

OREGON STATE UNIVERSITY

# Head-Up Display Alignment System

## Requirement Document

*Senior Software Engineering Project (CS461)*  
*Fall 2016*

*Authors:*

Krisna Irawan  
Jiongcheng Luo  
Drew Hamm

*Supervised by:*

Rockwell Collins, Inc.

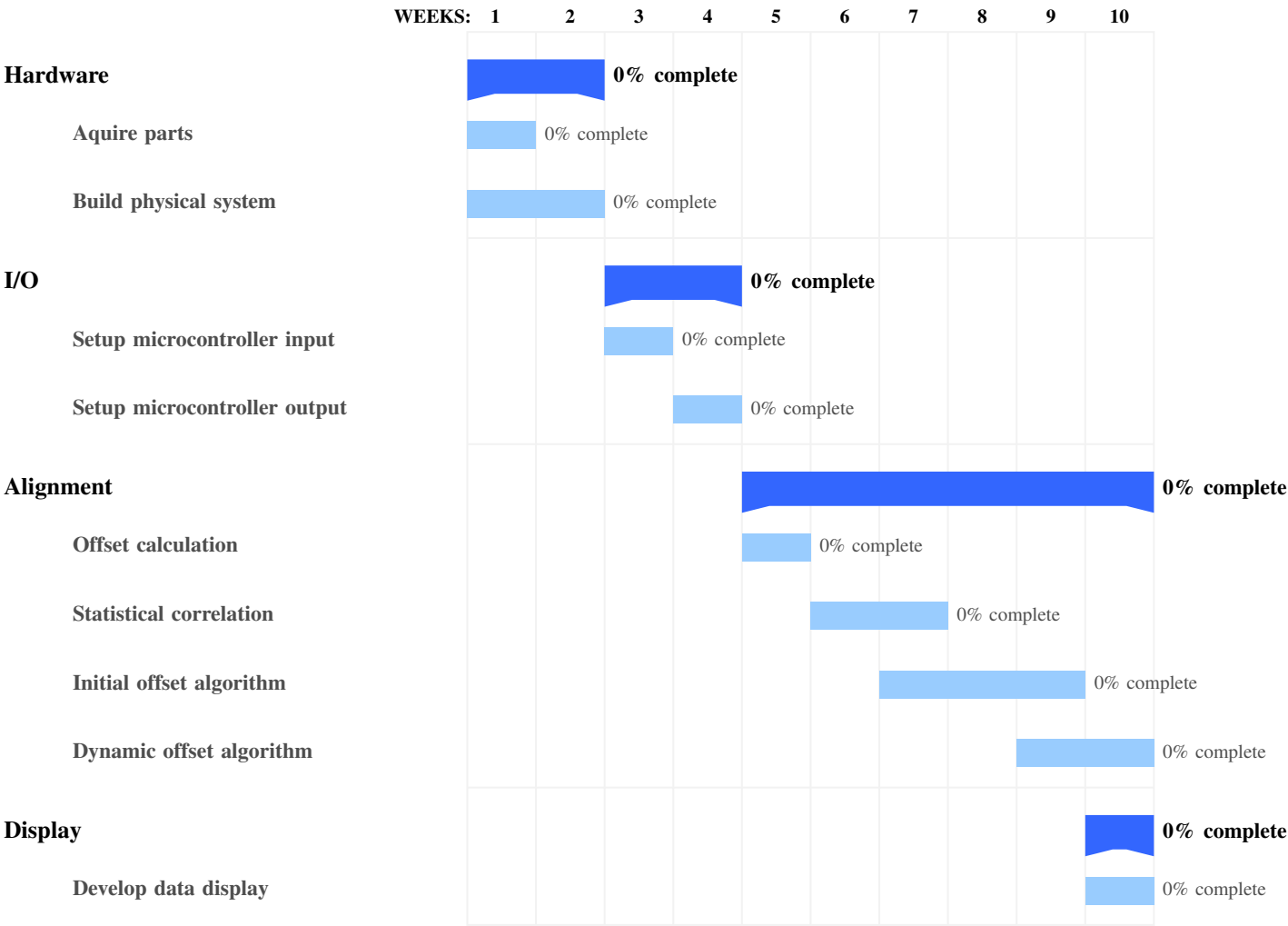
### Abstract

We are working with Rockwell Collins to explore potential technological innovations relating to their Head-Up Display (HUD) systems that present critical flight information to pilots. Our primary objective is to improve Rockwell Collins current HUD systems by reducing the cost and time required to precisely align flight information to the HUD. To meet our objective we will look into using a new alignment methodology in conjunction with the current HUD system as a proof of concept. The product being developed is a demonstration system that looks to include a MEMS IRU mounted onto the HUD and a new alignment algorithm that utilizes this additional sensor to determine accurate HUD alignment. This document will introduce the specific details of the demonstration system and describe the requirements for the development of the product.

# Contents

<b>I</b>	<b>Timeline for Project Plan</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>3</b>
II-A	Purpose	3
II-B	Scope	3
II-C	Definitions, Acronyms, and Abbreviations	3
II-D	Overview	4
<b>III</b>	<b>Overall Description</b>	<b>5</b>
III-A	Product Perspective	5
III-B	Product Functions	5
III-C	User Characteristics	5
III-D	Constraints	5
III-E	Assumption and Dependencies	6
<b>IV</b>	<b>Specific Requirements</b>	<b>6</b>
IV-A	External Interface Requirements	6
IV-A1	User Interface	6
IV-A2	Hardware Interface	6
IV-A3	Software Interface	6
IV-A4	Communication Interface	6
IV-B	Functional Requirements	7
IV-B1	Sensors	7
IV-B2	Algorithm	7
IV-B3	Statistics	7
IV-B4	Simulation	7
IV-C	Performance Requirements	7
IV-D	Design Constraints	7
IV-E	Software System Attributes	7
IV-E1	Reliability	7
IV-E2	Availability	7
IV-E3	Security	7
IV-E4	Maintainability	8
IV-E5	Portability	8
<b>V</b>	<b>Signatures</b>	<b>9</b>

I. TIMELINE FOR PROJECT PLAN



## II. INTRODUCTION

### A. Purpose

The purpose of this document is to describe a demonstration system that explores using an additional HUD mounted MEMS IRU to meet previous installation standards while reducing costs. This system will further be described by its use of sensor data to dynamically find alignment offset error during flight as a consequence to airframe droop. This document describes the hardware and software requirements along with the functional and non-function requirements of the project. This document is intended to be used by the Head-Up display system engineer team of Rockwell Collins as a proof of concept.

### B. Scope

This project aims to develop a demonstration system to work as a proof of concept for Rockwell Collins. This project is called the HUD Alignment System. 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.

Currently, the HUD obtains data from an aircrafts mounted device called inertial reference unit (IRU), this IRU outputs precise and aligned data to the HUD. 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 works on a solution that utilizes the data from both the inexpensive MEMS IRU and the aircraft mounted IRU to develop an algorithm, which aims to output precise and aligned data with reduced installation cost.

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. Definitions, Acronyms, and Abbreviations

- **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 [?].

- **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 [?].

- **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 [?].

- **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 [?].

- **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 [?].

- **MEMS Gyroscope**

An electromechanical device that measures rotational motion/angular velocity, in other word, the speed of rotation of the mounted object [?].

- **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 [?].

- **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.

- **Symbolology**

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**

AAAn 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 [?].

- **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 [?].

#### *D. Overview*

The second section of this document contains an overall description of the product, including the product perspectives, product functions, user characteristics, constraints, assumptions and dependencies. The third section of this document will specify the projects requirements.

### III. OVERALL DESCRIPTION

#### A. Product Perspective

Currently, the HUDs position is precisely aligned and determined during installation by a mechanical alignment method. Since the product will be developed as a proof of concept, this product will be a demonstrated system by using two separate MEMS IMUs respectively simulate for the add-on HUD IRU and Aircraft IRU. By including the HUD IRU, this alignment system will be able to determine the HUDs position as well as providing additional benefit to determine the HUDs shifting position during flight as a result of airframe droop.

#### B. Product Functions

- Both HUD and Aircraft IRUs will measure the acceleration and orientation of the mounted target object.
- The master boards (microcontrollers) will recognize both IRUs measurement data (raw) and output to the alignment algorithm.
- The alignment algorithm will compare the outputs from both IRUs (data from the aircraft IRU is assumed to be precise and aligned) and be able to quantify the alignment error as well as their correlation.
- The alignment algorithm will be able to calculate the target objects precise position based on the acceleration and orientation output from both IRUs.
- The algorithm will account for the target objects position and rotation to ensure that the conformal attitude, flight path information and other displayed data is accurately aligned.
- This algorithm will determine the alignment error to a desired accuracy level as 1.0 milliradian.

See figure 1 as the entire workflow for data transmission and connections between the hardware

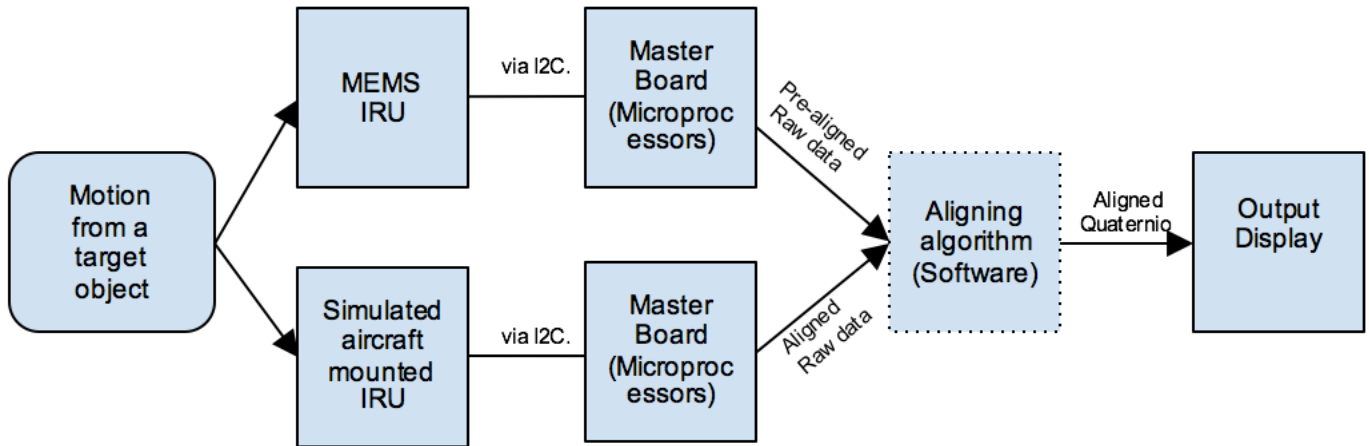


Figure 1: HUD Alignment System

#### C. User Characteristics

This alignment algorithm is intended to be applied to the HUD system, which will be used by Rockwell Collins to determine the availability of having an additional MEMS IRU in the for the new HUD alignment system. The intended users of this product are the technical expertise in this area, such as HUD system engineers in Rockwell Collins. The audience of this document will be familiar with the abbreviations and jargon used to describe the product. This familiarity reduces the requirement for more elaborate explanations of the project. As our expected user maintains technical expertise in the field, they will expect our project to meet a higher standard of competence.

#### D. Constraints

##### Hardware Limitations:

- This IMU model will be SparkFun MPU-9250 IMU MEMS sensors.
- The SparkFun MPU-9250 IMU is programmable to output 8,000 samples per second, down to 3.9 samples per second,
- The microcontroller model will used as the Pro Trinket chip to set up the sensors.
- The model ATmega328P as the core chip in Pro Trinket.

- The algorithm will account for the target objects position and rotation to ensure that the conformal attitude, flight path information and other displayed data is accurately aligned.
- This algorithm will determine the alignment error to a desired accuracy level as 1.0 milliradian.

#### **Signal Handshake Protocol:**

- I2C protocol is used to communicate between IMU MEMS IMUs.

#### **Higher order language Requirement:**

- C/C++ programming language is used for the development of the software.

#### *E. Assumption and Dependencies*

- This product will depend on both the microcontroller and the sensors that are being used within the system in terms of hardware limitations.
- We assume that the microcontroller can interact with our sensors as well as to meet any other hardware concerns.
- We assume that the sensors are sufficient to simulate the aircraft IRU and HUD mounted MEMS IRU when building our demonstration system.

### IV. SPECIFIC REQUIREMENTS

#### *A. External Interface Requirements*

This section describes the detailed breakdown of the entire system. A description of the user, hardware, software and communication interfaces are presented.

1) *User Interface:* Our user interface will be the physical manipulation of our demonstration system in order to generate movement data. The software user interface for this project is a display that will show the properly aligned symbology of the airplane. A generic HUD symbology picture which could be moved up/down, left/right will be used to illustrate the alignment.

#### *2) Hardware Interface:*

- A microcontroller will have access to data from two separate 9-axis sensors that are used to represent the aircraft IRU and HUD mounted MEMS IRU.
- The microcontroller will additionally be connected to an output device.
- The hardware is still not decided at this point. However, these are the product that we might use for the hardware.
- Product Name: Adafruit Pro Trinket - 5V 16MHz [?]
- Features:
  - ATmega328P onboard chip in QFN package
  - 16MHz clock rate, 28K FLASH available
- IMU Model: SparkFun IMU Breakout - MPU-9250 [?]
- Features:
  - Digital-output X-, Y-, and Z-axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of 250, 500, 1,000 and 2,000/sec and integrated 16-bit ADCs
  - Digital-output triple-axis accelerometer with a programmable full-scale range of 2g, 4g, 8g and 16g and integrated 16-bit ADCs
  - 3-axis silicon monolithic Hall-effect magnetic sensor with magnetic concentrator
  - Digitally programmable low-pass Gyroscope filter
  - Gyroscope operating current: 3.2mA
  - Accelerometer normal operating current: 450A
  - Magnetometer normal operating current: 280A at 8Hz repetition rate
  - VDD supply voltage range of 2.4 3.6V

3) *Software Interface:* At this point we are still not sure what additional library/software we will use in this project. We will use C-like language (C/C++) to program this project. We will also use the ATmega328P library since we will use the ATmega328P chip as our microcontroller board.

4) *Communication Interface:* I2C protocol will be used to communicate with MPU-9250 and Pro Trinket.

## B. Functional Requirements

### 1) Sensors:

a) *Sensor recognition*: This program will have the capability to recognize sensors. This program shall recognize and identify all sensors provided to the system. This functionality is the core to the input recognition capability.

b) *Input recognition*: This program will correctly interpret the data provided by the recognized sensors. This program will have the ability not only to recognize the outputs from the sensors, but also to interpret the outputs so it can be used as a proper input for the program. This functionality will be dependent on the success of the sensor recognition functionality.

### 2) Algorithm:

a) *Quaternion Manipulation*: This program will have the ability to do quaternion manipulation to the inputs. Inputs that are received from the sensors will be compared and manipulated to determine the alignment error. The desired accuracy level of the alignment error is within one milliradian. This functionality will execute in real-time, within 500 milliseconds.

### 3) Statistics:

a) *Statistical Correlation*: This program will have the capability to generate a confidence interval for each alignment error output being generated by the algorithm. Outputs in a form of alignment error will be used to determine the confidence interval of the data. This functionality will execute in real-time, within 500 milliseconds.

### 4) Simulation:

a) *Simulated HUD*: This program will have the capability to display the properly aligned symbology of the airplane. A generic HUD symbology picture which could be moved up/down, left/right will be used to illustrate the alignment of the airplane.

## C. Performance Requirements

- Number of terminals to be supported: Two terminals for both MEMS sensors.
- Amount of information to be handled: Programmable up to 3.9 to 8000 data per seconds. We will start with 10 data per seconds from both sensors.
- Type of information to be handled: Quaternion
- Valid Range of Accuracy: milliradians
- Timing: Currently not specified
- Destination of Output: Simulated HUD interface

## D. Design Constraints

This product is limited to the specification of the hardware. This product will get its input from MEMS IRU and aircraft IRU. The specification of the MEMS IRU and aircraft IRU will alter the quaternion manipulations being done to the data. Different MEMS IRUs will require a different quaternion manipulations to achieve the expected standard of accuracy. This hardware dependability makes this product dependent to one specific hardware. Thus, limiting our design to one specific hardware.

## E. Software System Attributes

1) *Reliability*: It is important to make sure the system in both software and hardware will be able to run properly and smoothly without abortion. As a proof of concept, reliability is important, a wrong data will lead to incorrect outcome in the determination of further development that based on this products.

2) *Availability*: The software piece of the entire system should be able to run without any crashes or other situation lead to system error (this is a critical issue during an real flight situation). Yet, the hardware pieces should be able to detect the failed components at the time when it happens. This product will have the capability to detect error and may restart itself when crashes.

3) *Security*: This product will be used internally by Rockwell Collins and since this product will be used for a proof of concept, security is not an issue.



4) *Maintainability*: This system does not work solely as a software neither a hardware product, which it has to be tested onto real flight situation. The current development is for a proof of concepts. Hence, maintainability will not be an issue for this project. This project will be used as a reference and basis of a future project in Rockwell Collins, but not directly used as the core/underlying code for any long-term product.

5) *Portability*: Portability of this system exists on the hardware level. Since hardware programming is required for this project to manipulate functionalities of the hardware components (e.g., sensors, microprocessors/microcontrollers), specific hardware programming language for this project is host-dependent. It depends on the specific controller boards we are using. In this project, we are using C-like languages (C/C++), which is one of the proven portable language for hardware programming for most general microcontrollers/microprocessors, therefore this system has high portability.

V. SIGNATURES

_____	_____
Client	Date

_____	_____
Author 1	Date

_____	_____
Author 2	Date

_____	_____
Author 3	Date