

Combined GPS and Inertial Navigation System

By Andrew Aubry

Advised by

Dr. In Soo Ahn

Dr. Yufeng Lu

January 17, 2016

Electrical and Computer Engineering Department

Bradley University

1501 W Bradley Ave

Peoria, IL 61625

ABSTRACT

Highly accurate real time navigation has recently been made possible due to advances in satellite and inertial navigation technology. GPS provides consumers access to inexpensive methods of navigation, but is not always locally available. These outages can be mitigated through the use of an Inertial Navigation System (INS) which uses dead reckoning instead of external signals to compute the navigation solution. Advances in MEMS technology have produced inertial measurement units (IMUs) that fit on integrated circuit chips, and greatly improved portability. These MEMS sensors are highly susceptible to noise, which left unchecked can cause severe drift in the navigation solution. The combination of an INS and GPS can mitigate the drift, and combined with a linear estimator, the Kalman filter, can produce a viable navigation solution. A system to collect IMU and GPS data was constructed. A strapdown solution in MATLAB was also built. A Kalman filter was attempted but unsuccessfully implemented.

ACKNOWLEDGEMENTS

I would like to thank my advisers, Drs. In Soo Ahn and Yufeng Lu, for all the help. The final report is long overdue but my advisors were kind enough to wait patiently. My years at Bradley have prepared me to meet the challenge of real-life problems and solve them. I also thank the entire ECE faculty and staff for all their help during my study at Bradley University.

CONTENTS

I.	Introduction	5
II.	Inertial Navigation	6
	A. Complete Project Description	6
	B. Inertial Navigation Background.....	6
	C. Reference Frames.....	7
III.	Mathematical Development	9
	A. Inertial Navigation System	9
	B. Kalman Filter	11
IV.	System Setup and Equipment	12
	A. System Design	12
	B. Inertial Measurement Unit	12
	C. GPS	15
	D. Data Logger Design	16
V.	Results	20
	A. Data Logger	20
	B. Inertial Navigation System	22
VI.	Conclusion	24
VII.	Appendix	25
	A. IMU Data Log Sample.....	25
	B. GPS Data Log Sample	26
	References	27

LIST OF FIGURES

1.	System Block Diagram	7
2.	Navigation Reference Frames.....	8
3.	Strapdown Inertial Navigation System	9
4.	VN-100 Attitude Heading and Reference System	14
5.	uBlox EVK-5T GPS Unit	16
6.	Sparkfun Logomatic V3 Data Logger.....	17
7.	Sample IMU Message.....	17
8.	Data Logger Initialization Flowchart.....	18
9.	Data Logger Main Routine Flowchart	19
10.	Data Logger UART Interrupt.....	20
11.	IMU and Data Logger	21
12.	Strapdown Solution Flowchart	23
13.	Strapdown Solution Drift.....	23

LIST OF TABLES

1.	VectorNav VN-100 Sensor Specifications (25 °C)	14
2.	Data Transmission Rates.....	15
3.	uBlox EVK-5T Sensor Specifications	16
4.	Final System Parameters.....	21
5.	Drift of Stationary Sensor Test	23

I. Introduction

Recent advances in satellite and inertial navigation technology have made determining position and attitude of a vehicle in real time a reality. The advent of GPS has provided consumers with an inexpensive receiver for civilian navigation. Yet, there are occurrences where GPS signals are unavailable. A different method, not subject to external signal blockage, must be implemented in conjunction with GPS for navigation. The standard solution before GPS based navigation was an Inertial Navigation System (INS). INS systems have been used for navigation on many major projects including systems such as space vehicles and aircraft. An INS system measures a body's accelerations and angular rates to determine the position and attitude of the vehicle. Recent advances in MEMS technology have brought cheap accelerometers to the market. INS systems are an attractive choice where cost or available space are a concern.

The combination of GPS and INS allows for a highly accurate system that can mitigate the drawbacks of both systems. An INS provides a description of the body's accelerations and angular rates, and can detect near instantaneous changes in orientation. Current MEMS sensors, however, tend to be very noisy and can only provide short term stability for an INS solution. Heavy statistical processing is required to make effective use out of the MEMS accelerometer outputs [1]. A GPS system on the other hand, also provides an accurate vehicle position, but has a much slower refresh rate than that of any typical INS. A coupled system can be constructed by taking advantages of each system's strength. Such system can survive frequent GPS signal outages and improve position and attitude estimates of the vehicle [2].

II. Inertial Navigation

A. Complete Project Description

The project consists of four components or stages. These stages are the hardware and data acquisition using a data logger, a fundamental navigation system, an advanced navigation system, and finally an embedded system. The first system part is the hardware integration and data acquisition system. Hardware must be chosen for accuracy and ease of interfacing. When designing a combined navigation system, the timing of the measurements must be carefully considered. Timing must be kept between individual sensor measurements, and also between GPS and IMU measurements. The data acquisition system must have the ability to accurately track and record the time at which a measurement is made in addition to the actual measurement.

The second component is the basic implementation of the combined INS and GPS system. The system will be implemented in MATLAB, and utilize a Kalman filter for data fusion. GPS position data from a GPS receiver is fed directly into the filter, and combined with the INS solution to form a loosely coupled system. In order to model the MEMS based IMU sensors, a linear Gauss-Markov process is to be used. These components provide the basis for a combined GPS - INS navigation system, and allow for future expansion and refinement of the system. A block diagram of these components is in Figure 1.

B. Inertial Navigation

Inertial navigation is a form of navigation which relies solely on the inertial measurements and the initial position of a vehicle to determine position. Accelerometers provide the magnitude of the acceleration on a vehicle body, which is then integrated to yield velocity and position. An additional set of gyroscopes augment the accelerometers and allow the resolution of the

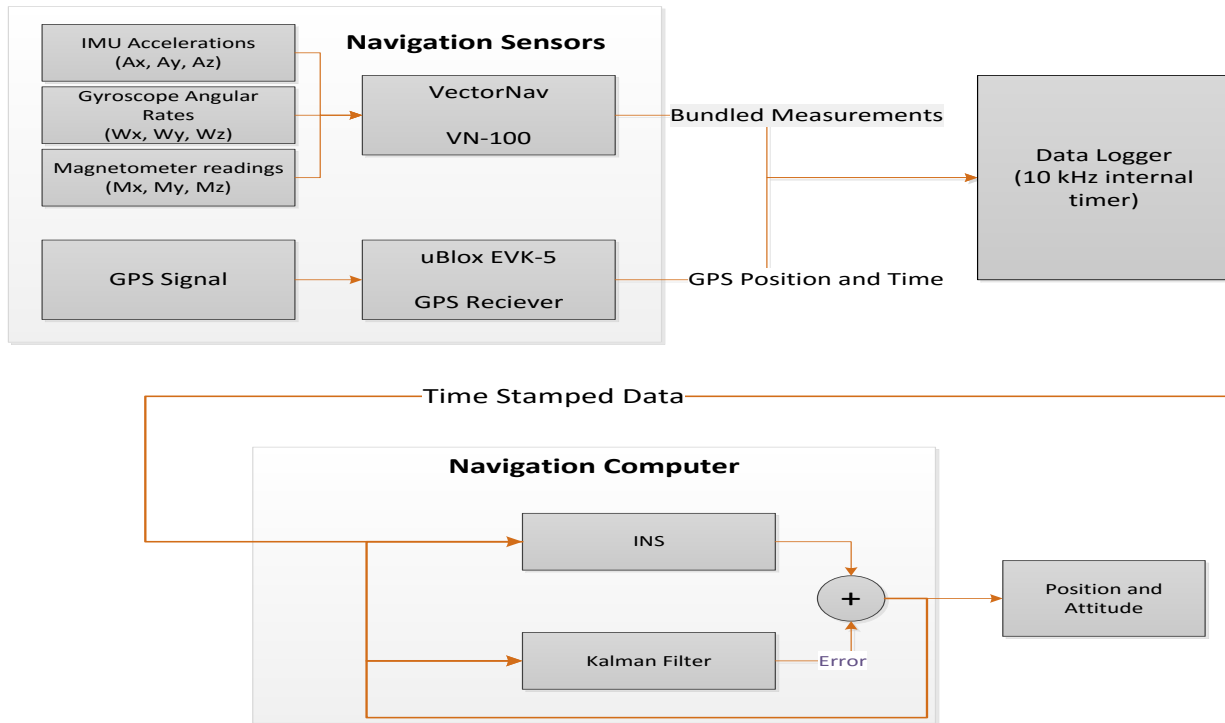


Figure 1: Complete system block diagram

directions of the accelerometers relative to a fixed axes or reference frame. Earlier inertial navigation systems utilized a set of gimbaled accelerometers to maintain the orientation of the measurements along fixed axes. The advent of more powerful embedded systems allowed the measurement units to be fixed directly to the body, creating a strapdown inertial navigation system (SINS).

C. Reference Frames

Multiple reference frames are used in a strapdown inertial navigation system (SINS). These frames include the inertial frame, the earth frame, the navigation frame, and the body frame. The inertial frame is defined by the axes Ox_i , Oy_i , and Oz_i , with the origin at the earth's center. The axes Ox_i and Oy_i are non-rotating with respect to fixed stars, and the Oz_i axis lies along the earth polar axis. The earth axes' origin is also located at the earth's center, with the Oz_e axis lying

along the earth polar axis, and the Ox_e axis pointing at the intersection of the equator and the Greenwich Meridian. The earth frame rotates about the inertial frame at a rate of Ω , defined as one revolution per sidereal day. The navigation frame is centered at the physical navigation system with axes aligned with the directions of north, east and the local vertical (down). Finally, the body frame is also located at the navigation system and coincident with the body's roll, pitch, and yaw axes. [3] An illustration of these frames relative to a globe is below in Figure 2.

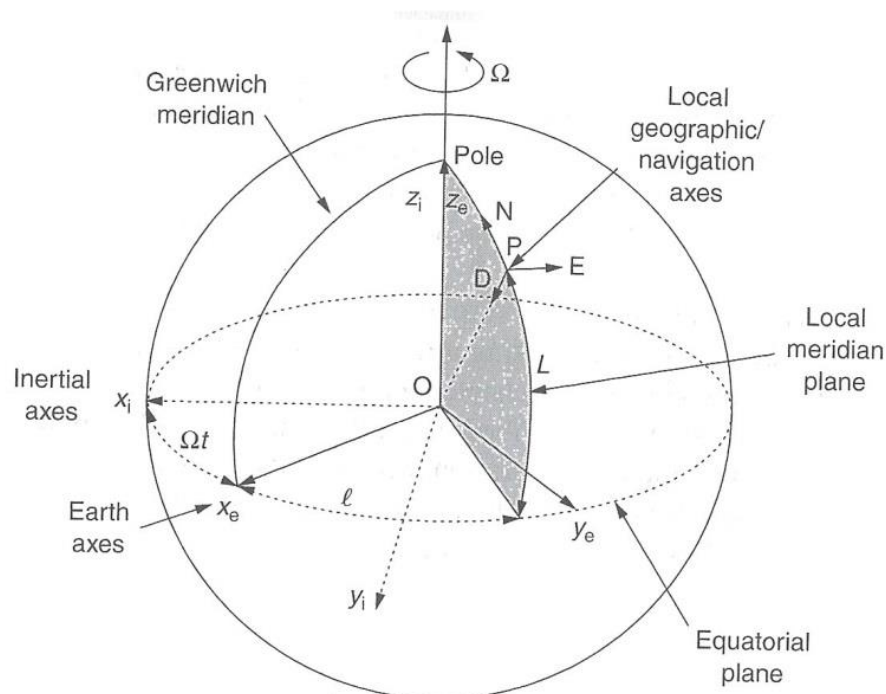


Figure 2: A representation of relative navigation frames used for inertial navigation [3]

The navigation frame is the primary frame of operation for a strapdown inertial navigation system. Measurements from the strapdown system components are taken from the body frame and must be transferred to the navigation frame. Additionally, the Coriolis Effect and local gravity forces are known in the earth and inertial frames respectively and must be transformed

accordingly. The translation of the navigation components is accomplished using methods that include a direction cosine matrix and a quaternion. A block diagram depicting the strapdown solution is shown in Figure 3.

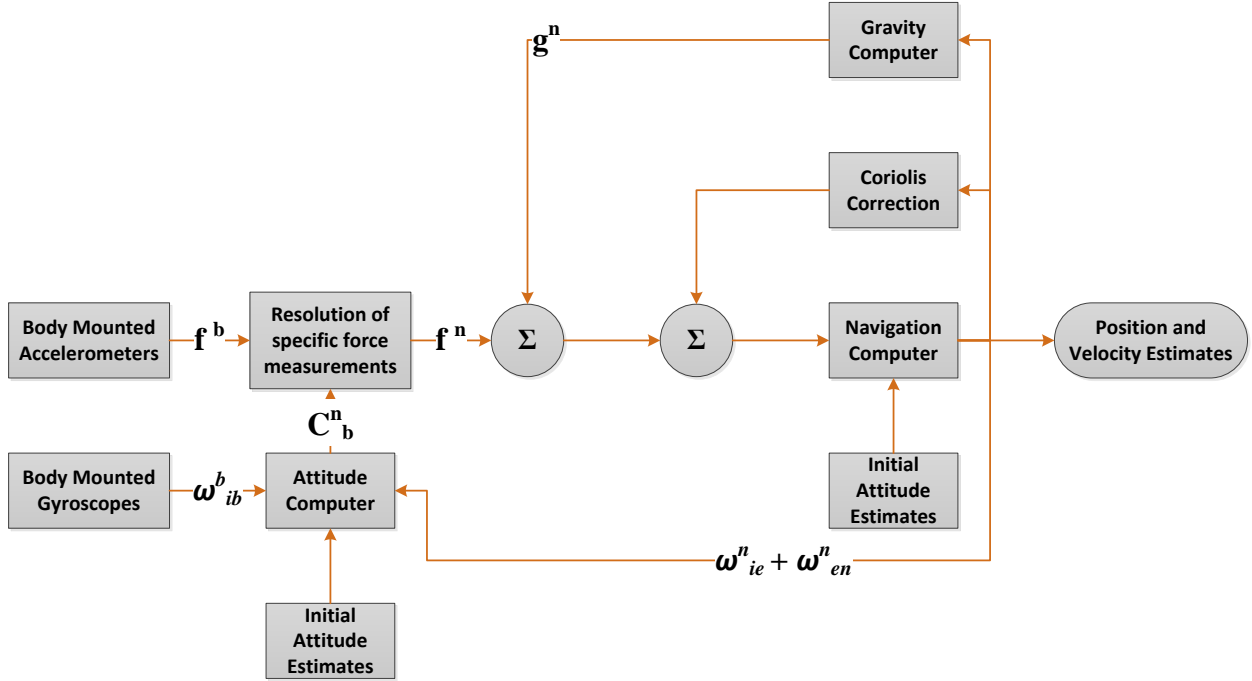


Figure 3: Block diagram of the navigation solution computed in the navigation frame [3]

III. Mathematical Development

A. Inertial Navigation System

The reference frame chosen for the strapdown solution is the Navigation Frame, also known as the North-East-Down frame (NED). The solution for the navigation frame is primarily drawn from [3]. The development of the strapdown requires vector and matrix notation, so let matrices be distinguished with upper-case bold font, and vectors denoted as lower-case bold font. A superscript on a vector or matrix signifies the current frame of navigation. When operating in the END frame, the navigation equation is expressed by:

$$\dot{\mathbf{v}}_e^n = \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_e^n + \mathbf{g}_l^n \quad (1)$$

Where the components of the equation are defined by the following:

$$\mathbf{v}_e^n = [v_N \quad v_E \quad v_D]^T \quad (2)$$

Representing the velocity with respect to Earth in the NED frame, with the components of velocity in the north, east, and local down directions. The accelerometers of the inertial measurement unit return the specific force vector, measured in the local body frame:

$$\mathbf{f}^b = [f_x \quad f_y \quad f_z]^T \quad (3)$$

In order to resolve the specific body forces into the navigation frame used in the solution, the direction cosine matrix, \mathbf{C}_b^n , is utilized. The direction cosine matrix is comprised of Euler angles to generate a rotation about the roll, pitch, and yaw axes in order to resolve the body frame components into navigation frame components. Next, the two rotation components are composed of the transport rate, $\boldsymbol{\omega}_{en}^n$, and the turn rate of the Earth in the navigation frame, $\boldsymbol{\omega}_{ie}^n$:

$$\boldsymbol{\omega}_{ie}^n = [\Omega \cos L \quad 0 \quad -\Omega \sin L]^T \quad (4)$$

The transport rate represents the turn rate of the local frame with respect to the Earth-fixed frame:

$$\boldsymbol{\omega}_{en}^n = [\dot{l} \cos L \quad -\dot{L} \quad -\dot{l} \sin L]^T \quad (5)$$

The variables L , l , h , and R_0 represent the latitude, longitude, altitude, and radius of the earth respectively. The local gravity vector, \mathbf{g}_l^n , is the final component of the navigation equation and is composed of the gravitation attraction of the earth, \mathbf{g} , and the centripetal acceleration of the earth:

$$\mathbf{g}_l^n = \mathbf{g} - \boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{R} = \mathbf{g} - \frac{\Omega^2(R_0+h)}{2} \begin{pmatrix} \sin 2L \\ 0 \\ 1 + \cos 2L \end{pmatrix} \quad (6)$$

In order to keep the resolved force components up to date, the direction cosine matrix is updated using the equation:

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^n \quad (7)$$

Where $\boldsymbol{\Omega}_{nb}^n$ is the skew symmetric form of the body rates in respect to the navigation frame, $\boldsymbol{\omega}_{nb}^n$. Because the body frame also rotates about the earth's axis and therefor the inertial frame, the gyroscope measurements are represented by $\boldsymbol{\omega}_{ib}^b$, and are converted to navigation rates by:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b [\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n] \quad (8)$$

To compute an equivalent direction cosine matrix, the quaternion may be used. The quaternion is an attitude representation that utilizes four parameters to represent a transformation by a single rotation about a vector defined with respect to the reference frame. The quaternion propagates through time by the equation:

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \mathbf{p} \quad (9)$$

Where

$$\mathbf{p} = \left[0, \boldsymbol{\omega}_{nb}^b \right]^T \quad (10)$$

B. Kalman Filter

In order to compensate for the noisy measurements of the inertial measurement unit and integrate the measurements from the GPS, a Kalman filter is used. The Kalman filter is a set of algorithms that are categorized as optimal state estimators, and is based on minimum mean-square error filtering with state space methods. Using a model of the dynamic system, combined with measurements of the system output, the Kalman filter is able to estimate the current internal states of a system. The Kalman filter is a discrete time algorithm. The development of the Kalman filter here is provided by [4] and [5]. A system is represented by its states and outputs:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (11)$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \quad (12)$$

With \mathbf{v}_k and \mathbf{w}_k representing Gaussian white noise, \mathbf{H}_k the observability matrix, and \mathbf{F}_k the plant dynamics. Using an initial estimate of the system, $\hat{\mathbf{x}}_0^-$, and an initial error covariance, P_0^- , an iterative process can be followed to estimate the internal states at discrete time points. Using the initial estimates, the Kalman Gain, \mathbf{K}_k , can be computed:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (13)$$

Using this gain, an *a posteriori* estimate of the current state can be found:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (14)$$

With the new estimate of the current states, a new covariance can be found for the current estimate:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (16)$$

Finally, using the current state estimate and covariance, *a priori* values for the next iteration can be calculated as:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_k \hat{\mathbf{x}}_k^+ + \mathbf{G}_k \mathbf{u}_k \quad (17)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k^+ \mathbf{F}_k^T + \mathbf{Q}_k \quad (18)$$

The Kalman filter presented here is a linear estimator. Use of this solution requires a linear system, which may not be present in an INS / GPS system. Another whole set of filters that overcome the linear requirement of the Kalman filter, and are classified as nonlinear filters. These nonlinear filters include adaptations of the Kalman filter including the unscented Kalman filter and the extended Kalman filters, as well as filters such as particle filters and batch filters. When properly tuned for a system, these filters can achieve a much higher degree of accuracy than the linear Kalman filter. For this project, only the linear Kalman filter presented above is explored.

IV. System Setup and Equipment

A. System Design

To achieve a fused sensor system, the GIN system consists of three major components: the navigation sensors, the data logger, and the navigation computer. A diagram of these components was presented in Figure 1. The development of the project followed a natural progression from the navigation sensors, to the data logging unit, and finally the navigation computer. The data logger interfaces with both sensors using a Universal Asynchronous Receiver/Transmitter (UART) communication setup. UART communication scheme transmits bytes of data in sequential bits, which are then reconstructed at the destination UART chip. The byte nature of

UART allows for easy character transmission between modules. The first block, navigation sensors, contains the IMU and GPS components.

B. Inertial Measurement Unit

The IMU selected for this project is the VectorNav VN-100 attitude and heading reference system (AHRS). The VN-100 contains an IMU nine sensors, including three accelerometers, three gyroscopes, and three magnetometers. Manufacturer specifications for the VN-100 can be found in TABLE I, and an image of the sensor in Figure 4. The VN-100 utilizes a register system for storing measurement messages.

TABLE I
VectorNav VN-100 Sensor Specifications (25 °C)

	Accelerometer	Gyroscope	Magnetometer
Range	$\pm 2g$ X Y, $\pm 6g$ Z	± 500 deg/sec	± 6 Gauss
Bias Stability	0.5mg X Y, 1.6mg Z	< 100 deg/hour	0.125 mGauss
Nonlinearity	$< 0.5\%$	$< 1\%$	$< 1\%$

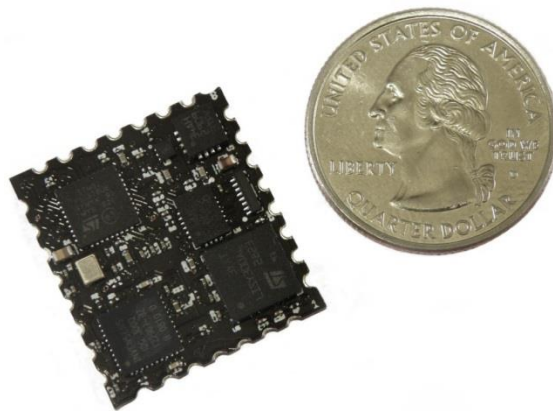


Figure 4: VN-100 Attitude Heading and Reference System

Because this system is an AHRS, a complete navigation solution can be obtained, however only the raw data of the IMU is recorded and utilized. Both unfiltered and filtered data are available from the unit. The message selected for use is the uncompensated earth magnetic field

readings, accelerations, and angular rates. This data can be found in register 252. The VN-100 can be set to either synchronous or asynchronous output mode. In synchronous operation, an external signal must transmit a read request to the VN-100, and the requested register values are returned. For asynchronous operation, the VN-100 is configured to output a specified register at a set frequency. The asynchronous output frequency and register can be changed using a separate control register. A list of asynchronous output frequencies is available in TABLE II. The VN-100 supports either USB or UART communication. UART communication connects the IMU to the data acquisition system, while the USB option provides convenient access to the board for debugging purposes through VectorNav’s software.

TABLE II
Data Transmission Rate Information

Frequency (Hz)	Minimum Baud (bps)	Max SD Write Time (ms)
1	9600	426.0
5	9600	426.0
10	19200	213.0
25	57600	71.1
50	115200	35.5
100	230400	17.7
200	460800	8.88

C. GPS

The GPS unit selected for this project is the uBlox EVK-5T timing GPS, with manufacturer’s specifications in Table III. The EVK-5T provides both USB and RS232 communication, and the uBlox u-Center software allows for quick interfacing for configuring the unit. The NMEA and UBX protocols are the two outputs available from the unit. NMEA is a standard ASCII based GPS protocol, and provides access to information such as the geographic position, satellite information, and dilution measurements. The UBX protocol is a uBlox proprietary format, and is a binary format. The UBX protocol provides access to an extended set of measurements and

information from the GPS, but is not in ASCII format. While the raw GPS measurements may be useful in future applications, the NMEA protocol provides the necessary information and works well with the text logger since it is in ASCII format. The EVK-5T allows for synchronous or asynchronous output, with an output frequency of up to 5 Hz. The selected output registers include the geographic position information, satellite information, and range residuals. The EVK-5T can be seen in FIGURE 5.

TABLE III
uBlox EVK-5T Sensor Specifications

Sample Rate	1-5 Hz
Tracking Sensitivity	-160 dBm (Indoor)
GPS Type	Galileo L1 (1575.42 Mhz)
Time to First Fix	< 1 s
Channels	50
Assisted GPS	SBAS, DGPS, Assist Now
Accuracy	2.5m (2.0m SBAS)



Figure 5: uBlox EVK-5T GPS Development Kit

D. Data Logger Design

Design considerations for the data logger included simultaneous logging of two UART sources (DUART) at two differing output frequencies, text based logging, and low cost. The data logger selected was the SparkFun Logomatic v3 Serial SD Data Logger. The Logomatic board contains an ARM LPC2148, microSD support, 2 UART channels, and up to eight analog logging channels. The Logomatic was selected for its dual UART channel, low cost, and high level of

flexibility. The SparkFun USB boot loader allows for firmware to be quickly changed by placing an updated build file onto the microSD card via USB. Out of the box, the initial Logomatic software provides single channel UART text logging, which saves the files to an SD card, which can be retrieved either via a USB port on the logger, or by removing the SD card. The code for the data logger is available for free from the manufacturer, utilizes open source libraries, written in C, and is easily modifiable. The data logger is shown in Figure 6.



Figure 6: Sparkfun Logomatic V3 Serial Data Logger

Ideally, the frequency of the system is limited by the output frequency of the IMU, requiring a sufficient data rate between the IMU and data logger. The message chosen for the IMU is the uncompensated magnetic, acceleration, and angular rate measurement message. An example of the IMU message is included below in Figure 7.

```
$VNCMV,-5.265888E-01,-7.150189E-01,+1.306738E+00,+6.423429E-02,+1.114974E-01,-
9.771467E+00,-1.795687E-01,+2.197948E-01,+8.457639E-02,+3.183125E+02*4A
```

Figure 7: Sample IMU Message

The message is broken into the following parts: message identification, three magnetometer measurements, three accelerometer measurements, three angular rate measurements, and a message checksum. Each message is 151 characters long, including the checksum and newline

characters. In binary UART communication, each symbol transmitted is a bit, meaning that the baud rate and bit rate are equal. A tabulation of minimum baud rate for a given IMU output frequency of the uncompensated measurement message is below in TABLE IV. The second data transmission consideration is the recording of messages to the SD card. The standard configuration of the data logger writes 512 character blocks at rate of 42.5 ms. The rate at which the SD card can store is dictated by the serial peripheral interface (SPI) clock. Also included in Table II is the maximum time a character array has to be written. The initialization routine for the data logger, as well as the main software loop can be seen in Figure 8 and Figure 9 respectively.

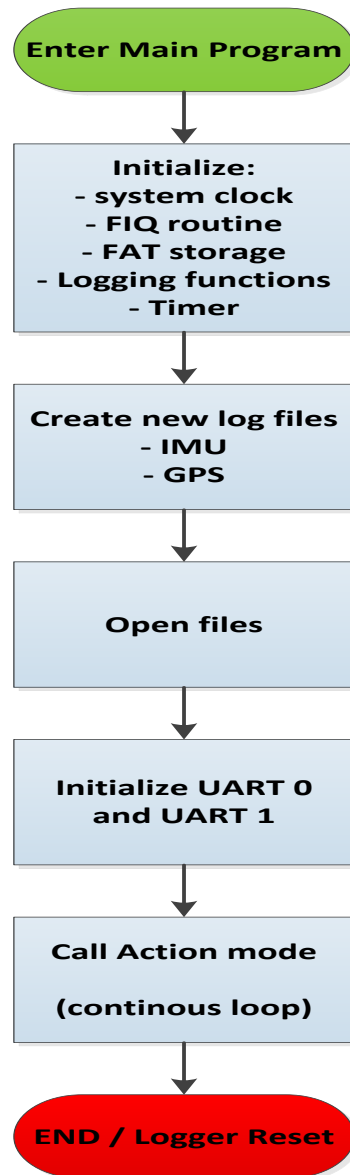


Figure 8: Initialization routine for Logomatic data logger

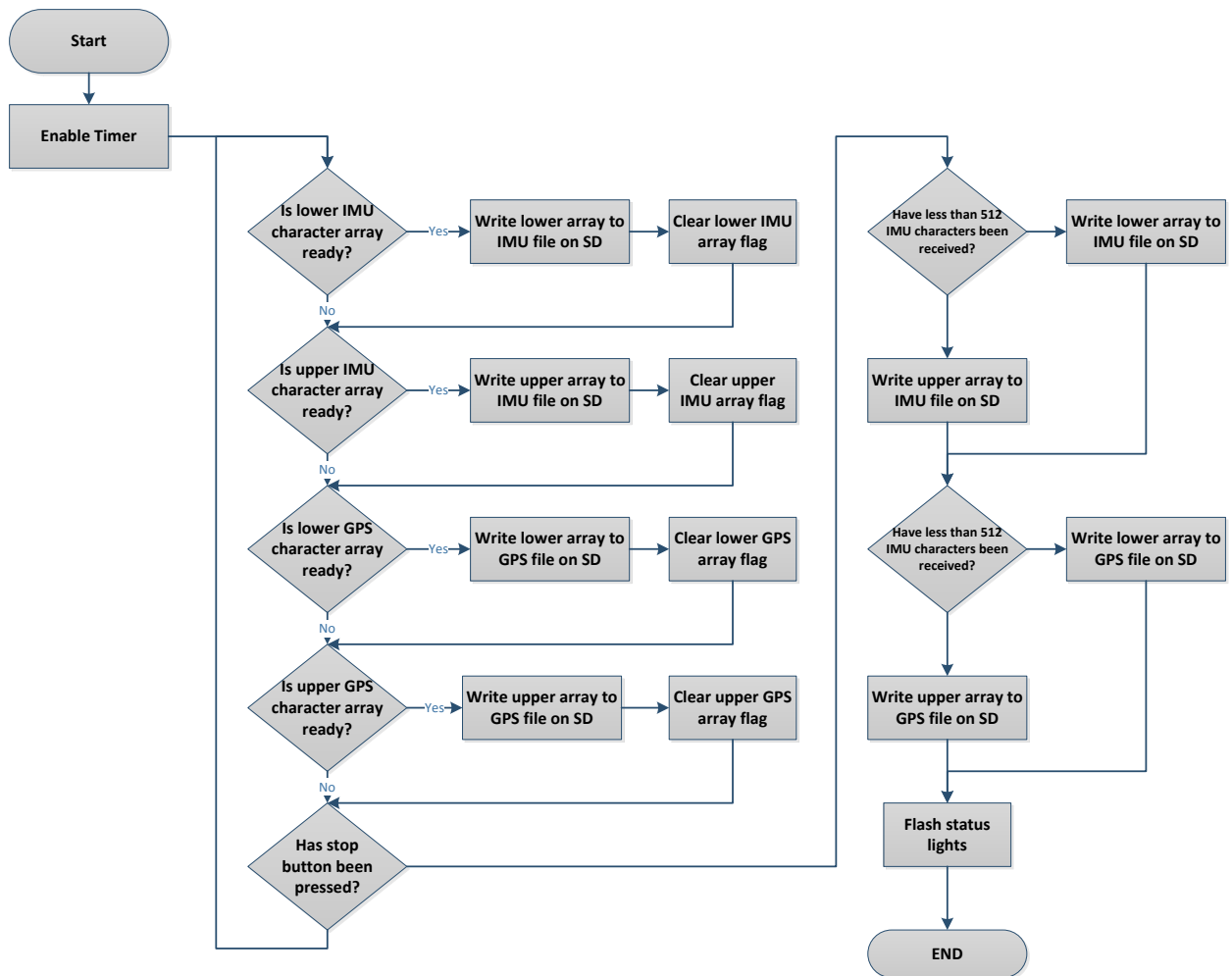


Figure 9: Main software loop of Logomatic data logger

An additional component of the data logger is time stamping the values that are received. Accurate timing data must be maintained in order to later correlate the INS and GPS data in post processing. Additionally, the time interval of the IMU messages are needed for the integrations performed in the INS. On the data logger, incoming characters are handled via two 512 character arrays. The two arrays alternate between storing received characters and writing to the SD card. When a character is received, an interrupt service routine is called by the UART interrupt and vectored to. This routine handles the character and returns it to the array. For time stamping purposes, a timer at a frequency 10 kHz runs in the background of the data logger chip. When a

new line character, '\$', is detected, the timer value is grabbed, converted to a string, and appended to the output array. Because of the 512 character limit on the arrays, overflow must be handled accordingly. The UART interrupt service routine flow char is below in Figure 10.

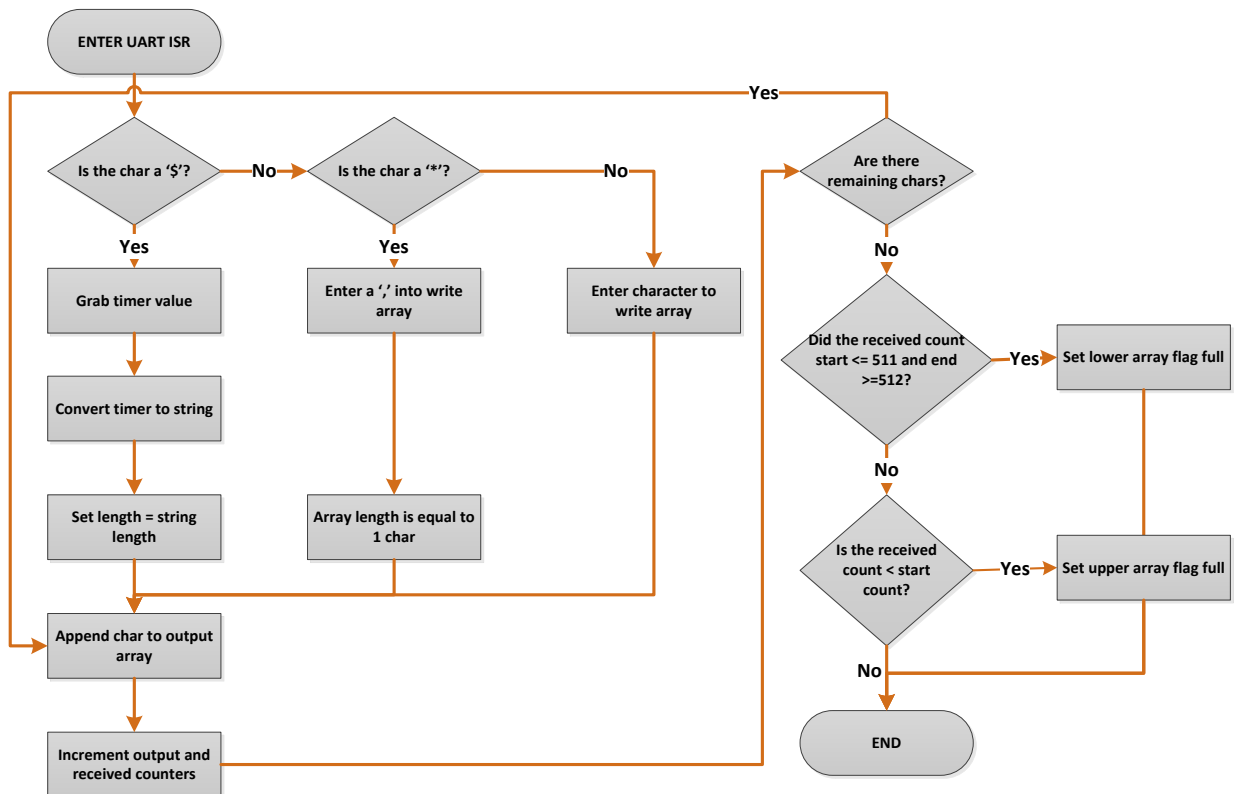


Figure 10: Interrupt service routine for handling UART characters on Logomatic data logger

V. Results

A. Data Logger

The first iterations of the data logger design focused primarily on interfacing solely with the IMU as it is run at much higher frequencies than the GPS. The limiting factor of the system data frequency was the SD write speed of the data logger; at frequencies higher than 50 Hz from the IMU, messages would get dropped during the SD write process. This result was somewhat

expected, as the minimum SD write time of 42.5 ms was already exceeded by 6 ms at 50 Hz, and 24.8 ms at 100 Hz. The SD write speed is dictated by the SPI clock, and altering that frequency impacts the write speed. Raising the frequency by a factor resulted in a write speed of approximately 12.5 ms. This speed was measured by starting a counter before a write operation of a full array, and stopping it once the operation was completed. This marked improvement of about 30 ms beat the 17.7 ms write time needed to run the IMU at 100 Hz. The final data collection system parameters are below in Table 5, with a picture of the system in operation in Figure 11.

TABLE IV
Final System Parameters

	Data Frequency	Baud	Format	Message(s)
IMU	100 Hz	230400	ASCII	Uncompensated earth magnetic field, accelerations, and angular rates
GPS	2 Hz	230400	ASCII	Geographic position, satellites in view, range residuals

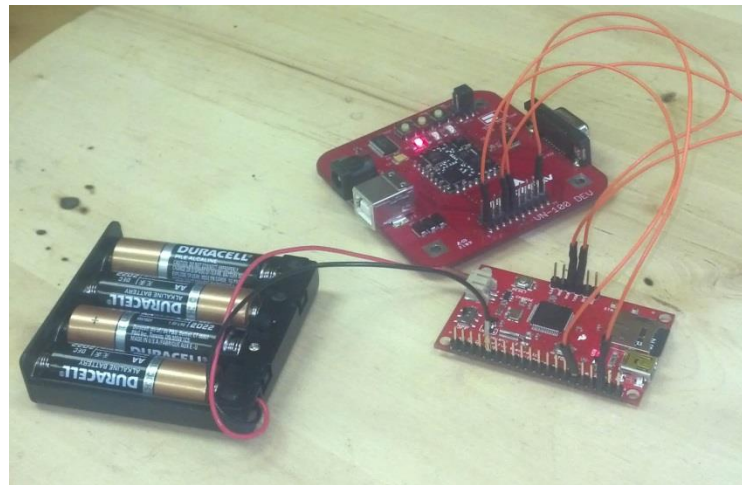


Figure 11: VN-100 IMU connected to the data logger

B. Inertial Navigation System

The raw text files generated by both the IMU and GPS were collected via the USB output of the data logger and fed into MATLAB. A script was written to parse and store all variables output from the sensors. Each line of data had simple formatting occur as characters were received in the data logger to offset each component of the message with a standard delimiter, in this case a comma. Using a delimiter allowed quicker and simpler parsing in MATLAB. The script was set up to expect messages with specific formats and lengths; if a message was found without not following that format, the message was marked as a bad measurement and replaced with an average of the measurement before and after it. Errors on the IMU data sets showed up approximately every 10,000 data points, or 100 seconds. These ‘hiccups’ in the data are attributed to running the data logging and SD writing at higher speeds, as the problem was not replicable at data frequencies lower than 25 Hz. The 1 in 10,000 bad data points was deemed acceptable, and the bad point averaged as mentioned before.

For the first iteration of the strapdown solution, MATLAB code was generated to compute the components of the solution, then combine and integrate the components to yield the Position, Velocity, Acceleration and Jerk (PVAJ) of the body. The iteration of the process can be seen in Figure 12.

The IMU was left to sit and record stationary data for fifteen minutes. This data was used to measure the biases inherent in the accelerometers and gyroscopes. Two data sets were run through the strapdown solution, the raw stationary data, and the same data set with the bias subtracted off. The starting positions, ending positions, and change in positions of both data sets can be seen in Table VI and Figure 13. Simply removing the biases of the sensors resulted in a 10X improvement over a fifteen minute period.

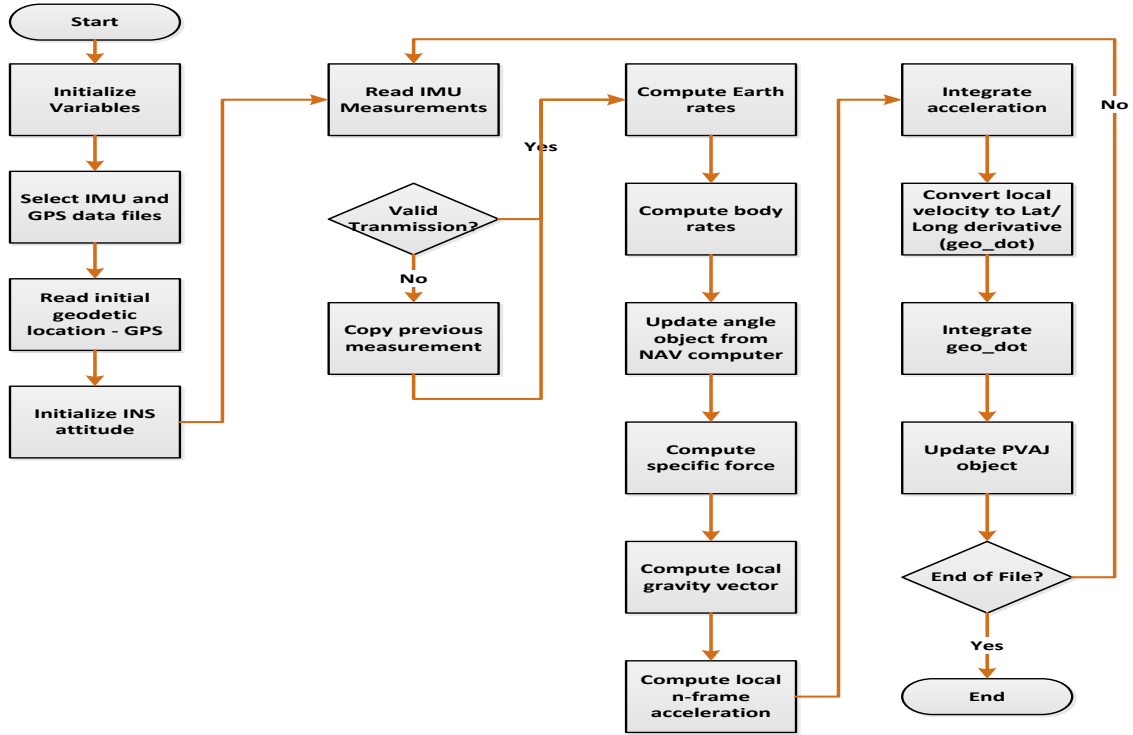


Figure 12: Flow chart of strapdown solution software iterations

TABLE V
Position Drift for 15 Minute Stationary Data Set

	Start Latitude	Start Longitude	End Latitude	End Longitude	Position Change
Biased	40.6955°	-89.6179°	40.5738°	-89.4462	20.37 km
No Bias	40.6955°	-89.6179°	40.7101	-89.6041	2.62 km

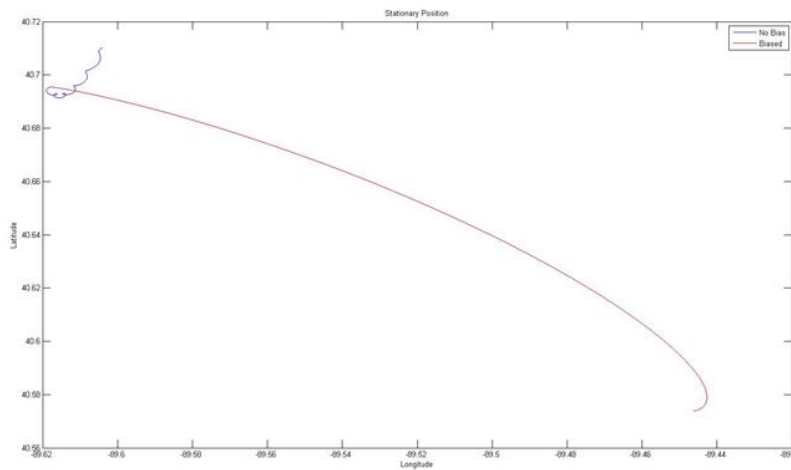


Figure 13: Drift in strapdown solution of 15 minute stationary solution

Despite the major improvement when removing the bias, a 2 km error is still unacceptable for modern navigation. The next step to improve the accuracy of the navigation solution is to integrate the position information measured by the GPS. The Kalman filter was not able to be successfully implemented and unable to be used to incorporate the GPS position updates into the INS position information.

VI. CONCLUSION

A navigation system combining an INS and GPS through Kalman filtering was proposed. The system was composed of three major components, the navigation sensors, data logging unit, and navigation computer. The data logging system has been successfully implemented on a standalone logging unit, and has achieved simultaneous logging of IMU data at 100 Hz and GPS data at 2 Hz. The data logging system has the ability to handle other IMU data rates also. The received data all contain timestamps for aligning data between two different sets of sensors. The SD write speed was found to be a limiting factor of the data acquisition system. A strapdown solution was achieved in MATLAB. Data processed through just the INS component of the navigation computer showed significant drift over a period of fifteen minutes for a stationary body. Subtracting the average sensor biases showed a marked improvement, but still had an error multiple orders of magnitude greater than necessary for modern navigation. The Kalman filter was unable to be successfully implemented, and the GPS data could not be combined with the INS data. Future work on the system would include fixing the Kalman filter, and investigating nonlinear filters that may offer additional improvement over the linear Kalman filter.

VII. Appendix

A. IMU data log sample

1842,VNCMV,-5.274759E-01,-7.151102E-01,+1.306457E+00,+8.982140E-02,+1.515975E-01,-9.889581E+00,-1.795986E-01,+2.166605E-01,+9.084704E-02,+3.184055E+02,45

2042,VNCMV,-5.238971E-01,-7.121478E-01,+1.308644E+00,+1.030171E-01,+1.572612E-01,-9.854329E+00,-1.805358E-01,+2.170784E-01,+9.176565E-02,+3.183986E+02,42

2242,VNCMV,-5.238209E-01,-7.085825E-01,+1.309394E+00,+9.304749E-02,+1.395615E-01,-9.776874E+00,-1.770675E-01,+2.162682E-01,+8.494401E-02,+3.183919E+02,4C

2442,VNCMV,-5.275813E-01,-7.093581E-01,+1.308815E+00,+7.542436E-02,+1.093737E-01,-9.733757E+00,-1.773376E-01,+2.173584E-01,+8.712844E-02,+3.183853E+02,46

2642,VNCMV,-5.279517E-01,-7.131570E-01,+1.308071E+00,+6.642105E-02,+1.116978E-01,-9.784008E+00,-1.799244E-01,+2.181875E-01,+8.413110E-02,+3.183789E+02,49

2842,VNCMV,-5.265755E-01,-7.134659E-01,+1.306714E+00,+7.675630E-02,+1.444039E-01,-9.861773E+00,-1.805489E-01,+2.185816E-01,+8.407921E-02,+3.183725E+02,47

3042,VNCMV,-5.235814E-01,-7.112780E-01,+1.307916E+00,+9.831433E-02,+1.611955E-01,-9.883721E+00,-1.809734E-01,+2.184756E-01,+9.345414E-02,+3.183663E+02,44

B. GPS data log sample

69435\$GPGGA,224822.00,4041.98884,N,08937.03790,W,2,08,1.31,183.3,M,-33.2,M,,0000*60#57234225
70059\$GPGSA,A,3,20,32,48,25,11,12,31,14,,,,,2.06,1.31,1.60*0E#57654378
70685\$GPGSV,4,1,14,01,23,270,,11,08,251,10,12,08,033,36,14,43,078,19*78#57956010
71311\$GPGSV,4,2,14,16,01,186,,20,27,311,26,22,24,151,08,23,02,285,26*70#58309609
71935\$GPGSV,4,3,14,25,33,057,30,30,27,176,,31,82,270,26,32,54,302,31*71#58663249
72560\$GPGSV,4,4,14,48,26,235,32,51,40,206,*75#59016842
73187\$GPGRS,224822.00,1,-0.3,0.3,11.3,5.0,-32.2,1.5,-24.1,-6.0,,,,*72#59235274
73811\$GPVTG,,T,,M,0.138,N,0.255,K,D*2E#116778602
74437\$GPGGA,224823.00,4041.98874,N,08937.03785,W,2,08,0.93,182.9,M,-33.2,M,,0000*68#116960618
75060\$GPGSA,A,3,20,32,48,22,25,12,31,14,,,,,1.74,0.93,1.47*04#117376617
75684\$GPGSV,4,1,14,01,23,270,,11,08,251,,12,08,033,36,14,43,078,17*77#117678250
76310\$GPGSV,4,2,14,16,01,186,,20,27,311,27,22,24,151,11,23,02,285,26*79#118021450
76935\$GPGSV,4,3,14,25,33,057,30,30,27,176,,31,82,270,26,32,54,302,31*71#118375089
77560\$GPGSV,4,4,14,48,26,235,32,51,40,206,*75#118732327
78188\$GPGRS,224823.00,1,-2.0,-2.8,11.5,-13.4,8.9,1.4,-26.2,-6.9,,,,*58#118950733
78812\$GPVTG,,T,,M,0.131,N,0.243,K,D*20#176380204
79438\$GPGGA,224824.00,4041.98867,N,08937.03786,W,2,08,0.93,182.4,M,-33.2,M,,0000*63#176562730
80060\$GPGSA,A,3,20,32,48,22,25,12,31,14,,,,,1.74,0.93,1.47*04#176978729
80685\$GPGSV,4,1,14,01,23,270,,11,08,251,,12,08,033,36,14,43,078,18*78#177285036

REFERENCES

- [1] Grewal, Mohinder S., and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Hoboken, NJ: Wiley, 2008. Print.
- [2] Lin, Ching-Fang. *Modern Navigation, Guidance, and Control Processing*. Englewood Cliffs, NJ: Prentice Hall, 1991. Print.
- [3] D.H. Titterton and J.L. Weston, *Strapdown Inertial Navigation Technology, 2nd Edition*, The Institution of Electrical Engineers, 2004
- [4] Simon, Dan, *Optimal State Estimation*. John Wiley & Sons, Inc., Hoboken, NJ: 2006. Print.
- [5] Brown, Robert Grover and Hwang, Patrick Y.C, *Introduction to Random Signals and Applied Kalman Filtering*. Hoboken, NJ: John Wiley & Sons, 1997. Print.