Computer Science Senior Software Engineering Project (CS461)
Fall 2016

Head-Up Display Alignment System

Design Document

Version 1.0

Authors:
Krisna Irawan
Jiongcheng Luo
Drew Hamm

Supervised by:
Rockwell Collins, Inc.

Abstract

A Head-up Display (HUD) Alignment system is developed as a proof of concept aims to explore a potential technological innovation for the HUD system that presents critical flight information to pilots. The primary objective of this project is to reduce the cost and time required to precisely align flight information to the HUD by introducing an additional sensor component to the system to make the alignment process more dynamic. This document is intended for use by Rockwell Collins and their HUD system development team. This document provides and explains an overall system framework, design viewpoints and specific design description for each viewpoint within the system.

December 2, 2016

Contents

I. Introduction

A. Purpose

The purpose of this software design document is to provide a description that sufficiently describes the system design. The system design must be specified completely to the point in which is needed for the development to proceed. The description will fully specify what is to be built, how it will be built and what expectations need to be met at completion.

B. Purpose

A HUD or a Head-Up Display provides critical information to the pilots during flight environment. Currently, the HUD obtains data from an aircrafts mounted device called an Inertial Reference Unit (IRU), which an IRU would output precise and aligned data to the HUD through an mechanical alignment system. However, the current alignment process requires specialized equipment and epoxy which is time consuming, costly, and interrupts production line progress for the original equipment manufacturer. In addition, the resulting HUD alignment, while precise, does not compensate for airframe droop during flight. Rockwell Collins looks forward to a new alignment methodology utilizing an inexpensive microelectromechanical systems (MEMS) IRU mounted onto the HUD to infer alignment data from the aircrafts precisely mounted and aligned IRU.

This project is to develop a feasible demonstration system as a proof of concept for Rockwell Collins, that will prove there is an algorithm that is able to output precise and aligned data with reduced installation cost utilizing the data from both the inexpensive MEMS IRU and the aircraft mounted IRU. The outcome (aligned-data) of this algorithm will compensate the alignment error correctly, and the alignment error should be within a range of one milliradian. The product will make the alignment process more dynamic and less time consuming. The dynamic alignment process also makes this new system compensate for the airframe droop that happens during the flight environment. As well as from the perspective of the industries, this product aims to improve the installation process by reducing cost and time for all parties involved.

C. Overview

The following second and third part of this document are glossary and references that we have for this document. The fourth part of this document cover the design and implementation details of this project. The design and implementation details of this document are split into six different viewpoints. These viewpoints contain context view, composition view, dependency view, interface view, interaction view, and algorithm view.

II. Definitions

- **HUD**
  A Head-Up Display (HUD) is a transparent display placed in front of a pilots head position in the cockpit of the aircraft. A HUD presents critical flight information to the pilots during the flight environment by using graphical, numerical and symbolical data [1].

- **Airframe Droop**
  As the center of the aircraft is pushed up due to lift, the nose of the aircraft is pulled down by its weight. The result of these conflicting forces causes a bend in the aircraft 's frame.

- **Conformal Attitude**
  A method of presenting flight information on the HUD. With conformal attitude presentation, the displayed information on the HUD is presented with respect to the real world (outside scene in front of the aircraft), and the presentation is dependent on the head position of the pilot [2].

- **Real-time**
  In this system, the algorithm should recognize and precisely align the alignment error for IMU output within minimal time (e.g., 500 milliseconds).

- **IRU/IMU**
  An Inertial Reference Unit (IRU), sometimes its called an Inertial Measurement Unit (IMU) is a device consists of inertial sensors typically including MEMS gyroscopes, accelerometers and compasses. These MEMS sensors measure and provide data of an aircraft 's velocity, acceleration and orientation [3].

- **MEMS**

An Inertial Reference Unit (IRU), sometimes its called an Inertial Measurement Unit (IMU) is a device consists of inertial sensors typically including MEMS gyroscopes, accelerometers and compasses. These MEMS sensors measure and provide data of an aircraft 's velocity, acceleration and orientation [4].

- **Accelerometer**
  An electromechanical device that measures the amount of static acceleration due to gravity, in other word, the rate of change in velocity of the mounted object [5].

- **MEMS Gyroscope**
  An electromechanical device that measures rotational motion/angular velocity, in other word, the speed of rotation of the mounted object [6].

- **I2C**
  The Inter-integrated Circuit (I2C) Protocol is a protocol intended to allow multiple slave digital integrated circuits (chips) to communicate with one or more master chips [7].

- **SPI**
  Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers [8].

- **UART**
  A universal asynchronous receiver/transmitter (UART) is a block of circuitry responsible for implementing serial communication. On one end of the UART is a bus of eight-or-so data lines (plus some control pins), on the other is the two serial wires - RX and TX [9].

- **Multiplexer**
  A MUX (Multiplexer) is a combinational logic circuit designed to switch one of several input lines through to a single common output line by the application of a control signal [10].

- **DMP**
  A Digital Motion Processor (DMP is a programmable chip within the SparkFun MPU-9250 IMU MEMS sensor. Allows for filtering output as well as converting to quaternion output without the need of an additional microcontroller [11].

- **Alignment Algorithm**
  The uses of mathematical and logical equations to align the input data from the IRUs based on the data from previous precisely pre-aligned IRUs.

- **Symbology**
  The use and interpretation of symbols or special characters, in order for representing the corresponding flight data.

- **Milliradian**
  1 Milliradian (MRAD) = 0.001 radian, approximately 0.057296 degrees.

- **Quaternion**
  Four-element vector that is used to encode any rotation in a 3D coordinate system.

- **Altitude**
  A distance measurement (usually in vertical) between ground and the aircraft.

- **Latitude**
  AAn angular distance shown as a horizontal line, in degrees, minutes, and seconds of a point north or south of the Equator. Lines of latitude are often referred to as parallels [12].

- **Longitude**
  An angular distance shown as a vertical line, in degrees, minutes, and seconds, of a point east or west of the

Prime (Greenwich) Meridian. Lines of longitude are often referred to as meridians [12].

III. Context View

A. Program Design

This section is the black box view for the system. This section covers the high-level, overall design description for this project including design concern, design elements, and HUD alignment system workflow are presented in this section.

1) Design Concern: The primary goal of this project will be to use sensor data to find the initial alignment offset after HUD installation. The alignment offset must be found within the same accuracy of the previous installation standards and will be used within the system as a hard coded value. The secondary goal will be to use this additional sensor to find the alignment offset during flight to be used within the system as a dynamic value. The dynamic alignment offset must also be found within the desired accuracy standards. The outcome of this project is to create an algorithm that will produced the correct alignment data for the HUD alignment system. The aligned-data will compensate the alignment error correctly, and the alignment error should be within a range of one milliradian.

*2) **Design Elements**:*

a) Stakeholders: Rockwell Collins HUD system engineers. This program is a proof of concepts that intended to be applied to the future HUD system. This program will be used by Rockwell Collins to determine the availability of having an additional MEMS IRU in the new HUD alignment system.

b) Design relationships: Users will be able to interact with the program via simulation or physical demonstration system. The users will be able to move and interact with the physical demonstration system to simulate the airframe droop and misalignment in the system.

c) Design constraints: This project is limited to its hardware, signal handshake protocol, and higher order language requirements:

- This IMU model will be *MPU-9250* MEMS sensors.
- The *MPU-9250* is programmable and it outputs 8,000 samples per second, down to 3.9 samples per second.
- The microcontroller model will used as the *Metro Mini 328* to set up the sensors
- The model *ATmega328P* as the core chip in *Metro Mini 328*.
- C/C++ programming language is used for the development of the software based on the selected microcontroller.

IV. Composition View

A. Hardware Configuration Design

This section will cover the specific design plan for assembling all hardware compositions of the system. This section is intended to summarize and explain all the necessary hardware components involved in the system design and their specification.

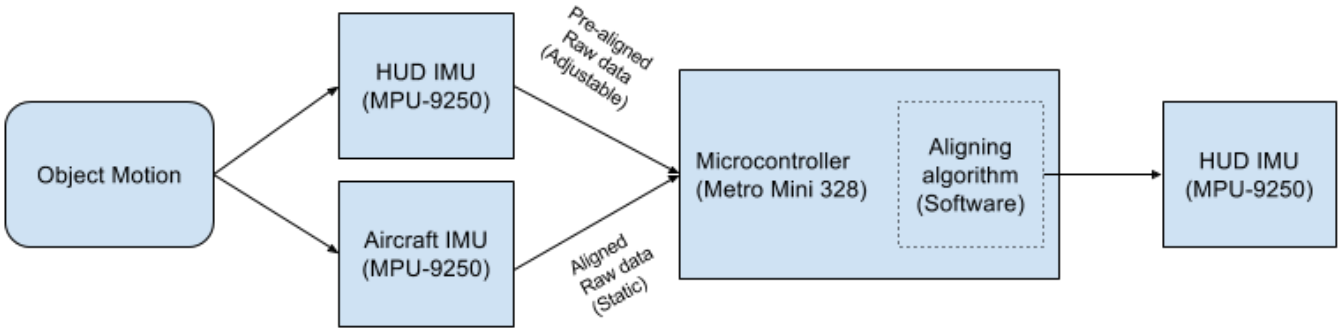
1) Design Concerns: One of the concerns in developing this demonstration system is to use inexpensive MEMS IMU instead of using the one from the original equipment manufacturer, which is costly. Hence, we chose to use the *Metro Mini 328* as the microcontroller that has a fair price in the market currently. *Metro Mini 328* provides an option that allows to use a 3.3V mode power supply. And since 3.3V is the standard operating power supply our selected IMU sensors model, *MPU-9250*, *Metro Mini 328* make it easy to connect with *MPU-9250* without any external circuit or additional resistors.

2) Design Elements:

- Simulated aircraft object: a remote controlled vehicle/drone
- HUD IMU: one or more *MPU-9250*s
- Aircraft IMU: one or more *MPU-9250*s
- Microcontroller: one *Metro Mini 328*
- Output Display: Graphical output by a computer software (for demonstration system only)

a) Design Relationships: Initially, all the hardware components will be mounted onto the simulated aircraft object. While the object is in a motion that could be a moving action by a remote vehicle, a flying action of a drone or simply a hand-made motion by manually adjusting, the IMU sensors (both HUD and aircraft IMUs) are simultaneously updating these motion data such as acceleration, angular velocity and orientation of the moving object and sending data to the connected microcontroller that is the *Metro Mini 328* though the *I2C* protocols. Here, the input from both IMUs to the microcontroller are raw data, and these two groups of data will be the input for the alignment algorithm. Refer Figure 1 in section **Component Diagram** for more detail about the relationship among all hardware components.

Fig. 1. Data Flow between Hardware Components



b) Function Attribute: The entire system is constructed based on four major entities including a simulated aircraft object, a microcontroller, a HUD IMU and an aircraft IMU. In addition, an output display is necessary in this demonstration system for testing purpose but it is not part of the original system, that means this display is used for simulating the real HUD output on an aircraft. Respectively, these two IMUs in this demonstration system represent and simulate the functionalities for the real IMU component mounted on a head-up display of an aircraft and the aircraft body itself. Within this demonstration system, both IMUs may consist of one or more *MPU-9250* (e.g., accelerometers). The simulated aircraft object provides motion input to the IMU sensors this object can be any kinds of vehicle or drone that is able to provide physical motion.

V. Dependency View

A. Dependency Design

This section will cover the interconnection of hardware and software throughout the system. This section is intended to summarize and explain all the necessary components in terms of their dependencies.

1) Design Concern: All components must have clearly defined relationships. All components within the system should be included in the dependency design, refer Figure 2 UML diagram for more detail.

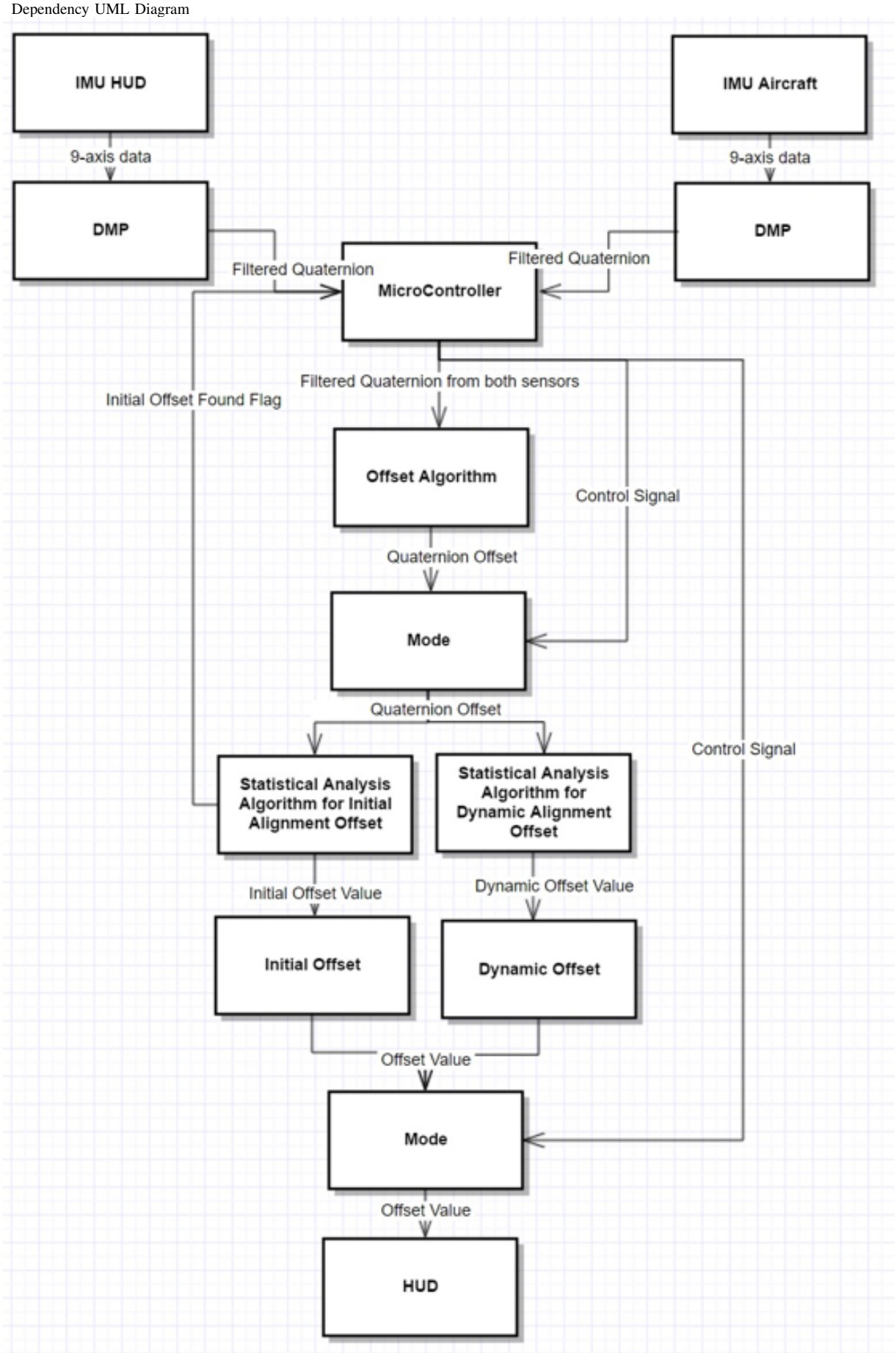
2) Design Elements:

- HUD IMU: *MPU-9250*.
- HUD IMU DMP: Provides on board filtering and 9-axis to quaternion conversion.
- Aircraft IMU: *MPU-9250*.
- Aircraft IMU DMP: Provides on board filtering and 9-axis to quaternion conversion.
- Microcontroller: *Metro Mini 328* .
- Offset Algorithm: Takes input from both IMUs through the microcontroller and returns the offset between each other.
- Statistical Analysis Algorithm for Initial Alignment Offset: Takes multiple offset inputs until a valid initial offset can be returned.
- Statistical Analysis Algorithm for Dynamic Alignment Offset: Takes multiple offset inputs until a valid dynamic offset can be returned.
- Initial Offset: The value found that represents the initial alignment offset.
- Dynamic Offset: The value found that represents the dynamic alignment offset.
- HUD: Graphical output by a computer software (for demonstration system only).

3) Dependency Attributes: See Figure 2.

4) Design Rationale: An overall view of the system is required to understand how each component fits into the system in relation to other components. By providing the dependency design, faults within the system can be determined by analyzing their specific scopes and connections. The dependency view will be used to determine the prioritization for component development to ensure that components are being developed by time of need to avoid blocking.

Fig. 2. Dependency UML Diagram



VI. Interface View

A. Software User Interface Design

This section will cover the specification of the software user interface portion for this project. This section is intended to guide the software development team to interact correctly with the software user interface.

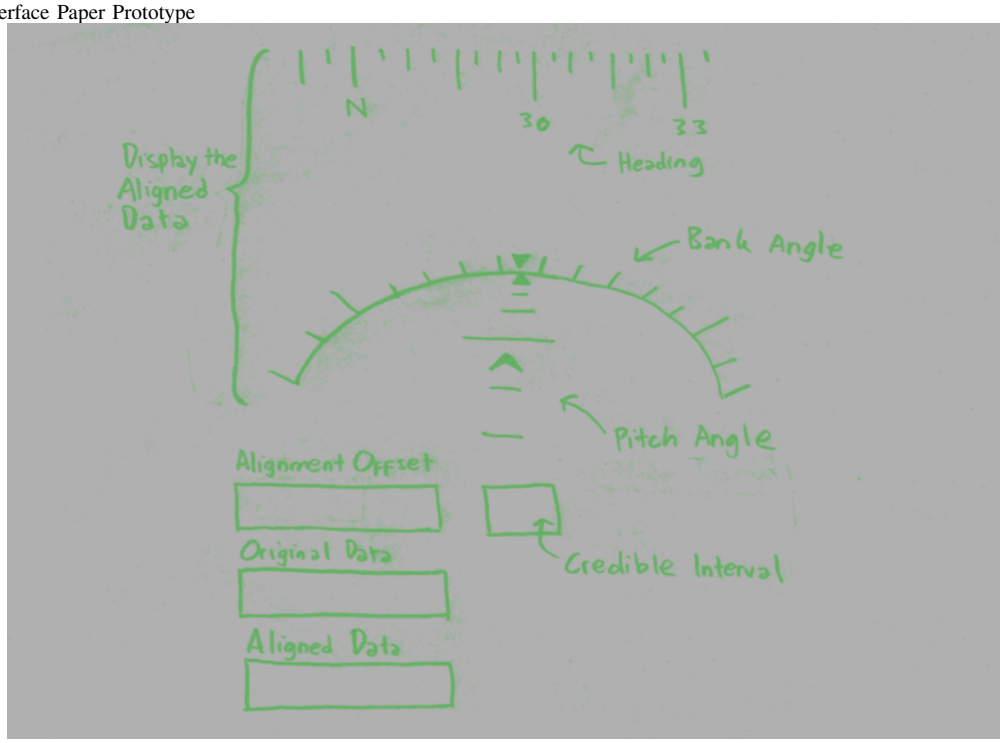
1) Design Concerns: The goal of this project is to have a demonstration system that can be used by Rockwell Collins HUD system engineers to determine the availability of having an additional MEMS IRU in the new HUD alignment system. Graphical user interface will be crucial for our project presentation and demonstration system. The algorithm from our project will output a raw alignment data and it will be hard to understand the output without the help of graphical user interface. The graphical user interface will put context to the data and present the correct symbology of the alignment data output. Rockwell Collins will provide us with a generic HUD symbology picture which could be moved up/down, left right to illustrate alignment.

2) Design Elements: This section will cover the interface attributes that defines each functionality on the display and a paper prototype as Figure 3.

a) Interface Attributes:

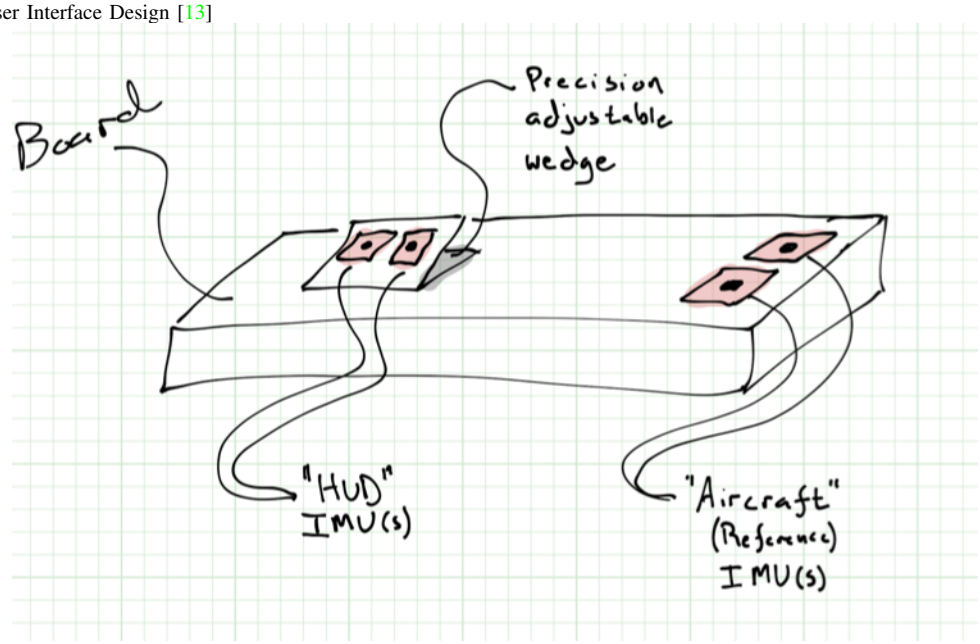
- *Heading:* Display the direction of where the aircraft is pointing
- *Bank Angle:* Display the vertical elevation angle of the aircraft
- *Pitch Angle:* Display the angle between the aircraft and the horizon. Pitch angle shows the horizontal elevation angle of the aircraft
- *Credible Interval:* Display the degree of certainty of the alignment offset
- *Alignment Offset:* Display the calculated alignment offset detected by the algorithm
- *Original Data:* Display the original alignment data from the sensors
- *Aligned Data:* Display the calculated alignment data from the algorithm

Fig. 3. Output Interface Paper Prototype



3) Design Rationale: An important piece of this project is the dynamic alignment algorithm and our clients encouraged us to spend most our time in developing and refining the dynamic alignment algorithm. Thus, using a full HUD display will also be out of scope for this project. However, we want to create a mini HUD display to make our demonstration system look astonishing. Our user interface has to be simple and clean. To make it simple and less time consuming, we will create our software user interface by using the generic HUD symbology picture provided by Rockwell Collins and a built in visual studio graphical user interface toolkit. The built in visual studio graphical user interface toolkit will help us to arrange the generic HUD symbology picture to its designated place and also display the appropriate data to the screen.

Fig. 4. Physical User Interface Design [13]



B. Physical User Interface Design

This section will cover the specification of the physical user interface portion for this project. This section is intended to guide the software development team to interact correctly with the physical user interface in the demonstration system.

1) Design Concern: The goal of this project is to have a demonstration system that can be used by Rockwell Collins HUD system engineers to determine the availability of having an additional MEMS IRU in the new HUD alignment system. The user of this product will be able to directly interact with the physical user interface of this product. The physical user interface is the most interactive piece of this product. Refer to Figure 4 The user will have the ability to adjust the precision adjustable wedge to simulate the HUD offset during flight. The user will also have the ability to move the board to simulate the airplanes movement. This physical user interface settings is arranged to simulate the flight environment. Rockwell Collins will help us in creating the board with the precision adjustable wedge in it.

2) Design Elements:

a) Interface Attributes:

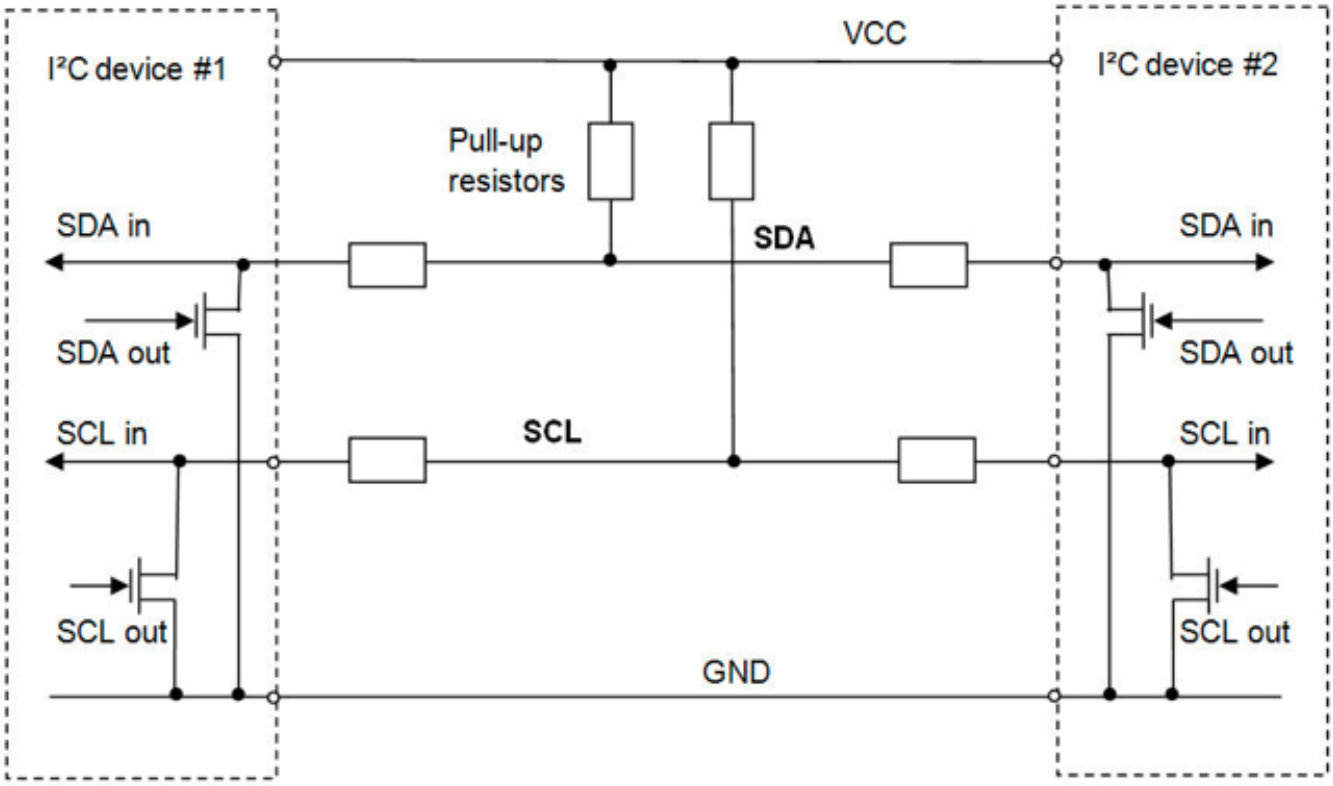
- Board: A platform that can be moved to simulate the airplane movement
- Precision adjustable wedge: An adjustable platform that can be moved to simulate the HUD alignment offset after initialization
- HUD IMU sensors: The sensors that act as the HUD IMU
- Aircraft IMU sensors: The sensors that act as the precisely aligned aircraft IMU.

3) Design Rationale: The physical user interface is a crucial piece of this project. This project algorithm is heavily based on the alignment and the set-up of our physical user interface. The physical user interface is an important part of our demonstration system. Our physical user interface is the most interactive piece of this product. Different than the software user interface, the user will be able to directly interact with our physical user interface. Our clients come up with the design for the physical user interface. We think that this is a great set-up to start our project. Since the wedge can be pre-aligned and adjusted, it eliminates some of our burden to find out the delta angle of the sensors. The precision adjustable wedge will also make our demonstration system more user friendly since it can be adjusted without requiring any additional tools.

VII. Interaction View

1) Hardware Communication Design: This section will describe the communication methods between hardware components within the entire system. This section is intended to list out and describe how each hardware component talks with others including communication protocols and connecting methods.

Fig. 5. *I2C* Connection between Slave Devices



2) Design Concerns: There are two necessary communications/connections for one input-output cycle, one is between the IMU and the microcontroller, the other is between the microcontroller and output display. Refer to the Technology Review for this project, there are three options for choosing the communication protocols between the IMU and the microcontroller: *I2C*, *SPI* and *UART*. Based on our project design, *I2C* protocol is the prior consideration for communicating between IMU and the microcontroller since *Metro Mini 328* has limited pin numbers and *I2C* only takes two pins (*SDA* and *SCL*) for wiring connection and *I2C* provide fair transmission speed. In addition, USB is the appropriate communication method between the microcontroller and output display (computer) since *Metro Mini 328* provides direct USB port and the makes the implementation simple and fast.

3) Design Elements:

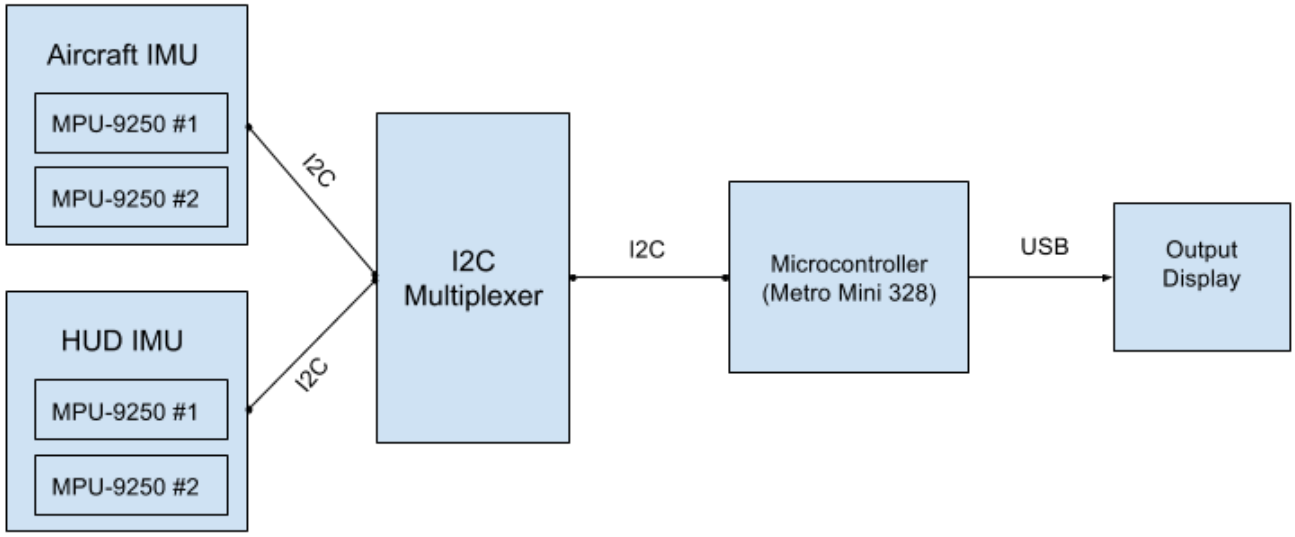
a) I2C Protocol: One *I2C* device (IMU) takes totally 4 pins to connect with a master board (microcontroller), besides VDD and GND pins, there are *SDA* and *SCL* which represent ”serial data” and ”serial clock” respectively. Both SDA and SCL are bi-direction wires, that means they can transfer sensor data to the master board, as well as transferring command data from the master board. The data rate has to be chosen between 100 kbps, 400 kbps and 3.4 Mbps, respectively called standard mode, fast mode and high speed mode. Some IC variants include 10 kbps (low speed mode) and 1 Mbps (fast mode +) as valid speeds [14].

In this system, there are more than one IMU sensors (*MPU-9250*) needed to be working together in order to retrieve a higher accuracy data output. Hence, we choose to use *I2C* protocol since it allows multi-devices communication. Additional devices are also known as the slave devices, referring to Figure 5, *I2C* allows *I2C* device #1 and #2 (these represent *MPU-9250* in this project) to communicate with each other by just using one SDA and one SCL wire (besides of VCC and GND), there is no need for extra chip select pins like SPI protocol for extra devices.

b) Universal Serial Bus: Metro Mini 328 uses Universal Serial Bus (USB) connection as default communication method between microcontroller and output display (computer). Specifically, *Metro Mini 328* provides a Micro-USB (B) port, the implementation of this connection is simple and fast.

4) Design Constraints: Either the Aircraft or HUD IMU consists of multiple *MPU-9250* sensors units, they can be put parallel together and act as an entity. See as Figure 6, both IMUs will be connected to an *I2C* Multiplexer separately by

Fig. 6. Communcation Method between System Hardware Components [14]



I2C protocol. An *I2C* multiplexer allows the microcontroller to select the particular IMU unit based on the address that the *I2C* Multiplexer assign to that IMU, so that we can easily split two groups of data (from HUD and Aircraft IMU) for the software program. Finally, an output display will retrieve a finalized aligned data from the microcontroller and allows for testing.

VIII. Algorithm View

A. Statistical Analysis Method for Initial Alignment Offset Design

This section will cover the specification of the statistical analysis method for initial alignment of this project. This section is intended to guide the software development team to develop a suitable algorithm. This algorithm is intended to calculate the proper initial alignment data from the sensors under an acceptable degree of certainty.

1) Design Concerns: Our primary goal is to find the initial alignment offset. Using a statistical analysis method is desired to ensure an accurate result. This algorithm requires the input of both sensors taken at the same time. A series of inputs will be received by this algorithm until the offset has been found to be within one milliradian of accuracy.

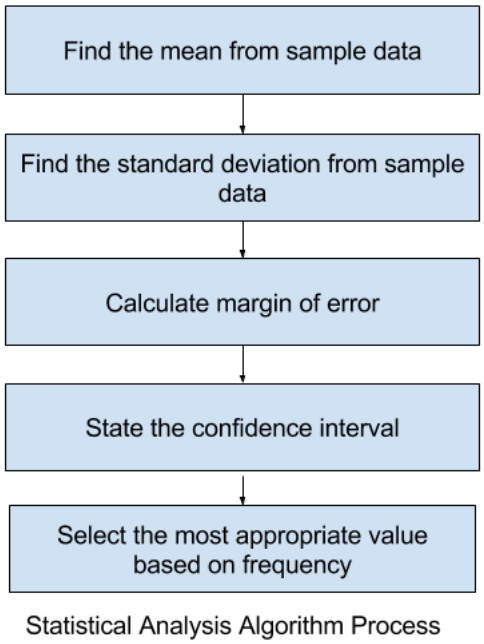
2) Design Elements: This section will cover the algorithm attributes that defines each functionality of the statistical analysis method for initial alignment offset

a) Processing Attributes:

- *Assumption:* The offset algorithm will calculate the correct alignment offset value
- *Prerequisite:* Both sensors must be transmitting data that corresponds to the same time values.
- *Input:* A series of sensor input values.
- *Output:* An alignment offset value that best represent the alignment error under an acceptable credible interval
- *Acceptable Range:* 95% credible interval
- *Acceptable Accuracy:* unit milliradian

3) Design Rationale: Confidence interval, credible interval, and tolerance interval are three possible options of statistical analysis method that we can use to gain credibility of our alignment data. The confidence interval will give us the frequency of the true value output being generated by the algorithm over a time period. However, until the time of calculation, we are not sure what kind of value that we will get from the algorithm. Hence, leaving us with no parameter to work on confidence interval. The tolerance interval will give us the spread of error in alignment data being generated by the algorithm. This statistical method can be useful for us to further refine our algorithm. Credible interval will give us the credibility to state the certainty of a true value within an interval of data. This method will give us the true value that might be a great representation of our alignment data. We think that credible interval method fits the most with our project. Thus, this makes us decide to do credible interval for our statistical analysis method.

Fig. 7. Statistical Analysis Algorithm Process



B. Statistical Analysis Method For Dynamic Offset Design

This section will cover the implementation details of the statistical analysis method for the offset error of this project. This section is intended to guide software developer to develop a suitable algorithm. This algorithm is intended to calculate the proper offset error value under an acceptable degree of certainty.

1) Design Concerns: The goal of using a statistical analysis method is to find the credibility of the output that we get from the algorithm. This algorithm will get its input from the offset algorithm. Then, the alignment error data that we get from the offset algorithm will be processed by this algorithm to find the credible interval for a range of offset error values. This algorithm will find the credible interval of the alignment error data and decide which value is best represent the alignment error for that instance. Referring to Figure 7 for specific process details.

2) Design Elements: Elements include assumption, prerequisite, expected input, expected output of this algorithm. This section will also cover the method and formula that we will be using to create this algorithm.

a) Processing Attributes:

- *Assumption:* The offset algorithm will calculate the correct alignment error value
- *Prerequisite:* An alignment error value produced by the offset algorithm
- *Input:* A range of alignment error values
- *Output:* An alignment error value that best represent the alignment error under an acceptable credible interval
- *Acceptable Range:* 95% credible interval
- *Acceptable Accuracy:* unit milliradian

3) Design Rationale: Confidence interval, credible interval, and tolerance interval are three possible options of statistical analysis method that we can use to gain credibility of our alignment data. The confidence interval will give us the frequency of the true value output being generated by the algorithm over a time period. However, until the time of calculation, we are not sure what kind of value that we will get from the algorithm. Hence, leaving us with no parameter to work on confidence interval. The tolerance interval will give us the spread of error in alignment data being generated by the algorithm. This statistical method can be useful for us to further refine our algorithm. Credible interval will give us the credibility to state the certainty of a true value within an interval of data. This method will give us the true value that might be a great representation of our alignment data. We think that credible interval method fits the most with our project. Thus, this makes us decide to do credible interval for our statistical analysis method.

C. Offset Algorithm Design

This section will cover the implementation details of the offset algorithm for this project. This section is intended to guide software developer to develop a suitable algorithm. This algorithm is intended to calculate the proper offset between two quaternion inputs.

1) Design Concerns: The resulting output must be the quaternion offset of two quaternion inputs. The algorithm will first take the input of the first input. The next step will be to multiply both inputs against each other. The resulting multiplication will be the desired output.

2) Design Elements:

a) Processing Attributes:

- Quaternion input A: The first of two quaternion inputs
- Quaternion input B: The second of two quaternion inputs
- Quaternion output offset: The resulting offset with the form of quaternion

3) Design Rationale: The design requires an offset in terms of a quaternion output in order to be used within the rest of the system.

IX. Conclusion

This design document has covered six main design views from the perspective of hardware and software, each view contains specific design viewpoints as well as relevant detailed explanations. This document would help and provide necessary plan information for our project implementation in terms of hardware set-up, algorithm design, testing and debugging. More importantly, this design document allows us to think thoroughly over the entire implementation process, and helps us move forward with in the development of our solutions to the problem.

REFERENCES

[1] "Head-up guidance system," Rockwell Collins. [Online]. Available: https://www.rockwellcollins.com/Data/Products/Displays/Head-Up_Displays-HUD/Head-up_Guidance_System.aspx

[2] D. R. Jones, T. S. Abbott, and J. R. B. II, "Concepts for conformal and body-axis attitude information for spatial awareness presented in a helmet-mounted display," The National Aeronautics and Space Administration, 1993. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930014219.pdf

[3] "Inertial navigation system, a.k.a inertial reference unit," Earth Observing Laboratory, 2005. [Online]. Available: https://www.eol.ucar.edu/instruments/inertial-navigation-system-aka-inertial-reference-unit

[4] "What is mems technology?" MEMS and Nanotechnology Exchange. [Online]. Available: https://www.mems-exchange.org/MEMS/what-is.html

[5] "A beginners guide to accelerometers dimension engineering," Dimension Engineering LLC. [Online]. Available: http://www.dimensionengineering.com/info/accelerometers

[6] A1RONZO, "Gyroscope," SparkFun Electronics. [Online]. Available: https://learn.sparkfun.com/tutorials/gyroscope

[7] SFUPTOWNMAKER, "I2c," SparkFun Electronics. [Online]. Available: https://learn.sparkfun.com/tutorials/i2c

[8] "A brief introduction to the serial peripheral interface (spi)," Arduino. [Online]. Available: https://www.arduino.cc/en/Reference/SPI

[9] Jimbo, "Serial communication: Uarts," SparkFun Electronics. [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication/uarts

[10] "The multiplexer," ElectronicsTutorials. [Online]. Available: http://www.electronics-tutorials.ws/combination/comb_2.html

[11] "Mpu-9250 product specification revision 1.0," InvenSense. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/5/5/0/MPU9250REV1.0.pdf

[12] "Latitude and longitude facts," Worldatlas. [Online]. Available: http://www.worldatlas.com/aatlas/imageg.htm

[13] W. Lahr, 2016.

[14] "Introduction to i2c and spi protocols," Byteparadigm. [Online]. Available: http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/

[15] "Ieee recommended practice for software requirements specification (ieee std 830-1998)," IEEE Computer Society, 1998.

[16] C. Rapids, "Head-up guidance system (hgs) for midsize and light business aircraft," Rockwell Collins. [Online]. Available: https://www.rockwellcollins.com/-/media/Files/Unsecure/Products/Product_Brochures/Displays/Head_up_displays/HGS-3500_White-Paper.ashx

[17] "Adafruit pro trinket - 5v 16mhz," Adafruit. [Online]. Available: https://www.adafruit.com/products/2000

[18] "Sparkfun imu breakout - mpu-9250," SparkFun Electronics. [Online]. Available: https://www.sparkfun.com/products/13762

X. Signatures

Client	Date
Author 1	Date
Author 2	Date
Author 3	Date