPS BERBASIS
*UNSCENTED KALMAN FILTER***

Muhamad Rizal Ridlo Prakoso
NRP 07111540000016

Dosen Pembimbing
Ir. Rusdhianto Effendi A.K., M.T.
Mochammad Sahal, S.T., M.Sc.

DEPARTMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - EE 184801

**INERTIAL NAVIGATION SYSTEM AND GPS
INTEGRATION ALGORITHM DESIGN BASED ON
*UNSCENTED KALMAN FILTER***

Muhamad Rizal Ridlo Prakoso
NRP 07111540000016

Supervisors
Ir. Rusdhianto Effendi A.K., M.T.
Mochammad Sahal, S.T., M.Sc.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2019

PERNYATAAN KEASLIAN

TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan **“Perancangan Algoritma Integrasi Sistem Navigasi Inersia dengan GPS Berbasis *Unscented Kalman Filter*”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka

Apabila pernyataan ini tidak benar, maka saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2019



Muhamad Rizal Ridlo P.

NRP 0711154000016

Halaman ini sengaja dikosongkan

**PERANCANGAN ALGORITMA INTEGRASI
SISTEM NAVIGASI INERSIA DENGAN GPS
BERBASIS *UNSCENTED KALMAN FILTER***


TUGAS AKHIR


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

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Rusdhianto Effendie, M.T.
NIP. 195704241985021001


Mochammad Sahal, S.T., M.Sc.
NIP. 197011191998021002



Halaman ini sengaja dikosongkan

PERANCANGAN ALGORITMA INTEGRASI SISTEM NAVIGASI INERSIA DENGAN GPS BERBASIS *UNSCENTED KALMAN FILTER*

Nama : Muhamad Rizal Ridlo Prakoso
NRP : 0711154000016
Dosen Pembimbing I : Ir. Rusdhianto Effendie AK., M.T
Dosen Pembimbing II : Mochammad Sahal ST., M.Sc.

ABSTRAK

Pada tugas akhir ini, dikembangkan algoritma integrasi antara sistem navigasi inersia (INS) dengan GPS menggunakan algoritma *Unscented Kalman Filter*. *Inertial Navigation System* atau sistem navigasi inersia adalah suatu sistem yang terdiri dari sensor, dan algoritma mekanisasi yang dapat digunakan untuk mengetahui 9 state, yaitu orientasi (*roll, pitch, yaw*), kecepatan, dan posisi wahana berdasarkan inialisasi dan penyelarasan, namun sensor pada INS ini sangat rentan terhadap gangguan dari luar maupun dari dalam sensor tersebut, sehingga ketidakakuratan terhadap solusi navigasi sangat besar. GPS sangat umum digunakan untuk mengoreksi kesembilan state tersebut, sehingga dibutuhkan state *estimator* berupa *Kalman Filter* nonlinier dikarenakan model kesalahan 15 state nonlinier, 9 state pada mekanisasi ditambah dengan 6 state bias akselerometer dan giro. Peran estimasi bias akselerometer dan giroskop sangat penting karena nilainya berubah setiap waktu. Integrasi INS-GPS berhasil meningkatkan performansi sebesar 40,30% , 6,73%, dan 48,85% untuk state *roll, pitch, yaw*. 82,44%, 94,66%, dan 90,85% untuk state kecepatan *north, east, down*, dan 91,52%, 97,11%, 98,18% untuk state *latitude, longitude*, dan ketinggian dibandingkan dengan tanpa integrasi GPS. Hasil menunjukkan perbedaan yang tidak signifikan antara performansi *UKF* dan *KF*. Peningkatan performansi yang terlihat adalah pada state *yaw* yaitu sebesar 0,1055%, kecepatan sumbu *down* sebesar 0,1403%, posisi lintang sebesar 0,006%, dan ketinggian sebesar 0,1%, dan waktu komputasi yang diperlukan *UKF* lebih lama jika dibandingkan dengan *KF*.

Kata Kunci: *Unscented Kalman Filter, Nonlinier, Navigasi, INS-GPS,*

Halaman ini sengaja dikosongkan

INERTIAL NAVIGATION SYSTEM AND GPS INTEGRATION ALGORITHM DESIGN BASED ON UNSCENTED KALMAN FILTER

Name : Muhamad Rizal Ridlo Prakoso
NRP : 07111540000016
Supervisor I : Ir. Rusdhianto Effendie AK., M.T.
Supervisor II : Mochammad Sahal S.T., M.Sc.

ABSTRACT

In this final project, the integration algorithm between INS (Inertial Navigation System) and GPS is developed using Unscented Kalman Filter. INS or Inertial Navigation System is a system which consists of sensors, and a mechanization algorithm used to know 9 states, which are attitude (roll, pitch, yaw), velocity, and position that is based on initialization and alignment processes, but the sensor of INS is prone to disturbance and noise both from external sources and internal sources, so that the inaccuracies of navigation solution is large. GPS is commonly used to correct nine states using the information of velocity, and position, so that the Kalman Filter is used as a non-linear state estimator because of the nonlinearity of the 15 states of error model, which are 9 states from mechanization, and 6 states from accelerometer and gyroscope biases. Bias role estimation affects the INS performance since the value changing with time. The INS-GPS integration successfully enhance the performance about 40,30% ,6,73%, and 48,85% for roll, pitch, yaw. 82,44%, 94,66%, dan 90,85% for north, east, down velocities, and 91,52%, 97,11%, 98,18% for latitude, longitude, and height compared to no GPS integration. But the results show that there are no significant difference of performance improvement between using UKF compared to KF. The improvements using UKF are the yaw state which is 0.1055%, velocity at down axis which is 0,1403%, latitude which is 0,006%, and the altitude which is 0,1%, and with more computational time consumption.

Keyword : *Unscented Kalman Filter, Nonlinear, Navigation, INS-GPS,*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul **“Perancangan Algoritma Integrasi Sistem Navigasi Inersia dengan GPS Berbasis *Unscented Kalman Filter*”** guna memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan Departemen Teknik Elektro Fakultas Teknologi Elektro Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Penulis juga mengucapkan terimakasih kepada pihak-pihak yang berjasa dalam pengerjaan tugas akhir ini yaitu:

1. Keluarga terutama orang tua, dan anggota keluarga lain yang selalu memberi dukungan, semangat, dan doa untuk keberhasilan penulis.
2. Bapak Ir. Rusdhianto Effendie AK., M.T dan Bapak Mochammad Sahal ST., M.Sc.. selaku dosen pembimbing atas bimbingan dan arahan selama proses penyelesaian tugas akhir ini
3. Dosen-dosen lain yang telah mengajar penulis selama penulis belajar di ITS.
4. Fachruddin Ari Setyawan yang telah menemani penulis melewati masa-masa sulit di Teknik Elektro ITS
5. M. Farih, Anan M., Tommy Pratama, dan Ilham Wicaksono yang telah menemani penulis selama menjalani riset laboratorium.
6. Moch. Oza A, dan Leonard Wihardi, yang telah menjadi tim yang hebat pada matakuliah PIS.
7. Teman-teman HIMATEKTRO yang telah menemani dan mewarnai masa-masa menjadi staff himpunan.
8. Alif Saiful Z, Nurika Andana, Asma'ul Khusna, atas bimbingannya dan dukungannya dalam penulisan tugas akhir ini.
9. Seluruh teman-teman Sistem Pengaturan yang saya cintai
10. Muhammad Azhar Ismail, untuk buku kalkulusnya yang sangat membantu.
11. Teman-teman e55 yang selalu memberikan inspirasi.

Laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Semoga buku laporan Tugas Akhir ini dapat memberikan manfaat bagi pembaca sebagai acuan penelitian selanjutnya.

Surabaya, 3 Juli 2019

Penulis

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vii
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4 Tujuan.....	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan	4
1.7 Relevansi	5
BAB 2 TEORI PENUNJANG	7
2.1 Sistem Navigasi [1]	7
2.2 Kerangka Koordinat[1].....	8
2.2.1 Kerangka Inersia.....	8
2.2.2 Kerangka Bumi	9
2.2.3 Kerangka Navigasi Lokal	10
2.2.4 Kerangka tubuh	11
2.3 Matriks Transformasi Koordinat[1].....	11
2.4 Model Bumi[1]	12
2.4.1 Model Elipsoid dari Permukaan Bumi	12

2.4.2 Posisi pada Lengkungan Bumi	15
2.4.3 Rotasi Bumi	15
2.4.5 Gravitasi dan Gaya Spesifik[1]	18
2.5 <i>Unscented Kalman Filter</i> [1,5].....	20
2.5.1. Algoritma <i>Unscented Kalman Filter</i>	22
2.5.2 <i>Closed-Loop Kalman Filter</i>	22
2.6 Sensor Inersia[1,2].....	23
2.6.1 Akselerometer	23
2.6.2 Giroskop.....	24
2.6.3 Karakteristik Kesalahan.....	24
2.7 Navigasi Inersia[1,2]	28
2.7.1 Pembaruan Orientasi	28
2.7.2 Pembaruan Kecepatan	29
2.7.3 Pembaruan Posisi	29
2.7.4 Inisialisasi dan Penyelarasan	30
2.6.5 Perambatan kesalahan INS.....	32
2.8 STIM 300[8].....	32
2.9 CNS 5000[9]	33
2.10 GPS (<i>Global Positioning System</i>)[6].....	34
2.11 Integrasi Sistem Navigasi Inersia dengan GPS[1,2].....	34
2.11.1 <i>Loosely Coupled Integration</i>	36
2.12 <i>Root Mean Square Error (RMSE)</i> [2]	37
BAB 3 PERANCANGAN SISTEM	39
3.1 Perancangan Model Kesalahan Sistem Navigasi Inersia.....	39
3.2 Perancangan <i>Unscented Kalman Filter</i>	41
3.3 Perancangan Mekanisasi	45

3.4 Perancangan Penyelarasan	46
3.5 Perancangan Integrasi INS dengan GPS	48
BAB 4 PENGUJIAN DAN ANALISIS	51
4.1 Pengujian Penyelarasan	51
4.1.1 Penyelarasan kasar (<i>coarse alignment</i>)	51
4.1.2 Penyelarasan halus (<i>Fine Alignment</i>)	55
4.2 Pengujian Sistem Navigasi Inersia	61
4.3 Pengujian GPS	67
4.4 Pengujian Sistem Navigasi Inersia dengan Integrasi GPS	71
BAB 5 KESIMPULAN	81
5.1. Kesimpulan	81
5.2. Saran	82
DAFTAR PUSTAKA	83
LAMPIRAN	83
RIWAYAT HIDUP PENULIS	105

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 Satelit GPS[1].....	7
Gambar 2. 2 Kerangka inersial[1].....	9
Gambar 2. 3 Kerangka Bumi[1].....	10
Gambar 2. 4 Kerangka navigasi lokal[1]	10
Gambar 2. 5 Kerangka tubuh[1].....	11
Gambar 2. 6 Model elipsoid sebagai representasi permukaan Bumi[1]	15
Gambar 2. 7 <i>Geocentric</i> dan <i>Geodetic Latitude</i> [1]	15
Gambar 2. 8 Ilustrasi Longitude[1]	16
Gambar 2. 9 Ketinggian geodesi dan lintang geodesi dari tubuh[1]	18
Gambar 2. 10 Rotasi Bumi dalam Kerangka ECI ECEF[1]	18
Gambar 2. 11 Gaya yang terjadi pada pegas	19
Gambar 2. 12 Pendekatan metode EKF dan UKF untuk estimasi state nonlinier[5].....	21
Gambar 2. 13 Skema INS dan sensor IMU[2]	23
Gambar 2. 14 Akselerometer statis dan mengalami percepatan[5]	24
Gambar 2. 15 Bias pada sensor IMU	25
Gambar 2. 16 Kesalahan faktor skala dan pengopelan silang	27
Gambar 2. 17 STIM300[8].....	33
Gambar 2. 18 CNS5000[9]	33
Gambar 2. 19 Integrasi INS-GPS[2]	35
Gambar 3. 1 Diagram alur sistem INSGPS	39
Gambar 3. 2 Alur mekanisasi.....	46
Gambar 3. 3 Diagram blok penyalarsan halus.....	47
Gambar 3. 4 Diagram blok integrasi INS GPS	49
Gambar 4. 1 Data akselerometer saat diam selama 20 detik	52
Gambar 4. 2 Data giroskop saat diam selama 20 detik	52
Gambar 4. 3 Hasil estimasi <i>roll</i> penyalarsan kasar	53
Gambar 4. 4 Hasil estimasi <i>pitch</i> penyalarsan kasar	53
Gambar 4. 5 Yaw penyalarsan kasar	54
Gambar 4. 6 Data akselerometer diam selama 210 detik	55
Gambar 4. 7 Data giroskop diam selama 210 detik	55
Gambar 4. 8 Perbandingan penyalarsan halus untuk <i>roll</i> dengan menggunakan UKF dan KF	56

Gambar 4. 9 Perbandingan penyelarasan halus untuk <i>pitch</i> dengan menggunakan UKF dan KF	57
Gambar 4. 10 Perbandingan penyelarasan halus untuk <i>yaw</i> dengan menggunakan UKF dan KF	58
Gambar 4. 11 Perbandingan hasil estimasi bias akseremoter dari penyelarasan halus antara UKF dan KF.....	59
Gambar 4. 12 Perbandingan hasil estimasi bias giroskop dari penyelarasan halus dengan menggunakan UKF dan KF.....	60
Gambar 4. 13 Data akselerometer STIM 300 saat bergerak selama 120 detik	62
Gambar 4. 14 Data akselerometer STIM 300 saat bergerak selama 120 detik.	62
Gambar 4. 15 Perbandingan <i>roll</i> INS denngan CNS	63
Gambar 4. 16 Perbandingan <i>pitch</i> INS dengan CNS	63
Gambar 4. 17 Perbandingan <i>yaw</i> INS dan CNS	64
Gambar 4. 18 Perbandingan kecepatan wahana antara INS dan CNS ..	65
Gambar 4. 19 Perbandingan posisi wahana INS dan CNS	66
Gambar 4. 20 Perbandingan ketinggian wahana INS dan CNS	66
Gambar 4. 21 Perbandingan antara kecepatan CNS dan GPS	68
Gambar 4. 22 Perbandingan antara posisi CNS dan GPS	69
Gambar 4. 23 Perbandingan antar ketinggian CNS dan GPS	70
Gambar 4. 24 Perbandingan koreksi <i>roll</i> dengan GPS menggunakan UKF dan KF	72
Gambar 4. 25 Perbandingan koreksi <i>pitch</i> dengan GPS menggunakan UKF dan KF	73
Gambar 4. 26 Perbandingan koreksi <i>yaw</i> dengan GPS menggunakan UKF dan KF	74
Gambar 4. 27 Perbandingan koreksi kecepatan dengan GPS menggunakan UKF dan KF	75
Gambar 4. 28 Perbandingan koreksi lintang dan bujur dengan GPS menggunakan UKF dan KF	76
Gambar 4. 29 Perbandingan koreksi ketinggian dengan GPS menggunakan UKF dan KF	77

DAFTAR TABEL

Tabel 2. 1 Nilai Parameter WGS84.....	14
Tabel 3. 1 Keterangan variabel dan parameter dari model kesalahan state pada integrasi sistem navigasi inersia dengan GPS	40
Tabel 3. 2 Parameter pada matriks P_0	42
Tabel 3. 3 Parameter pada matriks Q	43
Tabel 3. 4. Parameter pada matriks R	44
Tabel 3. 5 Parameter pada <i>Unscented Transformation</i>	44
Tabel 4. 1 Selisih kesalahan antara penyelarasan kasar IMU dengan referensi CNS5000.....	61
Tabel 4. 2 RMSE dari penyelarasan halus IMU dengan referensi CNS5000.....	61
Tabel 4. 3 RMSE pengujian sistem navigasi inersia dengan referensi CNS5000.....	67
Tabel 4. 4 RMSE GPS dengan referensi CNS5000.....	71
Tabel 4. 5 Perbandingan antara RMSE integrasi INS GPS dengan UKF dan KF dengan referensi CNS5000.....	78
Tabel 4. 6 Peningkatan performansi INS dengan GPS dan tanpa GPS dengan menyelisihkan kedua RMSE dan membandingkan dengan RMSE INS tanpa GPS.....	78
Tabel 4. 7 Peningkatan performansi INS dengan UKF dan KF dengan menyelisihkan kedua RMSE dan membandingkan dengan RMSE INS GPS dengan KF.....	79
Tabel 4. 8 Perbandingan waktu komputasi UKF dengan KF.....	79

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Pada bab ini akan dibahas latar belakang serta hal-hal yang mendasari pengerjaan penelitian diantaranya rumusan masalah, batasan masalah, tujuan serta tahapan dalam merancang algoritma integrasi sistem navigasi inersia dengan GPS berbasis *Unscented Kalman Filter*.

1.1 Latar Belakang

Sistem navigasi adalah *device* yang digunakan untuk menentukan posisi, kecepatan, maupun orientasi terhadap suatu *kerangka* tertentu. Pada sejarahnya, sistem navigasi inersia (INS) hanya digunakan untuk keperluan militer dan pesawat komersial, dikarenakan harganya yang mahal. Seiring berkembangnya teknologi, terutama pada bidang elektronik, membuat instrumen yang ada pada sistem navigasi semakin hari semakin berukuran kecil, hingga hanya millimeter dalam bentuk *chip* dan juga harganya semakin terjangkau. Namun sensor ini tidak bekerja sebaik sensor mahal tersebut, dikarenakan sangat dipengaruhi oleh beberapa sumber kesalahan, namun kecepatan informasi yang dapat tersedia sangat tinggi[1]. Sumber kesalahan dari INS berasal dari 3 sumber, yaitu pengukuran akselerometer dan giroskop, pendekatan dan kecepatan iterasi pada persamaan navigasi, dan kesalahan inisialisasi dan penyesuaian.

Untuk dapat memenuhi persyaratan spesifikasi hasil navigasi (posisi, kecepatan, dan orientasi) di beberapa kendaraan atau wahana, maka INS ini sering digabungkan dengan GPS saat navigasi. GPS dikenal dengan keakuratannya, namun juga sangat terpengaruh oleh keadaan lingkungan sekitar, seperti gedung, terowongan, dan lain-lain, juga kecepatan informasi yang dapat tersedia sangat rendah. Dengan menggabungkan kedua informasi tersebut, diharapkan mendapat hasil navigasi yang akurat dan presisi[2].

Terdapat beberapa macam metode integrasi INS-GPS, diantaranya adalah *loosely-coupled*, *tightly-coupled*, *ultra tightly-coupled*, yang mana masing-masing memiliki kelebihan dan kekurangan masing-masing. Semakin INS-GPS terhubung kuat maka algoritma integrasinya semakin rumit, namun jika satelit GPS mencapai jumlah kurang dari minimum satelit yaitu 4 buah, maka GPS akan tetap memberikan pembaruan informasi navigasi[11]. Pada tugas akhir ini, digunakan *loosely-coupled*

integration karena metode ini memiliki kelebihan yaitu algoritma integrasi yang lebih mudah dikarenakan GPS bisa diperlakukan seperti *black-box* yang menghasilkan informasi berupa posisi dan kecepatan yang akan digunakan sebagai integrasi INS-GPS tanpa memerlukan informasi *pseudorange* dan *delta range*.

Metode penggabungan dua informasi dari INS dan GPS biasa dilakukan dengan metode Kalman Filter maupun *Artificial Intelligence*. AI digunakan untuk mengatasi performansi integrasi INS dan GPS saat sinyal GPS tidak tersedia[12]. *Extended Kalman Filter(EKF)* dan *Linearised Kalman Filter* juga sering dipakai dalam integrasi INS dan GPS, namun ada satu kekurangan dari kedua metode tersebut, yaitu apabila estimasi *error* atau kesalahan yang terjadi tidak pada “area atau region linier”, yang mana dapat menyebabkan filter yang divergen. Pada tugas akhir ini digunakan metode *Unscented Kalman Filter*. *Unscented Kalman Filter* ini menggunakan sejumlah *sigma point* untuk mendekati distribusi *Gauss*, dibandingkan *Extended Kalman Filter* yang hanya menggunakan 1 titik, sehingga estimasi yang dihasilkan lebih akurat. *Unscented Kalman Filter* ini tidak menggunakan matriks Jacobian, maka sering disebut sebagai “*Derivative-Free Kalman Filter*” sehingga tidak membutuhkan linierisasi state pada sistem dengan matriks Jacobian[7]. Diharapkan dengan menggunakan *Unscented Kalman Filter* dapat meningkatkan keakuratan dan kepresisian dari solusi navigasi INS-GPS.

1.2. Perumusan Masalah

Pada tugas akhir ini yang menjadi masalah adalah pengintegrasian antara sistem navigasi inersia dengan GPS dengan menggunakan *Unscented Kalman Filter* sehingga didapatkan hasil yang sesuai dengan referensi atau nilai sesungguhnya. Karena *Unscented Kalman Filter* adalah filter nonlinier maka permodelan nonlinier dari integrasi sistem navigasi inersia perlu didapatkan.

1.3. Batasan Masalah

Agar persoalan yang dibahas dalam tugas akhir ini tidak terlalu meluas dan tanpa mengurangi tujuan yang dicapai, maka ditentukan batasan masalah sebagai berikut:

1. Tugas akhir ini menggunakan data yang diukur pada wahana darat yaitu menggunakan mobil. Pada mobil tersebut dipasang sensor IMU dan GPS untuk melakukan pengukuran.

2. Pengambilan data dilakukan di kota Surabaya, sehingga lintang dan bujur pada data yang digunakan pada Tugas Akhir ini hanya terbatas pada wilayah kota Surabaya.
3. Pada Tugas Akhir ini sinyal satelit dari GPS diasumsikan selalu tersedia yaitu berjumlah minimum empat buah satelit.
4. *Yaw alignment* atau sudut hadap awal didapat dari referensi yaitu CNS5000
5. Metode yang digunakan adalah metode integrasi INSGPS untai tertutup dan *loosely-coupled*.
6. Sensor IMU dan GPS diletakkan di titik pusat massa kendaraan *host*.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah untuk merancang algoritma komputasi INS sehingga dapat menghasilkan solusi navigasi dan juga merancang algoritma *Unscented Kalman Filter* sebagai metode integrasi INS-GPS untuk meningkatkan akurasi dan presisi dari solusi navigasi

1.5 Metodologi

Metodologi yang digunakan pada tugas akhir ini adalah:

- a. Studi Literatur
Studi literatur disini bertujuan untuk mendapatkan informasi mengenai model dinamika INS saat terintegrasi dengan GPS, baik saat dalam kondisi stasioner maupun bergerak. Informasi yang didapatkan dari *journal*, maupun buku-buku referensi.
- b. Pengambilan data
Pengambilan data ini adalah pengambilan data dari sensor IMU dan GPS menggunakan wahana darat yaitu mobil yang diambil di wilayah Surabaya yang kemudian diolah secara *offline* untuk mendapatkan solusi navigasi yang lebih akurat dan presisi.
- c. Perancangan Model Filter
Perancangan model filter ini bertujuan untuk pengintegrasian antara data INS dan GPS untuk mendapatkan solusi navigasi yang lebih akurat dan presisi. Model yang diharapkan adalah model nonlinier.
- d. Pengujian Sistem dan Analisis Hasil
Pengujian yang bertujuan untuk membandingkan solusi navigasi yang telah didapatkan menggunakan UKF dengan alat navigasi

lain sebagai referensi, kemudian dianalisa dan dibandingkan hasilnya dengan menggunakan KF biasa.

e. Evaluasi dan Penyusunan Laporan

Penyusunan laporan akan dilakukan bersamaan dengan tahap-tahap lainnya. Isinya berkaitan dengan tugas akhir yang dikerjakan meliputi pendahuluan, studi literatur, perancangan model Filter Kalman pada sistem navigasi, pengujian sistem, analisis, serta kesimpulan. Evaluasi dilakukan berdasarkan pertimbangan dosen maupun jika pada pengujian sistem masih belum sesuai dengan hasil yang diharapkan.

1.6 Sistematika Penulisan

Tahap terakhir dari sebuah penelitian adalah penulisan laporan. Pada penulisan laporan/buku tugas akhir ini disusun menjadi 5 bab yaitu:

BAB 1 : PENDAHULUAN

Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, dan sistematika penulisan.

BAB 2 : TEORI PENUNJANG

Bab ini membahas teori-teori yang membantu penelitian, diantaranya tentang definisi sistem navigasi inersia, kerangka koordinat, matriks transformasi koordinat, model bumi, algoritma *Unscented Kalman Filter*, sensor inersia, navigasi inersia, GPS, integrasi INS dengan GPS, dan perhitungan kesalahan dengan RMSE.

BAB 3 : PERANCANGAN SISTEM

Bab ini dijelaskan mengenai perancangan model kesalahan, perancangan penyalarsan, perancangan mekanisasi, dan perancangan integrasi INS dengan GPS dengan *Unscented Kalman Filter*.

BAB 4 : PENGUJIAN DAN ANALISIS

Bab ini memuat hasil dari perancangan keseluruhan sistem yang mengacu pada BAB III dan analisis yang berkaitan dengan hasil yang diperoleh yang kemudian dibandingkan dengan referensi atau hasil sesungguhnya.

BAB 5 : PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian tugas akhir yang telah dilakukan dan saran yang dapat digunakan sebagai acuan penelitian lebih lanjut.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat digunakan pada sistem *autonomous*, seperti *Unmanned Surface Vehicle* (USV) atau *Unmanned Aerial Vehicle* (UAV) dan juga terbentuknya suatu kajian yang dapat dijadikan acuan untuk membuat *prototype* sistem navigasi yang dapat diterapkan pada berbagai wahana baik di permukaan darat dan laut atau di udara.

Halaman ini sengaja dikosongkan

BAB 2

TEORI PENUNJANG

2.1 Sistem Navigasi [1]

Dalam dunia navigasi, dikenal terapat 2 metode navigasi, yaitu *position fixing* dan juga *dead-reckoning*. *Dead-reckoning* adalah teknik penentuan lokasi pemakai dengan mengetahui posisi awal, dan kecepatan awal. Contohnya adalah sistem navigasi inersia (INS) yang menggunakan informasi dari sensor inersia (akselerometer dan giroskop) dan juga posisi dan kecepatan awal agar bisa menentukan posisi, kecepatan, dan orientasi dari pemakai secara *real-time*. *Position-fixing* adalah navigasi yang berkaitan dengan penggunaan berbagai metode visual dan elektronik untuk menentukan posisi kendaraan atau orang di permukaan bumi. Contohnya adalah GPS.

Sistem navigasi inersia (INS) adalah sebuah bantuan navigasi dengan menggunakan sensor inersia berupa akselerometer dan giroskop dan menggunakan algoritma tertentu. Hasil dari sensor ini adalah berupa posisi, kecepatan, dan orientasi wahana relatif terhadap bumi. Sistem navigasi ini biasa digunakan pada wahana seperti kapal, pesawat terbang, pesawat luar angkasa, dan rudal.



Gambar 2. 1 Satelit GPS[1]

Sistem Pemosisi Global (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS Rusia,

Galileo Uni Eropa, IRNSS India. Gambar satelit ditunjukkan pada gambar 2.1.

2.2 Kerangka Koordinat[1]

Kerangka koordinat adalah suatu koordinat yang dipakai untuk menyatakan suatu posisi dari wahana terhadap suatu referensi tertentu. Pada keperluan navigasi, maka diperlukan 3-sumbu yang tidak sebidang dan tentunya saling tegak lurus, dan juga sebuah titik pusat untuk menyatakan posisi, kecepatan, dan orientasi. Hal inilah yang disebut kerangka koordinat.

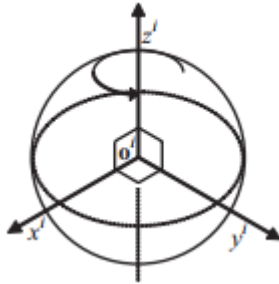
Dalam keperluan navigasi diperlukan sekurang-kurangnya 2 koordinat, yaitu sebagai kerangka objek, yang mana objek tersebut bergerak, dan lainnya adalah sebagai kerangka referensi, yang mana objek bergerak dan kemudian dinyatakan dalam kerangka referensi ini, seperti pada teori fisika relativitas. Pada navigasi, biasanya terdapat 3 kerangka koordinat untuk menyatakan, gerak, kecepatan, dan orientasi suatu wahana, yaitu kerangka objek, kerangka referensi, dan kerangka penyelesaian, dimana gerak objek terhadap kerangka referensi dinyatakan dalam kerangka penyelesaian ini. Kerangka referensi biasanya adalah Bumi, sedangkan kerangka penyelesaian biasanya adalah kerangka navigasi lokal, maupun kerangka Bumi. Sedangkan kerangka objek adalah kerangka tubuh. Berikut akan dijelaskan mengenai kerangka-kerangka tersebut.

2.2.1 Kerangka Inersia

Dalam fisika, setiap kerangka koordinat yang tidak berakselerasi atau berputar sehubungan dengan bagian lain. Semesta adalah kerangka inersia. Kerangka inersia yang berpusat di Bumi, dilambangkan dengan simbol i , yang mana berpusat di pusat massa bumi dan berorientasi sehubungan dengan poros putar Bumi dan bintang-bintang. Hal ini bukan sepenuhnya kerangka inersia karena Bumi mengalami percepatan dalam orbitnya di sekitar Matahari, poros putarannya bergerak perlahan, dan galaksi berputar. Namun, efek ini lebih kecil dari gangguan pengukuran yang diperlihatkan oleh sensor navigasi, sehingga kerangka ECI atau kerangka inersia ini dapat diperlakukan sebagai kerangka inersia sejati untuk tujuan praktis.

Asal dan sumbu kerangka ECI dan rotasi Bumi sehubungan dengan ruang ditunjukkan pada gambar 2.3. Sumbu z selalu menunjuk di sepanjang poros rotasi Bumi dari asal kerangka di pusat massa ke kutub

utara sebenarnya (bukan kutub magnet). Sumbu-x dan y terletak di dalam bidang ekuator, tetapi tidak berputar dengan Bumi. Sumbu y menunjuk 90° tegak lurus sumbu-x ke arah rotasi Bumi

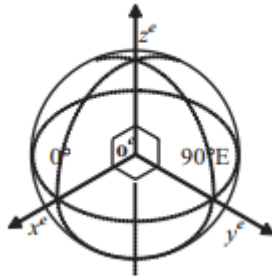


Gambar 2. 2 Kerangka inersial[1]

2.2.2 Kerangka Bumi

Kerangka Bumi yang berpusat di Bumi, biasanya disingkat dengan rangka Bumi, mirip dengan kerangka inersia yang berpusat di Bumi, kecuali bahwa semua sumbu tetap terfiksasi terhadap Bumi. Dua sistem koordinat memiliki asal yang sama, pusat elipsoid memodelkan permukaan Bumi yang kira-kira berada di pusat massa. Frame ECEF dilambangkan dengan simbol e .

Asal dan sumbu frame ECEF ditunjukkan pada gambar 2.3. Sumbu-z sama dengan sumbu ECI yang sesuai. Sumbu -z selalu menunjuk sepanjang poros rotasi Bumi dari pusat ke kutub utara (benar bukan magnet). Sumbu-x menunjuk dari pusat ke persimpangan khatulistiwa dengan *IERS Reference Meridian* (IRM) atau *Conventional Zero Meridian* (CZM), yang memiliki garis bujur 0° . Sumbu-y melengkapi set ortogonal tangan kanan, menunjuk dari pusat ke persimpangan khatulistiwa dengan 90° meridian timur. Sistem koordinat ECEF menggunakan IRP / CTP dan IRM / CZM juga dikenal sebagai Sistem Referensi Terrestrial Konvensional (CTRS), dan beberapa penulis menggunakan simbol t untuk menyatakannya. Sistem koordinat *Earth-fixed* yang berpusat di Bumi penting dalam navigasi karena pengguna ingin mengetahui posisinya relatif terhadap Bumi, sehingga realisasinya umumnya digunakan sebagai kerangka referensi dan kerangka penyelesaian.



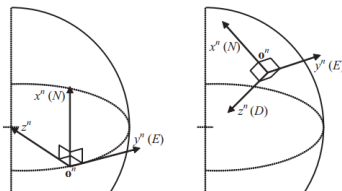
Gambar 2. 3 Kerangka Bumi[1]

2.2.3 Kerangka Navigasi Lokal

Kerangka navigasi lokal atau dalam bahasa Inggris disebut dengan *local-level frame* atau biasa disebut *LLF* biasa digunakan untuk merepresentasikan posisi, kecepatan, dan orientasi dari suatu wahana. Pada beberapa sumber, memiliki simbol n , atau l . Kerangka lokal didefinisikan sebagai berikut

- Titik pusatnya berimpit dengan pusat dari kerangka sensor (titik pusat sensor)
- Sumbu – x mengarah pada arah utara yang sebenarnya.
- Sumbu – y mengarah pada arah timur.
- Sumbu – z berlaku kaidah tangan kanan, sehingga sumbu – z mengarah ke bawah.

Kerangka ini berguna agar pengguna dapat mengetahui posisi, kecepatan, dan orientasi wahana terhadap arah utara, timur, dan bawah.

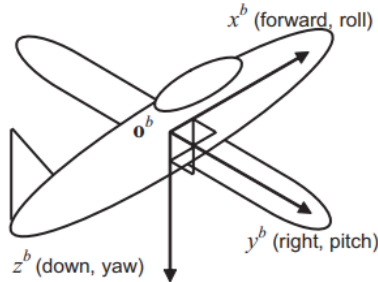


Gambar 2. 4 Kerangka navigasi lokal[1]

2.2.4 Kerangka tubuh

Kerangka tubuh atau dalam Bahasa Inggris disebut dengan *body frame* atau biasa disebut kerangka kendaraan adalah kerangka yang mana objek atau wahana dideskripsikan. Titik pusat dari kerangka tubuh berimpit dengan *local-level frame*, namun sumbu-sumbunya tetap relatif terhadap wahana. Sumbu $-x$ dideskripsikan sebagai arah depan, sumbu $-y$ dideskripsikan sebagai arah kanan, dan dengan menggunakan kaidah tangan kanan, maka sumbu $-z$ dideskripsikan ke arah bawah. Gerakan rotasi terhadap sumbu $-x$ disebut *roll*, sedang pada arah sumbu $-y$ disebut *pitch*, dan sumbu $-z$ disebut *yaw*.

Kerangka ini penting karena sensor inersia mengukur gerakan wahana pada kerangka tubuh. Simbol yang digunakan pada buku ini adalah \mathbf{b} . Berikut adalah ilustrasi kerangka tubuh pada gambar 2.5.



Gambar 2. 5 Kerangka tubuh[1]

2.3 Matriks Transformasi Koordinat[1]

Matriks transformasi koordinat ini adalah matriks 3×3 yang digunakan untuk mengubah sumbu penyelesaian dari suatu vektor ke sumbu resolving lain dengan cara mengalikan vektor tersebut dengan matriks transformasi ini.

$$x^\beta = C_\alpha^\beta x^\alpha \quad (2.1)$$

Keterangan :

C_α^β = matriks transformasi dari kerangka α ke kerangka β

Matriks transformasi koordinat bersifat *orthonormal*, yaitu

$$(C_\alpha^\beta)^{-1} = (C_\alpha^\beta)^T = C_\beta^\alpha \quad (2.2)$$

Matriks ini didapat dengan mengubah sudut euler menjadi berbentuk urutan matriks rotasi yang diputar dengan urutan Z-Y-X.

$$C_{\alpha}^{\beta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$C_{\alpha}^{\beta} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta c\psi & c\theta c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix} \quad (2.4)$$

Konversi dari matriks menjadi sudut euler adalah

$$\phi = \text{atan}_2(C_{\alpha 2,3}^{\beta}, C_{\alpha 3,3}^{\beta}) \quad (2.5a)$$

$$\theta = -\text{asin}(C_{\alpha 1,3}^{\beta}) \quad (2.5b)$$

$$\psi = \text{atan}_2(C_{\alpha 1,3}^{\beta}, C_{\alpha 1,1}^{\beta}) \quad (2.5c)$$

2.4 Model Bumi[1]

Untuk sebagian besar aplikasi dari INS, diperlukan solusi posisi dengan mempertimbangkan permukaan Bumi. Untuk mendapatkannya, diperlukan permukaan referensi yang harus diperhatikan yang berkaitan dengan pusat dan sumbu Bumi. Seperangkat koordinat untuk mengekspresikan posisi permukaan itu, lintang, bujur, dan ketinggian. Untuk itu diperlukan pemetaan, metode proyeksi koordinat ini ke permukaan rata. Untuk menransformasi pengukuran yang merujuk Bumi sebagai kerangka inersia, maka kecepatan rotasi Bumi juga harus diperhatikan.

2.4.1 Model Elipsoid dari Permukaan Bumi

Kerangka koordinat ECEF (*Earth-Centered Earth Fixed*) memungkinkan pengguna untuk melakukan navigasi dengan titik pusat bumi sebagai pusatnya. Namun, karena permukaan bumi tidak rata membuat pengguna mengalami kesulitan dalam menentukan posisi. Sedangkan memodelkan permukaan bumi yang tidak rata tersebut akan membutuhkan ruang penyimpanan data yang sangat besar dan algoritma yang rumit. Sehingga permukaan bumi didekatkan dengan sebuah bentuk umum yang nantinya akan dicocokkan dengan rata-rata tinggi permukaan laut (*mean sea level*).

Model dari permukaan bumi yang paling sering digunakan untuk navigasi adalah elipsoid, atau yang lebih dikenal dengan bentuk lingkaran yang datar pada kutub-kutubnya (*oblate spheroid*). Model elipsoid menggambarkan adanya sumbu *north-south* (z^e) yang simetris secara rotasional dan simetris terhadap planar equator. Planar equator (garis ekuator) sendiri terdefinisi menjadi dua macam, yaitu *Equatorial Radius* (R_0) atau jarak dari titik pusat bumi menuju titik manapun pada equator (lintang 0^0), yang merupakan bagian permukaan terjauh dari titik pusat bumi. Sedangkan *Polar Radius* (R_p) merupakan jarak dari titik pusat bumi menuju salah satu pole atau kutub, yang merupakan bagian permukaan terdekat dari titik pusat bumi.

Elipsoid biasanya didefinisikan dalam hubungan antara *Equatorial Radius* dan dengan esentrisme dari Elipsoid (*Eccentricity of Ellipsoid*, e), atau pemerataan dari elipsoid (*Flattening of Ellipsoid*, f).

$$e = \sqrt{1 - \frac{R_p^2}{R_0^2}} \quad (2.6)$$

$$f = \frac{R_0 - R_p}{R_0} \quad (2.7)$$

Dan keduanya saling berhubungan,

$$e = \sqrt{2f - f^2} \quad (2.8)$$

$$f = 1 - \sqrt{1 - e^2} \quad (2.9)$$

Posisi dalam kartesian dari sebuah titik, S , pada permukaan elipsoid adalah $r_{eS}^e = (x_{eS}^e, y_{eS}^e, z_{eS}^e)$. Jarak dari poin tersebut dari pusat bumi dikenal sebagai geosentrik radius dan ditulis sebagai berikut:

$$r_{eS}^e = |r_{eS}^e| = \sqrt{(x_{eS}^e)^2 + (y_{eS}^e)^2 + (z_{eS}^e)^2} \quad (2.10)$$

Kemudian akan lebih mudah untuk mendefinisikan magnitude dari proyeksi resultan r_{eS}^e ke dalam garis ekuatorial sebagai β_{eS}^e . Sehingga,

$$\beta_{eS}^e = \sqrt{(x_{eS}^e)^2 + (y_{eS}^e)^2} \quad (2.11)$$

Perpotongan dari Elipsoid pada gambar 2.6 adalah garis vertikal yang mengandung r_{eS}^e . Sehingga z_{eS}^e dan β_{eS}^e terikat oleh persamaan elips berikut:

$$\left(\frac{\beta_{eS}^e}{R_0}\right)^2 + \left(\frac{z_{eS}^e}{R_p}\right)^2 = 1 \quad (2.12)$$

$$\left(\frac{x_{eS}^e}{R_0}\right)^2 + \left(\frac{y_{eS}^e}{R_0}\right)^2 + \left(\frac{z_{eS}^e}{R_p}\right)^2 = 1 \quad (2.13)$$

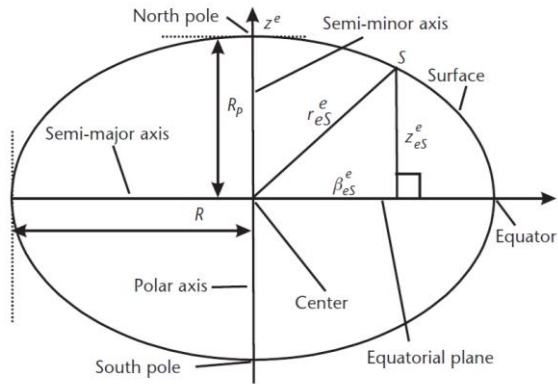
Selain digunakan sebagai referensi untuk menentukan posisi, model elipsoid juga digunakan sebagai basis untuk mendefinisikan kerangka navigasi lokal, sebagaimana arah bawah (sumbu *down*) dari kerangka ini didefinisikan sebagai garis normal dari elipsoid, mengarah menuju garis ekuatorial (khatulistiwa).

Perkembangan teknologi satelit navigasi telah memudahkan pengukuran posisi dari titik-titik yang berada di seluruh permukaan bumi berdasarkan referensi yang sama, yaitu konstelasi dari satelit, hal ini mengarah pada pengembangan model elipsoid global. Ada dua standart dalam pemodelan tersebut, yaitu *World Geodetic System 1984 (WGS84)* dan *International Terrestrial Reference Kerangka (ITRF)*. Kedua datum ini sama-sama memiliki titik origin berada pada titik pusat massa bumi dan mendefinisikan rotasi bumi menggunakan *IRP/CTP (IERS Reference Pole/Conventional Terrestrial Pole)*.

Selain mendefinisikan ECEF koordinat kerangka dan model elipsoid, WGS84 juga menyediakan model Geoid bumi, model medan gravitasi, dan beberapa konstanta-konstanta dasar. WGS84 mendefinisikan model elipsoid dalam hubungan antara *Equatorial Radius* dan *flattening*. *Polar Radius* serta *eccentricity* juga dapat diturunkan. Nilai-nilainya adalah sebagai berikut.

Tabel 2. 1 Nilai Parameter WGS84

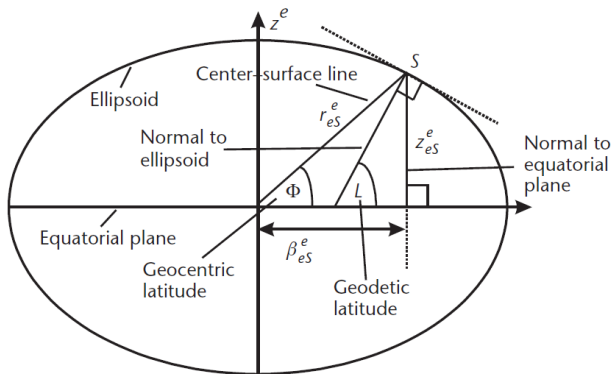
Parameter	Nilai WGS 84
<i>Equatorial Radius, R₀</i>	6,378,137.0 m
<i>Flattening, f</i>	0.003352810664747
<i>Polar Radius, R_p</i>	6,356,752.3142 m
<i>Eccentricity, e</i>	0.0818191908425



Gambar 2. 6 Model elipsoid sebagai representasi permukaan Bumi[1]

2.4.2 Posisi pada Lengkungan Bumi

Posisi wahana terhadap permukaan Bumi dideskripsikan dengan tiga sumbu koordinat yang saling tegak lurus, dan searah dengan sumbu-sumbu kerangka navigasi lokal.



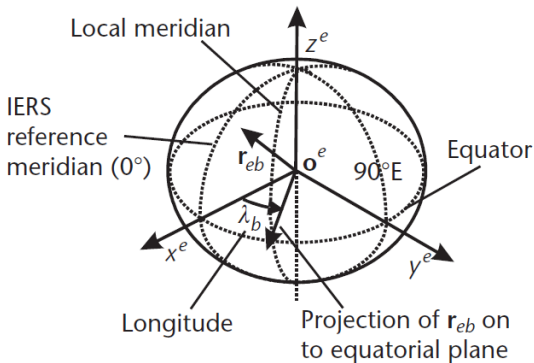
Gambar 2. 7 Geocentric dan Geodetic Latitude[1]

Jarak dari tubuh wahana dideskripsikan dari suatu permukaan bersama dengan garis normal menuju permukaan tersebut disebut dengan *height* atau *altitude*. Sumbu koordinat north-south pada suatu titik di

permukaan dimana garis normal tersebut berpotongan disebut *latitude* atau lintang, sedangkan koordinat pada titik tersebut dalam sumbu *West-East* disebut *longitude* atau bujur. Karena orientasi dari ketiga sumbu ini terhadap bumi bervariasi di setiap tempat, maka *latitude*, *longitude* dan *height* disebut sebagai posisi *curvilinear*.

Geocentric Latitude Φ seperti yang digambarkan pada gambar 2.7 adalah sudut dari perpotongan sebuah garis dari titik massa bumi (titik tengah) menuju ke sebuah titik di permukaan dari elipsoid dengan garis equatorial. Untuk setiap jenis latitude yang ada, konversi nilai latitude adalah positif di belahan bumi utara dan negatif di belahan bumi selatan.

Geodetic Latitude L juga seperti yang digambarkan pada gambar 2.7 adalah sudut antara perpotongan antara garis normal yang ditarik dari permukaan elipsoid dengan garis equatorial. Karena *geodetic latitude* didefinisikan dengan garis normal dari permukaan elipsoid, maka juga bisa didapat dari gradient dari permukaan.



Gambar 2. 8 Ilustrasi Longitude[1]

Longitude λ pada Gambar 2.8 adalah sudut yang diturunkan pada *plane equatorial* antara bidang meridian dengan titik yang diinginkan (point of interest) dan IRM/CZM (IERS Reference Meridian / Conventional Zero Meridian). IRM didefinisikan sebagai nilai rata-rata dari penentuan latitude nol dari *latitude-latitude* terobservasi di berbagai titik di bumi. Nilainya kurang lebih sama/setara dengan nilai meridian nol pada Greenwich, London. Longitude bernilai positif di belahan bumi bagian timur (menurut IRM) dan negatif di belahan bumi bagian barat.

Pada titik ini, sangat berguna untuk mendefinisikan jari-jari dari kelengkungan elipsoid. Radius dari kelengkungan untuk pergerakan

north-south juga dikenal dengan jari-jari meridian dari kelengkungan (*curvature*) R_N , dan menentukan laju dari perubahan geodetic latitude bersama dengan meridian.

$$R_N(L) = \frac{R_0(1 - e^2)}{(1 - e^2 \sin^2 L)^{3/2}} \quad (2.14)$$

Sedangkan jari-jari kelengkungan untuk pergerakan *west-east* dikenal sebagai jari-jari transvers dari kelengkungan R_E , dan menentukan perubahan dari longitude bersama dengan permukaan yang tegak lurus dengan meridian.

$$R_E(L) = \frac{R_0}{\sqrt{1 - e^2 \sin^2 L}} \quad (2.15)$$

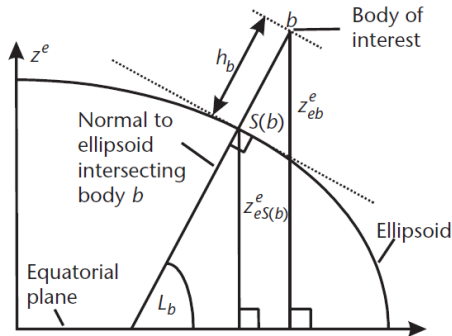
Posisi *curvilinear* dari body, b , dapat dituliskan dalam bentuk vektor sebagai $\mathbf{p}_b = [L_b \ \lambda_b \ h_b]$. Sedangkan turunan waktu dari posisi curvilinear adalah fungsi linier dari kecepatan di sumbu-sumbu kerangka navigasi lokal dengan referensi bumi.

$$\dot{L}_b = \frac{v_{eb,N}^n}{R_N(L_b) + h_b} \quad (2.16)$$

$$\dot{\lambda}_b = \frac{v_{eb,E}^n}{(R_E(L_b) + h_b) \cos L_b} \quad (2.17)$$

$$\dot{h}_b = -v_{eb,D}^n \quad (2.18)$$

Hal ini membuat posisi *curvilinear* dapat diintegalkan secara langsung dari kecepatan tanpa perlu menggunakan posisi kartesian sebagai penghubung.



Gambar 2. 9 Ketinggian geodesi dan lintang geodesi dari tubuh[1]

2.4.3 Rotasi Bumi

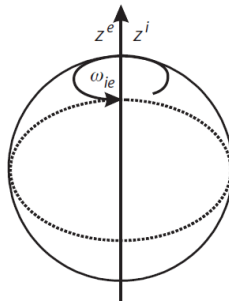
Bumi berrotasi dengan ruang angkasa sebagai referensinya, searah jarum jam dengan sumbu z yang sama untuk kerangka ECI dan ECEF hal ini ditunjukkan pada gambar 2.10. Sehingga vektor rotasi bumi yang terjadi pada sumbu-sumbu tersebut adalah sebagai berikut:

$$\boldsymbol{\omega}_{ie}^e = \begin{pmatrix} 0 \\ 0 \\ \omega_{ie} \end{pmatrix} \quad (2.19)$$

Vektor rotasi bumi yang terjadi pada sumbu-sumbu kerangka navigasi lokal adalah fungsi dari geodetic latitude:

$$\boldsymbol{\omega}_{ie}^n = \begin{pmatrix} \omega_{ie} \cos L_b \\ 0 \\ -\omega_{ie} \sin L_b \end{pmatrix} \quad (2.20)$$

Untuk keperluan navigasi, diasumsikan laju rotasi konstan. Nilai dari kecepatan sudut bumi dalam standar WGS84 adalah $\omega_{ie} = 7.292115 * 10^{-5}$ rad/s.



Gambar 2. 10 Rotasi Bumi dalam Kerangka ECI ECEF[1]

2.4.5 Gravitasi dan Gaya Spesifik[1]

Gaya spesifik adalah adalah gaya non-gravitasi per satuan massa dari body (benda), diukur dengan acuan terhadap inertial kerangka. Gaya ini tidak ada arti apapun pada kerangka lainnya, meskipun bisa

diekspresikan di sumbu manapun. Gravitasi adalah gaya tarik-menarik antar massa benda.

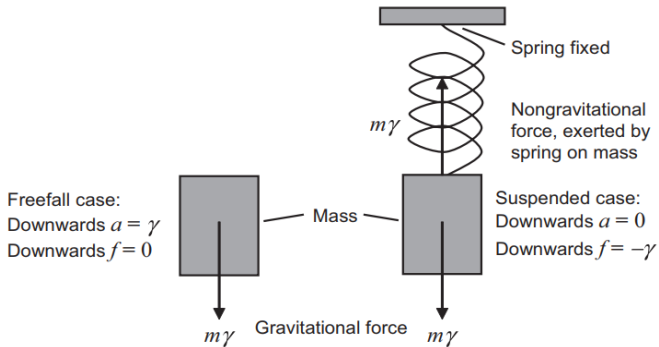
Gaya spesifik adalah apa yang manusia dan instrumen pengukuran rasakan, sementara gravitasi tidak bisa dirasakan karena bekerja pada setiap titik secara bersamaan ditunjukkan pada gambar 2.11. Sensasi bernama berat (*weight*) dirasakan karena adanya gaya yang melawan gravitasi. Pada keadaan benda jatuh bebas, tidak ada berat yang terasa karena gaya spesifik adalah nol. Sebaliknya, pada keadaan percepatan nol, reaksi terhadap gravitasi dapat terasa, dan gaya spesifik adalah sama dengan percepatan karena gravitasi namun memiliki arah yang berlawanan. Maka dari itu, gaya spesifik f bergantung pada percepatan a dan percepatan karena gaya gravitasi, γ

$$f_{ib}^{\gamma} = a_{ib}^{\gamma} - \gamma_{ib}^{\gamma} \quad (2.21)$$

Gaya spesifik adalah nilai yang terukur oleh akselerometer. Pengukuran terjadi pada kerangka tubuh oleh tiga akselerometer; sehingga, gaya spesifik yang terukur adalah f_{ib}^b .

Sebuah objek yang berada dalam state diam terhadap kerangka yang berotasi, semacam ECEF, memiliki $v_{eb}^e = 0$; $a_{eb}^e = 0$. Datum WGS84 menyediakan sebuah model sederhana dari percepatan karena gravity pada permukaan elipsoid sebagai fungsi dari *latitude*, yaitu

$$g_0(L) \approx 9.7803253359 \frac{(1+0.001931853 \sin^2 L) m}{\sqrt{1-e^2 \sin^2 L} s^2} \quad (2.22)$$



Gambar 2. 11 Gaya yang terjadi pada pegas[1]

2.5 Unscented Kalman Filter[1,5]

UKF adalah Filter Kalman yang berbasis *Unscented Transformation* yang digunakan untuk estimasi state non linier. UKF mengatasi kelemahan dari aproksimasi dari EKF, distribusi state dinyatakan dalam *Gaussian Random Variable*, namun direpresentasikan dengan menggunakan himpunan minimum titik sample yang dipilih secara hati-hati.

Titik sampel ini secara lengkap menangkap atau merepresentasikan mean sebenarnya, dan kovarians sebenarnya dari GRV (*Gaussian Random Variable*) dan saat berpropagasi melalui sistem nonlinier, dan menangkap atau merepresentasikan mean posterior, dan kovarians secara akurat sampai orde ke 3 dari deret Taylor untuk setiap ketidaklinieran sistem. Untuk memahami hal ini kita harus memahami tentang *Unscented Transformation*.

Unscented Transformation (UT) adalah suatu metode untuk menghitung statistika dari variable acak X (dimensi L) yang mengalami transformasi non linier. Meninjau variable acak yang berpropagasi fungsi nonlinier $y = f(x)$. Diasumsikan variable X memiliki mean \bar{x} dan kovarians P_x . Untuk menghitung statistika dari y , maka kita membuat sebuah matriks χ dari $2L + 1$ sigma vector χ_i (dengan bobot W_i yang sesuai), sesuai dengan persamaan berikut:

$$\chi_0 = \bar{x} \quad (2.23)$$

$$\chi_i = \bar{x} + \left(\sqrt{(L + \lambda) P_x} \right)_i, i = 1, 2, \dots, L \quad (2.24)$$

$$\chi_i = \bar{x} - \left(\sqrt{(L + \lambda) P_x} \right)_{i-L}, i = L + 1, \dots, 2L \quad (2.25)$$

$$W_0^{(m)} = \frac{\lambda}{L + \lambda} \quad (2.26)$$

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta \quad (2.27)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L + \lambda)}, i = 1, 2, \dots, 2L \quad (2.28)$$

Dimana $\lambda = \alpha^2(L + \kappa) - L$ adalah scaling parameter, α menentukan penyebaran *sigma points* dari \bar{x} dan biasanya diberi nilai kecil positif (seperti contoh, $1e-3$). κ adalah parameter sekunder yang biasanya di beri nilai 0, β menentukan pengetahuan prior dari distribusi

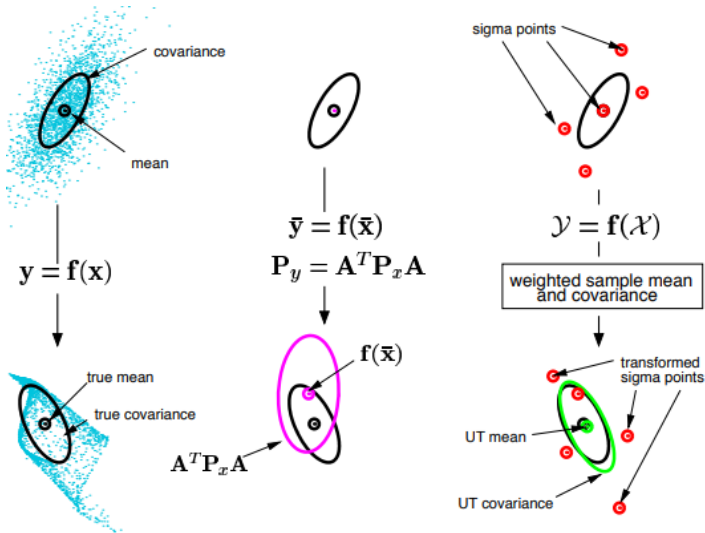
X (untuk distribusi Gauss, $\beta = 2$), $(\sqrt{(L + \lambda)P_x})_i$ adalah matriks kolom ke i . Sigma vector ini berpropagasi melalui fungsi nonlinear $y_i = g(x_i)$, $i = 0, \dots, 2L$.

Nilai mean dan kovarians di dekati dengan mean sample berbobot dan kovarians dari posterior *sigma point*.

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^{(m)} y_i \quad (2.29)$$

$$P_y \approx \sum_{i=0}^{2L} W_i^{(c)} \{y_i - \bar{y}\} \{y_i - \bar{y}\}^T \quad (2.30)$$

Unscented Transform dapat melakukan hal yang lebih baik dalam mengestimasi *mean* dan *sigma point* dari variable acak yang terkena transformasi nonlinear dibanding pendekatan linearisasi. Perlu digaris bawahi bahwa UT hanya benar-benar memilih *mean* dan sampel *sigma point* (dari matriks kovarians) sebelum mengalami transformasi, kemudian mentransformasikannya melalui persamaan nonlinear dan mengestimasi *mean* dan kovariansnya kembali seperti ditunjukkan gambar 2.12.



Gambar 2. 12 Pendekatan metode EKF dan UKF untuk estimasi state nonlinear[5]

2.5.1. Algoritma *Unscented Kalman Filter*

Berikut adalah algoritma dari *Unscented Kalman Filter*:

- a. Estimasi state (state) adalah berdistribusi Gauss $\hat{x} = N(x, P_x)$. Kemudian secara deterministik kita memilih sigma point.

$$\mathcal{X}_{k-1} = \left[\hat{x}_{k-1} \quad \hat{x}_{k-1} + \sqrt{(N+\lambda)P_{k-1}} \quad \hat{x}_{k-1} - \sqrt{(N+\lambda)P_{k-1}} \right] \quad (2.31)$$

- b. Untuk state prediksi, sigma points tersebut di propagasikan dengan menggunakan fungsi nonlinier $\chi_k^- = \phi(\chi_{k-1}^-)$ kemudian membangun ulang estimasi state *apriori* dan error kovariansnya.

$$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_k^- \quad (2.32)$$

$$P_k^- = \left\{ \sum_{i=0}^{2L} W_i^{(c)} \{ \chi_k^- - \hat{x}_k^- \} \{ \chi_k^- - \hat{x}_k^- \}^T \right\} + Q_k \quad (2.33)$$

- c. Kemudian kita sampling kembali dari state prediksi, untuk digunakan sebagai prediksi selanjutnya

$$\mathcal{X}_k = \left[\hat{x}_k \quad \hat{x}_k + \sqrt{(N+\lambda)P_k} \quad \hat{x}_k - \sqrt{(N+\lambda)P_k} \right] \quad (2.34)$$

- d. Untuk update state pengukuran, sama dengan update proses sistem, namun melalui fungsi nonlinier $\chi_k^- = h(\chi_{k-1}^-)$ kemudian membangun ulang estimasi state *apriori* dan error kovariansnya.

$$z_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_k^- \quad (2.35)$$

$$P_{zzk}^- = \left\{ \sum_{i=0}^{2L} W_i^{(c)} \{ \chi_k^- - z_k^- \} \{ \chi_k^- - z_k^- \}^T \right\} + R_k \quad (2.36)$$

$$P_{xz_k}^- = \left\{ \sum_{i=0}^{2L} W_i^{(c)} \{ \chi_k^- - \hat{x}_k^- \} \{ \chi_k^- - z_k^- \}^T \right\} \quad (2.37)$$

$$K_k = (P_{xz_k}^-)_k (P_{zzk}^-)^{-1} \quad (2.38)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - z_k^-) \quad (2.39)$$

$$P_k = P_k^- - K_k (P_{zzk}^-)_k K_k^T \quad (2.40)$$

2.5.2 *Closed-Loop Kalman Filter*

Model sistem linier adalah asumsi desain Filter Kalman standar. Namun, di banyak aplikasi navigasi, seperti integrasi, penyalarsan, dan kalibrasi dari INS, model sistem yang sebenarnya tidak linier. Salah satu

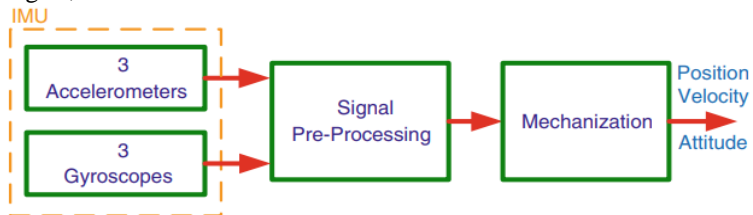
solusinya adalah menggunakan versi modifikasi dari algoritma Filter Kalman, seperti EKF atau UKF. Namun, sering kali diatasi dengan mengabaikan orde yang lebih tinggi dari model sistem.

Teknik yang umum untuk mendapatkan kinerja terbaik dari kesalahan state Filter Kalman dengan pendekatan linearitas yang diterapkan pada model sistem adalah implementasi untai tertutup. Di sini, kesalahan yang diperkirakan oleh Filter Kalman diumpangkan kembali disetiap iterasi, atau secara berkala, untuk memperbaiki sistem itu sendiri, mengonkkan setiap state Kalman disetiap proses. Proses umpan balik ini membuat nilai state dari Filter Kalman kecil, meminimalkan efek dari pengabaian produk orde tinggi di state model sistem. Sebaliknya, dalam implementasi untai terbuka, ketika tidak ada umpan balik, state pada umumnya akan semakin besar seiring berjalannya waktu.

2.6 Sensor Inersia[1,2]

Sensor inersia terdiri dari akselerometer dan giroskop. Sensor inersia ini berguna sebagai solusi navigasi Akselerometer mengukur gerakan linier pada tiga sumbu orthogonal, dan yang mana giroskop mengukur kecepatan angular pada tiga sumbu ortogonal. Pada umumnya sumbu-sumbu pada kedua sensor ini saling sejajar, saling berimpit pada titik pusat yang sama. Kedua sensor ini terpasang pada tubuh dari IMU (*Inertial Measurements Unit*).

Berikut adalah mekanisme INS dari sensor IMU menjadi solusi navigasi, serta skema IMU.



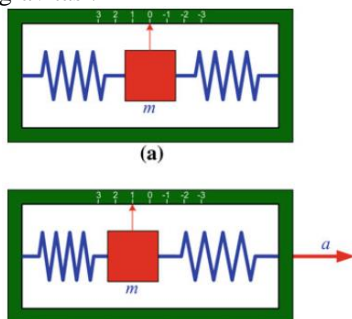
Gambar 2. 13 Skema INS dan sensor IMU[2]

2.6.1 Akselerometer

Akselerometer merupakan sensor inersia yang membaca data berupa percepatan yang dialami kendaraan atau benda bergerak yang diukur. Seperti yang terlihat pada gambar 2.14, akselerometer terdiri atas

proof mass, dari titik keseimbangan dikarenakan adanya percepatan yang terjadi.

Pada saat kondisi diam, akselerometer dengan sumbu vertikal akan membaca percepatan gravitasi. Pada kondisi wahana bergerak maka, data yang terbaca oleh akselerometer pada sumbu vertikal adalah $f = a - g$, dimana f adalah gaya spesifik, sedangkan a adalah percepatan wahana dan g adalah gaya gravitasi.



Gambar 2. 14 Akselerometer statis dan mengalami percepatan[5]

2.6.2 Girooskop

Girooskop merupakan sensor yang membaca perubahan sudut pada benda/kendaraan terhadap sumbu kartesian 3 dimensi. Apabila dibentuk sebuah sumbu kartesian tiga dimensi pada benda/kendaraan dengan titik pusat berada pada pusat massa kendaraan, dengan sumbu-x adalah arah depan (*forward*), sumbu-y adalah arah kanan kendaraan (*right*) dan sumbu-z adalah arah bawah (*down*) menciptakan sistem kartesian dengan kaedah tangan kanan.

Girooskop membaca perubahan sudut yang terjadi saat kendaraan bergerak, perubahan sudut tersebut terjadi terhadap *body* kendaraan. Pada kondisi diam, maka pada girooskop akan mengukur kecepatan rotasi Bumi.

2.6.3 Karakteristik Kesalahan

Kesalahan pada sensor IMU ini ada beberapa macam, bisa dikategorikan menjadi 3, yaitu: keterulangan, stabilitas, dan *drift*. Keterulangan adalah kemampuan sensor untuk menyediakan keluaran yang sama untuk aplikasi yang berulang dari input yang sama, dengan asumsi faktor-faktor pada lingkungan yang selalu konstan. Stabilitas adalah kemampuan sensor untuk menyediakan iutput saat mengukur input

yang konstan selama beberapa periode. *Drift* adalah juga sering disebut sebagai perubahan yang terjadi pada pengukuran sensor saat tidak ada perubahan pada masukan.

2.6.3.1 Bias

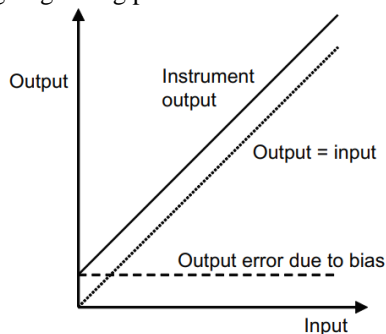
Bias adalah kesalahan konstan yang ditunjukkan oleh semua akselerometer dan gyro seperti pada gambar 2.15. Itu independen dari gaya spesifik yang mendasari dan kecepatan sudut. Dalam kebanyakan kasus, bias adalah istilah dominan dalam kesalahan keseluruhan instrumen inersia. Akselerometer dan bias gyro dari IMU, mengikuti kalibrasi sensor dan kompensasi, dilambangkan dengan vektor $\mathbf{b}_a = (b_{ax}, b_{ay}, b_{az})$ dan $\mathbf{b}_g = (b_{gx}, b_{gy}, b_{gz})$. Kesalahan IMU selalu dinyatakan dalam sumbu body, jadi superscript \mathbf{b} dapat dihilangkan. Mudah untuk membagi kedua bias ini menjadi 2, yaitu :

$$\mathbf{b}_a = \mathbf{b}_{as} + \mathbf{b}_{ad} \tag{2.41}$$

$$\mathbf{b}_g = \mathbf{b}_{gs} + \mathbf{b}_{gd} \tag{2.42}$$

Komponen statis, disebut juga sebagai bias konstan, bias saat on, atau bias berulang meliputi variasi *run to run* dari bias instrumen ditambah residual dari bias konstan setelah kalibrasi sensor. Hal ini terjadi konstan, selama IMU dalam kondisi beroperasi tapi bervariasi dari setiap kali menyalakan IMU.

Komponen dinamik, disebut juga variasi bias *in-run* atau bias tak stabil, bervariasi setiap periode dalam hitungan menit, dan juga meliputi residu yang tergantung pada suhu setelah kalibrasi sensor.



Gambar 2. 15 Bias pada sensor IMU[1]

2.6.3.2. Kesalahan Faktor Skala dan Pengopelan Silang

Semua jenis akselerometer dan giroskop menunjukkan kesalahan bias, faktor skala dan *crosscoupling* dan *noise* acak pada batas tertentu., Setiap sumber kesalahan sistematis memiliki empat komponen: kontribusi tetap, variasi tergantung suhu, variasi *run-to-run*, dan variasi *in-run*. Kontribusi kesalahan tetap hadir setiap kali sensor digunakan dan dikoreksi oleh prosesor IMU menggunakan data kalibrasi laboratorium. Komponen yang bergantung pada suhu juga dapat diperbaiki oleh IMU menggunakan data kalibrasi laboratorium. Ketika ini tidak diperbaiki, sensor biasanya akan menunjukkan variasi kesalahan sistematis selama beberapa menit pertama operasi sementara sensor memanaskan ke suhu operasi normal.

Variasi *run-to-run* dari setiap sumber kesalahan menghasilkan kontribusi terhadap kesalahan total, yang berbeda setiap kali sensor digunakan, tetapi tetap konstan dalam setiap proses. Hal ini tidak dapat dikoreksi oleh prosesor IMU, tetapi dapat dikalibrasi oleh algoritma penyalarsan INS dan / atau integrasi setiap kali IMU digunakan.

Kesalahan faktor skala adalah eror yang mengindikasikan eror kecepatan sudut atau percepatan linear pada IMU ketika sedang bergerak. Kesalahan ini bersifat proporsional terhadap percepatan linear dan kecepatan sudut sebenarnya dari IMU seperti ditunjukkan pada gambar 2.16. Berikut ini merupakan matriks kesalahan faktor skala dan pengopelan silang:

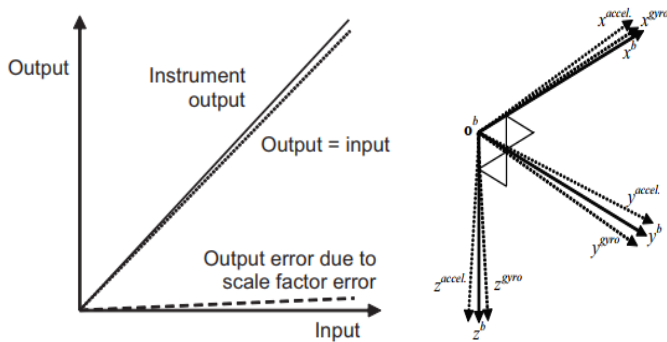
$$\mathbf{M}_a = \begin{pmatrix} s_{a,x} & m_{a,xy} & m_{a,xz} \\ m_{a,yx} & s_{a,y} & m_{a,yz} \\ m_{a,zx} & m_{a,zy} & s_{a,z} \end{pmatrix} \quad (2.43)$$

$$\mathbf{M}_g = \begin{pmatrix} s_{g,x} & m_{g,xy} & m_{g,xz} \\ m_{g,yx} & s_{g,y} & m_{g,yz} \\ m_{g,zx} & m_{g,zy} & s_{g,z} \end{pmatrix} \quad (2.44)$$

Sehingga didapatkan koreksi percepatan a dan kecepatan sudut ω adalah

$$\hat{\mathbf{f}}_{ib}^b = (\mathbf{I}_3 + \hat{\mathbf{M}}_a)^{-1} (\tilde{\mathbf{f}}_{ib}^b - \hat{\mathbf{b}}_a) \approx (\mathbf{I}_3 + \hat{\mathbf{M}}_a)^{-1} \tilde{\mathbf{f}}_{ib}^b - \hat{\mathbf{b}}_a \quad (2.45)$$

$$\begin{aligned} \hat{\boldsymbol{\omega}}_{ib}^b &= (\mathbf{I}_3 + \hat{\mathbf{M}}_g)^{-1} (\tilde{\boldsymbol{\omega}}_{ib}^b - \hat{\mathbf{b}}_g - \hat{\mathbf{G}}_g \hat{\mathbf{f}}_{ib}^b) \\ &\approx (\mathbf{I}_3 + \hat{\mathbf{M}}_g)^{-1} \tilde{\boldsymbol{\omega}}_{ib}^b - \hat{\mathbf{b}}_g - \hat{\mathbf{G}}_g \hat{\mathbf{f}}_{ib}^b \end{aligned} \quad (2.46)$$



Gambar 2. 16 Kesalahan faktor skala dan pengopelan silang[1]

2.6.3.3. Kesalahan Acak

Spektrum dari akselerometer dan *noise* giroskop untuk frekuensi dibawah 1 Hz di dekati dengan *white noise*, dan standar deviasi dari rata-rata gaya spesifik dan kecepatan sudut bervariasi pada proporsi invers dari akar kuadrat dari rata-rata waktu. Gangguan sensor inersia adalah dinyatakan dalam bentuk akar dari PSD. Satuan umumnya adalah mg/\sqrt{Hz} untuk akselerometer dimana $\frac{1mg}{\sqrt{Hz}} = 9.80665e - 06ms^{-1.5}$, dan untuk giro adalah $^{\circ}/\sqrt{hr}$ dimana $\frac{1^{\circ}}{\sqrt{hr}} = 2.909e - 04rads^{-0.5}$. Standar deviasi dari gangguan acak didapatkan dengan mengalikan dengan akar kuadrat PSD dengan akar dari kecepatan sample kemudian dibagi dengan interval sampling. *White noise* tidak bisa dikalibrasi dan dikompensasi karena tidak ada korelasi antara nilai lalu dan nilai yang akan datang.

Sensor MEMS dapat juga menunjukkan *noise* frekuensi tinggi, Sensor MEMS juga dapat menunjukkan *noise* frekuensi tinggi yang signifikan. Dalam bingkai tubuh IMU, *noise* ini rata-rata melebihi urutan sedetik, jadi lewat output sensor melalui persamaan navigasi inersia akan menghilangkan sebagian besar efek dari *noise* ini. Namun, jika IMU sudah menyala, gangguan tidak akan keluar ke batas yang sama dalam bingkai yang digunakan untuk menghitung persamaan navigasi. Akibatnya, berhati-hati harus dilakukan dalam memilih sensor ini untuk aplikasi yang sangat dinamis.

Menerapkan filter *lowpass* langsung ke sensor atau output IMU mengurangi *noise* berfrekuensi tinggi terlepas dari dinamika. Metode

menggunakan teknik penyaringan wavelet atau jaringan saraf tiruan (JST) dapat memberikan kinerja yang lebih baik daripada penyaringan lowpass konvensional. Namun, semua teknik ini sama-sama memperkenalkan *time lag* dan mengurangi bandwidth sensor yang efektif. Salah satu solusi untuk masalah yang terakhir adalah untuk memvariasikan filter *passband*

2.7 Navigasi Inersia[1,2]

Sistem navigasi akan memberikan informasi tentang posisi dengan mengintegrasikan nilai kecepatan, yang mana, diperoleh dari mengintegrasikan percepatan, begitu juga dengan orientasi yang diperoleh dengan cara mengintegrasikan kecepatan sudut. Hal inilah yang disebut dengan mekanisasi. Pada mekanisasi ini, terdapat beberapa kerangka yang bisa digunakan sebagai kerangka penyelesaian, yaitu kerangka ECI, kerangka ECEF, dan kerangka lokal. Pada tugas akhir ini menggunakan kerangka lokal dikarenakan kerangka lokal lebih mudah dipahami oleh pengguna, namun juga lebih rumit dalam algoritmanya.

2.7.1 Pembaruan Orientasi

Pada pembaruan orientasi, persamaan navigasi menggunakan solusi posisi dan kecepatan begitu pula pengukuran kecepatan sudut yang digunakan untuk memperbarui C_b^n . Hal ini diperlukan karena orientasi *north*, *east*, dan *down* berubah sebagaimana sistem navigasi bergerak terhadap bumi. Turunan terhadap waktu dari matriks transformasi koordinat adalah sebagai berikut:

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b \quad (2.49)$$

$$\dot{C}_b^n = C_b^n \Omega_{ib}^b (\Omega_{ie}^n + \Omega_{en}^n) C_b^n \quad (2.50)$$

Dimana Ω_{ib}^b adalah kecepatan sudut yg diukur sensor giro, kemudian Ω_{ie}^n berhubungan dengan rotasi bumi, dan Ω_{en}^n dikenal juga sebagai kecepatan transpor.

$$\Omega_{ie}^n = \omega_{ie} \begin{pmatrix} 0 & \sin L_b & 0 \\ -\sin L_b & 0 & -\cos L_b \\ 0 & \cos L_b & 0 \end{pmatrix} \quad (2.51)$$

$$\Omega_{en}^n = \omega_{ie} \begin{pmatrix} 0 & \frac{v_{eb,E}^n \tan L_b}{R_E L_b + h_b} & -\frac{v_{eb,N}^n}{R_N L_b + h_b} \\ -\frac{v_{eb,E}^n \tan L_b}{R_E L_b + h_b} & 0 & -\frac{v_{eb,E}^n}{R_N L_b + h_b} \\ \frac{v_{eb,N}^n}{R_N L_b + h_b} & \frac{v_{eb,E}^n}{R_N L_b + h_b} & 0 \end{pmatrix} \quad (2.52)$$

Sehingga dengan pendekatan orde 1, persamaan 2.48 menjadi

$$\mathbf{C}_b^n(+)\tau_i \approx \mathbf{C}_b^n(-)(\mathbf{I}_3 + \Omega_{ib}^b \tau_i) - (\Omega_{ie}^n(-) + \Omega_{en}^n(-))\mathbf{C}_b^n(-)\tau_i \quad (2.53)$$

2.7.2 Pembaruan Kecepatan

Kemudian untuk kecepatan, persamaanya adalah sebagai berikut

$$\mathbf{v}_{eb}^n = \mathbf{C}_e^n \mathbf{v}_{eb}^e \quad (2.54)$$

$$\dot{\mathbf{v}}_{eb}^n = \dot{\mathbf{C}}_e^n \mathbf{v}_{eb}^e + \mathbf{C}_e^n \dot{\mathbf{v}}_{eb}^e \quad (2.55)$$

$$\dot{\mathbf{v}}_{eb}^n = \dot{\mathbf{C}}_e^n \mathbf{v}_{eb}^e + \mathbf{C}_e^n \dot{\mathbf{v}}_{eb}^e \quad (2.56)$$

$$\dot{\mathbf{v}}_{eb}^n = \mathbf{f}_{ib}^n + \mathbf{g}_b^n(L_b, h_b) - (\Omega_{en}^n + 2\Omega_{ie}^n)\mathbf{v}_{eb}^n \quad (2.57)$$

$$\mathbf{v}_{eb}^n(+)=\mathbf{v}_{eb}^n(-)+(\mathbf{f}_{ib}^n \quad (2.58)$$

$$+\mathbf{g}_b^n(L_b(-), h_b(-)) - (\Omega_{en}^n(-) \quad (2.59)$$

$$+2\Omega_{ie}^n(-))\mathbf{v}_{eb}^n(-)\tau \quad (2.60)$$

2.7.3 Pembaruan Posisi

Untuk pembaruan posisi, persamaannya adalah sebagai berikut.

$$L_b(+)\approx L_b(-)+\frac{\tau_i}{2}\left(\left(\frac{v_{eb,N}^n(-)}{R_N(-)+h_b(-)}\right)+\left(\frac{v_{eb,N}^n(+)}{R_N(+)+h_b(+)}\right)\right) \quad (2.58)$$

$$\lambda(+)=\lambda(-)+\frac{\tau_i}{2}\left(\left(\frac{v_{eb,E}^n(-)}{(R_E(-)+h_b(-))\cos L_b(-)}\right)+\left(\frac{v_{eb,E}^n(+)}{(R_E(+)+h_b(+))\cos L_b(+)}\right)\right) \quad (2.59)$$

$$h_b(+)=h_b(-)-\frac{\tau_i}{2}(v_{eb,D}^n(-)+v_{eb,D}^n(+)) \quad (2.60)$$

2.7.4 Insialisasi dan Penyelarasan

Inisialisasi ini meliputi inisialisasi posisi dan kecepatan dari wahana, karena sistem navigasi inersia menggunakan metode *dead-reckoning* yang hanya mengolah informasi berupa perubahan-perubahan yang terjadi dari inisialisasi awal. Maka dari itu, inisialisasi biasanya dilakukan dengan bantuan sensor dari luar seperti perangkat GPS, radio navigasi terestrial, Posisi INS dapat diinisialisasi dari sistem navigasi lain.

Kecepatan dapat diinisialisasi hanya dengan mempertahankan sistem navigasi inersia stasioner dengan Bumi. Atau, sistem navigasi lain, seperti GNSS, radar Doppler, atau INS lainnya, dapat digunakan sebagai referensi. Masalah lebih lanjut untuk inisialisasi kecepatan adalah gangguan, getaran, dan tekstur. Sebagai contoh, ketika sistem navigasi inersia diasumsikan stasioner pada Bumi, wahana dapat terganggu oleh angin atau aktivitas manusia, seperti pengisian bahan bakar dan pemuatan. Untuk kapal, gerakan air juga menjadi masalah. Untuk inisialisasi dalam gerakan, lengan tuas antara INS dan navigasi referensi sistem dapat dipengaruhi oleh getaran dan getaran. Solusinya adalah melakukan pengukuran inisialisasi selama beberapa detik dan merata-ratanya. Posisi juga dapat dipengaruhi oleh tekstur dan getaran, tetapi besarnya biasanya kurang dari akurasi yang diperlukan.

Penyelarasan adalah inisialisasi orientasi awal dari suatu wahana pada saat kondisi diam. Inisialisasi pada penyelarasan sangatlah penting, dikarenakan kesalahan pada penyelarasan ini mengakibatkan kesalahan yang terakumulasi terus bila tidak ada sensor pengukuran yang dapat mengoreksi. Inisialisasi ini terbagi atas 2 tahap, yaitu *self-alignment* atau disebut juga *coarse-alignment*, dan *fine-alignment*. Pada *self-alignment* ini dibagi lagi menjadi beberapa tahap yaitu *leveling* dan *gyrocompassing*.

Pada *leveling*, saat wahana diam atau stasioner, maka yang dirasakan oleh akselerometer hanyalah gaya spesifik dari gravitasi. Oleh karena itu orientasi *roll* dan *pitch* awal dapat diketahui dengan metode ini. Namun keakurasian dari metode ini tergantung pada kesalahan pengukuran dari akselerometer. Getaran mesin, efek angin, dan kegiatan manusia, dapat mengganggu dalam proses *leveling* ini. Proses *leveling* dilakukan dengan metode seperti berikut

$$\theta_{nb} = \text{atan} \frac{f_{ib,x}^b}{\sqrt{f_{ib,y}^b{}^2 + f_{ib,z}^b{}^2}} \quad (2.61)$$

$$\phi_{nb} = \text{atan}_2 \left(-f_{ib,x}^b, -f_{ib,y}^b \right) \quad (2.62)$$

Pada *gyrocompassing*, saat wahana diam atau stasioner, maka yang dirasakan oleh giroskop hanyalah rotasi Bumi. Oleh karena itu orientasi *yaw* dapat ditentukan dengan rumus sebagai berikut

$$\sin(\psi_{nb}) = -\omega_{ib_y}^b \cos(\phi_{nb}) + \omega_{ib_z}^b \sin(\phi_{nb}) \quad (2.63)$$

$$\begin{aligned} \cos(\psi_{nb}) = & -\omega_{ib_x}^b \cos(\theta_{nb}) + \omega_{ib_y}^b \sin(\phi_{nb}) \sin(\theta_{nb}) \\ & + \omega_{ib_z}^b \cos(\phi_{nb}) \sin(\theta_{nb}) \end{aligned} \quad (2.64)$$

$$\psi_{nb} = \text{atan}_2(\sin(\psi_{nb}), \cos(\psi_{nb})) \quad (2.65)$$

Hanya dengan giroskop dengan kelas penerbangan maupun kelas kelautan saja yang mampu melakukan metode *gyrocompassing*. Sedangkan giro yang memiliki bias yang melebihi $5^\circ/hr$ tidak akan bisa melakukan metode ini.

Fine-alignment digunakan untuk memperbaiki kesalahan dari *self-alignment*. Pada metode ini kesalahan residual pada orientasi dirasakan dari pertumbuhan kesalahan pada kecepatan. Pada tugas akhir ini digunakan *quasi-stationary alignment*, yaitu wahana diasumsikan diam

Kebanyakan aplikasi navigasi inersia membutuhkan orientasi dengan akurasi 1 mrad atau lebih baik, jika ingin meminimalisasi drift posisi dan kecepatan. Kebanyakan teknik inisialisasi attitude tidak dapat mencapai akurasi ini. Sehingga diperlukan untuk melanjutkan inisialisasi dengan sebuah periode kalibrasi attitude yang biasa disebut *fine alignment*.

Pada teknik *fine alignment* atau penyelarasan halus, kesalahan *attitude* residu dideteksi melalui peningkatan kesalahan kecepatan. Misalnya, sebuah eror attitude 1 mrad pada *pitch* dan *roll* akan menyebabkan kesalahan kecepatan horizontal berkembang dengan kecepatan ~ 10 mm s⁻² dikarenakan kesalahan gravitasi.

Ada tiga teknik utama penyelarasan halus, yang masing-masing memiliki referensi berbeda.

a. *Quasi-stationary Alignment*

Quasi-stationary alignment mengasumsikan posisi telah di inisialisasi dan INS dalam kondisi stasioner terhadap bumi dan menggunakan *Zero Velocity Updates* (ZVUs).

b. *GNSS Alignment*

GNSS Alignment atau integrasi INS/GNSS, menggunakan posisi dan kecepatan yang didapat dari GNSS dan dapat beroperasi selama fasa navigasi ataupun fasa alignment.

c. *Transfer Alignment*

Transfer alignment menggunakan posisi atau kecepatan, dan bahkan attitude dari INS atau INS/GNSS yang lain. Biasanya digunakan untuk aligning INS senjata terpandu ketika power-up dan *launch*.

Tipe teknik *fine alignment* yang paling cocok bergantung pada pengaplikasiannya. Ketika INS dalam kondisi stasioner di permukaan Bumi, *Quasi-stationary alignment* biasanya paling bagus karena tingkat derau yang paling rendah. Sedangkan apabila diaplikasikan pada saat kondisi terbang misalnya, pilihan terbaik adalah *transfer alignment* menggunakan referensi INS/GNSS karena mengombinasikan akurasi waktu pendek dan kecepatan update dari INS dengan akurasi waktu panjang dari GNSS. Dalam aplikasi ini, diterapkan teknik *quasi-stationary alignment*.

2.6.5 Perambatan kesalahan INS

Kesalahan dalam solusi navigasi inersia muncul dari tiga sumber: sensor inersia, kesalahan inisialisasi, dan persamaan pembaruan solusi navigasi, termasuk model gravitasi. Kesalahan ini terintegrasi melalui persamaan navigasi untuk menghasilkan kesalahan posisi, kecepatan, dan orientasi yang tumbuh seiring waktu. Sebagai contoh, kesalahan inisialisasi kecepatan menghasilkan kesalahan posisi tumbuh. Perambatan kesalahan juga dipengaruhi oleh lintasan kendaraan host. Sebagai contoh, pengaruh faktor skala dan kesalahan lain cross-coupling tergantung pada dinamika kendaraan host, seperti halnya kopling kesalahan orientasi, khususnya sudut hadap, dan kecepatan dan posisi.

2.8 STIM 300[8]

STIM 300 adalah IMU dari perusahaan Sensoror, dengan performansi yang tinggi yang terdiri dari 3 giro MEMS yang berakurasi tinggi, 3 akselerometer dengan stabilitas tinggi dan 3 inklinometer, dan termasuk dalam kelas taktis. IMU sudah dikalibrasi oleh pabrikan dan terkompensasi untuk efek pemanasan suhu. Kelebihan lain dari teknologi MEMS Sensoror tidak sensitif terhadap vibrasi dan guncangan. STIM 300 cocok untuk stabilisasi, pemanduan, dan aplikasi navigasi dalam skala industri, penerbangan, dalam lingkup pertahanan. Sensor ini sangat dibutuhkan pada sistem navigasi UAV, AUV, AGV, UGV, ROV, robotika, dan lain sebagainya. STIM 300 dapat secara kompetitif

mengganti FOG (*Fiber Optic Gyro*) dan meningkatkan performansi sistem terhadap ketahanan, keandalan, ukuran, berat, daya, dan harga. Gambar STIM300 ditunjukkan pada gambar 2.17.



Gambar 2. 17 STIM300[8]

2.9 CNS 5000[9]

CNS 5000 membawa 2 teknologi mutakhir, KVH FOG (*Fiber Optic Gyro*) berdasarkan IMU dan NovAtel OEM 6 GNSS receiver yang memiliki presisi tinggi untuk INS yang memiliki akurasi pada harga yang terjangkau. Pada paket yang kompak, teknologi ini adalah *deeply coupled* untuk memproses data mentah dari GPS dan IMU, dan mendapatkan solusi navigasi yang baik dan andal serta akurat dalam posisi, kecepatan, dan informasi orientasi. Gambar CNS5000 ditunjukkan pada gambar 2.18.



Gambar 2. 18 CNS5000[9]

CNS-5000 menyediakan navigasi inersia yang akurat bahkan ketika GNSS diblokir, macet atau tidak tersedia. Sistem navigasi inersia yang tangguh (INS) ini adalah produk komersial yang tersedia di pasaran, sehingga menyederhanakan logistik untuk pelanggan yang sistemnya melintasi batas internasional

2.10 GPS (*Global Positioning System*)[6]

Global Positioning System, atau yang biasa disebut GPS merupakan sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika Serikat di bawah program NAVSTAR (*Navigation by Satellite Ranging and Timing*). GPS merupakan sistem navigasi berbasis satelit yang menghasilkan solusi navigasi berupa posisi dimensi tiga (x,y,z) menggunakan sinyal radio yang dipancarkan melalui satelit yang mengorbit [3].

GPS merupakan salah satu bagian dari sistem navigasi berbasis satelit. Sistem navigasi berbasis satelit yang pertama kali dikembangkan adalah Sistem transit milik Angkatan Laut Amerika Serikat (U.S. Navy's Transit System) [2]. Pengembangan dilakukan pada tahun 1958 dengan satelit percobaan pertama diluncurkan pada tahun 1961 dan sistem beroperasi secara optimal pada tahun 1964. Sistem navigasi ini mulai terbuka untuk umum pada tahun 1967. Sedangkan pengembangan dari GPS dimulai pada tahun 1973. Satelit prototip pertama yang beroperasi diluncurkan pada tahun 1978 dan kemampuan operasional awal (*Initial Operational Capability/IOC*) dari sistem GPS penuh diumumkan pada tahun 1993. Meskipun dikembangkan untuk keperluan militer, kini GPS dapat digunakan untuk kepentingan umum.

Pada praktiknya, jumlah satelit yang terpantau oleh receiver bisa lebih dari empat, hal ini membuat akurasi dari estimasi posisi dapat lebih baik dan konsisten saat sistem navigasi digunakan. Namun, jumlah satelit yang terlihat oleh receiver bisa juga kurang dari 4, hal ini dikarenakan beberapa hal, seperti gedung-gedung atau pepohonan tinggi yang menghalangi sinyal transmisi sampai pada receiver. Kondisi ini merupakan salah satu gangguan yang terjadi pada GPS dan dapat menurunkan akurasi GPS dalam mengestimasi jarak dan posisi receiver di permukaan Bumi.

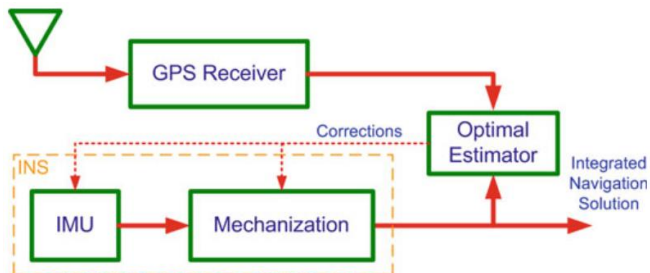
2.11 Integrasi Sistem Navigasi Inersia dengan GPS[1,2]

Sebuah INS mengandung sistem navigasi *autonomous* dengan frekuensi lebih dari 200 Hz, memiliki akurasi waktu singkat yang bagus,

dan menyediakan informasi posisi, kecepatan, dan *attitude*. Akan tetapi ketika digunakan waktu lama, kesalahan ini tumbuh tanpa batasan karena kesalahan sensor inersial terakumulasi karena proses integrasi pada algoritma navigasi. Berkebalikan dengan INS, GPS memiliki akurasi waktu lama yang bagus dan kesalahan yang terbatas hanya beberapa meter dan memiliki harga hardware yang murah. Akan tetapi GPS memiliki akurasi waktu singkat yang kurang bagus dan memiliki frekuensi yang lebih rendah dari INS. GPS membutuhkan *Line of Sight* lurus terhadap sekurang-kurangnya empat satelit, yang mana tidak selalu mungkin karena sinyal dari satelit terpengaruh dari berbagai hal misal bangunan tinggi, pohon, terowongan, serta degradasi karena atmosfer dan juga interferensi multipath.

Dengan karakteristik yang saling berkebalikan, integrasi antara keduanya menjadikan akurasi yang lebih bagus dan solusi navigasi yang lebih tahan terhadap gangguan dibandingkan secara individu. Integrasi keduanya menghasilkan sebuah system dengan frekuensi tinggi yang menyediakan solusi navigasi penuh (*position*, *velocity*, dan *attitude* atau orientasi atau orientasi) dengan peningkatan akurasi baik itu dalam waktu singkat maupun lama.

Teknik estimasi yang optimal, dominannya berbasis Filter Kalman digunakan untuk menggabungkan secara optimal antara *positioning* GPS dan INS dan informasi navigasi dengan solusi navigasi yang dapat diandalkan. GPS mencegah *drifting* dan INS menyediakan kontinuitas dalam solusi navigasi, informasi *attitude*, dan menjembatani keterbatasan sinyal GPS. Integrasi ini ditunjukkan pada gambar 2.19. Permodelan ini dimulai dari kesalahan orientasi, kesalahan kecepatan, dan kesalahan posisi



Gambar 2. 19 Integrasi INS-GPS[2]

Berikut persamaan model kesalahan dari INS untuk kesalahan orientasi.

$$\begin{aligned} \delta \boldsymbol{\psi}_{nb}^n &= (\mathbf{I} - \delta \mathbf{C}_{nb}^n) \boldsymbol{\omega}_{in}^n + (\tilde{\boldsymbol{\omega}}_{ie}^n - \boldsymbol{\omega}_{ie}^n) \\ &+ (\tilde{\boldsymbol{\omega}}_{en}^n - \boldsymbol{\omega}_{en}^n) \end{aligned} \quad (2.66)$$

Sedangkan kesalahan perubahan kecepatan didapatkan dari persamaan berikut

$$\begin{aligned} \delta \hat{\mathbf{v}}_{eb}^n &= (\tilde{\mathbf{f}}_{ib}^n - \mathbf{f}_{ib}^n) + \mathbf{g}_b^n(\tilde{L}_b, \tilde{h}_b) - \mathbf{g}_b^n(L_b, h_b) \\ &- (\tilde{\boldsymbol{\Omega}}_{en}^n + 2\tilde{\boldsymbol{\Omega}}_{ie}^n) \tilde{\mathbf{v}}_{eb}^n + (\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n) \mathbf{v}_{eb}^n \end{aligned} \quad (2.67)$$

Dan untuk kesalahan dari perubahan posisi terhadap kerangka lokal yaitu

$$E(\delta \hat{L}_b) = \frac{\delta v_{eb,N}^n}{R_N(\hat{L}_b) + \hat{h}_b} - \frac{v_{eb,N}^n \delta h_b}{(R_N(\hat{L}_b) + \hat{h}_b)^2} \quad (2.68)$$

$$\begin{aligned} E(\delta \hat{\lambda}_b) &= \frac{\delta v_{eb,E}^n}{(R_E(\hat{L}_b) + \hat{h}_b) \cos(\hat{L}_b)} \\ &+ \frac{v_{eb,E}^n \sin \hat{L}_b \delta L_b}{(R_N(\hat{L}_b) + \hat{h}_b) \cos^2(\hat{L}_b)} \\ &- \frac{\hat{v}_{eb,E}^n \delta h_b}{(R_E(\hat{L}_b) + \hat{h}_b) \cos(\hat{L}_b)} \end{aligned} \quad (2.69)$$

$$E(\delta \hat{h}_b) = -\delta v_{eb,D}^n \quad (2.70)$$

$$\mathbf{b}_a = \mathbf{0}_{3 \times 1} \quad (2.71)$$

$$\mathbf{b}_g = \mathbf{0}_{3 \times 1} \quad (2.72)$$

$$f(x) = [\delta \boldsymbol{\psi}_{nb}^n \ \delta \hat{\mathbf{v}}_{eb}^n \ \delta \hat{L}_b \ \delta \hat{\lambda}_b \ \delta \hat{h}_b \ \mathbf{b}_a \ \mathbf{b}_g]^T \quad (2.73)$$

2.11.1 Loosely Coupled Integration

Dalam integrasi ini, GPS dan INS beroperasi secara independen dan menyediakan solusi navigasi yang terpisah. Untuk meningkatkan solusi, posisi dan / atau kecepatan dari GPS diumpankan ke beberapa estimator optimal, biasanya Filter Kalman. Solusi INS adalah juga diumpankan ke filter yang mengambil nilai selisih antara keduanya dan, berdasarkan pada model kesalahan, perkiraan kesalahan INS. Solusi INS diperbaiki untuk kesalahan ini untuk menghasilkan solusi navigasi terintegrasi dalam bentuk posisi, kecepatan dan sikap.

Pendekatan ini juga disebut pendekatan desentralisasi karena ada filter terpisah yang digunakan untuk GPS. Integrasi ini mudah diterapkan

dan *robust*. Integrasi ini menyediakan tiga solusi navigasi terpisah untuk untai terbuka (*raw INS*, *raw GPS*, dan solusi terintegrasi) dan dua untuk *closed-loop* (tidak ada *raw INS*). Masalah utama adalah itu ketidakmampuan untuk memberikan bantuan GPS ketika jumlah efektif satelit berada di bawah minimum. Karena output GPS KF berkorelasi dengan waktu, asumsi KF *noise* pengukuran tidak berkorelasi adalah membahayakan, sehingga mengganggu kinerja sistem.

2.12 Root Mean Square Error (RMSE)[2]

Root Mean Square Error atau akar kesalahan kuadrat rata-rata merupakan suatu ukuran kesalahan yang didasarkan pada selisih antara dua buah nilai yang bersesuaian atau dengan kata lain merupakan selisih antara data *output* referensi dengan data *output* sebenarnya untuk menyatakan tingkat kemiripan antara nilai yang diinginkan dengan nilai fakta. Semakin kecil eror yang dihasilkan maka semakin mendekati dengan sistem sebenarnya. Dengan kata lain RMSE merupakan standar deviasi dari residual (kesalahan prediksi). Residual adalah ukuran seberapa jauh dari titik-titik data garis regresi. Pada umumnya metode ini digunakan dalam analisis regresi untuk memverifikasi hasil eksperimen.

Error yang akan didapatkan untuk analisa data antara sinyal referensi dengan sinyal aktual dapat dinyatakan sebagai berikut

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Y_{in} - Y_{out})^2}{N}} \quad (2.74)$$

di mana,

- Y_{in} = Data *ouput* syal referensi
- Y_{out} = Data *ouput* syal sebenarnya
- N = Banyaknya data

Halaman ini sengaja dikosongkan

BAB 3 PERANCANGAN SISTEM

Pada bab ini akan dipaparkan mengenai perancangan sistem dari integrasi sistem navigasi inersia dengan GPS atau INS-GPS dengan menggunakan algoritma *Unscented Kalman Filter*. Berikut diagram alur dari perancangan sistem integrasi INS dengan GPS dengan menggunakan algoritma *Unscented Kalman Filter*.



Gambar 3. 1 Diagram alur sistem INS-GPS

Pada tahap pertama, adalah tahap perancangan penyalarsan yang terbagi menjadi dua tahap, yaitu penyalarsan kasar (*coarse alignment*), dan penyalarsan halus (*fine alignment*). Penyalarsan ini menggunakan data wahana yang diam. Untuk *coarse alignment* atau penyalarsan kasar ini menggunakan dua prinsip, yaitu *levelling* dan *gyrocompassing* untuk menentukan sudut kemiringan wahana dan sudut hadap awal pada wahana terhadap kerangka navigasi lokal. Sedangkan untuk *fine alignment* atau penyalarsan halus menggunakan *Unscented Kalman Filter* untuk mengestimasi lebih baik sudut kemiringan serta sudut hadap wahana terhadap kerangka navigasi lokal. Pada tahap ketiga dilakukan perancangan algoritma *Unscented Kalman Filter* untuk integrasi antara sistem navigasi inersia dengan GPS dengan menggunakan data wahana bergerak atau navigasi.

3.1 Perancangan Model Kesalahan Sistem Navigasi Inersia

Perancangan model kesalahan pada sistem navigasi inersia merupakan tahap awal dari seluruh perancangan integrasi sistem navigasi inersia dengan GPS. Perancangan model kesalahan ini digunakan untuk 2

tahap, yaitu penyelarasan awal halus (*fine alignment*) dan juga integrasi INS/GPS. Persamaan model kesalahan ini diperlukan sebagai model yang dipergunakan Filter Kalman sebagai estimasi state (*state estimation*). Model kesalahan ini merupakan hasil propagasi dari kesalahan, baik sistemik maupun acak dari sensor akselerometer dan giroskop yang kemudian menyebabkan kesalahan – kesalahan pada variabel lain yaitu orientasi atau sikap atau *attitude*, kecepatan, maupun posisi dari suatu wahana. Sehingga dengan pengukuran dari GPS ini dapat dilakukan estimasi terhadap kesalahan dari akselerometer dan giroskop serta estimasi dari kesalahan-kesalahan pada variabel lain.

Untuk *script* program dapat dilihat pada lampiran 1. Pada *script* pada lampiran 1, terlihat bahwa terdapat 15 state meliputi [$\delta\phi, \delta\theta, \delta\psi, \delta v_N, \delta v_E, \delta v_D, \delta L_b, \delta\lambda_b, \delta h_b, \mathbf{b}_a, \mathbf{b}_g$] pada bagian *modelling* secara berurutan dengan variabel x_new dan x . Variabel x_new ini merupakan propagasi dari $f(x)$, sedangkan x disini adalah *sigma point*. Masukan dari fungsi tersebut adalah estimasi orientasi, kecepatan, dan posisi, hasil dari mekanisasi, juga *sigma point*, dan juga pengukuran akselerometer, dengan state nonlinier pada state ketiga yaitu $\delta\psi$ yang diasumsikan dengan kesalahan yang nilainya besar sehingga persamaannya menjadi nonlinier. Pada state ke sepuluh sampai dengan limabelas adalah state bias akselerometer dan giroskop 3 sumbu yang dimodelkan dengan *random walk*. Pada bagian terakhir terdapat diskritisasi karena model yang dipakai adalah model diskrit dari Filter Kalman. Semua *script* pada model kesalahan mengacu pada bab 2.11

Tabel 3. 1 Keterangan variabel dan parameter dari model kesalahan state pada integrasi sistem navigasi inersia dengan GPS

Simbol	Keterangan	Satuan
R_E	Radius dari lengkungan bumi untuk arah timur -barat	m
R_N	Radius dari lengkungan bumi untuk arah utara - selatan	m
\hat{L}_b	Estimasi garis lintang dari wahana.	rad
$\hat{\lambda}_b$	Estimasi garis bujur dari wahana.	rad
\hat{h}_b	Estimasi ketinggian wahana terhadap terhadap permukaan	m

Simbol	Keterangan	Satuan
	elipsoid yang dihitung secara tegak lurus terhadap permukaan.	
$v_{eb,N}^n$	Kecepatan wahana dalam kerangka lokal pada arah utara	ms^{-1}
$v_{eb,E}^n$	Kecepatan wahana dalam kerangka lokal pada arah timur	ms^{-1}
$v_{eb,D}^n$	Kecepatan wahana dalam kerangka lokal pada arah bawah	ms^{-1}
ω_{ie}	Kecepatan rotasi bumi	$rad\ s^{-1}$
ω_{ie}^n	Kecepatan rotasi bumi dalam kerangka lokal	$rad\ s^{-1}$
ω_{en}^n	<i>Transport-rate</i> wahana dalam kerangka lokal	$rad\ s^{-1}$
Ω_{ie}^n	<i>Skew-symmetric</i> dari ω_{ie}^n	—
Ω_{en}^n	<i>Skew-symmetric</i> dari ω_{en}^n	—
g_0	Percepatan gravitasi pada bidang elipsoid	$m\ s^{-2}$
b_a	Bias akselerometer	ms^{-2}
b_g	Bias giroskop	$rad\ s^{-1}$
f_{ib}^b	Nilai akselerometer	$m\ s^{-2}$
r_{eS}^e	Radius geocentric	m
\wedge	Estimasi	—
\sim	Terukur	—

3.2 Perancangan *Unscented Kalman Filter*

Unscented Kalman Filter ini digunakan di dua proses, yaitu penyalarsan halus dengan *rate filter* sebesar 2.000 Hz, dan saat navigasi sebesar 10 Hz. Pada algoritma Kalman Filter sendiri terdapat 2 model, yaitu model sistem yang berupa model nonlinier diskrit yang sudah dibahas perancangannya pada subab 3.1, serta model pengukuran yaitu sensor atau sumber informasi dari luar sebagai sumber pengukuran, dan modelnya adalah model linier. Pengukuran ini menggunakan GPS dimana menggunakan informasi kecepatan (m/s) dan posisi (dalam $^{\circ}$) wahana

yang mana kecepatan dan posisi ini sudah terukur dalam kerangka navigasi lokal pada saat navigasi atau bergerak, dalam hal ini GPS menggunakan waktu cuplik sebesar 0.1 detik atau 10 Hz. Pengukuran saat penyelarasan halus menggunakan inisialisasi sendiri untuk kondisi diam.

Pada algoritma *Unscented Kalman Filter* juga terdapat beberapa parameter yang perlu diatur, yaitu $\alpha, \beta, \kappa, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}$, dimana \mathbf{P}_0 adalah matriks kovarians kesalahan awal, \mathbf{Q} adalah matriks gangguan sistem (*process noise*), dan \mathbf{R} adalah matriks gangguan pengukuran (*measurement model*). Pada matriks \mathbf{P}_0 ini diberi nilai seperti berikut:

$$\mathbf{P}_0 = \text{diag}([\sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2, \sigma_{v_{N,E,D}}, \sigma_L^2, \sigma_\lambda^2, \sigma_h^2, \delta b_a, \delta b_g]) \quad (3.1)$$

Sedangkan untuk persamaan diatas dilakukan penyetelan parameter-parameter seperti berikut.

Tabel 3. 2 Parameter pada matriks \mathbf{P}_0

Simbol	Keterangan	Nilai	Satuan
σ_ϕ	Ketidakpastian kesalahan awal orientasi x	0.0175	rad
σ_θ	Ketidakpastian kesalahan awal orientasi y	0.0175	rad
σ_ψ	Ketidakpastian kesalahan awal orientasi z	0.524	rad
$\sigma_{v_{N,E,D}}$	Ketidakpastian kesalahan awal kecepatan utara, timur, dan bawah	2	ms^{-1}
$\sigma_{L,\lambda,h}$	Ketidakpastian kesalahan awal posisi lintang dan bujur, serta ketinggian	3.005	m
$\sigma_{b_{a_x}}$	Ketidakpastian bias awal dari akselero sumbu x	4.9033e-05	ms^{-2}
$\sigma_{b_{a_y}}$	Ketidakpastian bias awal dari akselero sumbu y	4.9033e-05	ms^{-2}

Simbol	Keterangan	Nilai	Satuan
σ_{ba_z}	Ketidakpastian bias awal dari akselero sumbu z	9.8066e-05	ms^{-2}
σ_{bg_x}	Ketidakpastian bias awal dari giro sumbu x	4.8481e-06	$rad s^{-2}$
σ_{bg_y}	Ketidakpastian bias awal dari giro sumbu y	2.4240e-05	$rad s^{-2}$
σ_{bg_z}	Ketidakpastian bias awal dari giro sumbu z	4.8481e-06	$rad s^{-2}$

Pada matriks **Q** berlaku seperti berikut

$$Q = diag([\mathbf{S}_{rg}, \mathbf{S}_{ra}, 0, \mathbf{S}_{bad}, \mathbf{S}_{bgd}])\tau \quad (3.2)$$

Tabel 3. 3 Parameter pada matriks **Q**

Simbol	Keterangan	Nilai	Satuan
S_{rg_x}	<i>PSD</i> dari gangguan giro sumbu x	4.2921e-07	$rad^2 s^{-3}$
S_{rg_y}	<i>PSD</i> dari gangguan giro sumbu y	1.7940e-07	$rad^2 s^{-3}$
S_{rg_z}	<i>PSD</i> dari gangguan giro sumbu z	1.4695e-07	$rad^2 s^{-3}$
S_{ra_x}	<i>PSD</i> dari gangguan akselero sumbu x	1.1765e-07	$m^2 s^{-3}$
S_{ra_y}	<i>PSD</i> dari gangguan akselero sumbu y	1.4909e-08	$m^2 s^{-3}$
S_{ra_z}	<i>PSD</i> dari gangguan akselero sumbu z	6.9444e-05	$m^2 s^{-3}$
S_{bad_x}	<i>PSD</i> dari bias akselero dinamis sumbu x	2.4043e-08	$m^2 s^{-5}$
S_{bad_y}	<i>PSD</i> dari bias akselero dinamis sumbu y	2.4043e-08	$m^2 s^{-5}$

Simbol	Keterangan	Nilai	Satuan
S_{bad_x}	<i>PSD</i> dari bias akselero dinamis sumbu z	1.2021e-10	m^2s^{-5}
S_{bgd_x}	<i>PSD</i> dari bias giro dinamis sumbu x	2.1154e-15	rad^2s^{-1}
S_{bgd_y}	<i>PSD</i> dari bias giro dinamis sumbu y	2.1154e-13	rad^2s^{-1}
S_{bgd_z}	<i>PSD</i> dari bias giro dinamis sumbu z	2.1154e-14	rad^2s^{-1}

Pada matriks \mathbf{R} berlaku seperti berikut

$$\mathbf{R} = \text{diag}([\sigma_{L_b}, \sigma_{\lambda_b}, \sigma_{h_b}, \sigma_{v_{NE,D}}]) \quad (3.3)$$

Tabel 3. 4 Parameter pada matriks \mathbf{R}

Simbol	Keterangan	Nilai	Satuan
σ_{L_b}	Ketidakpastian pengukuran posisi lintang dari wahana	5.2360e-07	<i>rad</i>
σ_{λ_b}	Ketidakpastian pengukuran posisi bujur dari wahana	2.6180e-07	<i>rad</i>
σ_h	Ketidakpastian pengukuran ketinggian wahana	0.5	<i>m</i>
$\sigma_{v_{N,E,D}}$	Ketidakpastian pengukuran kecepatan utara, timur, dan bawah	0.1	ms^{-1}

Tabel 3. 5 Parameter pada *Unscented Transformation*

Simbol	Keterangan	Nilai
α	<i>Alpha</i>	0.25
β	<i>Beta</i>	2
κ	<i>Kappa</i>	0

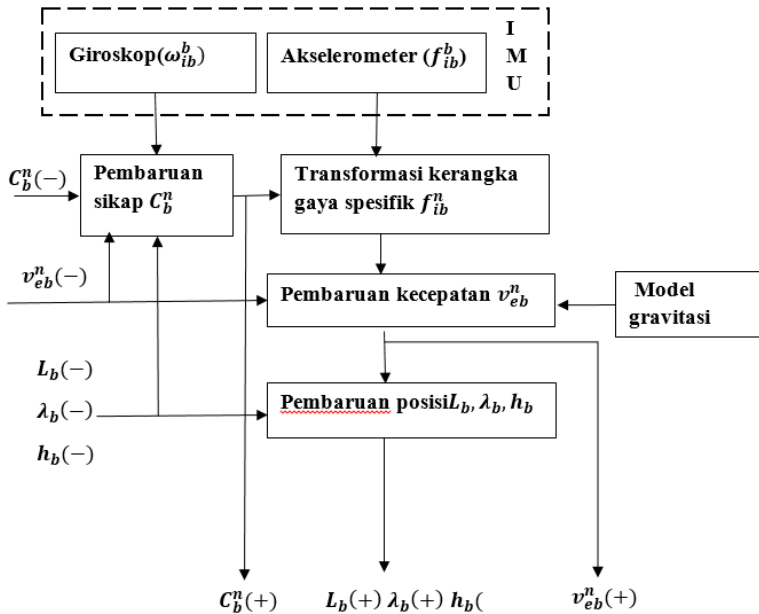
Script parameter dapat dilihat pada lampiran 2, dalam bentuk *struct*.

Script UKF terlampir pada lampiran 3. Pada *script* tersebut diuraikan tentang program *Unscented Kalman Filter* yang mengacu pada subbab 2.5., yang dimulai dengan mendefinisikan matriks \mathbf{Q} berdimensi 15×15 , \mathbf{R} berdimensi 6×6 , matriks \mathbf{H} berdimensi 6×15 yang terdiri dari 6 state terukur yaitu kesalahan kecepatan, dan posisi dalam kerangka navigasi lokal sebagai pengukuran vektor \mathbf{z} . Pada model $f\mathbf{x}$ merupakan model nonlinier dari model kesalahan bagian 3.1. Kemudian α, β, κ didefinisikan nilainya, dan algoritma. Penyetelan parameter ini didasarkan dengan menggunakan referensi CNS5000, sehingga menghasilkan kesalahan yang kecil (*trial and error*).

3.3 Perancangan Mekanisasi

Mekanisasi bertujuan untuk mengubah keluaran sensor menjadi suatu solusi navigasi berupa orientasi, kecepatan, dan posisi. Pada mekanisasi digunakan inisialisasi orientasi dari penyelerasan halus, dan bias yang diestimasi dari Filter Kalman yang akan dikurangkan kepada keluaran sensor akselerometer dan giroskop, dan posisi serta kecepatan awal yang berasal dari GPS.

Mekanisasi ini dilakukan pada kerangka navigasi, sehingga semua hasil keluaran sensor yang menghitung gaya spesifik dan kecepatan sudut dari wahana harus diubah ke kerangka lokal yang tentunya harus mengompensasi kecepatan bumi, *transport rate*, dan juga gravitasi. Blok diagram mekanisasi ini ditunjukkan pada gambar 3.2. Keluaran dari sensor memiliki waktu cuplik sebesar $\tau_i = 0.0005$ detik atau sebesar 2.000 Hz. Perancangan mekanisasi seperti yang telah dijelaskan pada bab 2.7.



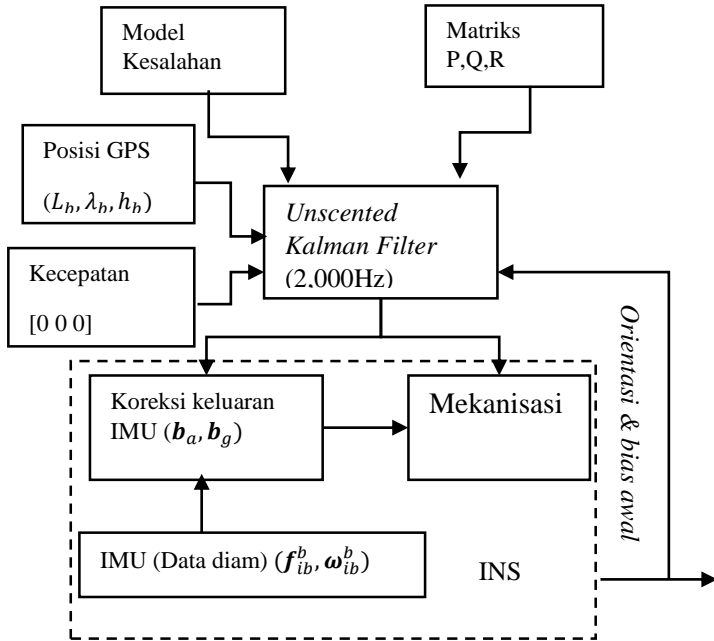
Gambar 3. 2 Alur mekanisasi

Script mekanisasi dapat dilihat pada lampiran 4, dengan masukan orientasi, kecepatan, dan posisi pada iterasi sebelumnya, dan juga pengukuran akselerometer dan giroskop yang sudah terkoreksi, yang kemudian keluarannya adalah orientasi, kecepatan, dan posisi pada iterasi sekarang.

3.4 Perancangan Penyelarasan

Secara garis besar penyelarasan ini adalah suatu prosedur yang mana digunakan untuk mengestimasi orientasi awal suatu wahana terhadap kerangka lokal saat kondisi wahana diam. Dikarenakan untuk proses tanpa bantuan dari sumber eksternal, maka sistem navigasi inersia akan sangat mudah terpengaruh oleh bias, dan sumber kesalahan internal, seperti gangguan acak, dan juga kesalahan inisialisasi yang berakibat pada kesalahan yang terus menerus terakumulasi setiap cuplik waktu, sehingga penyelarasan ini merupakan bagian yang sangat penting agar kesalahan inisialisasi dan bias dapat dikurangi. Penyelarasan ini

dilakukan dalam waktu 240 detik yang terdiri dari 2 tahap, yaitu penyeselarasan kasar dan penyeselarasan halus. Bagan untuk penyeselarasan halus terdapat pada gambar 3.3.



Gambar 3. 3 Diagram blok penyeselarasan halus

Penyeselarasan kasar adalah suatu metode yang mana menggunakan nilai akselerometer untuk menghitung sudut kemiringan wahana dan giroskop untuk menghitung sudut hadap wahana terhadap kerangka lokal pada kondisi wahana tidak bergerak. Pada penyeselarasan tahap ini dilakukan selama 20 detik, dengan merata-rata nilai akselero dan gyro dengan menggunakan metode rata-rata bergerak, setelah didapatkan rata-rata nilai akselerometer dan giroskop, maka dapat diketahui nilai sudut kemiringan dari wahana secara kasar.

Pada *script* lampiran 5, penyeselarasan kasar dilakukan dengan menggunakan rata-rata bergerak antara rata-rata sekarang dengan rata-rata 1 detik sebelumnya, agar didapatkan data yang semakin lama

konvergen dari data diam. Kemudian dengan persamaan pada bagian 2.6.5 untuk mengetahui *roll, pitch, yaw*.

Penyelarasan halus dilakukan dengan menggunakan *Unscented Kalman Filter*, dengan pengukuran dari GPS untuk posisi (L_b, λ_b, h_b) dan kecepatan inialisasi stasioner bernilai $\mathbf{0}_{3 \times 1}$, yang dilakukan dengan data diam. Pengukuran Filter Kalman menggunakan INS ditunjukkan pada persamaan berikut ini. \mathbf{p}_{INS} dan \mathbf{v}_{INS} ini didapat dari persamaan mekanisasi, seperti yang ditunjukkan pada subbab 2.7

$$\mathbf{z} = \begin{bmatrix} \mathbf{p}_{GPS} - \mathbf{p}_{INS} \\ \mathbf{v}_{stasioner} - \mathbf{v}_{INS} \end{bmatrix} \quad (3.4)$$

$$\begin{bmatrix} \mathbf{p}_{GPS} - \mathbf{p}_{INS} \\ \mathbf{v}_{stasioner} - \mathbf{v}_{INS} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \dots \\ \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ \dots & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ \dots & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \end{bmatrix} \begin{bmatrix} \delta\psi \\ \delta\mathbf{v} \\ \delta\mathbf{p} \\ \mathbf{b}_a \\ \mathbf{b}_g \end{bmatrix} \quad (3.5)$$

Pada *script* yang terlampir pada lampiran 6, pada bagian awal adalah bagian inialisasi baik dari penyelarasan kasar, inialisasi posisi dari GPS, dan kecepatan [0 0 0]. Posisi dari GPS dan kecepatan [0 0 0] adalah sebagai pengukuran dari *Unscented Kalman Filter*. Kemudian nilai dari akselerometer dan giroskop terukur akan masuk ke mekanisasi dan kemudian masuk kedalam algoritma *Unscented Kalman Filter*, yang nilai estimasi dari UKF ini akan diumpun kembali ke mekanisasi, sebagai nilai yang lalu sehingga terjadi koreksi, akibat dari koreksi ini, maka nilai estimasi disetiap iterasi diatur ulang menjadi $\mathbf{0}_{15 \times 1}$, algoritma ini disebut integrasi untai tertutup (*closed-loop integration*).

3.5 Perancangan Integrasi INS dengan GPS

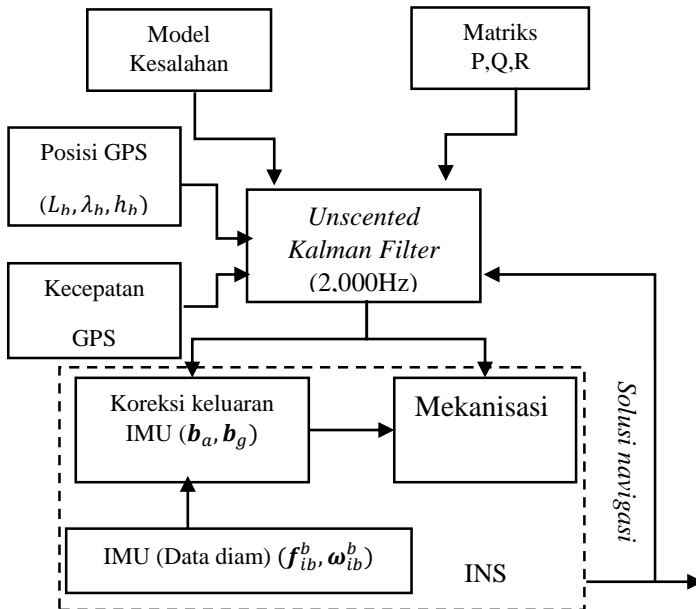
Perancangan integrasi sistem navigasi inersia atau INS dengan GPS ini menggunakan *Unscented Kalman Filter*, yang mengasumsikan bahwa sistem tidak linier atau nonlinier. Data pada penyelarasan halus adalah data diam, sedangkan data pada integrasi INS-GPS adalah data bergerak atau navigasi. Waktu untuk penyelarasan halus sendiri adalah 210 detik, sedangkan pada saat navigasi adalah selama 120 detik saja. *Rate* dari Filter Kalman untuk penyelarasan halus adalah 2.000 Hz, sedangkan pada

navigasi mengikuti dari *rate* dari data GPS yaitu sebesar 10 Hz. Bagan integrasi INS GPS terdapat pada gambar 3.4 berikut ini.

Filter Kalman menggabungkan kedua informasi INS dan GPS, dari prediksi dan pengukuran. Prediksi menggunakan model kesalahan INS dari persamaan pada subbab 2.11, dan pengukuran Filter Kalman menggunakan pengukuran GPS dan estimasi dari INS ditunjukkan pada persamaan berikut ini.

$$\mathbf{z} = \begin{bmatrix} \mathbf{p}_{GPS} - \mathbf{p}_{INS} \\ \mathbf{v}_{GPS} - \mathbf{v}_{INS} \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} \mathbf{p}_{GPS} - \mathbf{p}_{INS} \\ \mathbf{v}_{GPS} - \mathbf{v}_{INS} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \dots \\ \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ \dots & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ \dots & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \end{bmatrix} \begin{bmatrix} \delta\psi \\ \delta\mathbf{v} \\ \delta\mathbf{p} \\ \mathbf{b}_a \\ \mathbf{b}_g \end{bmatrix} \quad (3.7)$$



Gambar 3. 4 Diagram blok integrasi INS GPS

Setelah mendapatkan nilai kesalahan masing – masing dengan menggunakan Filter Kalman, maka koreksi dapat dilakukan dengan formula sebagai berikut:

$$\hat{\mathbf{C}}_b^n(+)=\delta\hat{\mathbf{C}}_b^n\hat{\mathbf{C}}_b^n(-) \quad (3.8)$$

$$\hat{\mathbf{v}}_{eb}^n(+)=\hat{\mathbf{v}}_{eb}^n(-)-\delta\hat{\mathbf{v}}_{eb}^n \quad (3.9)$$

$$\hat{\mathbf{L}}_b(+)=\hat{\mathbf{L}}_b(-)-\delta\hat{\mathbf{L}}_b \quad (3.10)$$

$$\hat{\lambda}_b(+)=\hat{\lambda}_b(-)-\delta\hat{\lambda}_b \quad (3.11)$$

$$\hat{\mathbf{h}}_b(+)=\hat{\mathbf{h}}_b(-)-\delta\hat{\mathbf{h}}_b \quad (3.12)$$

Simbol (+) menyatakan estimasi yang sudah terkoreksi, sedangkan (-) menyatakan estimasi yang belum terkoreksi.

Pada *script* pada lampiran 7, bagian awal adalah bagian inialisasi hasil dari penyelarasan halus berupa orientasi, kecepatan, dan posisi awal. GPS digunakan sebagai pengukuran dari *Unscented Kalman Filter*. Kemudian nilai dari akselerometer dan giroskop terukur akan masuk ke mekanisasi dan selanjutnya masuk ke dalam algoritma *Unscented Kalman Filter*. Nilai estimasi dari UKF ini akan diumpan kan kembali ke mekanisasi sebagai nilai yang lalu sehingga terjadi koreksi. Akibat dari koreksi ini, maka nilai estimasi disetiap iterasi diatur ulang menjadi $\mathbf{0}_{15 \times 1}$.

Perbedaanya dengan *script* penyelarasan halus adalah pengukuran pada penyelarasan halus ini tersedia setiap saat (karena wahana diam sehingga asumsi posisi dan kecepatan tidak berubah), namun pada saat integrasi INS-GPS ini pengukuran dari GPS tersedia setiap 10 Hz, sedangkan kecepatan *sampling data* pada STIM300 adalah 2.000 Hz, sehingga setiap 10 Hz saja terjadi integrasi INSGPS dengan *Unscented Kalman Filter*.

BAB 4

PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas mengenai pengujian penyelarasan, pengujian sistem navigasi inersia (*standalone*), pengujian GPS, dan pengujian integrasi sistem navigasi inersia dengan GPS (INS/GPS) baik menggunakan KF maupun UKF.

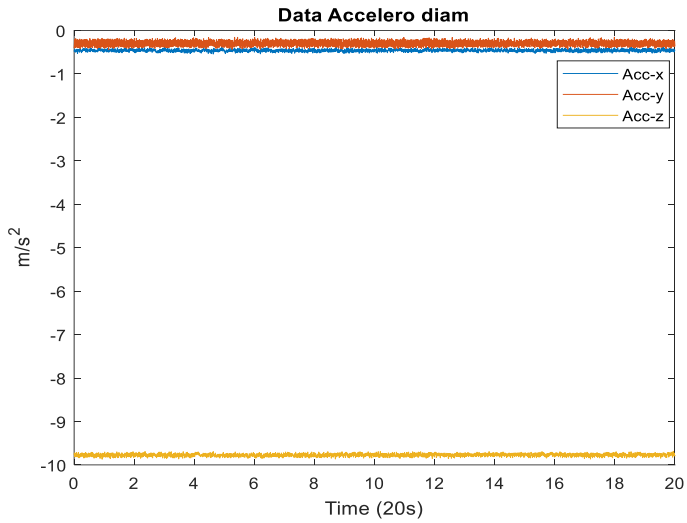
4.1 Pengujian Penyelarasan

Pengujian penyelarasan ini dibagi menjadi dua tahap, yaitu penyelarasan kasar, dan penyelarasan halus. Penyelarasan ini menggunakan data diam, yaitu data terukur dari STIM 300 saat kendaraan dalam kondisi diam dalam waktu 240 detik dengan 10 detik untuk membuang data (*ignore data*), hal ini disebabkan karena pada saat awal IMU dinyalakan sinyal IMU belum stabil. Penyelarasan adalah tahap inisialisasi awal dari INS yaitu berupa orientasi awal wahana yang akan menentukan presisinya INS saat bergerak.

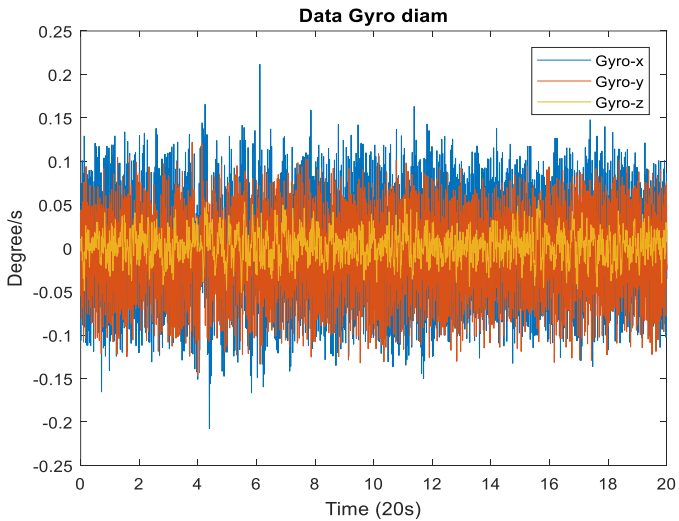
4.1.1 Penyelarasan kasar (*coarse alignment*)

Penyelarasan kasar ini menggunakan sinyal mentah dari akselerometer STIM 300 yang digunakan pada metode *leveling* untuk mengetahui *roll* dan *pitch*. Dengan merata-rata nilai akselerometer tiap detik (selama 20 detik), dan kemudian hasil tersebut dirata-rata dengan nilai pada detik sebelumnya, maka tiap detik didapatkan 1 nilai. Kemudian *roll* dan *pitch* didapatkan dengan menggunakan rumus *leveling*. Gambar 4.1 dan 4.2 adalah sinyal IMU STIM 300 dalam kondisi diam selama 20 detik.

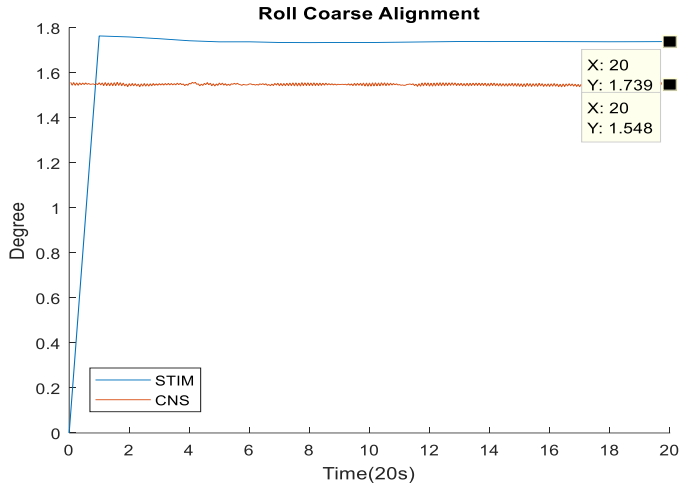
Sinyal mentah IMU terlihat sangat terpengaruh oleh *noise* acak baik akselerometer maupun giroskop pada gambar 4.1 dan 4.2 secara berurutan.



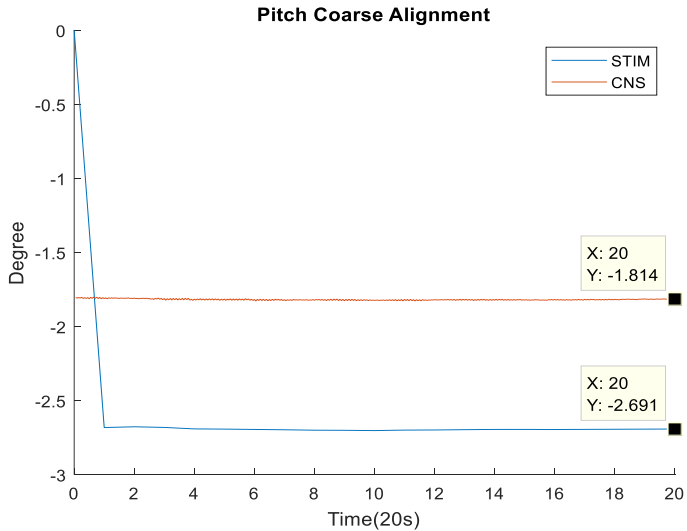
Gambar 4. 1 Data akselerometer saat diam selama 20 detik



Gambar 4. 2 Data giroskop saat diam selama 20 detik

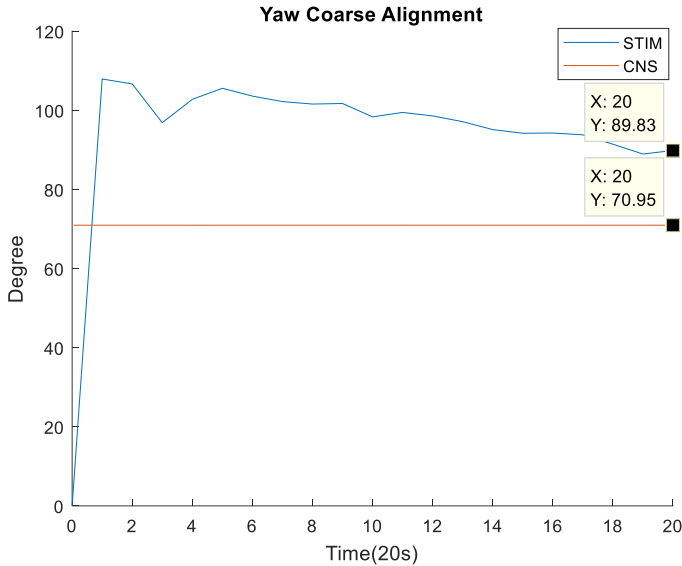


Gambar 4. 3 Hasil estimasi *roll* penyelarasan kasar



Gambar 4. 4 Hasil estimasi *pitch* penyelarasan kasar

Pada gambar 4.3 nilai estimasi kasar dari *roll* serta gambar 4.4 nilai estimasi untuk *pitch*, terlihat semakin lama waktu maka kedua nilai akan konvergen akibat dari mekanisme rata-rata bergerak. Bila dibandingkan dengan nilai referensi maka pada nilai terakhir (detik ke-20), selisihnya adalah sebesar $0,191^\circ$ untuk *roll* dan sebesar $0,887^\circ$ untuk *pitch*.

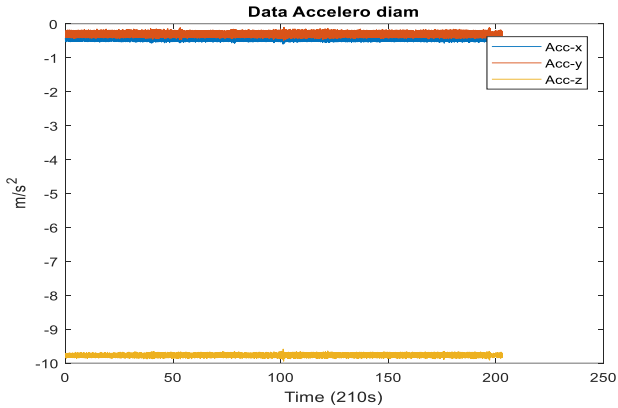


Gambar 4.5 *Yaw* penyelerasan kasar

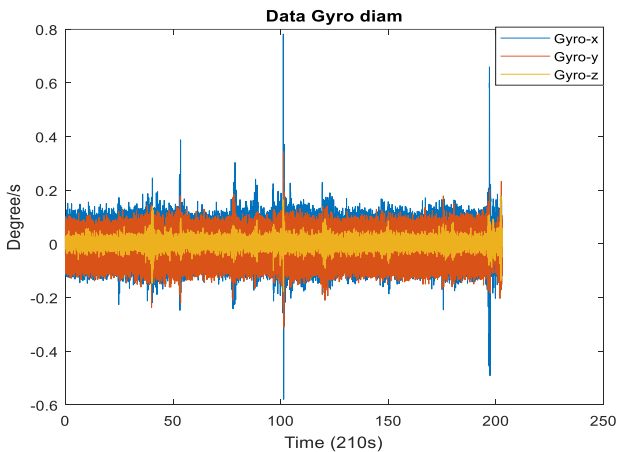
Untuk *yaw* sendiri karena *noise* yang terdapat pada giroskop sangatlah besar (melebihi kecepatan rotasi bumi yang dirasakan oleh giroskop) dengan variansi sebesar [0,0013 rad/s 0,0018 rad/s 0.0004 rad/s], maka *gyrocompassing* tidak dapat dilakukan. Hal ini disebabkan karena *noise* melebihi nilai kecepatan rotasi Bumi yaitu sebesar $7,272205 \times 10^{-5}$ rad/s, sehingga kecepatan rotasi Bumi tidak bisa dideteksi dengan baik. Terdapat selisih sebesar 18.88° pada saat akhir penyelerasan, sehingga butuh inisialisasi dari magnetometer atau dari sumber pengukuran lain untuk menghasilkan estimasi *yaw* awal. Pada tugas akhir ini digunakan dari referensi CNS5000.

4.1.2 Penyelarasan halus (*Fine Alignment*)

Mekanisme *fine alignment* menggunakan *Unscented Kalman Filter* dengan nilai kecepatan (v_N, v_E, v_D) pada pengukuran adalah nol. Serta untuk posisi lintang, bujur, dan ketinggian (*latitude, longitude, height*) menggunakan inisialisasi dari GPS.

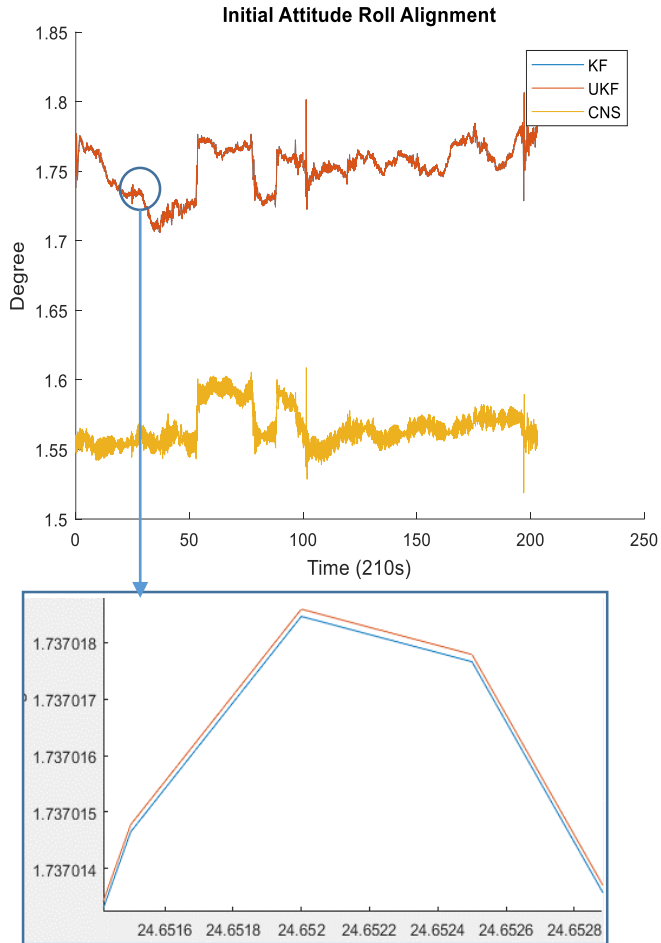


Gambar 4. 6 Data akselerometer diam selama 210 detik



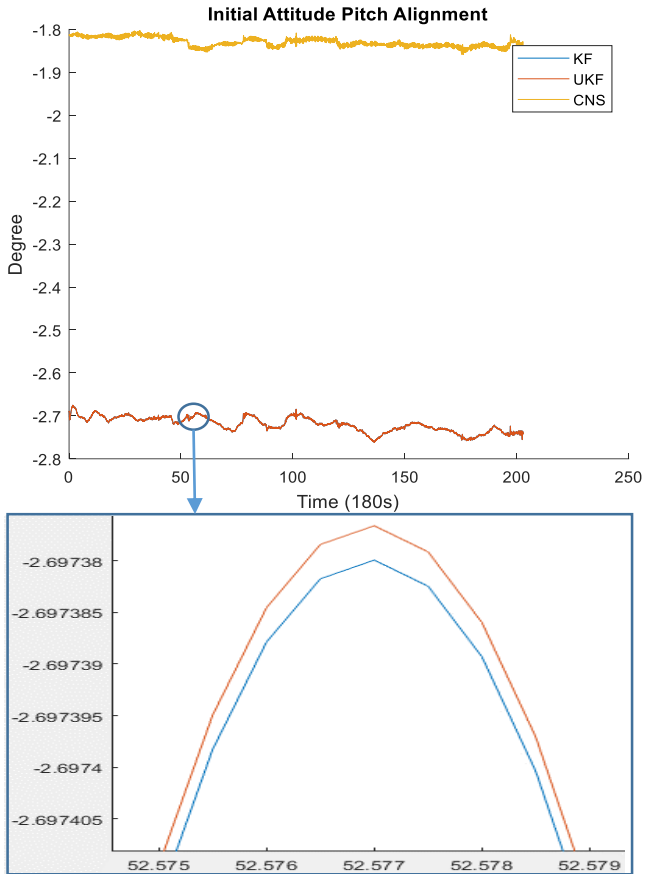
Gambar 4. 7 Data giroskop diam selama 210 detik

Berdasarkan pada gambar 4.6 dan 4.7 terlihat bahwa nilai akselerometer dan giroskop pada STIM 300 sangat tidak konstan, hal ini dipengaruhi oleh derau atau *noise* acak berupa *velocity random walk* dan *angular random walk* atau representasi dari *white noise*.



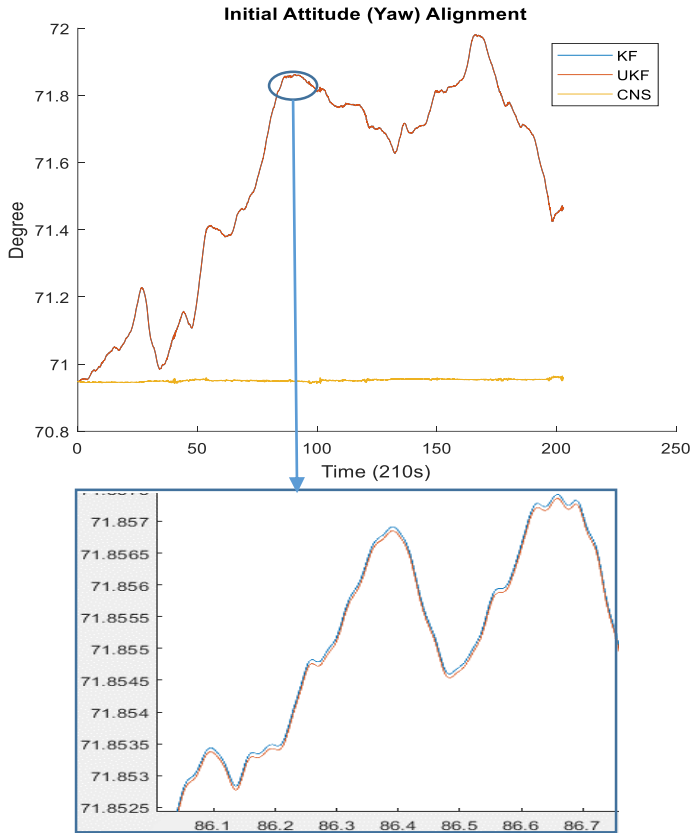
Gambar 4.8 Perbandingan penyelarasan halus untuk *roll* dengan menggunakan UKF dan KF

Berdasarkan gambar 4.8 dapat disimpulkan bahwa hasil estimasi dari *roll* kedua metode memiliki selisih yang cukup kecil dengan RMSE keduanya adalah sama, yaitu $0,1879^\circ$ terhadap referensi. Bias pada akselerometer tidak terkoreksi pada saat penyelarasan ini, sehingga terlihat pada gambar bahwa RMSE estimasi *roll* masih sebesar $0,1879^\circ$



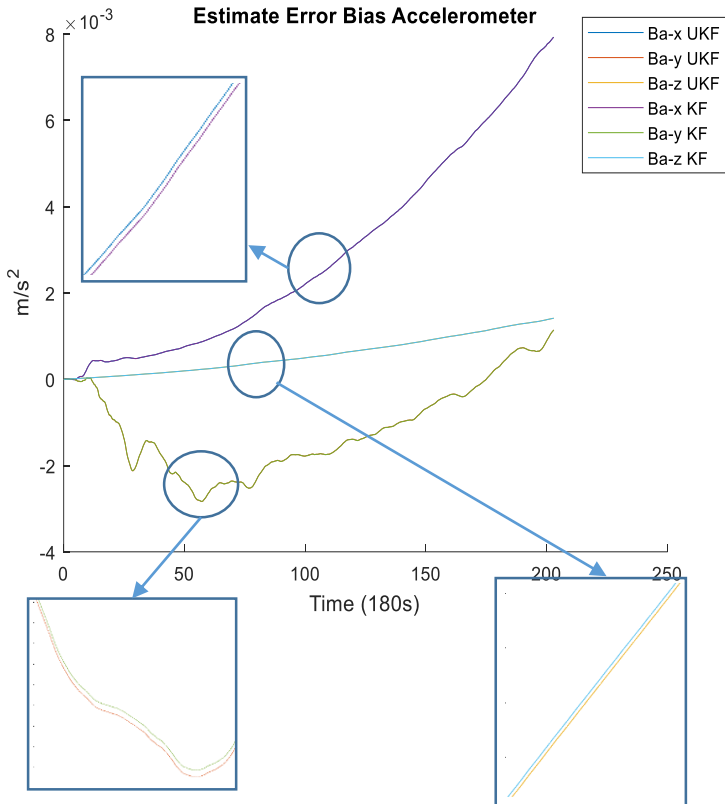
Gambar 4. 9 Perbandingan penyelarasan halus untuk *pitch* dengan menggunakan UKF dan KF

Berdasarkan gambar 4.9 dapat disimpulkan bahwa hasil estimasi dari *pitch* kedua metode memiliki selisih yang cukup kecil dengan RMSE keduanya adalah sama yaitu $0,8904^\circ$ terhadap referensi. Bias pada akselerometer juga tidak terkoreksi pada saat penyalarsan ini, sehingga terlihat pada gambar bahwa estimasi *pitch* masih sebesar $0,8904^\circ$.



Gambar 4. 10 Perbandingan penyalarsan halus untuk *yaw* dengan menggunakan UKF dan KF

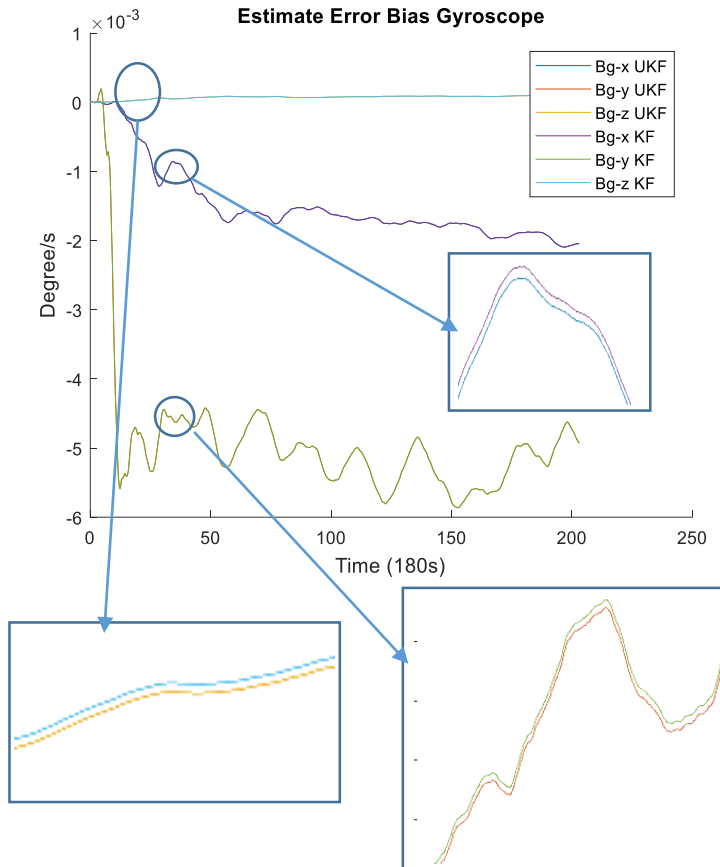
Berdasarkan gambar 4.10, estimasi *yaw* memiliki RMSE sebesar $0,6539^\circ$ untuk UKF , $0,6540^\circ$ untuk KF.



Gambar 4.11 Perbandingan hasil estimasi bias akselerometer dari penyesuaian halus antara UKF dan KF

Dengan membandingkan nilai *roll*, *pitch*, *yaw* referensi kemudian dicari nilai estimasi nilai f_{ib}^b , selanjutnya dikurangi dengan nilai rata-rata akselerometer terukur, maka dapat dicari nilai bias dari akselerometer. Nilai bias tersebut adalah sebesar $b_a \approx [-0,1485 \ -0,0330 \ 0,0047]$, atau $b_a \approx [32,2519\% \ 11,08\% \ -0,04719\%]$ dari rata-rata nilai terukur masing-masing sumbu pada akselerometer, sedangkan nilai rata-rata dari masing-masing akselerometer sendiri adalah $[-0,4604 \ -0,2998 \ -9,7615]$. Berdasarkan pada gambar 4.11, bias akselerometer tidak bisa terestimasi

dengan baik, atau tidak sama dengan nilai bias yang dicari dengan metode diatas.



Gambar 4. 12 Perbandingan hasil estimasi bias giroskop dari penyelarasan halus dengan menggunakan UKF dan KF.

Dikarenakan *noise* giroskop yang besar, maka estimasi bias giroskop pada dengan cara mencari nilai rata-ratanya dan dikurangi dengan nilai giroskop tanpa *noise* dan tanpa bias bila diketahui *roll*, *pitch*,

yaw tidak dapat dilakukan. Sehingga kita tidak dapat melakukan validasi pada gambar 4.12.

Pada estimasi ke-sembilan state orientasi atau *attitude* serta bias akselerometer dan giroskop, memiliki estimasi yang cukup mirip dengan perbedaan yang tidak signifikan antara UKF dan KF. Hal ini diakibatkan oleh mekanisme *closed-loop* dari algoritma INS, pada kondisi kendaraan diam perubahan akselerometer dan giroskop juga tidak terlalu signifikan.

Pada algoritma INS, metode *closed-loop* ini me-reset setiap nilai estimasi ke-15 state pada iterasi berikutnya menjadi 0, sehingga propagasi state yang selalu dikoreksi ini tidak tumbuh seiring berjalannya waktu, sehingga estimasi state tersebut tidak pernah membesar. Artinya UKF tetap akan memberikan nilai estimasi yang kecil (walaupun nilai nonlinier pada kesalahan sudut seharusnya bernilai besar, sehingga dianggap tidak linier).

Tabel 4. 1 Selisih kesalahan antara penyelarasan kasar IMU dengan referensi CNS5000

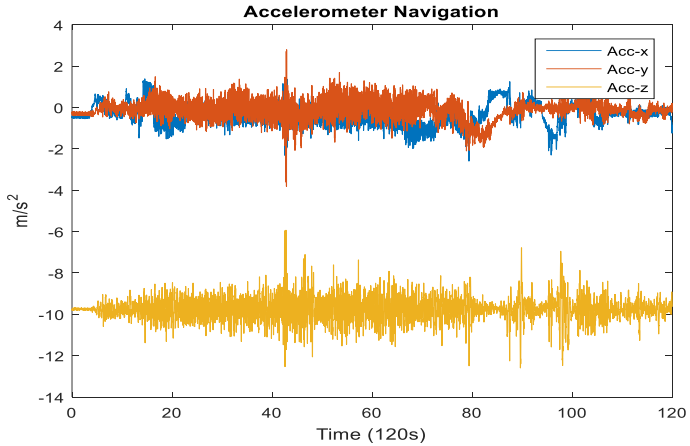
<i>Roll</i> (°)	<i>Pitch</i> (°)	<i>Yaw</i> (°)
0,191	0,887	18,88

Tabel 4. 2 RMSE dari penyelarasan halus IMU dengan referensi CNS5000

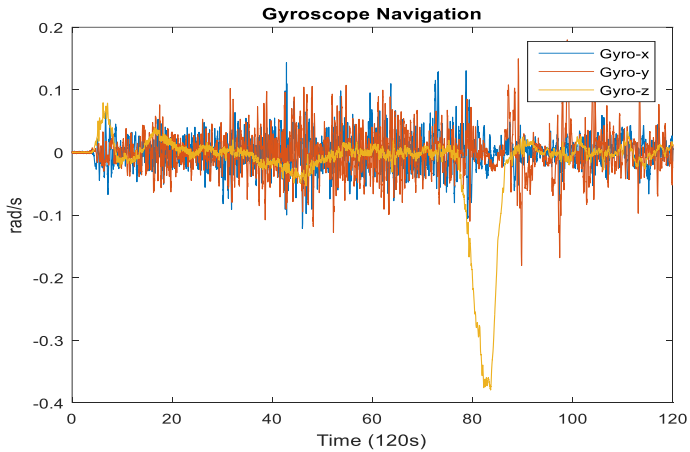
	<i>Roll</i> (°)	<i>Pitch</i> (°)	<i>Yaw</i> (°)
KF	0,1879	0,8904	0,6540
UKF	0,1879	0,8904	0,6539

4.2 Pengujian Sistem Navigasi Inersia

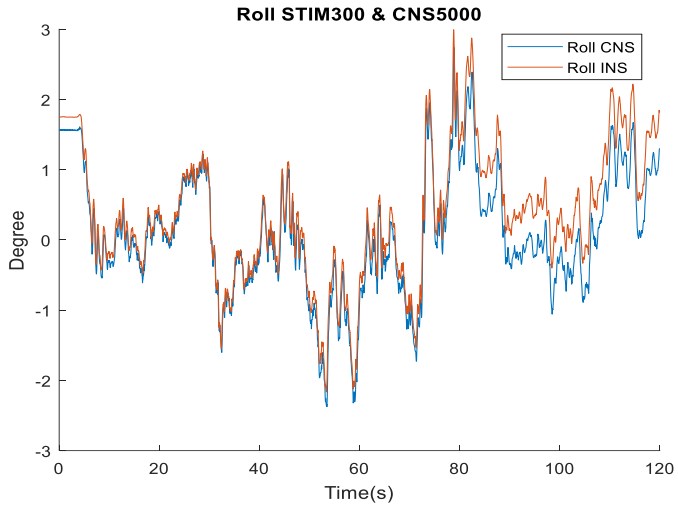
Pengujian sistem navigasi inersia tanpa integrasi ini menggunakan nilai awal estimasi orientasi *roll*, *pitch*, *yaw*, dan juga bias hasil dari penyelarasan, dan inisialisasi dari GPS untuk posisi, dan kecepatan. Gambar 4.13 adalah data akselerometer dari STIM300 saat bernavigasi selama 120 detik. Sedangkan gambar 4.14 adalah data giroskop akselerometer dari STIM300 saat bernavigasi selama 120 detik.



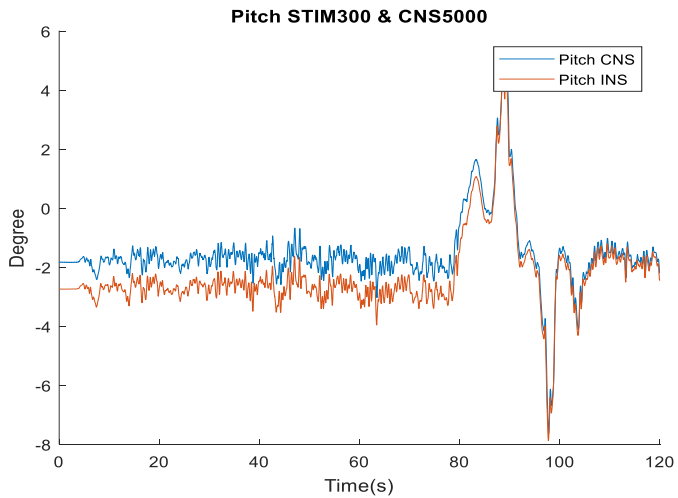
Gambar 4. 13 Data akselerometer STIM 300 saat bergerak selama 120 detik



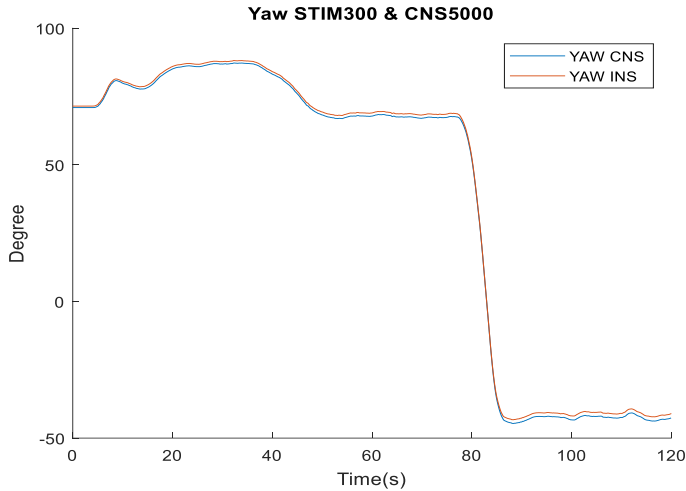
Gambar 4. 14 Data akselerometer STIM 300 saat bergerak selama 120 detik.



Gambar 4. 15 Perbandingan *roll* INS dengan CNS



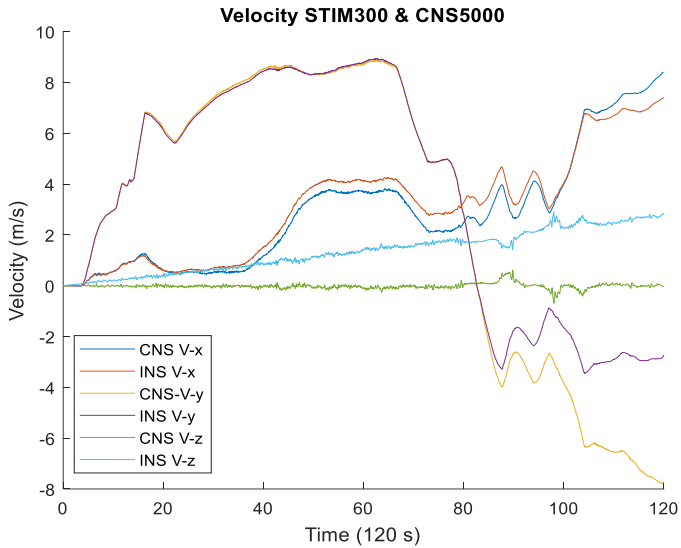
Gambar 4. 16 Perbandingan *pitch* INS dengan CNS



Gambar 4. 17 Perbandingan yaw INS dan CNS

Berdasarkan gambar 4.15, terlihat bahwa estimasi *roll* dari hasil mekanisasi menunjukkan bahwa tidak ada koreksi, sehingga kesalahan dari IMU dan inialisasi *roll* dari penyelarasan tetap ada. Hal ini menyebabkan hasil estimasi pada detik ke- 80 berbeda dengan referensi. RMSE *roll* sebesar $0,3432^\circ$. Berdasarkan gambar 4.16, hasil mekanisasi *pitch* terlihat tidak ada koreksi, sehingga kesalahan pada awal inialisasi tetap ada, dan terlihat tidak ada reduksi. RMSE *pitch* mekanisasi sebesar $0,8009^\circ$. Berdasarkan gambar 4.17 hasil mekanisasi *yaw* terlihat bahwa kesalahan inialisasi tidak terkoreksi, hal ini dikarenakan tidak ada reduksi kesalahan selama 120 detik. RMSE *yaw* sebesar $1,1105^\circ$

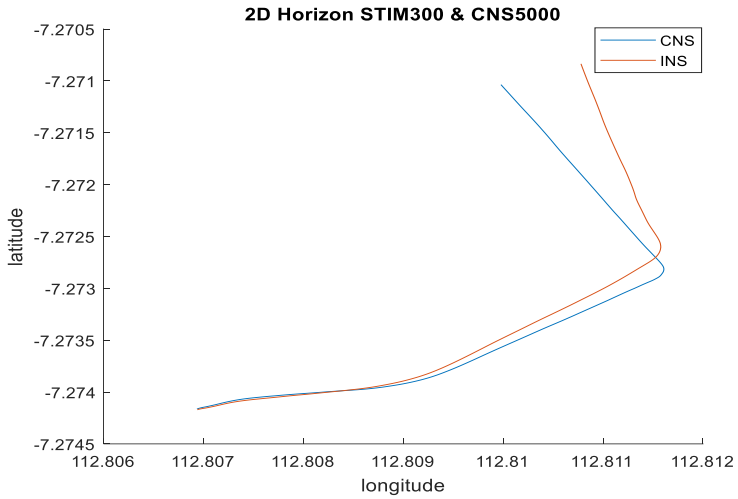
Untuk kecepatan (v_N, v_E, v_D) hasilnya seperti berikut.



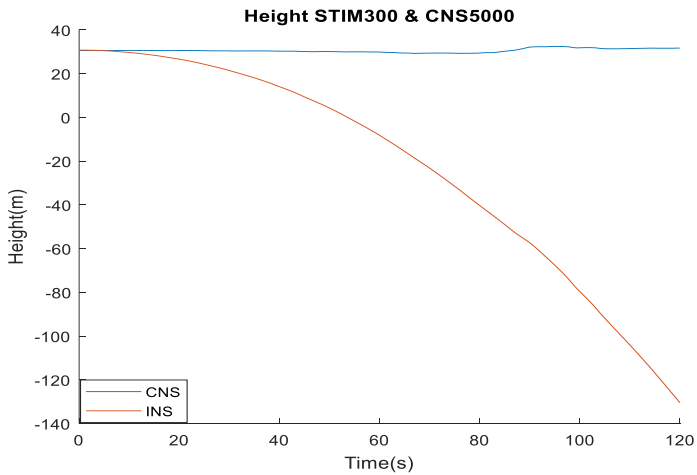
Gambar 4. 18 Perbandingan kecepatan wahana antara INS dan CNS

Berdasarkan gambar 4.18, terlihat bahwa kecepatan menghasilkan kesalahan yang terus tumbuh seiring berjalannya waktu. Hal ini disebabkan karena adanya bias dan gangguan yang tidak terkoreksi, sehingga menyebabkan tumbuhnya kesalahan dari waktu ke waktu yang terakumulasi pada mekanisasi. RMSE hasil mekanisasi dari kecepatan adalah [0,4478 m/s 1,5850 m/s 1,5562 m/s]

Untuk posisi hasilnya adalah sebagai berikut



Gambar 4. 19 Perbandingan posisi wahana INS dan CNS



Gambar 4. 20 Perbandingan ketinggian wahana INS dan CNS

RMSE posisi lintang, bujur, dan ketinggian dari hasil mekanisasi adalah [17,9484 m 21,6933 m 71,4158 m]. Dari gambar 4.19 dan

gambar 4.20 terlihat bahwa semua state memiliki RMSE yang cukup besar terutama untuk posisi karena hasil mekanisasi mengakibatkan terakumulasinya kesalahan dari waktu ke waktu, sehingga kesalahan akan tumbuh seiring berjalannya waktu.

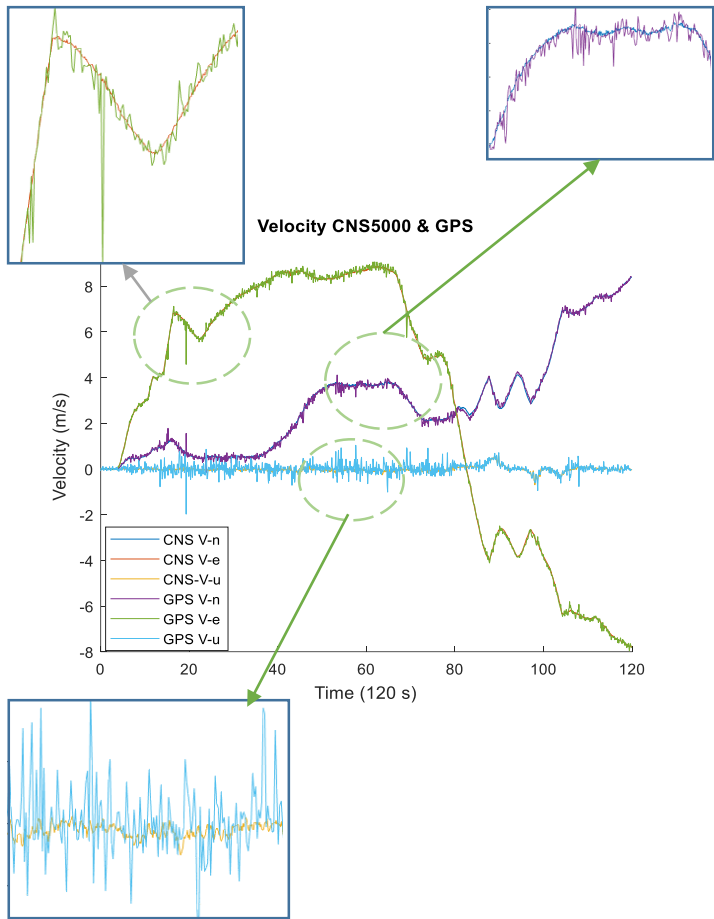
Dalam tabel 4.3. disajikan perbandingan RMSE dari detik ke – 1 sampai dengan detik ke- 20 dengan detik ke- 80 sampai detik ke-100, Kedua RMSE terlihat lebih besar di hampir semua state di detik ke-80 sampai dengan detik ke-100. Hal ini menunjukkan bahwa kesalahannya berubah-ubah seiring berjalannya waktu dan dalam state tertentu bertambah besar. Sehingga dalam kesalahan jangka panjang sangat buruk, namun dalam jangka pendek masih stabil

Tabel 4. 3 RMSE pengujian sistem navigasi inersia dengan referensi CNS5000

State	RMSE	RMSE dari detik ke- 1 sampai ke-20	RMSE dari detik ke- 80 sampai ke-100
<i>Roll</i> (°)	0,3432	0,1451	0,5319
<i>Pitch</i> (°)	0,8009	0,9282	0,45
<i>Yaw</i> (°)	1,110	0,7704	1,18
Kecepatan sumbu <i>north</i> (m/s)	0,4478	0,0376	0,5804
Kecepatan sumbu <i>east</i> (m/s)	1,5850	0,0353	1,17
Kecepatan sumbu <i>down</i> (m/s)	1,556	0,2329	2,032
Lintang (m)	17,948	1,2874	28,6312
Bujur (m)	21,693	0,2624	6,29316
Ketinggian (m)	71,4158	1,718	90,4424

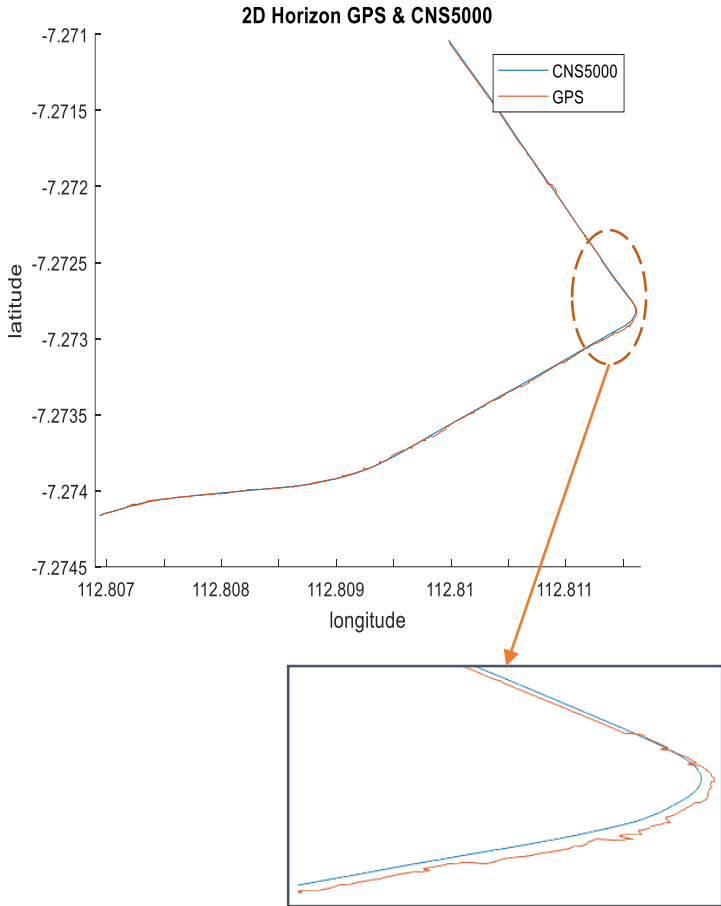
4.3 Pengujian GPS

Pengujian GPS ini dilakukan dengan membandingkan data GPS yang didapat dengan data referensi. Data ini di ambil dengan wahana mobil oleh PT. Infoglobal di Surabaya. Hasilnya data GPS memiliki tingkat variasi yang cukup tinggi untuk jangka pendek, namun untuk jangka panjang sensor ini tidak memiliki *drift*, sehingga cocok untuk dijadikan sebagai pengukuran pada Filter Kalman.



Gambar 4. 21 Perbandingan antara kecepatan CNS dan GPS

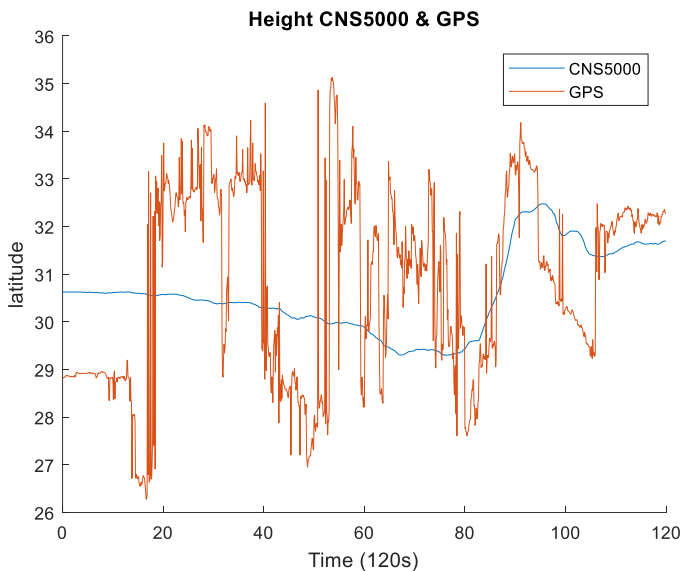
GPS hanya menyediakan state kecepatan, lintang, bujur, dan ketinggian, yang semua sudah dalam kerangka navigasi. GPS ini akan memberikan informasi sebanyak 6 state yaitu kecepatan yang teramati di kerangka navigasi (*north, east, down*) dan posisi terhadap lengkung bumi (lintang, bujur, dan ketinggian).



Gambar 4. 22 Perbandingan antara posisi CNS dan GPS

Pada state kecepatan (v_N, v_E, v_D) yang ditunjukkan pada gambar 4.21 terlihat bahwa GPS memiliki frekuensi *noise* yang cukup tinggi untuk jangka pendek. Walaupun demikian sensor ini dapat melakukan *tracking* terhadap nilai referensi untuk jangka panjang. Pada state kecepatan GPS memiliki RMSE sebesar [0,1116 m 0,1271 m 0,2071 m]

Pada state posisi (L_b, λ_b) yang ditunjukkan gambar 4.22 terlihat bahwa GPS memiliki frekuensi *noise* yang lebih rendah untuk jangka pendek, dan sensor ini dapat melakukan *tracking* terhadap nilai referensi. Pada state ketinggian gambar 4.23 sensor GPS memiliki *noise* berfrekuensi tinggi. Pada state posisi, terdapat RMSE sebesar [0,6982 m 1,1995 m 1,9930 m].



Gambar 4. 23 Perbandingan antar ketinggian CNS dan GPS

Tabel 4. 4 RMSE GPS dengan referensi CNS5000

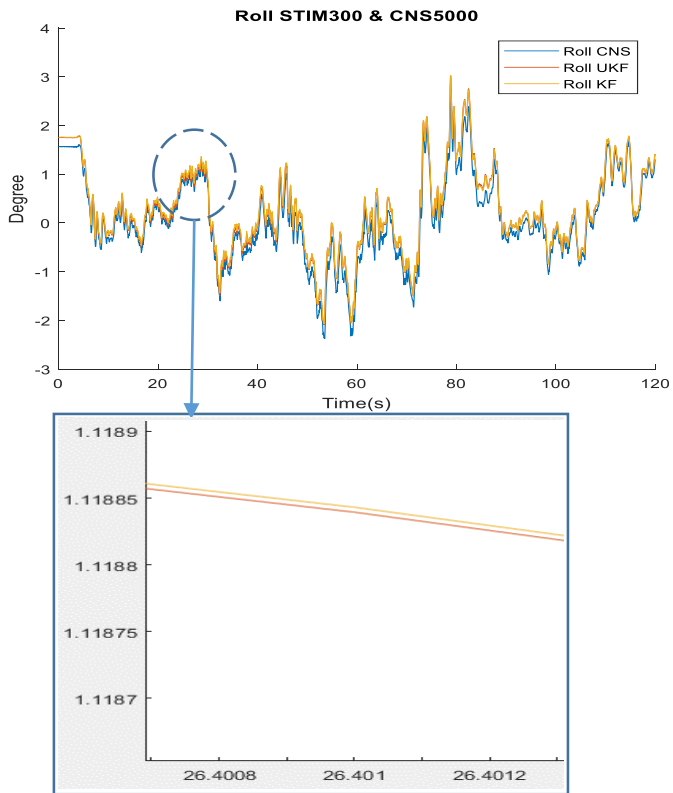
State	RMSE	RMSE dari detik ke- 1 sampai ke- 20	RMSE dari detik ke- 80 sampai ke- 100
Kecepatan sumbu <i>north</i> (m/s)	0,116	0,1116	0,1144
Kecepatan sumbu <i>east</i> (m/s)	0,1271	0,1633	0,0899
Kecepatan sumbu <i>down</i> (m/s)	0,2071	0,2308	0,1015
Lintang (m)	0,6982	0,8570	0,8570
Bujur (m)	1,1195	0,8028	0,8028
Ketinggian (m)	1,9930	1,2383	1,2383

Berdasarkan tabel 4.4. perbandingan RMSE dari detik ke-1 sampai dengan detik ke- 20 dengan detik ke- 80 sampai dengan detik ke- 100 terlihat stabil. Hal ini menunjukkan tidak ada *drift* atau kesalahan yang menumbuh dari GPS seiring berjalannya waktu. Namun, berdasarkan gambar 4.21, gambar 4.22, dan gambar 4.23 dalam jangka pendek sinyal GPS terlihat buruk

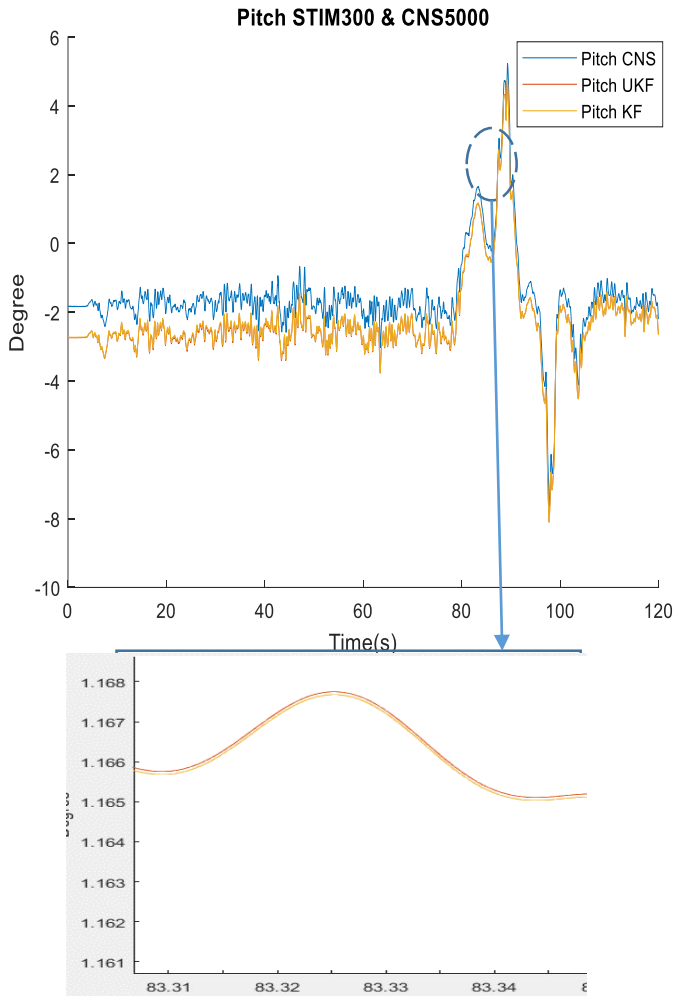
4.4 Pengujian Sistem Navigasi Inersia dengan Integrasi GPS

Pada pengujian sistem navigasi inersia dengan integrasi ini adalah menggunakan mekanisme hasil penyalarsan awal dengan UKF dan KF dan mekanisasi yang diintegrasikan menggunakan *Unscented Kalman Filter* dengan pengukuran dari GPS menggunakan mekanisme *closed loop*, yaitu koreksi pada waktu iterasi sekarang akan digunakan untuk iterasi selanjutnya.

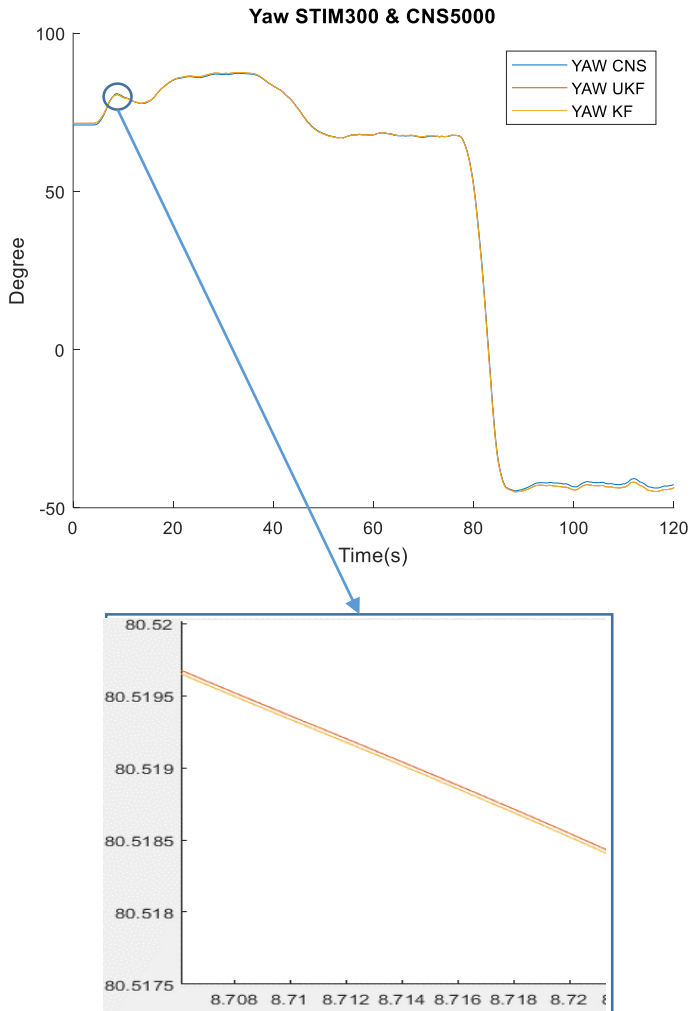
Adapun estimasi *roll*, *pitch*, *yaw* dengan UKF cukup baik dengan RMSE masing- masing sebesar $[0,2049^\circ \ 0,747^\circ \ 0,5677^\circ]$ $[0,2049^\circ \ 0,7468^\circ \ 0,5683^\circ]$ untuk KF secara berurutan.



Gambar 4. 24 Perbandingan koreksi *roll* dengan GPS menggunakan UKF dan KF



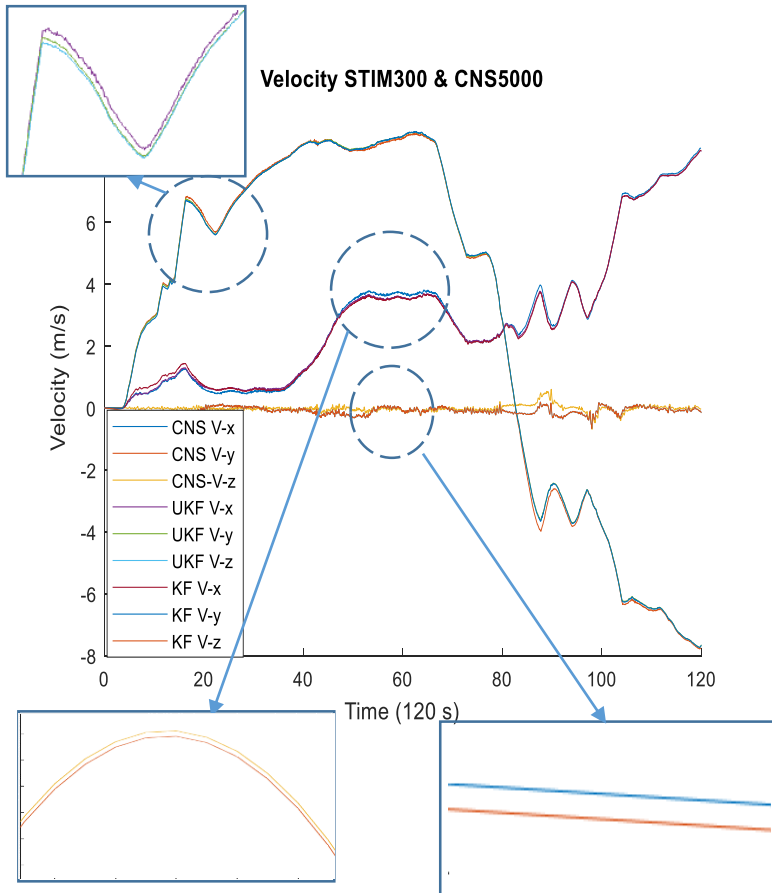
Gambar 4. 25 Perbandingan koreksi *pitch* dengan GPS menggunakan UKF dan KF



Gambar 4. 26 Perbandingan koreksi yaw dengan GPS menggunakan UKF dan KF

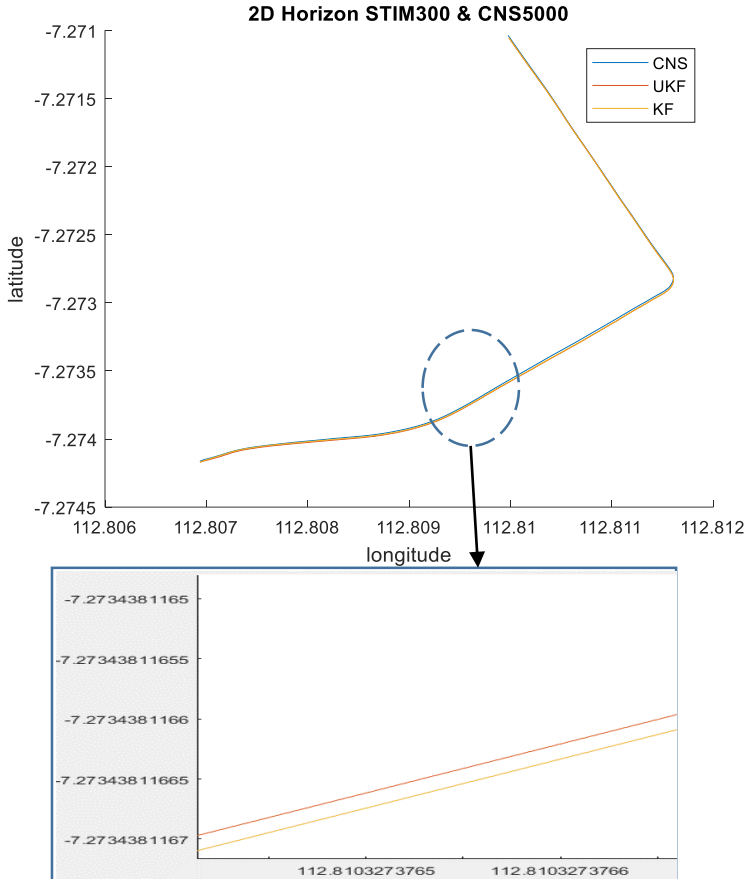
Untuk kecepatan (v_N, v_E, v_D) hasilnya ditunjukkan pada gambar 4.27. Terlihat sudah ada koreksi dari GPS namun masih ada kesalahan,

tetapi kesalahan ini terbatas (*bounded*) atau tidak menumbuh seiring berjalannya waktu RMSE masing-masing metode adalah [0,0786 m/s 0,0845 m/s 0,1424 m/s] untuk UKF , dan [0,0786 m/s 0,0845 m/s 0,1426 m/s] untuk KF.

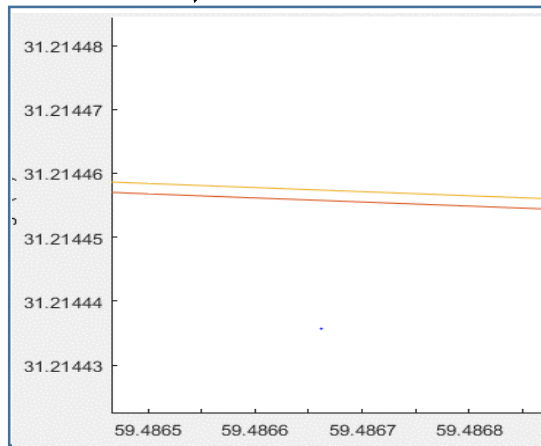
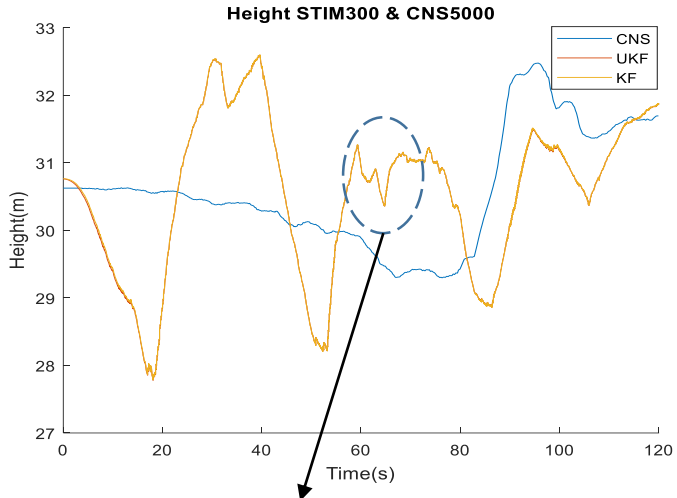


Gambar 4. 27 Perbandingan koreksi kecepatan dengan GPS menggunakan UKF dan KF

Untuk state posisi, hasilnya seperti gambar 4.28 dan gambar 4.29. Terlihat sudah ada koreksi dari GPS walaupun masih ada kesalahan, tetapi kesalahan ini juga terbatas seiring berjalannya waktu. RMSE masing-masing metode adalah [1,5214 m 0,6279 m 1,2978 m] untuk UKF dan [1,5215 m 0,6279 m 1,2991 m] untuk KF.



Gambar 4. 28 Perbandingan koreksi lintang dan bujur dengan GPS menggunakan UKF dan KF



Gambar 4. 29 Perbandingan koreksi ketinggian dengan GPS menggunakan UKF dan KF

Tabel 4. 5 Perbandingan antara RMSE integrasi INS GPS dengan UKF dan KF dengan referensi CNS5000

State	RMSE UKF	RMSE KF
<i>Roll</i> (°)	0,2049	0,2049
<i>Pitch</i> (°)	0,747	0,747
<i>Yaw</i> (°)	0,5677	0,5683
Kecepatan sumbu <i>north</i> (m/s)	0,0786	0,0786
Kecepatan sumbu <i>east</i> (m/s)	0,0846	0,0846
Kecepatan sumbu <i>down</i> (m/s)	0,1424	0,1426
Lintang (m)	1,5214	1,5215
Bujur (m)	0,6279	0,6279
Ketinggian (m)	1,2978	1,2991

Tabel 4. 6 Peningkatan performansi INS dengan GPS dan tanpa GPS dengan menyelisihkan kedua RMSE dan membandingkan dengan RMSE INS tanpa GPS

State	%Peningkatan
<i>Roll</i>	40,30
<i>Pitch</i>	6,73
<i>Yaw</i>	48,85
Kecepatan sumbu <i>north</i>	82,44
Kecepatan sumbu <i>east</i>	94,66
Kecepatan sumbu <i>down</i>	90,85
Lintang	91,52
Bujur	97,11
Ketinggian	98,18

Tabel 4. 7 Peningkatan performansi INS dengan UKF dan KF dengan menyelisihkan kedua RMSE dan membandingkan dengan RMSE INS GPS dengan KF

State	%Peningkatan
<i>Roll</i>	0%
<i>Pitch</i>	0%
<i>Yaw</i>	0,1055%
Kecepatan sumbu <i>north</i>	0%
Kecepatan sumbu <i>east</i>	0%
Kecepatan sumbu <i>down</i>	0,1403%
Lintang	0,006%
Bujur	0%
Ketinggian	0,1%

Berdasarkan tabel 4.6 integrasi INS dengan GPS saat navigasi dapat meningkatkan kepresisian dan keakuratan navigasi, dengan peningkatan performansi yang cukup besar pada kecepatan dan posisi. Hal ini dikarenakan kesalahan pada INS tanpa integrasi semakin meningkat seiring berjalannya waktu, sehingga tingkat koreksi yang diberikan oleh Filter Kalman juga semakin besar.

Sedangkan pada tabel 4.5 integrasi INS GPS dengan UKF terlihat lebih baik dengan RMSE lebih kecil pada state *yaw*. Sedangkan pada tabel 4.7 terlihat terdapat peningkatan performansi, namun peningkatan performansi ini tidak terlalu signifikan. Hal ini dikarenakan sistem *closed-loop* yang digunakan sudah memiliki performansi yang baik dan mencegah tumbuhnya nilai state di setiap propagasi dengan mereset nilai semua statenya .

Tabel 4. 8 Perbandingan waktu komputasi UKF dengan KF dalam detik

UKF	KF
2.762,02859	282,174773

Berdasarkan tabel 4.8 terlihat bahwa waktu komputasi dari UKF mencapai 9.8 kali lipat dari waktu komputasi KF. Hal ini disebabkan oleh UKF yang menggunakan *sigma point* sebanyak 31 yang harus di iterasikan disetiap state, sehingga konsumsi waktu pada saat penyelarasan

halus lebih banyak dan juga disebabkan pula karena juga *rate* dari Kalman Filter sebesar 2.000 Hz pada saat penyetaraan halus.

BAB 5 KESIMPULAN

Bab ini menjelaskan hasil dari pengujian dan simulasi yang telah dilaksanakan yang dinyatakan berupa kesimpulan dan saran. Selama proses penelitian terdapat kendala yang dihadapi. Kendala tersebut telah penulis rangkum menjadi saran sebagai acuan penyempurnaan penelitian lebih lanjut.

5.1. Kesimpulan

Adapun kesimpulan dari penelitian ini adalah sebagai berikut:

- a. Bias dan *white noise* sangat mempengaruhi proses penyelerasan awal, baik untuk proses *leveling* dan *gyrocompassing*. Bias akselerometer sebesar $[-0,1485\text{m/s}^2 - 0,0330\text{m/s}^2 \quad 0,0047\text{m/s}^2]$ menyebabkan kesalahan sebesar $0,1875^\circ$ untuk *roll* dan $0,8899^\circ$ pada *pitch*, sedangkan pada giroskop, nilai bias tidak dapat diestimasi dikarenakan varians *noise* yang besar, yaitu $[0,0013\text{ rad/s} \quad 0,0018\text{ rad/s} \quad 0,0004\text{ rad/s}]$ yang mana lebih besar dari kecepatan rotasi Bumi yaitu $7,272205\text{e-}5\text{ rad/s}$.
- b. Estimasi bias dengan *Unscented Kalman Filter* saat penyelerasan halus belum dapat mendekati nilai sebenarnya.
- c. INS merupakan sistem navigasi yang bersifat kontinyu yang sifatnya bagus dalam jangka pendek namun buruk dalam jangka panjang, untuk posisi *latitude* pada INS pada detik ke 1-20 memiliki RMSE sebesar 1,2874 meter sedangkan pada detik ke 80-100 memiliki RMSE yang membesar yaitu sebesar 28,6312 meter, berkebalikan dengan GPS, untuk posisi *latitude* pada detik ke 1-20 memiliki RMSE sebesar 0,8570 meter, dan pada detik ke 80-100 memiliki RMSE yang sama yaitu 0,8570 meter.
- d. Perancangan integrasi INS-GPS dengan menggunakan UKF berhasil. Hal ini dibuktikan dengan RMSE integrasi INS-GPS pada setiap state lebih kecil dibandingkan dengan tanpa GPS, untuk posisi *latitude* dengan integrasi GPS memiliki RMSE sebesar 1,5214 meter, sedangkan untuk *latitude* tanpa GPS memiliki RMSE sebesar 17,948 meter.
- e. Performansi *Unscented Kalman Filter* dipengaruhi oleh nilai setiap propagasi state. Dikarenakan nilai estimasi yang *direset*

disetiap iterasi (akibat dari *closed-loop implementation*) membuat nilai statenya tetap kecil, sehingga hasil estimasi antara KF dan UKF tidak jauh berbeda, untuk state *yaw* memiliki peningkatan sebesar 0,1055%, kecepatan sumbu *down* sebesar 0,1403% untuk posisi lintang sebesar 0,0006%, dan ketinggian sebesar 0,1%.

- f. Waktu komputasi yang diperlukan UKF lebih lama, yaitu 2.762,0285 detik dan untuk KF sebesar 282,174 detik, disebabkan oleh adanya 15 state yang harus diestimasi, sehingga *sigma points* yang dihasilkan pada UKF sebanyak $2n+1$, atau sebanyak 31 *sigma points* yang harus diiterasikan disetiap state. Pada penyelarasan memakan waktu lama karena UKF memiliki *rate* 2.000 Hz sedangkan saat navigasi hanya 10 Hz.
- g. Integrasi INS-GPS berhasil dilakukan dengan metode UKF namun peningkatan yang dihasilkan tidak terlalu signifikan bila dibandingkan dengan KF biasa akibat dari *closed-loop implementation*, dan juga membutuhkan waktu komputasi yang lebih lama.

5.2. Saran

Adapun saran dari penelitian ini adalah sebagai berikut :

- a. Penyederhanaan model penyelarasan agar UKF dapat memberikan waktu komputasi lebih cepat.
- b. Menggunakan metode lain agar meningkatkan kepresisian estimasi penyelarasan awal dengan komputasi yang lebih cepat.

DAFTAR PUSTAKA

- [1] Groves, Paul D. (2013), *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems* (Fourth ed.). Artech House.
- [2] Noureldin, Aboelmagd, Karamat, Tashfeen B., Georgy, Jacques (2013). *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Springer.
- [3] Jackson, Paul B., "Overview of missile flight control systems," Johns Hopkins Apl Technical Digest, Volume 29, Number 1 (2010).
- [4] Aydin, Gunes, "Aerodynamic Parameter estimation of a missile in closed loop control and validation with flight data," Thesis, The Graduate School of Natural and Applied Sciences of the Middle East Technical University (2010).
- [5] Robert Grover Brown, P. Y. "Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises" (Fourth ed). John Wiley & Sons, Danvers, 2012
- [6] Maliki, Maulana, "Perancangan Instrumen Sistem Navigasi dengan Kalman Filter dan Algoritma *Smoothing Savitzky Golay*", ITS, 2018
- [7] Van der Merwe R, Wan EA. The square-root unscented Kalman filter for state and parameter estimation, Oregon Graduate Institute of Science and Technology, Oregon, 2000.
- [8] "CNS 5000." www.kvh.com/Military-and-Government/Gyros-and-Inertial-Systems-and-Compasses/Gyros-and-IMUs-and-INS/INS/CNS-5000.aspx.
- [9] *Sensoror AS - High Precision MEMS Sensors*, www.sensoror.com/media/1132/ts1524r9-datasheet-stim300.pdf.
- [10] J. L. Crassidis, "Sigma-point Kalman filtering for integrated GPS and inertial navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 2, pp. 750–756, 2006.
- [11] A. M. H. Hasan, K. Samsudi, A. R. Ramli, R. S. Azmir, and S. A.

Ismaeel, "A Review of Navigation Systems (Integration and Algorithms," *Aust. J. Basic Appl. Sci.*, vol. 3, no. 2, pp. 943–959, 2009.

- [12] A. Nouredin, A. El-shafie, and M. Bayoumi, "GPS / INS integration utilizing dynamic neural networks for vehicular navigation," *Inf. Fusion*, vol. 12, no. 1, pp. 48–57, 2011.

LAMPIRAN

Berikut ini adalah lampiran berupa program yang digunakan pada Tugas Akhir ini

Lampiran 1 : Model kesalahan nonlinier INS

```
function x_new =
fx(x,est_L_b,est_h_b,est_v_eb_n,est_C_b_n,f_i
b_b,tor_s)
R0 = 6378137; % WGS84 Equatorial radius in
meters
e = 0.0818191908425; % WGS84 eccentricity
omega_ie = 7.292115*10^(-5); % Earth rotation
rate in rad/s

%% Contruction on input (Done)
est_v_eb_n_N = est_v_eb_n(1);
est_v_eb_n_E = est_v_eb_n(2);
est_v_eb_n_D = est_v_eb_n(3);

%% Variable Function (Done)
% R_E(est_L_b)
R_E_L = R_E(e,R0,est_L_b);
% R_N(est_L_b)
R_N_L = R_N(e,R0,est_L_b);
% g_0(est_L)
g0_L = getg0L(est_L_b);
% r_eS_e(est_L)
r_eS_e_L = r_eS_e(e,R_E_L,est_L_b);
% (Done)
errorOmega_ie_n = -
omega_ie*[sin(est_L_b);0;cos(est_L_b)]*x(7);
errorOmega_en_n =[x(5)/(R_E_L+est_h_b);-
x(4)/(R_N_L+est_h_b);...
;-
x(5)*tan(est_L_b)/(R_E_L+est_h_b)] -...
```

```

[0;0;1]*est_v_eb_n_E*x(7)/((R_E_L+est_h_b)*(
cos(est_L_b))^2) +...
    [-
est_v_eb_n_E/((R_E_L+est_h_b)^2);
est_v_eb_n_N/((R_N_L+est_h_b)^2)];...

;est_v_eb_n_E*tan(est_L_b)/((R_E_L+est_h_b)^
2)]*x(9);
errorCbn =
(Euler_to_CTM2([0,0,x(3)])*(eye(3) +
Skew_symmetric([x(1:2);0]))) ;
[Omega_ie_n, ~] = GetOmega_ie_n(est_L_b);
[Omega_en_n, ~] = GetOmega_en_n(est_L_b,
est_h_b, est_v_eb_n);
Omega_in_n = Omega_ie_n + Omega_ie_n;
O_3 = zeros(3,1);
cosLat=cos(est_L_b);
sinLat=sin(est_L_b);
omega_ie_n = [omega_ie*cosLat; 0; -
omega_ie*sinLat];
omega_en_n = [
est_v_eb_n(2)/(transv_rad_curv(est_L_b)+est_
h_b);-
est_v_eb_n(1)/(meridian_rad_curv(est_L_b) +
est_h_b);
est_v_eb_n(2)*tan(est_L_b)/(transv_rad_curv(
est_L_b)+est_h_b)];
omega_in_n = omega_ie_n+omega_en_n;

%% MODELLING
x_new      = zeros(15,1);
x_new(1:3) = -((eye(3)-errorCbn)*omega_in_n
+ errorOmega_ie_n + errorOmega_en_n) +
est_C_b_n*x(13:15);
x_new(4:6) = (eye(3)-errorCbn)*-
(est_C_b_n*f_ib_b) -

```

```

(Omega_en_n+2*Omega_ie_n)*x(4:6) +
Skew_symmetric([est_v_eb_n])*errorOmega_en_n
+2*Skew_symmetric([est_v_eb_n])*errorOmega_ie
_n - [0;0;1]*(2*g0_L/r_eS_e_L)*x(9) +
est_C_b_n*x(10:12);
x_new(7) = (x(4)/(R_N_L+est_h_b)) -
(est_v_eb_n_N*x(9)/(R_N_L+est_h_b)^2);
x_new(8) =
x(5)/((R_E_L+est_h_b)*cos(est_L_b)) +
est_v_eb_n_E*sin(est_L_b)*x(7)/((R_E_L+est_h_
b)*(cos(est_L_b))^2) -
est_v_eb_n_E*x(9)/((R_E_L+est_h_b)^2*cos(est_
L_b));
x_new(9) = -x(6);
x_new(10:12) = O_3;
x_new(13:15) = O_3;

%% Diskritisasi
x_new = x_new*tor_s+x;

```

Lampiran 2 : Perancangan Parameter Unscented Kalman Filter

```

% Constants
deg_to_rad = 0.01745329252;
rad_to_deg = 1/deg_to_rad;
micro_g_to_meters_per_second_squared =
9.80665E-6;
% Initial velocity uncertainty per axis (m/s)
LC_KF_config.init_vel_unc = 2;
% Initial position uncertainty per axis (m)
LC_KF_config.init_pos_unc = 3.005;
Initial accelerometer bias uncertainty per
instrument (micro-g, converted
% to m/s^2)
LC_KF_config.init_b_a_uncX = 5 *
micro_g_to_meters_per_second_squared;
LC_KF_config.init_b_a_uncY = 5 *
micro g to meters per second squared;

```

```

LC_KF_config.init_b_a_uncZ = 10 *
micro_g_to_meters_per_second_squared;
% Initial gyro bias uncertainty per
instrument (deg/hour, converted to rad/sec)
LC_KF_config.init_b_g_uncX = 1* deg_to_rad /
3600;
LC_KF_config.init_b_g_uncY = 5* deg_to_rad /
3600;
LC_KF_config.init_b_g_uncZ = 1* deg_to_rad /
3600;
% Gyro noise PSD (deg^2 per hour, converted
to rad^2/s)
LC_KF_config.init_att_uncX = .5*deg_to_rad;
LC_KF_config.init_att_uncY = 1*deg_to_rad;
LC_KF_config.init_att_uncZ = 0.5*deg_to_rad;
LC_KF_config.gyro_noise_PSDX =
(rad2deg(gyro_std(1))*0.01*2.9094e-04)^2;
LC_KF_config.gyro_noise_PSDY =
(rad2deg(gyro_std(2))*0.01*2.9094e-04)^2;
LC_KF_config.gyro_noise_PSDZ =
(rad2deg(gyro_std(3))*10*2.9094e-04)^2;
LC_KF_config.accel_noise_PSDX =
(acc_std(1)*0.5/60)^2;
LC_KF_config.accel_noise_PSDY =
(acc_std(2)*0.1/60)^2;
LC_KF_config.accel_noise_PSDZ = (0.5/60)^2;
% Gyro bias random walk PSD (rad^2 s^-3)
LC_KF_config.gyro_bias_PSDX =
((deg2rad(0.3/3600))^2/1000);%; (gyro_std(1))*0
.000001)^2; %2.0e-12;%(9*deg_to_rad)^2;
LC_KF_config.gyro_bias_PSDY =
((deg2rad(0.3/3600))^2/10);
LC_KF_config.gyro_bias_PSDZ =
((deg2rad(0.3/3600))^2/100);%; (gyro_std(3))*0.
00001)^2;

```



```

%% Position measurement noise SD per axis
(rad)
LC_KF_config.pos_meas_SDX =
deg2rad(0.000030); % 0.00001 deg = 1.1132 m
LC_KF_config.pos_meas_SDY =
deg2rad(0.000015);
LC_KF_config.pos_meas_SDZ = 0.5;
% Velocity measurement noise SD
per axis (m/s)
LC_KF_config.vel_meas_SD = 0.1;

```

Lampiran 3 : Algoritma *Unscented Kalman Filter*

```

function
[est_C_b_n_new,est_v_b_n_new,est_L_b_new,est_
lambda_b_new,est_h_b_new,est_IMU_bias_new,...

P_matrix_new,x_est_new]=coba_SPKF(GNSS_L_b,GN
SS_lambda_b, GNSS_h_b,GNSS_v_b_n,...

tor_s,est_C_b_n_old,est_v_b_n_old,...

est_L_b_old,est_lambda_b_old,est_h_b_old,est_
IMU_bias_old,...

P_matrix_old,meas_f_ib_b,LC_KF_config,x_est_o
ld)

% UKF Unscented Kalman Filter for nonlinear
dynamic systems
Q_prime_matrix = zeros(15);
Q_prime_matrix(1,1) =
LC_KF_config.gyro_noise_PSDX * tor_s;
Q_prime_matrix(2,2) =
LC_KF_config.gyro_noise_PSDY * tor_s;

```

```

Q_prime_matrix(3,3) =
LC_KF_config.gyro_noise_PSDZ * tor_s;
Q_prime_matrix(4,4) =
LC_KF_config.accel_noise_PSDX * tor_s;
Q_prime_matrix(5,5) =
LC_KF_config.accel_noise_PSDY * tor_s;
Q_prime_matrix(6,6) =
LC_KF_config.accel_noise_PSDZ * tor_s;
Q_prime_matrix(10,10) =
LC_KF_config.accel_bias_PSDX * tor_s;
Q_prime_matrix(11,11) =
LC_KF_config.accel_bias_PSDY * tor_s;
Q_prime_matrix(12,12) =
LC_KF_config.accel_bias_PSDZ * tor_s;
Q_prime_matrix(13,13) =
LC_KF_config.gyro_bias_PSDX * tor_s;
Q_prime_matrix(14,14) =
LC_KF_config.gyro_bias_PSDY * tor_s;
Q_prime_matrix(15,15) =
LC_KF_config.gyro_bias_PSDZ * tor_s;

% MEASUREMENT UPDATE PHASE
% Set-up measurement matrix using (12.115)
H_matrix = zeros(6,15);
H_matrix(1:3,7:9) = -
eye(3);%Skew_symmetric([0 0 omega_ie]');
H_matrix(4:6,4:6) = -eye(3);

%Set-up measurement noise covariance matrix

R_matrix(1,1) = LC_KF_config.pos_meas_SDx^2;
R_matrix(2,2) = LC_KF_config.pos_meas_SdY^2;
R_matrix(3,3) = LC_KF_config.pos_meas_SdZ^2;
R_matrix(1:3,4:6) = zeros(3);
R_matrix(4:6,1:3) = zeros(3);
R_matrix(4:6,4:6) = eye(3) *
LC_KF_config.vel_meas_SD^2;

```

```

%INISIALISATION OF WEIGHT
alpha=0.25;
beta =2;
kappa=0;
xEst = x_est_old;
n=length(xEst);%size of state vector
lamda=alpha^2*(n+kappa)-n;
%calculate weights
wm = zeros(2*n+1,1);
wc = zeros(2*n+1,1);
for i=2:2*n+1
    wm(i,1)=1/(2*(n+lamda));
    wc(i,1)=1/(2*(n+lamda));
end
wm(1)=lamda/(lamda+n);
wc(1)=(lamda/(lamda+n))+(1-alpha^2+beta);
gamma=sqrt(n+lamda); %c

%Compute the sigma point for Estimated States
sigma =
GenerateSigmaPoints(x_est_old,P_matrix_old,gamma);%15x31

%Unscented Transformation for the Sigma Point
Propagation of Estimated
%State
sigma=PredictMotion(sigma,est_L_b_old,est_h_b_
_old,est_v_b_n_old,est_C_b_n_old,meas_f_ib_b,
tor_s); %unscented transformation of
process

%Compute the mean of propagated sigma point
for Prediction States
xPred=sigma*wm;

```

```

%12. Compute the error variance of propagated
sigma point for Prediction States
PPred=CalcSigmaPointsCovariance(xPred, sigma, wc,
Q_prime_matrix);

%13. Compute the sigma point for Prediction
States
sigma=GenerateSigmaPoints(xPred, PPred, gamma);
%transformed cross-covariance

%14. Propagating the sigma points through h(x)
zSigma=PredictObservation(sigma, H_matrix);

%16. Calculating the mean of zsigma
zb=zSigma*wm;
delta_z = delta_z - zb;

%17. Calculating the error covariance of zsigma
St=CalcSigmaPointsCovariance(zb, zSigma, wc, R_mat
rix);

%18. Calculating cross covariance in order to
compute Kalman Gain
Pxz=CalcPxz(sigma, xPred, zSigma, zb, wc);

%19. Compute Kalman Gain
K=Pxz*inv(St);

%20. Calculating the mean
x_est_propagated = xPred;
x_est_new=x_est_propagated+K*(delta_z) ;
%state update

```

```

%17. Calculating P_matrix_new
P_matrix_new=PPred-K*St*K';

% Correct attitude, velocity, and position
using (12.7-9)
est_C_b_n_new =
(Euler_to_CTM2([0,0,x_est_new(3)])*(eye(3) +
Skew_symmetric([x_est_new(1:2);0])))' *
est_C_b_n_old;

est_v_b_n_new = est_v_b_n_old -
x_est_new(4:6);
est_L_b_new = est_L_b_old - (x_est_new(7));
est_lambda_b_new = est_lambda_b_old -
(x_est_new(8));
est_h_b_new = est_h_b_old - x_est_new(9);

% Update IMU bias estimates
est_IMU_bias_new = est_IMU_bias_old +
x_est_new(10:15);

function
sigmakuy=PredictMotion(sigma,est_L_b_old,est
_h_b_old,est_v_b_n_old,est_C_b_n_old,meas_f
_ib_b,tor_s)

% Sigma Points prediction with motion model
sigmakuy = zeros(size(sigma));
for i=1:length(sigma(1,:))
sigmakuy(:,i)=fx(sigma(:,i),est_L_b_old,est
_h_b_old,est_v_b_n_old,est_C_b_n_old,meas_f
_ib_b,tor_s);
end

```

```

function
sigma=PredictObservation(sigma,H_matrix)
% Sigma Points prediction with observation model
sigma = H_matrix*sigma;

function
P=CalcSigmaPointsCovariance(xPred,sigma,wc,N)
nSigma=length(sigma(1,:));
d=sigma-repmat(xPred,1,nSigma);
P = N+d*diag(wc)*d';

function P=CalcPxz(sigma,xPred,zSigma,zb,wc)
nSigma=length(sigma(1,:));
dx=sigma-repmat(xPred,1,nSigma);
dz=zSigma-repmat(zb,1,nSigma);
P = dx*diag(wc)*dz';

function
sigma=GenerateSigmaPoints(xEst,PEst,gamma)
n=length(xEst);
sigma = zeros(n,2*n+1);
sigma(:,1)=xEst;
Psqrt=sqrtm(PEst);
for i=1:n
    sigma(:,i+1)=xEst+gamma*Psqrt(:,i);
    sigma(:, n+i+1) = xEst-gamma*Psqrt(:,i);
end

```

Lampiran 4 : Perancangan Mekanisasi

```

function [L_b,lambda_b,h_b,v_eb_n,C_b_n] =
mekanisasi_baru(tor_i,
old_L_b,old_lambda_b,old_h_b,old_v_eb_n,old_C
_b_n,f_ib_b,omega_ib_b)
omega_ie = 7.292115E-5; % Earth rotation
rate (rad/s)
omega_ie_n = omega_ie * [cos(old_L_b); 0; -
sin(old_L_b)];
[old_R_N,old_R_E] =
Radii_of_curvature(old_L_b);
old_omega_en_n = [old_v_eb_n(2) / (old_R_E +
old_h_b);...
-old_v_eb_n(1) / (old_R_N + old_h_b);...
-old_v_eb_n(2) * tan(old_L_b) / (old_R_E
+ old_h_b)];
omega_in_n_k = omega_ie_n +
old_omega_en_n;
omega_in_b = old_C_b_n' * omega_in_n_k;
omega_lb_b = omega_ib_b - omega_in_b;
%rad/s
OMEGA_lb_b = [0 -
omega_lb_b(3) omega_lb_b(2);
omega_lb_b(3) 0
-omega_lb_b(1); -omega_lb_b(2)
omega_lb_b(1) 0];
C_b_n = old_C_b_n +
old_C_b_n*OMEGA_lb_b*tor_i;
v_eb_n = old_v_eb_n + tor_i * (f_ib_n +
Gravity_NED(old_L_b,old_h_b) -...
Skew_symmetric(old_omega_en_n + 2 *
omega_ie_n) * old_v_eb_n);

% Transform specific force to ECEF-frame
resolving axes using (5.86)
f_ib_n = C_b_n * f_ib_b;

```

```

% UPDATE VELOCITY
v_eb_n = old_v_eb_n + tor_i * (f_ib_n +
Gravity_NED(old_L_b,old_h_b) -...
    Skew_symmetric(old_omega_en_n + 2 *
omega_ie_n) * old_v_eb_n);

% UPDATE CURVILINEAR POSITION
h_b = old_h_b - 0.5 * tor_i * (old_v_eb_n(3) +
v_eb_n(3));

% Update latitude using (5.56)
L_b = old_L_b + 0.5 * tor_i * (old_v_eb_n(1) /
(old_R_N + old_h_b) +...
    v_eb_n(1) / (old_R_N + h_b));

% Calculate meridian and transverse radii of
curvature
[R_N,R_E]= Radii_of_curvature(L_b);

% Update longitude using (5.56)
lambda_b = old_lambda_b + 0.5 * tor_i *
(old_v_eb_n(2) / ((old_R_E +...
    old_h_b) * cos(old_L_b)) + v_eb_n(2) /
((R_E + h_b) * cos(L_b)));

```

Lampiran 5 : Perancangan Penyelarasan Kasar

```

phiAngle=zeros(20,1);
thetaAngle=zeros(20,1);
psiAngle=zeros(10,1);
prev_acc_mean=[0 0 0]';
prev_gyro_mean=[0 0 0]';
ignoreData = 10*2000;
init_L_b = deg2rad(cns01(1,1));
init_h_b = cns01(1,3);
init_g = Gravity_NED(init_L_b, init_h_b);

```



```

for coarse=1:20
startCoarseAlignment = (coarse-
1)*2000+ignoreData;
timeCoarseAlignment = 1;
fs=2000;
nData = 210*fs;
accCoarse=ned01(startCoarseAlignment+1:startC
oarseAlignment+nData,4:6);
gyroCoarse=ned01(startCoarseAlignment+1:start
CoarseAlignment+nData,1:3);
gyroCoarse=deg2rad(gyroCoarse);

if coarse == 1
    omega_ib_b_mean=mean(gyroCoarse);
    f_ib_b_mean =
(mean(accCoarse))*init_g(3); % (5.89)
else

omega_ib_b_mean=(mean(gyroCoarse)*fs+prev_gyr
o_mean*nDataCoarse)/(nDataCoarse+fs);
    f_ib_b_mean =
(((mean(accCoarse))*init_g(3))*fs+prev_acc_me
an*nDataCoarse)/(nDataCoarse+fs);
end

theta2=atan(f_ib_b_mean(1)/(sqrt(f_ib_b_mean(
2)^2+f_ib_b_mean(3)^2)));
phi2=atan2(-f_ib_b_mean(2), -f_ib_b_mean(3));
psi2=atan2(omega_ib_b_mean(3)*sin(phi2)-
omega_ib_b_mean(2)*cos(phi2),
omega_ib_b_mean(1)*cos(theta2)+...
omega_ib_b_mean(2)*sin(theta2)*sin(phi2)+omeg
a_ib_b_mean(3)*sin(theta2)*cos(phi2));
phiAngle(coarse)=rad2deg(phi2);
thetaAngle(coarse)=rad2deg(theta2);

```

```

psiAngle(coarse)=rad2deg(psi2);
nDataCoarse=coarse*fs;
prev_acc_mean=f_ib_b_mean;
prev_gyro_mean=omega_ib_b_mean;

end
theta=deg2rad(cns01(1,8));
phi=deg2rad(cns01(1,7));
psi=deg2rad(cns01(1,9));
eul_b_Coarse=[phi2 theta2 psi]';
gyro_std=std(gyroCoarse);
acc_std=std(accCoarse*init_g(3));

```

Lampiran 6 : Perancangan Penyelarasan Halus

```

startFineAlignment = startCoarseAlignment
+ timeCoarseAlignment * fs;
timeAlignment =210;
ignore = (210-timeAlignment);
fs=2000;
nData = timeAlignment*fs;
acc=ned01(startFineAlignment+1:startFineAl
ignment+nData,4:6);
gyro=ned01(startFineAlignment+1:startFineA
lignment+nData,1:3);
gyro=deg2rad(gyro);

% Initialize LC KF Config
Init_KF_config_uji_mobil;
%Initialize LC KF NED config;
Init_KF_config_uji_mobil;
true_L_b = init_L_b;
true_lambda_b = init_lambda_b;
true_h_b = init_h_b;
true_v_b_n = zeros(3,1);
true_eul_nb = eul_b_Coarse; % deg2rad([0 0
0]');
true_C_b_n = Euler_to_CTM2(true_eul_nb);

```

```

tor_s = 5e-4;%1e-2;
tor_i = 5e-4;
est_L_b = true_L_b;
est_lambda_b = true_lambda_b;
est_h_b = init_h_b;
t_corr=5e-4;
est_v_b_n = [0;0;0];
est_C_b_n = Euler_to_CTM2(eul_b_Coarse);
% Initialize estimated attitude solution
old_est_C_b_n =
Initialize_NED_attitude(true_C_b_n,initialization_errors);
old_est_v_b_n = [0;0;0];%true_v_b_n;
old_L_b = true_L_b;
old_lambda_b = true_lambda_b;
old_h_b = init_h_b;
P_matrix =
Initialize_LC_P_matrix(LC_KF_config);
est_IMU_bias = zeros(6,1);
est_IMU_bias_Corr = zeros(6,1); % for
correcting sensor
out_IMU_bias_est=zeros(nData, 6);
out_IMU_bias_est_Corr=zeros(nData/(t_corr*fs), 6);
out_IMU_bias_est(1,1:6) = est_IMU_bias';

GNSS_L_b = true_L_b;
GNSS_lambda_b = true_lambda_b;
GNSS_h_b = true_h_b;
GNSS_v_b_n = true_v_b_n;
% Data Output (PVA)
nDataOut = timeAlignment*(1/tor_s);
AttitudeKF=zeros(nDataOut,3);
% Attitude2=zeros(nDataOut,3);
VelocityKF=zeros(nDataOut,3);
PositionKF=zeros(nDataOut,3);
j=1;idCorr=1;
p = 1;

```

```

for k=1:nData
    % Collect data IMU
    % g0L=getg0L(est_L_b);
    gbn=Gravity_NED(est_L_b, est_h_b);
    meas_f_ib_b = acc(k,:)'*gbn(3);
    meas_omega_ib_b = gyro(k,:);

    if mod(k,20000) == 0
        p = p+1;
        % x_est;
    end
    % Correct IMU errors
    meas_f_ib_b = meas_f_ib_b -
est_IMU_bias_Corr(1:3);
    % est_IMU_bias_Corr(6)=0;
    meas_omega_ib_b = meas_omega_ib_b-
est_IMU_bias_Corr(4:6);

    % Update estimated navigation solution
    [est_L_b, est_lambda_b,
est_h_b,est_v_b_n,est_C_b_n] =
Nav_equations_NED(tor_i,...

old_L_b,old_lambda_b,old_h_b,old_est_v_b_n,
old_est_C_b_n,meas_f_ib_b,...
    meas_omega_ib_b);
x_est = zeros(15,1);

[est_C_b_n,est_v_b_n,est_L_b,est_lambda_b,e
st_h_b,est_IMU_bias,P_matrix,x_est] =...

coba_SPKF(GNSS_L_b,GNSS_lambda_b,GNSS_h_b,G
NSS_v_b_n,...
    tor_s,est_C_b_n,est_v_b_n,...
est_L_b,est_lambda_b,est_h_b,est_IMU_bias,P
_matrix,meas_f_ib_b,...
    LC_KF_config,x_est);

```

Lampiran 7 : Perancangan Integrasi INS-GPS

```
startNav = startFineAlignment +
(timeAlignment+ignore)*fs;
timeNav = 120;
fs=2000;
nDataNav = timeNav*fs;
accNAV=ned01(startNav+1:startNav+nDataNav,4
:6);
gyroNAV=ned01(startNav+1:startNav+nDataNav,
1:3);
gyroNAV=deg2rad(gyroNAV);
f_ib_b = 9.80665*accNAV;
omega_ib_b = rad2deg(gyroNAV);
j=240*10+1; % setelah 4 menit
dataGPSNAV=gps01(j:120*10+j,:);
tor_s = 0.1;
old_L_b = true_L_b;
old_lambda_b = true_lambda_b;
old_h_b = true_h_b;
old_est_v_b_n = [0;0;0];
old_est_C_b_n = est_C_b_n;

%prepare data output
AttitudeNavKF=zeros(nDataNav,3);
AttitudeNav2KF=zeros(nDataNav,3);
VelocityNavKF=zeros(nDataNav,3);
PositionNavKF=zeros(nDataNav,3);
GPS_v_b_n = zeros(120*10,3);
p =1;
m=1;
est_IMU_bias = est_IMU_bias_Corr;
chok = zeros(12,15);
for k=1:nDataNav
    % Collect data IMU
    % g0L=getg0L(est_L_b);
    gbn=Gravity_NED(est_L_b, est_h_b);
    meas_f_ib_b = accNAV(k,:)'*gbn(3);
    meas_omega_ib_b = gyroNAV(k,:)';
```

```

meas_f_ib_b = meas_f_ib_b -
est_IMU_bias_Corr(1:3);
    meas_omega_ib_b = meas_omega_ib_b -
est_IMU_bias_Corr(4:6);

    % Update estimated navigation solution
    [est_L_b, est_lambda_b,
est_h_b, est_v_b_n, est_C_b_n] =
Nav_equations_NED(tor_i, ...

old_L_b, old_lambda_b, old_h_b, old_est_v_b_n, o
ld_est_C_b_n, meas_f_ib_b, ...
    meas_omega_ib_b);

    % Run Kalman filter 200Hz
    if mod(k, (1/tor_i)/(1/tor_s)) == 0

        GNSS_L_b = deg2rad(dataGPSNAV(m,2));
        GNSS_lambda_b =
deg2rad(dataGPSNAV(m,3));
        GNSS_h_b = dataGPSNAV(m,4);
        GNSS_hor_speed = dataGPSNAV(m,5);
        GNSS_ground_track =
deg2rad(dataGPSNAV(m,6));
        GNSS_ver_speed = dataGPSNAV(m,7);
        GNSS_v_north = GNSS_hor_speed *
cos(GNSS_ground_track);
        GNSS_v_east = GNSS_hor_speed *
sin(GNSS_ground_track);
        GNSS_v_b_n = [GNSS_v_north;
GNSS_v_east; GNSS_ver_speed];
        GPS_v_b_n(m,:) = GNSS_v_b_n';
        x_est
x_est = zeros(15,1);
[est_C_b_n, est_v_b_n, est_L_b, est_lambda_b, es
t_h_b, est_IMU_bias, P_matrix, x_est] = ...

```

```

        est_IMU_bias_Corr = est_IMU_bias;
        m=m+1;
    end % end if Run kalman

    % Reset old values

        old_L_b = est_L_b;
    old_lambda_b = est_lambda_b;
    old_h_b = est_h_b;
    old_est_v_b_n = est_v_b_n;
    old_est_C_b_n = est_C_b_n;
    % SAVE OUTPUT

AttitudeNavKF(k,:) = rad2deg(CTM_to_Euler2(est_C_b_n));
    if(AttitudeNavKF(k,3) < 0)

AttitudeNavKF(k,3) = AttitudeNavKF(k,3) + 360;
    end
    VelocityNavKF(k,:) = est_v_b_n;
    PositionNavKF(k,1) = rad2deg(est_L_b);

PositionNavKF(k,2) = rad2deg(est_lambda_b);
    PositionNavKF(k,3) = est_h_b;
end

```

Lampiran 8 : Gravity model

```

function g = Gravity_NED(L_b,h_b)
% Parameters
R_0 = 6378137; %WGS84 Equatorial radius
in meters
R_P = 6356752.31425; %WGS84 Polar radius
in meters
e = 0.0818191908425; %WGS84 eccentricity
f = 1 / 298.257223563; %WGS84 flattening
mu = 3.986004418E14;

```

Lampiran 9 : Radius Lengkungan Bumi

```
function [R_N,R_E]= Radii_of_curvature(L)
R_0 = 6378137; %WGS84 Equatorial radius
in meters
e = 0.0818191908425; %WGS84 eccentricity

temp = 1 - (e * sin(L))^2;
R_N = R_0 * (1 - e^2) / temp^1.5;
R_E = R_0 / sqrt(temp);
```

Lampiran 10 : Transport rate

```
function [Omega_en_n,
omega_en_n]=GetOmega_en_n(L_b, h_b,
v_eb_n)
omega_en_n = [
v_eb_n(2)/(transv_rad_curv(L_b)+h_b);
v_eb_n(1)/(meridian_rad_curv(L_b) + h_b);
v_eb_n(2)*tan(L_b)/(transv_rad_curv(L_b)+
h_b)];
Omega_en_n = Skew_symmetric(omega_en_n);
end
```

Lampiran 11 : Kecepatan rotasi bumi dalam kerangka navigasi

```
function [Omega_ie_n,
omega_ie_n]=GetOmega_ie_n(L_b)
omega_ie=7.292115e-5;
cosLat=cos(L_b);
sinLat=sin(L_b);
omega_ie_n = [omega_ie*cosLat; 0; -
omega_ie*sinLat];
Omega_ie_n = omega_ie*[0 sinLat 0; -
sinLat 0 -cosLat; 0 cosLat 0];
End
```


RIWAYAT HIDUP PENULIS



Penulis, dengan nama lengkap Muhamad Rizal Ridlo Prakoso merupakan mahasiswa tahun angkatan 2015 Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember yang lahir di Kediri pada tanggal 3 Agustus 1997 sebagai putra bungsu dari 3 bersaudara. Riwayat pendidikan penulis adalah SDN Banjaran IV Kediri, SMPN 1 Kediri, dan SMAN 2 Kediri. Penulis pernah menjadi Ketua Bakti Sosial semasa SMA, dan pernah menjadi juara olimpiade sains tingkat kota di bidang Kebumihan. Semasa kuliah, penulis aktif sebagai staf HIMATEKTRO ITS dalam departemen KESMA. Penulis ingin menjadi bermanfaat untuk lingkungan sekitar, ingin membantu perekonomian kedua orangtua, dan ingin menjunjung derajat kedua orangtua. Di bidang pendidikan, penulis ingin melanjutkan program master sehingga nantinya ilmu yang diperoleh dapat bermanfaat bagi diri penulis dan lingkungan sekitar. Penulis sangat ramah, dan mudah tersenyum, bila ingin menghubungi silakan di rzlske48@gmail.com