Hindawi
Journal of Sensors
Volume 2019, Article ID 1345202, 13 pages
https://doi.org/10.1155/2019/1345202

Research Article

Design and Real-Time Implementation of a 3-Stage CnW Heading System on an Ubuntu Linux-Embedded Board

Felipe P. Vista IV and Kil To Chong

Electronic Engineering Department, Chonbuk National University, Jeonju City 54896, Republic of Korea

Correspondence should be addressed to Kil To Chong; kitchong@jbnu.ac.kr

Received 27 December 2018; Accepted 6 February 2019; Published 24 March 2019

Academic Editor: Fanli Meng

This paper describes the design and real-time implementation of a proposed algorithm for deriving an accurate heading system by fusing data from various inexpensive sensor devices that is comparable to more expensive maritime navigation systems. The proposed algorithm is a 3-Stage Classification N' Weighing (CnW) Heading System with forward azimuth (FAz) and extended Kalman filter (EKF). Data from three Global Positioning System devices, an inertial measurement unit, and an electronic compass were fed into the algorithm that can be generally described as Classification N' Weighing-Stage 1 → forward azimuth → Classification N' Weighing-Stage 2 → extended Kalman filter → Classification N' Weighing-Stage 3. The proposed algorithm is shown to be comparably accurate as an expensive marine navigation system, and it has less processing time compared to our previous work. The Qt-anywhere-based system developed on a Linux desktop was successfully downloaded onto an Ubuntu Linux-embedded board for real-time implementation. Important notes related to device naming problems when deploying the system on a Linux-embedded board are also given as reference for those interested to address it.

1. Introduction

Sensor data fusion is integral in deriving better and more accurate data from varied sources such as Global Navigation Satellite System- (GNSS-) Global Positioning System (GPS) [1], cellular handoff [2], electronic nose/electronic tongue [3], webcam [4], radar, smartphone [5], StarGazer [6], and a host of other sensors (compass, inertial measurement, pressure, accelerometer, gyroscope, wheel speed, lateral acceleration, steering wheel angle, tilt, microwave, ultrasonic, vision, altimeter, etc.).

It is applied in various settings such as navigation/localization/positioning for both indoor [7, 8] or outdoor [9–11] as well as in pedestrian dead-reckoning [12, 13], mobile robots [14, 15], and even for research works related to extraterrestrial navigational purposes [16, 17]. Sensor fusion was used in the design of an assistive instrument for paraglider and hang-glider pilots [18]. There are also research works that were focused mainly on deriving a more accurate heading information [1, 19].

In the field of safety and security, there are systems that were designed to detect and track road barriers such as tunnels or guardrails [20], estimating the speed of vehicles on the freeway [2], as well as detection, monitoring, and intelligent alarm related to public security [21].

In the aspect of maritime industry, several systems were designed to assist the ship navigator such as leaving or entering a harbor [22]. There is also the Advanced Sensor Module part of the MUNIN project that dealt with unmanned and autonomous shipping [23] while another group of works dealt with marine collision avoidance [24, 25]. Other maritime navigation-related works focused on obtaining a more accurate heading value such as that of Hu and Huang [26] and Juang and Lin [27] while others concentrated on improving the attitude and position apart from the heading such as those of Feng-de et al. [28], Bryne [29], Jaroś et al. [30], and Núñez et al. [31].

Sensor fusion is utilized in food industry such as in improving the dependability of quality assessment and verification of food and beverages [32]. It has also been utilized for

identifying the blending ratio between old frying oil and the new edible oil to control production cost [3]. It has even been successfully applied in real-time recognition of human action [33] as well as recognizing the activity of daily living (ADL) in helping indicate one's capability for quality living and health status [5].

Various fusion methods have been used in previous works just like ad hoc [34], dead reckoning [35], fuzzy logic [36], Kalman filter [6] and its variations [10], neural networks [2], or numerical discretization [37]. Most of these sensor fusion systems are usually developed for embedded systems just like the ones that used Field-Programmable Gate Arrays (FPGA) with digital signal processors (DSP) [38].

The current work given in this paper describes the design, development, and successful deployment onto an embedded board of a system for deriving an accurate heading value through a 3-Stage Classification and Weighing (3-Stage CnW) algorithm with forward azimuth (FAz) and extended Kalman filter (EKF) by fusing data from multiple inexpensive devices (multiple GPS, an electronic compass, and an IMU). This algorithm is based on our previously proposed work that was checked through postprocessing [39]. The system was developed using Qt-anywhere on a desktop with an Ubuntu Linux system and then deployed onto an Ubuntu Linux-based embedded board using another previous work of ours [40]. The proposed algorithm was tested and validated to generate a more accurate heading value compared to the individual GPS COG values and is even comparable to the more expensive Furuno Satellite Compass (GPS Compass) Model SC-50. This paper is arranged with the proposed 3-Stage Classification and Weighing proposed algorithm in Section 2. The experimental tests composed of the postprocessing and real-time implementation subsections are given in Section 3 followed by the concluding remarks.

2. Proposed 3-Stage Classification and Weighing Sensor Fusion Algorithm

The system design for the proposed sensor fusion system is given in Figure 1. The proposed algorithm is composed of several steps that includes three stages of Classification and Weighing which was inspired by the work of Ercan et al. [34], initial heading calculation for each GPS using forward azimuth [41], and then multiple extended Kalman filters. The simplified algorithm for the whole process is given in Figure 2.

2.1. Classification and Weighing-Stage 1 (CnW-S1). The first stage classifies and assigns weight to the calculated FAz heading for each GPS device. This is done by evaluating the FIX and HDOP (Horizontal Dilution of Precision) values. The GGA NMEA sentence provides crucial fix data in terms of the 3D accuracy and location while the FIX value from the GGA NMEA sentence indicates the quality of the system fix. The FIX value classification based on the description given by DePriest [42] assigns values for corresponding measurements (Code 1).

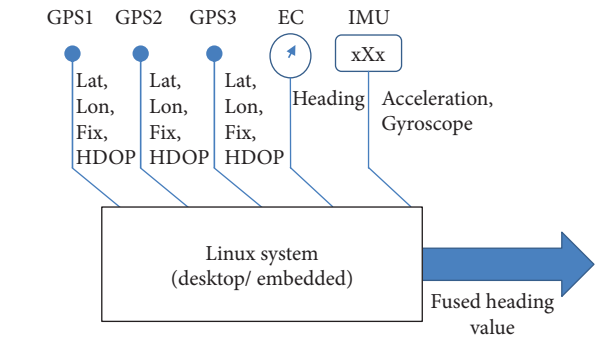


Figure 1: System design.

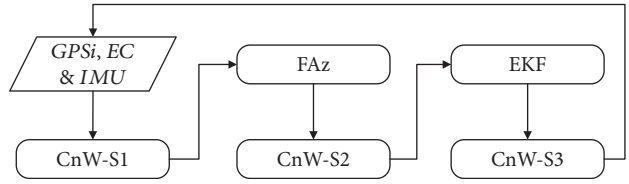


Figure 2: Complete simplified algorithm.

The GSA NMEA sentence provides information on the inherent characteristics of the FIX wherein the HDOP variable of the GSA sentence is available to telemetry users in overseeing control on the quality of the location. Dussault et al. [43] showed that the probability of attaining an inaccurate location value increases when the satellite geometry is poor. The classification for the HDOP values based on Yuen's [44] work is noted in Code 2.

The results of the previous classifications for FIX and the classified HDOP values are then further evaluated using our own classification method. The respective weight values assigned relative to the resulting descriptive classifications will then be utilized as inputs for both Classification and Weighing Stage-2 and Classification and Weighing Stage-3. Hence, the values generated are very integral in deriving the overall heading value for the system. The first classification and weighing stage is given in Code 3.

2.2. Forward Azimuth. The initial calculated heading for each GPS is derived using the forward azimuth (FAz) method by taking the current (lat_{curr}, $\text{long}_{\text{curr}}$) and previous ($\text{lat}_{\text{prev}}$, $\text{long}_{\text{prev}}$) values. The formula for deriving the heading (ψ_{FAz}) through FAz (1) is given as follows:

$$x_1 = \sin\Delta\text{long}^* \cos \text{lat}_{\text{curr}},$$

$$x_2 = \cos \text{lat}_{\text{prev}}{}^* \sin \text{lat}_{\text{curr}} - \sin \text{lat}_{\text{prev}} \cos \text{lat}_{\text{curr}}\Delta\text{long},$$

$$\psi_{\text{FAz}} = \tan^{-1}(x_1, x_2) + 180. \tag{1}$$

2.3. Classification and Weighing-Stage 2 (CnW-S2). The second stage fuses heading data from the GPS devices and the EC using the equation given in (2). The computed FAz GPS

(i) If Invalid then "$FIX_{Gi\_class} = 0$"
(ii) If GPS mode or Standard Positioning Service then "$FIX_{Gi\_class} = 1$"
(iii) If Differential GPS mode (DGPS) then "$FIX_{Gi\_class} = 2$"
(iv) If Precise Positioning System then "$FIX_{Gi\_class} = 3$"
(v) If Real Time Kinematics then "$FIX_{Gi\_class} = 4$"
(vi) If Float Real Time Kinematics then "$FIX_{Gi\_class} = 5$"
(vii) If Estimated Fix or Dead Reckoning then "$FIX_{Gi\_class} = 6$"
(viii) If Manual Input mode then "$FIX_{Gi\_class} = 7$"
(ix) If Simulation mode then "$FIX_{Gi\_class} = 8$"

Code 1

(i) If (0 < HDOP <= 1) then "$HDOP_{Gi\_class}$ = IDEAL"
(ii) If (1 < HDOP <= 2) then "$HDOP_{Gi\_class}$ = EXCELLENT"
(iii) If (2 < HDOP <= 5) then "$HDOP_{Gi\_class}$ = GOOD"
(iv) If (5 < HDOP <= 10) then "$HDOP_{Gi\_class}$ = MODERATE"
(v) If (10 < HDOP <= 20) then "$HDOP_{Gi\_class}$ = FAIR"
(vi) If (20 < HDOP) then "$HDOP_{Gi\_class}$ = POOR"

Code 2

(i) If (2 <= FIX_{Gi_class} <= 5) and (0 < HDOP <= 2) then IDEAL, with weight "3"
(ii) If (FIX_{Gi_class} == 1) and (0 < HDOP <= 2) then EXCELLENT, with weight "2"
(iii) If (2 <= FIX_{Gi_class} <= 5) and (2 < HDOP <= 5) then EXCELLENT, with weight "2"
(iv) If (2 <= FIX_{Gi_class} <= 5) and (5 < HDOP <= 10) then GOOD, with weight "1"
(v) If (FIX_{Gi_class} == 1) and (2 < HDOP <= 5) then GOOD, with weight "1"
(vi) Else BAD, with weight "0"

Code 3

heading and measured EC heading are fused while taking into consideration the derived weights from CnW-S1.

$$\begin{aligned} h_{\text{fused}} &= (h_{\text{allGPS}})(w_{\text{allGPS}}) + (h_{\text{EC}})(w_{\text{EC}}), \\ h_{\text{allGPS}} &= \frac{\sum_1^n (w_i h_i)}{\sum_1^n w_i}, \\ w_{\text{allGPS}} &= \frac{\sum_1^n w_i}{n^* \text{idealValue}}, \\ w_{\text{EC}} &= (1 - w_{\text{allGPS}}), \end{aligned} \tag{2}$$

where n is the total number of GPS devices, idealValue is the IDEAL weight value from CnW-S1, h_{fused} is the fused EC and GPS heading value, h_{allGPS} is the fused calculated heading for all the "n" number of GPS, w_{allGPS} is the weight given to calculated combined GPS heading, h_{EC} is the measured EC heading, w_{EC} is the weight given to EC heading, w_i is the weight given to "i^{th}" GPS, h_i is the calculated FAz heading of "i^{th}" GPS, and ${}^*h_{\text{fused}}$ is labeled as "*GPSEC_yaw*" in the later parts of algorithm and it will also be used in lieu of *GPSn_FAz* during the EKF stage when GPS data is invalid.

2.4. Extended Kalman Filter. The state estimate is derived in the extended Kalman filter through the following steps:

(1) State and error covariance prediction

(2) Kalman gain computation

(3) Time updating of the estimate

(4) Time updating of the error covariance

The controllability and observability of the extended Kalman filter or Kalman filter have been extensively researched and proven in several works such as those by Elizabeth and Jothilakshmi [45], Kamrani et al. [46], and Southall et al. [47]. The dynamic and measurement models which are nonlinear in nature are as follows:

$$\begin{aligned} x_{k+1} &= f(x_k) + w_k, \\ z_k &= h(x_k) + v_k, \end{aligned} \tag{3}$$

where x_k and z_k are the state estimate and measurement. The nonlinearity of the system is the reason why EKF

was utilized for this work. The predicted state $(\widehat{x}_{k+1}^{-})$ and error covariance (P_{k+1}^{-}) can be derived through

$$\begin{aligned}\widehat{x}_{k+1}^{-} &= f(\widehat{x}_{k}^{+}),\\ P_{k+1}^{-} &= F_k P_k F_k^T + Q_k,\end{aligned} \tag{4}$$

where F_k is the Jacobian matrix of the nonlinear dynamic model and P_k is the updated state covariance matrix of previous time step k. Q_k is the process noise, and R_k is the measurement noise covariance matrix which are assumed to be positively definite. The predicted measurement $\widehat{z}_{k+1}^{-}$ is derived as

$$\widehat{z}_{k+1}^{-} = h(\widehat{x}_{k+1}^{-}). \tag{5}$$

The innovation covariance P_{k+1}^{vv} of the residual error between observed and predicted measurements is

$$P_{k+1}^{vv} = P_{k+1}^{yy} + R_{k+1} = H_{k+1} P_{k+1}^{-} H_{k+1}^T + R_{k+1}, \tag{6}$$

where $P_{k+1}^{yy} = H_{k+1} P_{k+1}^{-} H_{k+1}^T$ is the output covariance matrix and H_{k+1} is the Jacobian matrix of the measurement function $h(\widehat{x}_{k+1}^{-})$ that is evaluated about the predicted state $\widehat{x}_{k+1}^{-}$. The Kalman gain is calculated as

$$K_{k+1} = P_{k+1}^{xy} (P_{k+1}^{vv})^{-1} = P_{k+1}^{-} H_{k+1}^T (P_{k+1}^{vv})^{-1}. \tag{7}$$

The state vector x_k of the dynamic model which has the elements latitude, longitude, and heading, $x_k = [x_N, y_E, \psi]^T$, is defined by

$$\begin{aligned}\dot{x} &= \begin{Bmatrix}\dot{x}_N\\ \dot{y}_E\\ \dot{\psi}\end{Bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix} \begin{Bmatrix}u\\ v\\ r\end{Bmatrix} + w\\ &= \begin{bmatrix}u\cos\psi - v\sin\psi\\ u\sin\psi + v\cos\psi\\ r\end{bmatrix} + w = f(x) + w.\end{aligned} \tag{8}$$

On the other hand, the measurement model $h(x_k)$ is described as

$$y_k = \begin{Bmatrix}x_{N,k}\\ y_{E,k}\\ \psi_k\end{Bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} \begin{pmatrix}x_{N,k}\\ y_{E,k}\\ \psi_k\end{pmatrix} = H_k x_k + v_k. \tag{9}$$

The measurements for estimating the state are taken from three GPS devices and one EC. The covariance for process noise (Q) and measurement noise (R) was set dependent on the measured values as

$$\begin{aligned}Q &= \begin{bmatrix}0.01 & 0 & 0\\ 0 & 0.01 & 0\\ 0 & 0 & 10\end{bmatrix},\\ R_k &= \begin{bmatrix}0.3 & 0 & 0\\ 0 & 0.07 & 0\\ 0 & 0 & 0.03\end{bmatrix}.\end{aligned} \tag{10}$$

2.5. Classification and Weighing-Stage 3 (CnW-S3). The third stage was developed to address the difficulty in deploying to an embedded board the previously proposed algorithm that utilized the Covariance Intersection algorithm [39, 40]. This stage fused the heading values from the multiple EKF processes with their corresponding weight values from the first stage (CnW-S1). The equation for this step is given as follows:

$$h_{\text{CnW-S3}} = \frac{\sum_1^n w_i h_{\text{EKFi}}}{\sum_1^n w_i}, \tag{11}$$

where n is the total number of EKF's; w_i is the weight for GPSi from CnW-S1; h_{EKFi} is the heading value from EKF number i; $h_{\text{CnW-S3}}$ is the overall fused heading value.

The graphical representations of these steps are given in Figures 3 and 4.

3. Experimental Tests

3.1. Postprocessing Simulation. The proposed algorithm was initially tested through postprocessing via MATLAB using the device setup given in Figure 5 to capture the data. There were two sets of data captured two days apart using three inexpensive GPS, an electric compass, an IMU, and Furuno Satellite Compass (GPS Compass) Model SC-50 which was for comparative performance. The SC-50 has a heading accuracy of 0.5 degrees RMS. The reference heading is 18.01 degrees based on Google Earth (Figure 6).

The results of the experimental tests via postprocessing are given in Figure 7 with the left column showing the derived heading value while the heading error is given in the right column, as compared to EKF-CI, GPS#1 COG, GPS#2 COG, GPS#3 COG, and EC. The 1st and 2nd data sets have artificially introduced data blackouts for all the three GPS devices hence the very noticeable increase in their respective COG RMSEs. The RMSE for electronic compass on the other hand is a result of the actual measured value during the second data capture run. Additional plots of postprocessing results without missing GPS data are given in Figure 8.

The RMSEs for both data sets with complete GPS data and with missing GPS data are given in Tables 1 and 2, respectively. It can be seen that the proposed 3-Stage CnW performed better compared to EKF-CI for both situations. The postprocessing time for both methods with the same set of data is given in Table 3. It shows that the proposed 3-Stage CnW compared with EKF-CI has a factor of

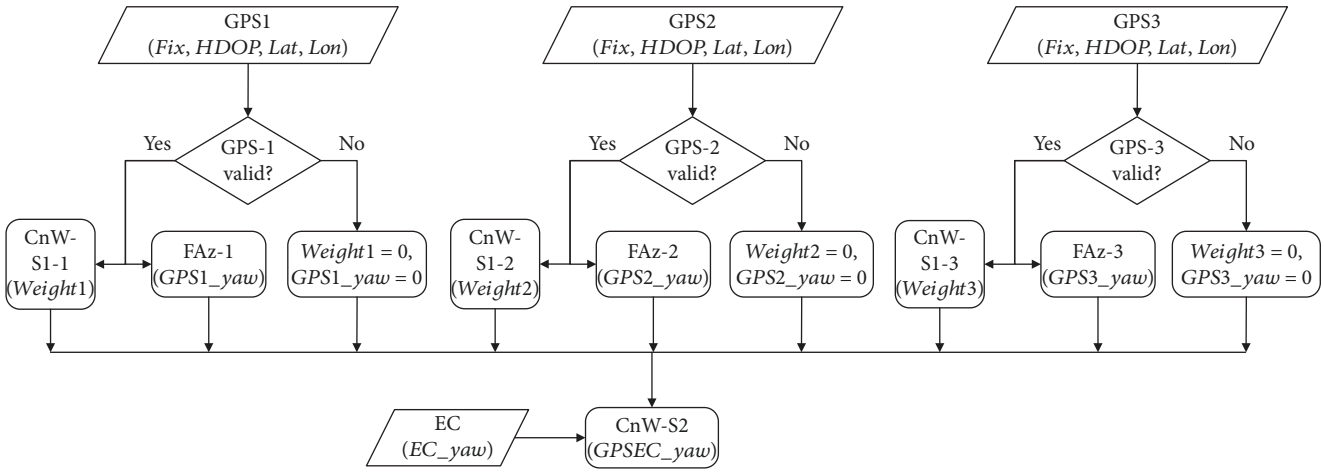


Figure 3: CnW-S1, FAz, and CnW-S2.

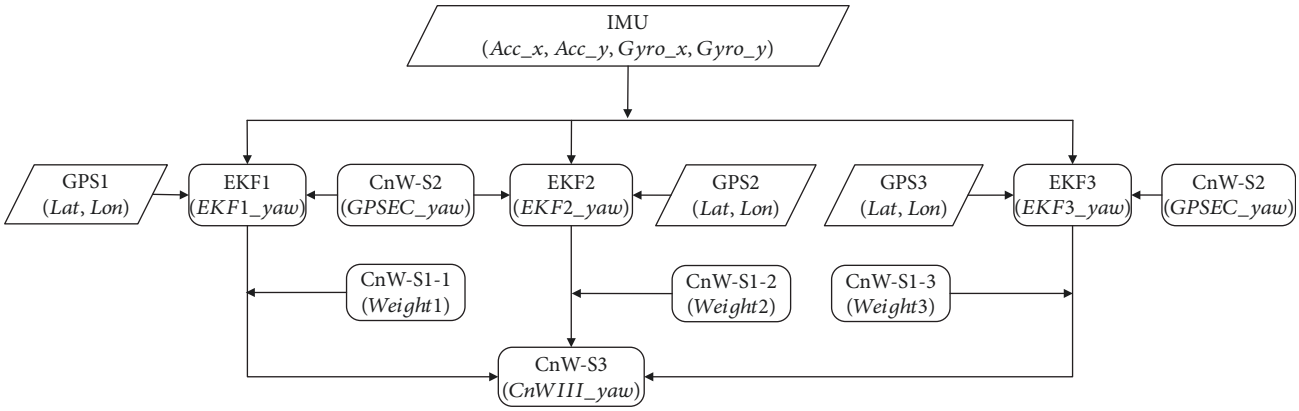


Figure 4: EKF and CnW-S3.

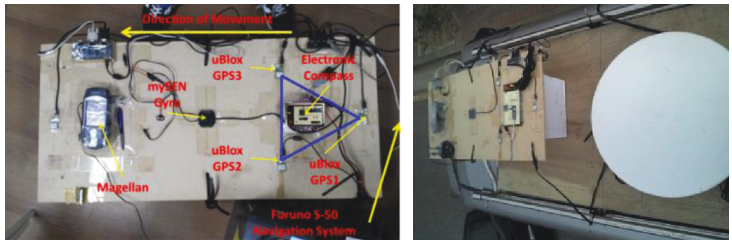


Figure 5: Data gathering setup for postprocessing.

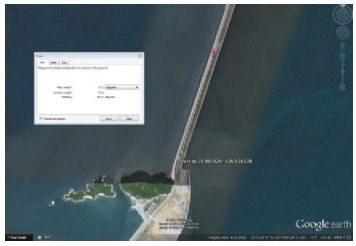
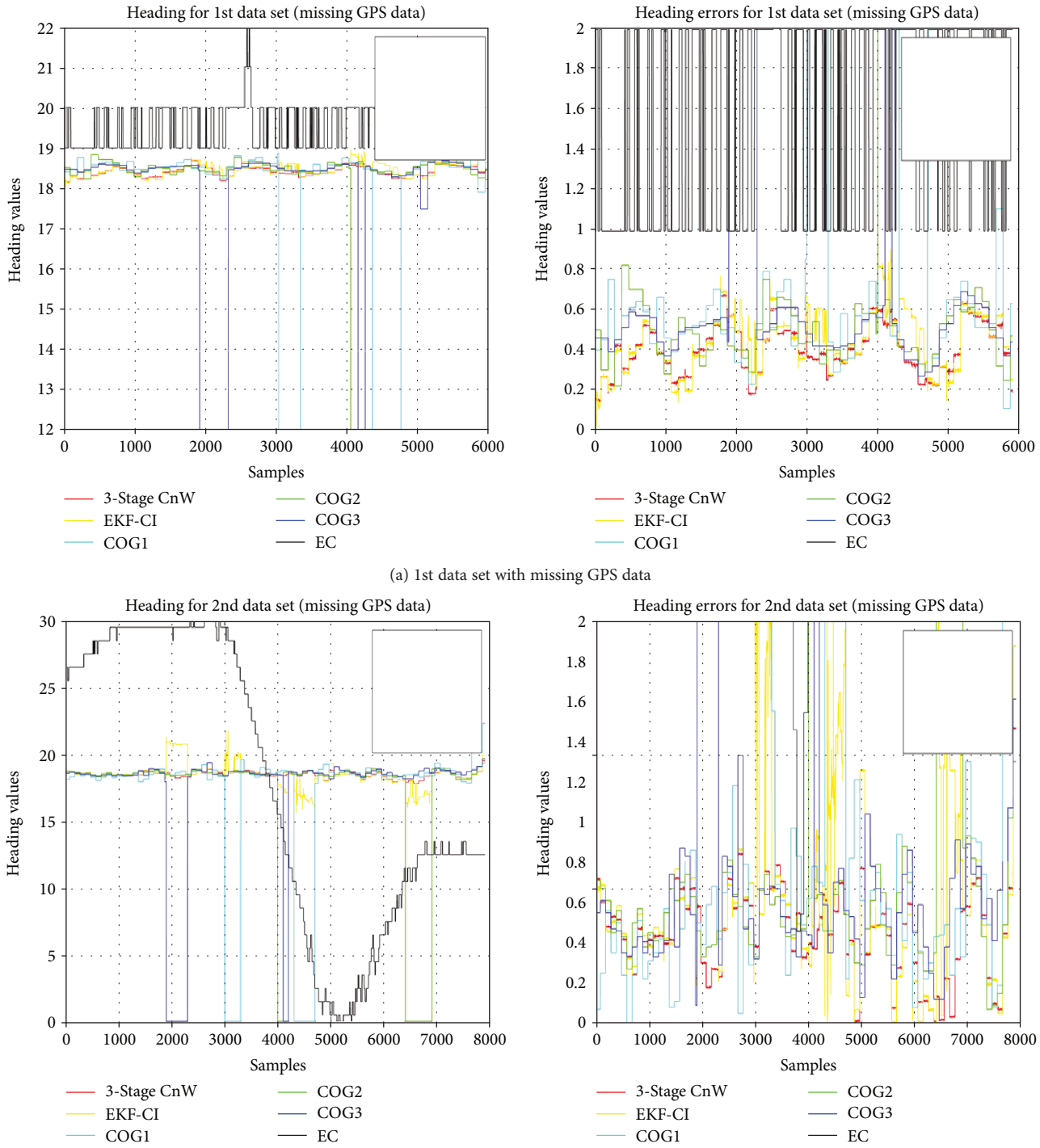


Figure 6: Reference heading via Google Earth of 18.01 degrees.

approximately 1 is to 27 in terms of processing time. This reduction in processing time opens the opportunity for more real-time application of the proposed heading system aside from marine vehicles such as land vehicles or even aerial vehicles.

The decrease in processing time is directly due to the lesser complexity of the proposed algorithm compared to the previous algorithm especially the use of the CnW-Stage 3 (13) instead of the Covariance Intersection (CI) [48] algorithm which is more complicated in itself and has to be performed thrice. The first CI implementation processes the EKF1_output and EKF2_output, and the second CI works on



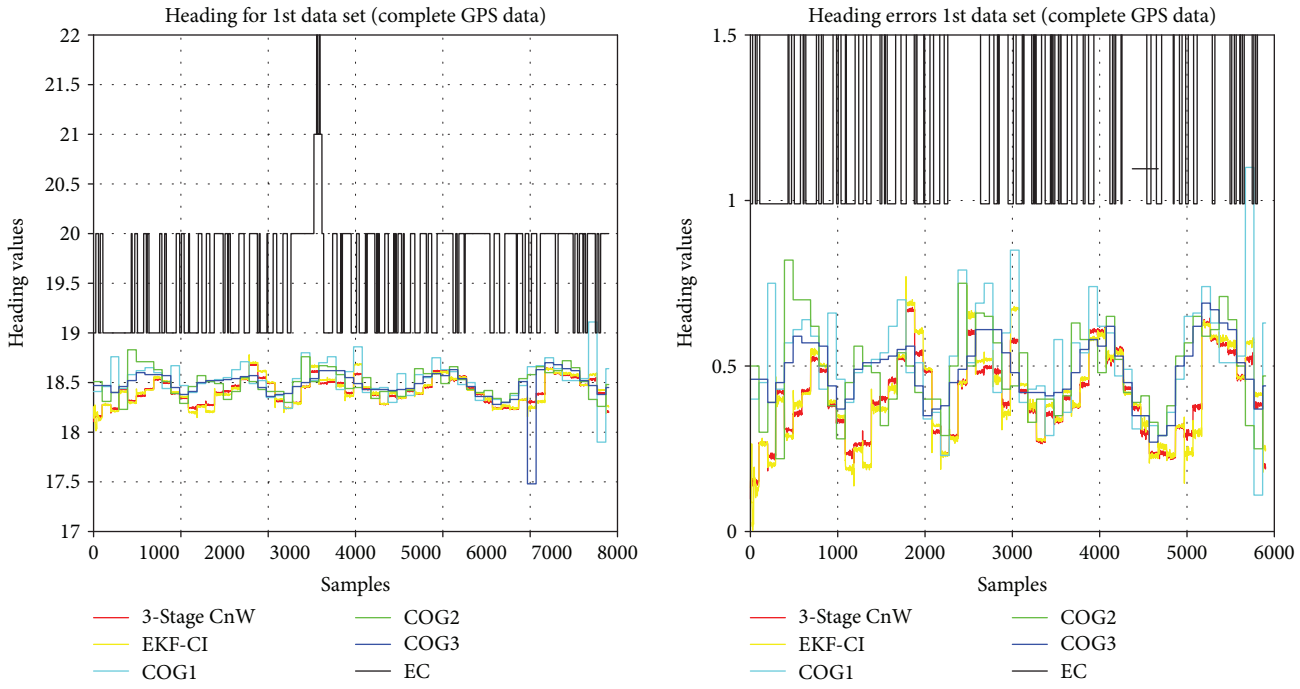
(a) 1st data set with missing GPS data

(b) 2nd data set with missing GPS data

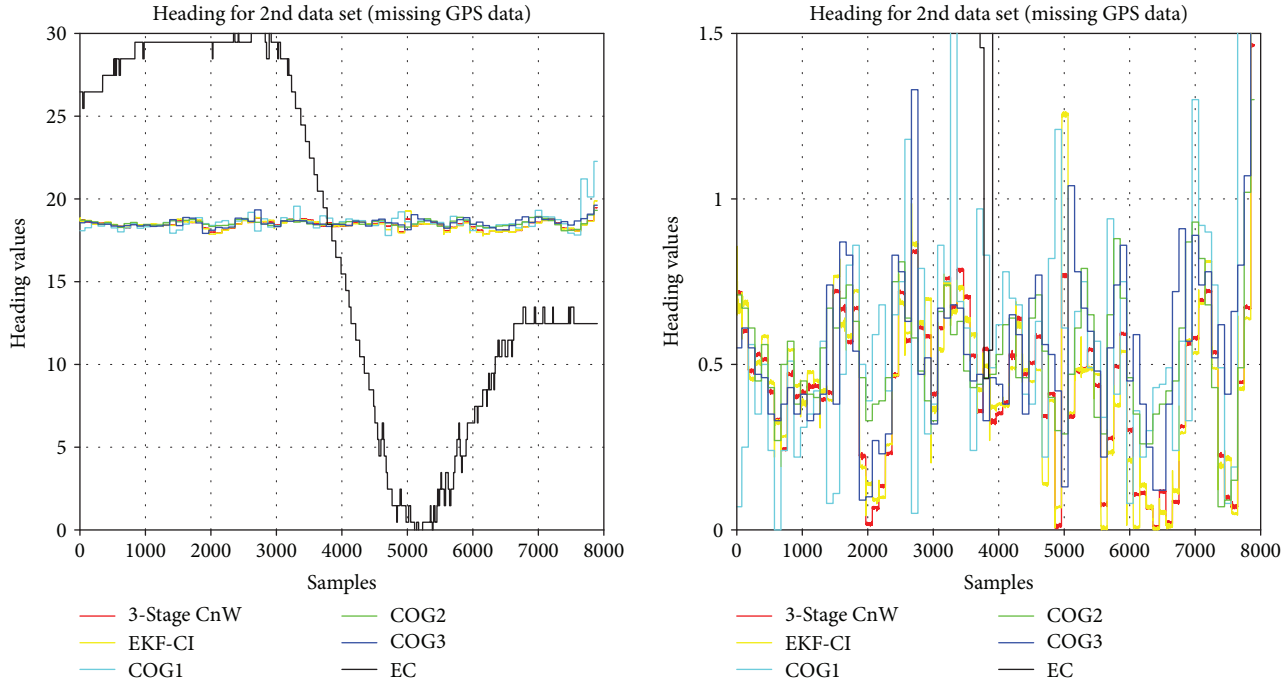
Figure 7: 1st and 2nd data set postprocessing results.

the EKF2_output and EKF3_output, while the third CI implementation utilizes the outputs of CI_1 and CI_2 as its input. There is no perceived trade-off since the derived heading accuracy also improved with the decrease in processing time.

3.2. Real-Time Implementation. The proposed algorithm was initially developed using Qt-everywhere on a desktop running Ubuntu Linux. The TinyEKF library of Simon D. Levy which can be accessed at GitHub [49] was utilized for the EKF aspect of the algorithm when deployed on an embedded board. It was chosen due to it being a C/C++ implementation of EKF that can be deployed on Arduinos, STM32, and other embedded systems. The devices in Figure 9 were utilized during the real-time implementation. The program was then uploaded to the



(a) 1st data set, no missing GPS data



(b) 2nd data set, no missing GPS data

Figure 8: 1st and 2nd data set postprocessing results, with no missing GPS data.

Beagle Bone Black (BBB) board which has the LCD cape attached for display purposes.

3.2.1. Persistent Naming. Persistent naming was implemented in the Linux system in order to address the problem of a device getting a new assigned name when disconnecting and connecting. This is very important in deploying the system especially when developing the system using a desktop and then deploying it onto an embedded board. We will be presenting a more detailed account of its implementation in this section. The connected device naming convention for the UBUNTU Linux system that is being currently utilized is "/dev/ttyACM0" for GPS#1, "/dev/ttyACM1" for GPS#2, "/dev/ttyACM2" for GPS#3, "/dev/ttyACM3" for EC (Arduino), and "/dev/ttyUSB0" for IMU. The device information can be retrieved through the following commands

Table 1: RMSE with complete GPS data.

	1st data set	2nd data set
GPS#1/#2/#3 (COG)	0.5483, 0.5037, 0.4966	0.8183, 0.5616, 0.5997
Average of GPS#1/#2/#3 (COG)	0.5052	0.6211
Electronic compass	1.6553	10.5455
EKF-CI	0.4330	0.5122
3-Stage CnW	0.4262	0.4946

Table 2: RMSE with the all three GPS missing some data.

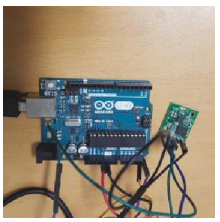
	1st data set	2nd data set
GPS#1/#2/#3 (COG)	6.2169, 2.3857, 5.2544	5.4118, 4.9847, 4.5608
Average of GPS#1/#2/#3 (COG)	2.7220	2.7520
Electronic compass	1.6553	10.5455
EKF-CI	0.4633	0.9467
3-Stage CnW	0.4222	0.5012

Table 3: Comparative postprocessing time.

	Complete GPS data		With missing GPS data	
	1st set (6001 data set)	2nd set (8001 data set)	1st set (6001 data set)	2nd set (8001 data set)
EKF-CI	27.082907 secs	36.388161 secs	28.040673 secs	36.065577 secs
3-Stage CnW	0.990270 secs	1.268189 secs	1.005248 secs	1.264782 secs
FACTOR	27	28	27	28

(a) GPS

(b) IMU



(c) Arduino with EC

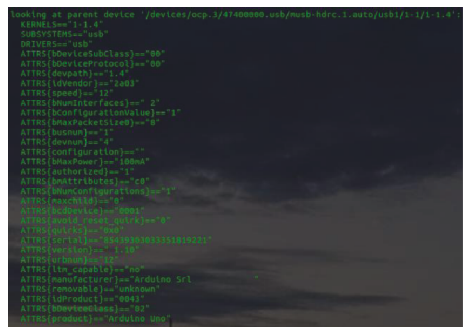
(d) BBB

(e) LCD cape

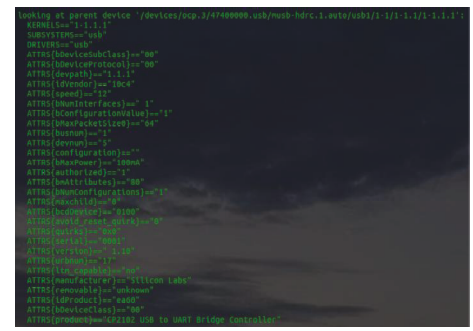
Figure 9: Devices utilized for real-time development.

```
For EC (Arduino) :   "udevadm info –a –n /dev/ttyACM0"
For IMU          :   "udevadm info –a –n /dev/ttyUSB0"
For GPS#1        :   "udevadm info –a –n /dev/ttyACM1"
For GPS#2        :   "udevadm info –a –n /dev/ttyACM2"
For GPS#3        :   "udevadm info –a –n /dev/ttyACM3"
```

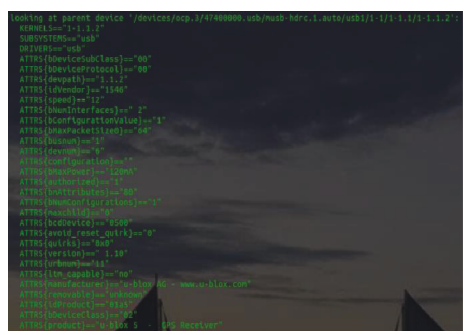
Code 4



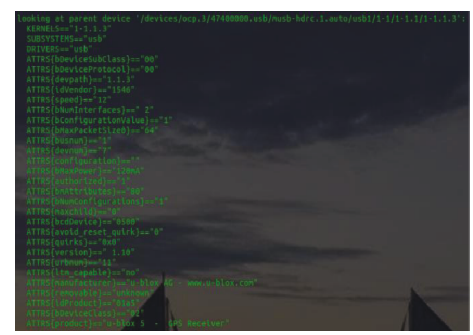
(a) EC (Arduino Uno) (/dev/ttyACM0)



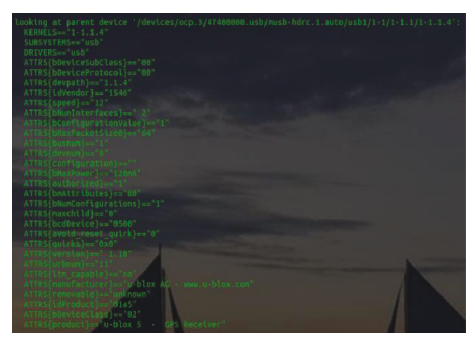
(b) IMU (/dev/ttyUSB0)



(c) GPS #1 (/dev/ttyACM1)



(d) GPS #2 (/dev/ttyACM2)



(e) GPS #3 (/dev/ttyACM3)

Figure 10: Results for "udevadm info –a –n/dev/ttyXXX#" (Ubuntu Linux on BBB).

```
For IMU      :   SUBSYSTEM, ATTRS{idVendor}, ATTRS{idProduct},
                 ATTRS{serial}
For EC via Arduino, GPS#1, GPS#2, and GPS#3: SUBSYSTEM, KERNELS, ATTRS{idVendor},
                 ATTRS{idProduct}, ATTRS{Product}, ATTRS{serial}
```

Code 5

```
For EC (Arduino)  :   KERNELS==“1-1.4”
For IMU           :   KERNELS==“1-1.1.1”
For GPS #1        :   KERNELS==“1-1.1.2”
For GPS #2        :   KERNELS==“1-1.1.3”
For GPS #3        :   KERNELS==“1-1.1.4”
```

Code 6

```
For EC (Arduino)  :   SYMLINK+=“fjEC”
For IMU           :   SYMLINK+=“fjIMU”
For GPS #1        :   SYMLINK+=“fjGPS1”
For GPS #2        :   SYMLINK+=“fjGPS2”
For GPS #3        :   SYMLINK+=“fjGPS3”
```

Code 7

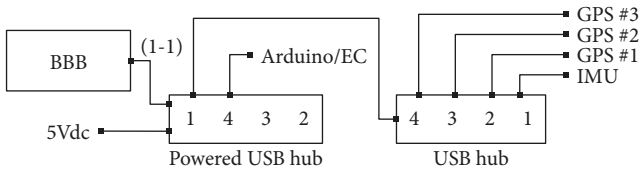


Figure 11: USB hub mesh.

Figure 12: Implemented `/etc/udev/rules.d/99-usb-serial.rules` file (Ubuntu Linux on BBB).

```
For EC (Arduino)  :
serialPort->setPortName(“/dev/ttyACM0”)to
serialPort->setPortName(“/dev/fjEC”)
For IMU           :
serialPort->setPortName(“/dev/ttyUSB0”)to
serialPort->setPortName(“/dev/fjIMU”)
For GPS #1        :
serialPort->setPortName(“/dev/ttyACM1”)to
serialPort->setPortName(“/dev/fjGPS1”)
For GPS #2        :
serialPort->setPortName(“/dev/ttyACM2”)to
serialPort->setPortName(“/dev/fjGPS2”)
For GPS #3        :
serialPort->setPortName(“/dev/ttyACM3”)to
serialPort->setPortName(“/dev/fjGPS3”)
```

Code 8

executed at a terminal command line in the Ubuntu Linux setup on the BBB embedded board.

Screen captures of significant portions of the output of the command can be seen in Figure 10. The implemented rules file has different configurations for the GPS, EC, and IMU. Given below is the pertinent information needed when running the `udevadm` command on each of the device that is needed for the persistent naming to work in our system.

The four items of information for IMU are enough to uniquely identify it while the EC and GPS need the additional information of `ATTRS{Product}` and especially `KERNELS` to help us identify the device when writing the rules file. It must be noted that the exact information including the number of spaces given by the `udevadm info a n` command must be used, such as `ATTRS{product}==`“Arduino Uno” for the EC/Arduino while `ATTRS{product}==`“u-blox 5 - GPS Receiver” for the GPS. In terms of having the same devices on the system such as three GPS devices, this is where the `KERNELS` information is necessary to uniquely identify each of them. This is in order to avoid the system from mistakenly polling the same GPS device as another GPS when there are actually three GPS devices. The KERNELS information for EC/Arduino and multiple GPS devices are given as follows:

The persistent naming is affected though by the utilization of multiple USB hubs in order to accommodate the several sensor devices that was used in the real-time implementation of the sensor fusion system onto an embedded board. The BBB board that was utilized has only one USB port hence the need for USB hubs with the first one being a powered hub due to the power requirement of the various sensor devices that cannot be sufficiently supplied by the BBB board itself. The ports where the devices are connected matter when it comes to setting up the persistent names through a rules file in the Linux system. This information can be fully understood through the help of the diagram given in Figure 11. Taking for example the `KERNELS` information for GPS #1, the first set of numbers (“`1-1.`”) refers to the USB port of the BBB itself. The next number (“**.1.**”) refers to the port number of the powered USB hub that the second USB hub is connected to. The last number (“**.2.**”) refers to the port number of the second USB hub that GPS #1 is connected. The same logic is given for all the other devices, but it is worth noting that EC via Arduino only has (“1-1.4”) since it is directly connected to the fourth port of the powered hub itself.

With respect to the rules file, another thing that is necessary but must be supplied by the developer is the symbolic link (SYMLINK) information which will be used in the sensor fusion system to refer to the specified device. The developer has the freedom to assign any name preferred. A screen capture of the implemented `.rules` file is given in Figure 12 wherein the following symbolic links were utilized:

The proper implementation of persistent naming would then enable the fusion system to continue to operate properly even if the device(s) get(s) disconnected and then reconnected again. Given below is the update done in the Qt program code for the real-time implementation of the system.

3.2.2. Multithreading. The multithreading feature of the embedded board and the installed Ubuntu system were exploited since there was a need for the simultaneous polling of data from the various sensors as well as the individual processes of the sensor fusion algorithm. The current

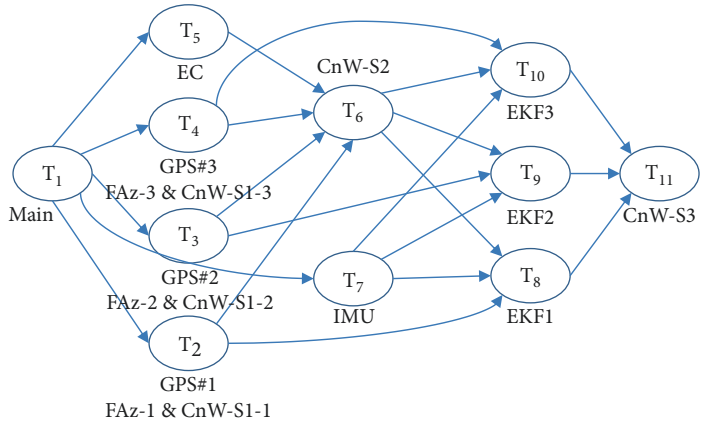
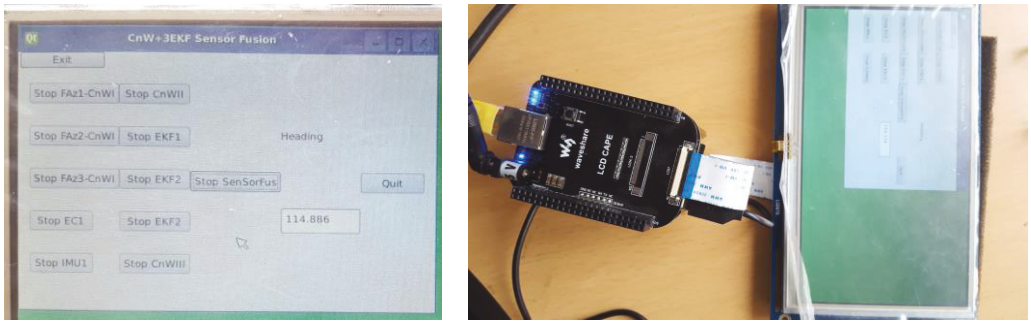


Figure 13: Implementation threads.

Thread 1 - main thread for sensor fusion system;
Thread 2 - poll data, execute CnW-S1 and then FAz on GPS#1;
Thread 3 - poll data, execute CnW-S1 and then FAz on GPS#2;
Thread 4 - poll data, execute CnW-S1 and then FAz on GPS#3;
Thread 5 - poll data from EC;
Thread 6 - execute CnW-S2;
Thread 7 - poll data from IMU;
Thread 8 - execute EKF on data from GPS#1 and IMU;
Thread 9 - execute EKF on data from GPS#2 and IMU;
Thread 10 - execute EKF on data from GPS#3 and IMU;
Thread 11 - execute CnW-S3 on data from EKF#1, EKF#2, and EKF#3

Code 9



(a) (b)

Figure 14: System on BBB.

implementation has a total of eleven (11) threads that are running simultaneously. The multithreading design and their interconnection are graphically represented in Figure 13. The thread and the specific operation they execute are given as follows:

The resulting real-time implementation of the 3-Stage CnW that was programmed on an Ubuntu desktop and then deployed onto a BBB with LCD cape is shown in Figure 14. The test program has several buttons to individually start/stop the subprocesses related to each of the threads (Threads # 2-11) as well as a single button to start/stop the complete sensor fusion algorithm. The fusion system was able to operate as desired even when the sensors got disconnected and then reconnected as well as when the ports they were connected to were changed. This was successfully addressed by the use of persistent naming that would have otherwise broken the system.

4. Conclusion

Presented in this paper is the design, development, and real-time deployment of the 3-Stage Classification and

Weighing algorithm with the forward azimuth (FAz) and extended Kalman filter (EKF) that generates an accurate heading value. The resulting heading is comparable to more expensive navigation systems such as the Furuno Satellite Compass (GPS Compass) Model SC-50 by fusing data from multiple inexpensive sensors. The current algorithm is based on our previous work [39] that has been further improved.

The accuracy of the proposed algorithm is tested and validated as well as shown to have faster processing time compared to our previously proposed algorithm. The algorithm was initially tested through MATLAB and then was programmed on an Ubuntu-Linux desktop using Qt-anywhere. The multithreading capability of the desktop and target BBB-embedded board was utilized in order to simultaneously implement the fusion processes as well as polling data from the individual sensor devices. The complete sensor fusion was successfully deployed on an embedded board following the general steps given in our previous work [40] with additional focus on persistent naming to address the problem of device name reassignment in the Linux system.

Data Availability

The GPS, IMU, and EC data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the Brain Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, & Future Planning (Grant no. NRF-2017M3C7A1044815). This was also supported by the "Research Base Construction Fund Support Program" funded by Chonbuk National University in 2018.

References

[1] Z. Dai, R. Ziebold, A. Born, and E. Engler, "Heading-determination using the sensor-fusion based maritime PNT unit," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pp. 799–807, South Carolina, USA, 2012.

[2] S. He, J. Zhang, Y. Cheng, X. Wan, and B. Ran, "Freeway multisensor data fusion approach integrating data from cellphone probes and fixed sensors," *Journal of Sensors*, vol. 2016, Article ID 7269382, 13 pages, 2016.

[3] H. Men, D. Chen, X. Zhang, J. Liu, and K. Ning, "Data fusion of electronic nose and electronic tongue for detection of mixed edible-oil," *Journal of Sensors*, vol. 2014, Article ID 840685, 7 pages, 2014.

[4] J. Li, J. A. Besada, A. M. Bernardos, P. Tarrío, and J. R. Casar, "A novel system for object pose estimation using fused vision and inertial data," *Information Fusion*, vol. 33, pp. 15–28, 2017.

[5] J. Wu, Y. Feng, and P. Sun, "Sensor fusion for recognition of activities of daily living," *Sensors*, vol. 18, no. 11, 2018.

[6] S. W. Yoon, S.-B. Park, and J. S. Kim, "Kalman filter sensor fusion for Mecanum wheeled automated guided vehicle localization," *Journal of Sensors*, vol. 2015, Article ID 347379, 7 pages, 2015.

[7] Q. Li, D. C. Li, Q. F. Wu et al., "Autonomous navigation and environment modeling for MAVs in 3-D enclosed industrial environments," *Computers in Industry*, vol. 64, no. 9, pp. 1161–1177, 2013.

[8] M. C. P. Santos, L. V. Santana, A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "Indoor low-cost localization system for controlling aerial robots," *Control Engineering Practice*, vol. 61, pp. 93–111, 2017.

[9] M. Barisic, N. Miskovic, and A. Vasilijevic, "Fusing hydroacoustic absolute position fixes with AUV on-board dead reckoning," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 211–217, 2012.

[10] C. Melendez-Pastor, R. Ruiz-Gonzalez, and J. Gomez-Gil, "A data fusion system of GNSS data and on-vehicle sensors data for improving car positioning precision in urban environments," *Expert Systems with Applications*, vol. 80, pp. 28–38, 2017.

[11] J. H. Lim, W. Yoo, L. Kim, Y. Lee, and H. Lee, "Augmentation of GNSS by low-cost MEMS IMU, OBD-II, and digital altimeter for improved positioning in urban area," *Sensors*, vol. 18, no. 11, 2018.

[12] S. Qiu, Z. Wang, H. Zhao, K. Qin, Z. Li, and H. Hu, "Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion," *Information Fusion*, vol. 39, pp. 108–119, 2018.

[13] Widyawan, G. Pirkl, D. Munaretto et al., "Virtual lifeline: multimodal sensor data fusion for robust navigation in unknown environments," *Pervasive and Mobile Computing*, vol. 8, no. 3, pp. 388–401, 2012.

[14] M. D. N. Forte, W. B. Correia, F. G. Nogueira, and B. C. Torrico, "Reference tracking of a nonholonomic mobile robot using sensor fusion techniques and linear control," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 364–369, 2018.

[15] H. Lee and S. Jung, "Balancing and navigation control of a mobile inverted pendulum robot using sensor fusion of low cost sensors," *Mechatronics*, vol. 22, no. 1, pp. 95–105, 2012.

[16] J. Kowalski, P. Linder, S. Zierke et al., "Navigation technology for exploration of glacier ice with maneuverable melting probes," *Cold Regions Science and Technology*, vol. 123, pp. 53–70, 2016.

[17] J. Delaune, G. Le Besnerais, T. Voirin, J. L. Farges, and C. Bourdarias, "Visual–inertial navigation for pinpoint planetary landing using scale-based landmark matching," *Robotics and Autonomous Systems*, vol. 78, pp. 63–82, 2016.

[18] B. Brzozowski, Z. Rochala, K. Wojtowicz, and P. Wieczorek, "Measurement data fusion with cascaded Kalman and complementary filter in the flight parameter indicator for hang-glider and paraglider," *Measurement*, vol. 123, pp. 94–101, 2018.

[19] F. Abyarjoo, A. Barreto, J. Cofino, and F. R. Ortega, "Implementing a sensor fusion algorithm for 3D orientation detection with inertial/magnetic sensors," in *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, Lecture Notes in Electrical Engineering, pp. 305–310, Springer, 2015.

[20] T. Kim and B. Song, "Detection and tracking of road barrier based on radar and vision sensor fusion," *Journal of Sensors*, vol. 2016, Article ID 1963450, 8 pages, 2016.

[21] J. Feng, W. Zhou, and K. Sun, "Multimedia fusion for public security in heterogeneous sensor networks," *Journal of Sensors*, vol. 2014, Article ID 273210, 12 pages, 2014.

[22] A. Trzuskowsky, C. Hoelper, and D. Abel, "ANCHOR: navigation, routing and collision warning during operations in harbors," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 220–225, 2016.

[23] W. C. Bruhn, H.-C. Burmeister, M. T. Long, and J. A. Moræus, "Conducting look-out on an unmanned vessel: Introduction to the advanced sensor module for MUNIN's autonomous dry bulk carrier," in *Proceedings of International Symposium Information on Ships—ISIS 2014*, pp. 141–154, Hamburg, Germany, 2014.

[24] A. L. Flåten and E. F. Brekke, "Performance prediction of tracking sensors for surface vehicle collision avoidance," in *2016 19th International Conference on Information Fusion (FUSION)*, pp. 1894–1900, Heidelberg, Germany, 2016.

[25] D. Chen, C. Dai, X. Wan, and J. Mou, "A research on AIS-based embedded system for ship collision avoidance," in *2015 International Conference on Transportation Information and Safety (ICTIS)*, pp. 512–517, Wuhan, China, 2015.

[26] P. Hu and C. Huang, "Shipborne high-accuracy heading determination method using INS- and GPS-based heading determination system," *GPS Solutions*, vol. 21, no. 3, pp. 1059–1068, 2017.

[27] J. C. Juang and C. F. Lin, "A sensor fusion scheme for the estimation of vehicular speed and heading angle," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 7, pp. 2773–2782, 2015.

[28] Q. Feng-de, W. Feng-wu, L. Jiang, and T. Guan-jun, "A method for the measurement of ship attitude based on multi-sensor data fusion," in *2015 Ninth International Conference on Frontier of Computer Science and Technology*, pp. 196–199, Dalian, China, 2015.

[29] T. H. Bryne, *Nonlinear Observer Design for Aided Inertial Navigation of Ships, [Ph.D. thesis]*, NTNU, Trondheim, Norway, 2017.

[30] K. Jaroś, A. Witkowska, and R. Śmierzchalski, "Data fusion of GPS sensors using particle Kalman filter for ship dynamic positioning system," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 89–94, Miedzyzdroje, Poland, 2017.

[31] J. M. Núñez, M. G. Araújo, and I. García-Tuñón, "Real-time telemetry system for monitoring motion of ships based on inertial sensors," *Sensors*, vol. 17, no. 5, p. 948, 2017.

[32] E. Borràs, J. Ferré, R. Boqué, M. Mestres, L. Aceña, and O. Busto, "Data fusion methodologies for food and beverage authentication and quality assessment – a review," *Analytica Chimica Acta*, vol. 891, pp. 1–14, 2015.

[33] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, 2016.

[34] Z. Ercan, V. Sezer, H. Heceoglu et al., "Multi-sensor data fusion of DCM based orientation estimation for land vehicles," in *2011 IEEE International Conference on Mechatronics (ICM)*, pp. 672–677, Istanbul, Turkey, 2011.

[35] Y. L. Zhang, J. H. Park, N. O. Sel, and K. T. Chong, "Robot navigation using a DR/GPS data fusion," *Applied Mechanics and Materials*, vol. 392, pp. 261–266, 2013.

[36] M. A. A. Akhoundi and E. Valavi, "Multi-sensor fuzzy data fusion using sensors with different characteristics," 2010, http://arxiv.org/abs/1010.6096.

[37] T. Yairi, K. Hori, and S. Nakasuka, "Sensor space discretization in autonomous agent based on entropy minimization of behavior outcomes," in *Proceedings. 1999 IEEE/SICE/RSJ. International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI'99 (Cat. No.99TH8480)*, pp. 111–116, Taipei, Taiwan, 1999.

[38] P. Scherz, A. Haderer, K. Pourvoyeur, and A. Stelzer, "Embedded sensor fusion system for unmanned vehicle navigation," in *2008 IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications*, pp. 192–197, Beijing, China, 2008.

[39] F. P. Vista IV, D.-J. Lee, and K. T. Chong, "Design of an EKF-CI based sensor fusion for robust heading estimation of marine vehicle," *International Journal of Precision Engineering and Manufacturing*, vol. 16, no. 2, pp. 403–407, 2015.

[40] F. P. Vista IV and K. T. Chong, "Design, development, and deployment of real-time sensor fusion (CnW + EKF) for a Linux-based embedded system using Qt-anywhere," *Journal of Sensors*, vol. 2018, Article ID 8695397, 14 pages, 2018.

[41] US Army, *Advanced Map and Aerial Photograph Reading*, U.S. Army Headquarters, War Department, 1941.

[42] D. DePriest, *NMEA Data*, GPS NMEA Information, 2019, https://www.gpsinformation.org/dale/nmea.htm.

[43] C. Dussault, R. Courtois, J.-P. Ouellet, and J. Huot, "Influence of satellite geometry and differential correction on GPS location accuracy," *Wildlife Society Bulletin*, vol. 29, no. 1, pp. 171–179, 2001.

[44] M. F. Yuen, *Dilution of Precision (DOP) Calculation for Mission Planning Purposes*, Naval Postgraduate School, Monterey, CA, Californina, 2009.

[45] S. Elizabeth and R. Jothilakshmi, "Convergence analysis of extended Kalman filter in a noisy environment through difference equations," *International Journal of Differential Equations and Applications*, vol. 14, no. 2, 2015.

[46] E. Kamrani, A. N. Foroushani, M. Vaziripour, and M. Sawan, "Detecting the stable, observable and controllable states of the human brain dynamics," *Open Journal of Medical Imaging*, vol. 2, no. 4, pp. 128–136, 2012.

[47] B. Southall, B. F. Buxton, and J. A. Marchant, "Controllability and observability: tools for kalman filter design," in *Proceedings of the British Machine Vision Conference 1998*, pp. 17.1–17.10, Southampton, UK, 1998.

[48] S. Julier and J. K. Uhlmann, "General decentralized data fusion with covariance intersection," in *Handbook of Multisensor Data Fusion: Theory and Practice*, pp. 319–344, CRC Press, Boca Raton, FL, USA, 2009.

[49] S. D. Levy, "TinyEKF, GitHub," 2019, https://github.com/simondlevy/TinyEKF.