

Plume Estimation Using a Gas-Sensing Mobile Robot

A Major Qualifying Project Report

Submitted to the Faculty of the

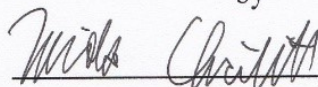
WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

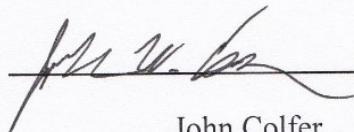
Degree of Bachelor of Science

in Aerospace Engineering

by



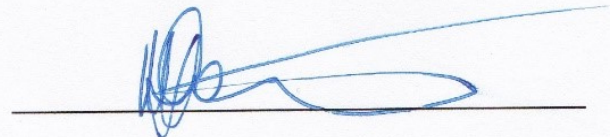
Nicholas Christie



John Colfer

March 25, 2016

Approved by:



Professor Michael A. Demetriou, Advisor

Aerospace Engineering Program

WPI

Certain materials are included under the fair use Exemption of the U.S. Copyright Law and have been prepared according to the fair use guidelines and are restricted from further use

Abstract

The goal of this project was to design an experiment to detect a plume of carbon dioxide in a controlled environment using a gas-sensing mobile robot. Research was conducted in the areas of differential-drive robotics, navigation, estimation, and structural analysis. The Khepera IV robot was equipped with four gas concentration sensors and a wind velocity sensor for plume detection and robot guidance. An experimental mount was designed and built to hold the concentration and wind sensors with specifications given from plume estimation simulations. A navigation program was developed using an Extended Kalman Filter to estimate the state of the robot during the plume detection experiment. Testing was conducted on each subsystem to confirm that the experiment was feasible for plume detection.

Acknowledgements

We would like to thank the following individuals and organizations for the support they provided our team. Without them, we would not have been able to achieve what we have during our time working on this project. Our advisor Professors Michael Demetriou have helped guide us through the project process every step of the way. We worked alongside another MQP group advised by Professor Nikolas Gatsonis, Michael Barney and Stephanie Rivard, during the late stages of the project. Ioannis Skourtis, a summer research student who worked with us in the early stages of the project. Randy Robinson, the Computer Operations Manager for Higgins Laboratory who assisted us with any of our computer hardware needs. Barbara Furhman, an Administrative Assistant in the Mechanical Engineering Department assisted us in the purchasing of all necessary materials for our project. Donna Hughes, an Administrative Assistant for the Aerospace Department who aided with scheduling and room reservations. The Washburn machine shop for allowing us to use the CNC machines for the construction of each aluminum part. The WPI Rapid Prototyping Laboratory for the production of all plastic parts. Lastly, we would like to thank the entire Worcester Polytechnic Institute Aerospace Faculty. Over the past four years, they have taught us the material necessary to work on the project at hand, and gave us feedback and asked questions during our weekly department wide meetings.

Authorship

Abstract	Christie, Colfer
Acknowledgements	Christie
Authorship	Colfer
Table of Contents	Christie, Colfer
Table of Figures	Christie, Colfer
List of Tables	Christie, Colfer
1 Introduction.....	Christie, Colfer
1.1 Overview	Christie, Colfer
1.2 Research	Christie, Colfer
1.3 Goals	Christie, Colfer
2 Literature Review.....	Christie, Colfer
2.1 Khepera IV Robot.....	Christie, Colfer
2.2 CO2 Meter COZIR Gas Sensors	Christie
2.3 Wind Sensor	Christie
2.4 Arduino Uno Rev 3	Christie
2.5 Power Requirements	Christie
2.6 State Estimation.....	Colfer
2.6.1 Discrete – Time Kinematic Equations of Motion.....	Colfer
2.6.2 Dynamic Model Representation	Colfer
2.6.3 Measurement Model Representation	Colfer
2.6.4 Linearization.....	Colfer
2.7 The Extended Kalman Filter	Colfer
2.8 Control	Colfer
3 Goals, Objectives, & Approach	Christie, Colfer
3.1 New Exterior Chassis Proposal	Christie
3.2 Transmitting Sensor Data to the Base Station.....	Christie, Colfer
3.3 Calibration of Sensors.....	Christie
3.4 State Estimation.....	Colfer
3.5 State Estimate Simulations.....	Colfer
3.5.1 Numerical Simulation in MATLAB	Colfer

3.5.2 Path Following Navigation on the Khepera IV	Colfer
4 Results	Christie, Colfer
4.1 Experimental Sensor Mount Design	Christie
4.1.1 Sensor Mount Design.....	Christie
4.1.2 Raised Structural Support	Christie
4.1.3 Component Holding Shelf.....	Christie
4.2 Symbolic Block Diagram.....	Colfer
5 Experiment Outcomes	Christie, Colfer
5.1 Experimental Sensor Mount Assembly	Christie
5.2 Extended Kalman Filter Testing.....	Colfer
6 Recommendations	Christie, Colfer
7 Conclusions.....	Christie, Colfer
8 References	Christie, Colfer
Appendix A: Chosen Khepera Mount Stress and Strain Tests	Christie, Colfer
Appendix B: Weight of Chosen Mount with Different Materials	Christie, Colfer
Appendix C: Frequency Analysis Data for Experimental Mount Assembly	Christie, Colfer
Appendix D: Bill of Materials for Project.....	Christie, Colfer
Appendix E: Technical Drawings, Generated Using SOLIDWORKS	Christie, Colfer

All members participated in the revising and editing of all chapters of the report.

All photos are by the authors unless otherwise noted.

Table of Contents

Abstract	2
Acknowledgements	3
Authorship	4
Table of Contents	6
Table of Figures	8
List of Tables	9
1 Introduction.....	10
1.1 Overview	10
1.2 Research	10
1.3 Goals	12
2 Literature Review	13
2.1 Khepera IV Robot.....	13
2.2 CO2 Meter COZIR Gas Sensors	15
2.3 Wind Sensor	15
2.4 Arduino Uno Rev 3.....	16
2.5 Power Requirements	17
2.6 State Estimation	17
2.6.1 Discrete – Time Kinematic Equations of Motion.....	18
2.6.2 Dynamic Model Representation	21
2.6.3 Measurement Model Representation	22
2.6.4 Linearization.....	23
2.7 The Extended Kalman Filter	26
2.8 Control	28
3 Goals, Objectives, & Approach	29
3.1 New Exterior Chassis Proposal	29
3.2 Transmitting Sensor Data to the Base Station.....	29
3.3 Calibration of Sensors.....	30
3.4 State Estimation.....	30
3.5 State Estimate Simulations.....	31
3.5.1 Numerical Simulation in MATLAB	31

3.5.2 Path Following Navigation on the Khepera IV	31
4 Results	33
4.1 Experimental Sensor Mount Design	33
4.1.1 Sensor Mount Design.....	33
4.1.2 Raised Structural Support	35
4.1.3 Component Holding Shelf.....	36
4.2 Symbolic Block Diagram.....	37
5 Experiment Outcomes	39
5.1 Experimental Sensor Mount Assembly	39
5.2 Extended Kalman Filter Testing.....	40
6 Recommendations	42
7 Conclusions.....	44
8 References	45
Appendix A: Chosen Khepera Mount Stress and Strain Tests.....	46
Appendix B: Weight of Chosen Mount with Different Materials	49
Appendix C: Frequency Analysis Data for Experimental Mount Assembly	51
Appendix D: Bill of Materials for Project.....	57
Appendix E: Technical Drawings, Generated Using SOLIDWORKS	59

Table of Figures

FIGURE 1. KHEPERA IV MOBILE ROBOT [1]. COPYRIGHT © 2015 K-TEAM S.A.	14
FIGURE 2: KORE IO EXTENSION BOARD [1]. COPYRIGHT © 2015 K-TEAM S.A.	14
FIGURE 3: COZIR AMBIENT 10K CO2 SENSOR [7]. COPYRIGHT © 2015 CO2 METER.....	15
FIGURE 4: MODERN DEVICE WIND SENSOR REV P [10]. COPYRIGHT © 2015 MODERN DEVICE. ...	16
FIGURE 5: ARDUINO UNO REV 3 [9]. COPYRIGHT © 2015 ARDUINO INC.	16
FIGURE 6: DEGREES OF FREEDOM ON THE KHEPERA IV ROBOT [1]. COPYRIGHT © 2015 K-TEAM S.A.....	19
FIGURE 7 SYSTEM BLOCK DIAGRAM FOR PLUME DETECTION EXPERIMENT	28
FIGURE 8. PREVIOUS PROJECT'S ACRYLIC MOUNT DESIGN [3].....	29
FIGURE 9: ALUMINUM 6061 ALLOY FLAT SHEET DESIGN.....	34
FIGURE 10: ALUMINUM 6061 ALLOY WELD DESIGN.	35
FIGURE 11: ROD STABILIZER (LEFT), 0.25 M ROD (RIGHT).	36
FIGURE 12: ELECTRONIC SHELF WITH ROD ACCESSIBLE MOUNTING HOLES.....	37
FIGURE 13: SYMBOLIC BLOCK DIAGRAM FOR EXPERIMENTAL SETUP WITH STATE ESTIMATION ON THE KHEPERA IV.	38
FIGURE 14: FINAL ASSEMBLY.....	40
FIGURE 15: TRUE PATH VS ESTIMATED PATH, EXTENDED KALMAN FILTER SIMULATION	41
FIGURE 16: CROSS MOUNT, STRESS TEST	46
FIGURE 17: CROSS MOUNT, STRAIN TEST	47
FIGURE 18: CROSS MOUNT, DISPLACEMENT TEST.....	48
FIGURE 19: CROSS MOUNT MASS PROPERTIES, ABS	49
FIGURE 20: CROSS MOUNT MASS PROPERTIES, ACRYLIC	49
FIGURE 21 CROSS MOUNT MASS PROPERTIES, ALUMINUM 6061 – T6	50
FIGURE 22: EXTENDER ROD, 48.3 CM	59
FIGURE 23: EXTENDER ROD, 25 CM	60
FIGURE 24: COMPONENT SHELF.....	61
FIGURE 25: SENSOR-HOLDING CROSS MOUNT	62
FIGURE 26: EXTENDER ROD STABILIZER.....	63
FIGURE 27: WIND SENSOR HOLDER.....	64

List of Tables

TABLE 1: REQUIRED VOLTAGE AND CURRENT DRAW OF EACH SENSOR AND COMPONENT.....	17
TABLE 2: FREQUENCY ANALYSIS OF EXPERIMENTAL MOUNT ASSEMBLY, MODE 1	51
TABLE 3: FREQUENCY ANALYSIS OF EXPERIMENTAL MOUNT ASSEMBLY, MODE 2	52
TABLE 4: FREQUENCY ANALYSIS OF EXPERIMENTAL MOUNT ASSEMBLY, MODE 3	53
TABLE 5: FREQUENCY ANALYSIS OF EXPERIMENTAL MOUNT ASSEMBLY, MODE 4	54
TABLE 6: FREQUENCY ANALYSIS OF EXPERIMENTAL MOUNT ASSEMBLY, MODE 5	55
TABLE 7: MEASUREMENTS FOR EACH FREQUENCY NUMBER OF THE EXPERIMENTAL MOUNT ASSEMBLY	56
TABLE 8: FREQUENCY AND DISPLACEMENT VALUES FOR EACH MODE OF THE EXPERIMENTAL MOUNT ASSEMBLY	56
TABLE 9: MAD 1601 BILL OF MATERIALS.....	57

1 Introduction

1.1 Overview

The detection of hazardous gas in an air space using various methods of air concentration sensing equipment is a critical step in the prevention of harm caused by the gas. One prospective method for detecting harmful gas plumes that WPI has devoted resources to is the development of sensing autonomous vehicles (SAVs). The SAVs can track a plume's position and source and create a projection of the gas plume's size, shape, makeup, and heading within the air space autonomously. This data generated from the gas plume estimation is delivered back to a fixed base station to determine further actions and potential damages to the area involved. The objective of this Major Qualifying Project (MQP) is to design a ground-based experiment to test the plume estimation software. The experiment will utilize a ground based vehicle equipped with olfactory gas sensors and a wind velocity sensor to localize a controlled gas plume within a predefined space. The MQP is divided into two sub groups in order to perform the experiments on the SAVs properly. One sub group of the MQP is The Design of a Plume Generation and Detection (NAG 1602) which focuses on the construction of a plume generation unit for controlled and measurable releases of a gas source. This group is focused on the Plume Detection Using a Gas Sensing Mobile Robot, which develops the computer program for the mobile ground robot to track and locate the gas plume. Along with the program, the group will create an external chassis for proper sensor set up.

1.2 Research

This study is a continuation of an ongoing project; as such, we draw heavily from the work done by a previous MQP team. The team of Anglin, Hunt, and Myles conducted a study of the necessary elements in making the plume detection experiment work [3]. It was this team that purchased the Khepera IV robot and its accessories, and developed some preliminary equations modeling the robot's motion. The Khepera IV User's Manual is provided much of the technical information on the robot. Academic journal articles focused on topics such as differential drive motion and Extended Kalman Filters formed the rest of this background.

The MQP began by reviewing the material regarding this project from the previous project by WPI students. The project report from the previous year focused on the selection of a mobile

ground robot for that fit the parameters, the selection of gas concentration and wind sensors, and an external chassis design. Along with the research within those fields, the project group developed some preliminary equations of motion, and created motion simulations within Simulink. Using the information from Anglin, Hunt, and Myles reports, the group continued to research literature regarding robot dynamics and programming, and reevaluated the design of the external chassis. A brief summary of last year's report and two prominent experiments are given below

A study from Anglin, et. al. concluded that the project would need a small to medium sided mobile robot due to the area in which the experiments will be conducted [3]. The area in which the robot will be tested is an enclosed box of dimensions 9m by 9m with a height of 3m. The robot also needed to have the proper equipment to accommodate the required sensors and have wireless communication from the bay station. The three robots that were considered were the ATRV-Jr, the Koala Bot 2.5, and Khepera IV. The ATRV-Jr has dimensions of 77.5cm by 55cm and weighs 50kg. The ATRV-Jr has a maximum payload of 25kg, offers wireless communication, and a variety of compatible sensors for experimentation. The Koala Bot has dimensions 32cm by 32cm by 14.5cm and has similar qualities to the ATRV-Jr except for a max payload of 3.5kg. The mobile robot that was chosen was the Khepera IV model due to its small size and similar capabilities as the two previous robots, which the Khepera IV has a diameter of 14cm and a height of 5.8cm [3].

The next objective the previous group completed was to determine the correct gas sensor to be used on the Khepera IV and develop the proper arrangement of the chosen gas sensor on the external chassis. For the experiment, CO₂ was chosen as a gas and the COZIR Ambient 10K CO₂ sensor by CO₂ Meter was chosen by the counterpart MQP sub group Gas Plume Generation and Detection (NAG 1501). The gas sensors needed to be arranged along two primary axes of the robot and separated by two feet. Keeping these parameters in mind, the external chassis needed to extend each sensor one foot away from the body of the robot. The weight of the mount that would be placed on top of the robot was taken heavily into consideration. The Khepera IV has a max payload of 2kg; anything over that parameter would hinder the robot's mobility. The group concluded that making the mount of acrylic would be the best option, which the final mount design weighed 564.66 grams, falling within the range of the max payload of 2kg [3].

1.3 Goals

The goal of the project group is to program a mobile robot to detect, localize, and estimate a controlled gas plume within a pre-defined area. The robot is to be fitted with an external chassis (which does not hinder the motion of the robot) to accommodate the sensing equipment; we define this chassis as the experimental setup. The experimental setup must be capable of transmitting data from its five gas sensors and sending the data to the base station wirelessly. The robot must be able to receive commanded velocities to locate the source of the gas plume and to estimate its current state. Once these are completed, this project group will reconnect with our partner project group, The Design of a Plume Generation and Detection, and create the environment where experiments on the robots systems may occur. Within the environment, the robot and the plume generation unit will be tested for functionality.

2 Literature Review

2.1 Khepera IV Robot

The Khepera IV Robot was selected by Anglin, et. al. as the ideal robot platform to perform the gas sensing experiment [3]. Khepera IV was selected for its small size and maneuverability, as well as for the sizable processor power and many extension options. The team purchased two Khepera IV robots for the experiment [2].

Khepera IV (Figure 1) is a very maneuverable robot featuring two differential drive wheels (meaning the wheels are driven independently) and two caster wheels for stability. The drive wheels are driven by two DC motors that use Pulse Width Modulation (PWM) to achieve different speeds. The robot has eight infrared sensors mounted evenly spaced around the robot for short distance ranging, and five ultrasonic sensors facing toward the front for medium distance ranging.

The Khepera IV has a built in ST LSM330DLC three axis accelerometer and gyroscope [1]. The accelerometer can be used to measure the robot's x- and y-acceleration, and the gyroscope can measure the angular rate about the z-axis, which is the orientation of the robot. The standard deviation of the accelerometer is necessary to measure the distribution of error in the sensor reading. The spectral noise density (given from the ST LSM330DLC datasheet) of the accelerometer is $220 \mu\text{g}/\sqrt{\text{Hz}}$, and the normal mode operating bandwidth is given as 149 Hz [11]. The standard deviation of the sensor is therefore the noise density multiplied by the square root of the operating bandwidth and divided by g (9.81 m/s^2): $\sigma = 0.000274$ [12].

There are four mounting screw holes on the top of the robot, which can be attached to some of the Khepera IV peripheral devices or used to add custom hardware to the robot. The maximum load of the Khepera is 2000 grams (2kg); if the load on the robot exceeds the maximum value, then the robot will experience strain on the wheels, requiring more power to operate. The increase in power in the motors of the wheels will take power away from the robot's many sensors and drain its battery life at a rapid rate [1].

Khepera is equipped with a Gumstix Overo FireSTORM processor board. The processor is an ARM Cortex 8 with 800 MHz nominal clock speed and 512 MB NAND flash memory. The processor runs an embedded Linux operating system distribution called Angstrom. The robot is

capable of Bluetooth and Wi-Fi wireless communication. There are two extension connectors mounted on the top of the robot for I/O. Figure 1 shows the Khepera IV robot and its features [1].

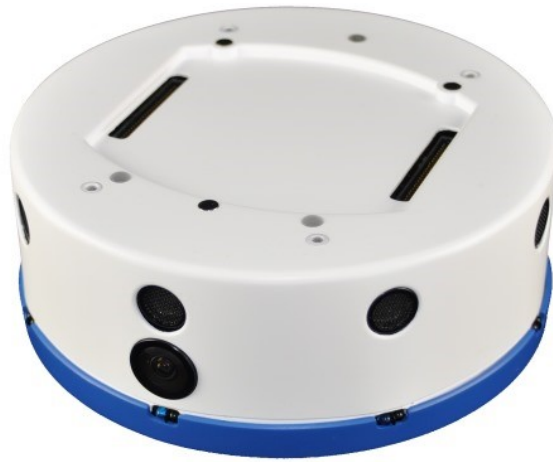


Figure 1. Khepera IV Mobile Robot [1]. Copyright © 2015 K-Team S.A.

The Kore IO extension board is an external accessory to the Khepera IV. The Kore IO is a bus that includes many I/O ports for extra inputs and communications to the Khepera IV. It can communicate with the Khepera IV through I2C and CAN serial communication methods. The Kore IO connects to the robot through the two extension connectors on the top of the Khepera IV. It is therefore important to that any external apparatus design onto the Khepera IV considers the size and shape of the Kore IO [1]. The Kore IO is pictured in Figure 2.



Figure 2: Kore IO Extension Board [1]. Copyright © 2015 K-Team S.A.

2.2 CO2 Meter COZIR Gas Sensors

The MQP team of Anglin, et. al. selected the gas sensor to be used in the experiment [3]. The COZIR gas sensor from the CO2 Meter company is a carbon dioxide sensor with a sensitivity of up to 10,000 parts-per-million (ppm). It is lightweight and has a low profile, so as to disturb the gas flow as little as possible. The power consumption of each sensor is around 3.5mW, and they have a measurement frequency of one measurement per 0.5s [7]. The experiment will require four of the COZIR sensors. The COZIR sensor is shown in Figure 3.



Figure 3: COZIR Ambient 10K CO2 Sensor [7]. Copyright © 2015 CO2 Meter.

2.3 Wind Sensor

One of the first tasks of this study was to determine a wind sensor that could operate at low wind velocities. The wind sensor chosen to record the wind velocity is the Modern Device Wind Sensor Rev. P developed by Modern Device (Figure 4). The wind sensor is a small and inexpensive anemometer that reads the change in resistance to calculate the wind speed. The Rev P can read wind speeds up to 150mph and sense the ambient temperature surrounding it. The voltage required to operate the sensor ranges from 8V to 12V with a current around 40mA depending on the wind speed [10].

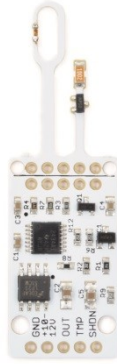


Figure 4: Modern Device Wind Sensor Rev P [10]. Copyright © 2015 Modern Device.

2.4 Arduino Uno Rev 3

The Arduino Uno Rev 3 is a microcontroller board that has 14 digital input/output pins, 6 analog inputs, and USB connection (Figure 5). The Arduino Uno can be powered via USB connection to a computer, a wall adapter, or a battery. The board is recommended to run at a voltage range of 7V to 12V; any higher or lower would create noise or possibly damage the board. The Arduino uses the external voltage to power electrical equipment attached to the board pins. The Arduino has a built in memory of 32 kilobytes and 2 kilobytes of ram. Communication between the Arduino and the computer can happen in a variety of ways. The connection can be established via wired USB or wireless communications like Wi-Fi or Bluetooth. To have the wireless communication, additional hardware needs to be connected and programmed to the board [9].



Figure 5: Arduino Uno Rev 3 [9]. Copyright © 2015 Arduino Inc.

2.5 Power Requirements

The electrical components of the experiment require a variety of voltages and currents which will require a unique set up for the power supply. The specifications of each component are giving in Table 1.

Table 1: Required Voltage and Current Draw of Each Sensor and Component.

Components	Required Voltage (V)	Current Draw (mA)
Arduino Uno Rev 3 [9]	7.0 – 12.0	46.5
COZIR CO2 Sensors (x4) [7]	3.3 (13.2)	1.5 – 33.0
Wind Sensor [10]	8.0 – 12.0	40.0
Total:	30.1 - 40.8	102 - 133.5

The sensors above were chosen for use within the experiment due to a variety of characteristics. The primary characteristic for both the concentration and wind sensor is their small size and functionality. There are, however, long measurement times associated with the COZIR CO2 sensor, which places constraints on the experiment. The COZIR CO2 receives a gas concentration reading every 0.5 seconds [7]. When four COZIR sensors operate at the same time on the same system, the full CO2 concentration reading can take as long as 2 seconds. This is extremely long time period for the base station to wait to receive data that contributes to the robots movement and will probably produce a large amount on noise for the plume estimation. The Wind Sensor outputs an analog voltage [10], which must be converted to digital values using an Analog-to-Digital Converter (ADC) on the Arduino Uno microprocessor [9].

2.6 State Estimation

An analysis of the motion of the Khepera IV robot is detailed in this section. In describing the motion of the robot, two models are defined: the dynamic model and the measurement model. The dynamic model is represented in state-space and is derived from the kinematic equations of motion for this robot. The measurement model deals with measurements of the robot's position from sensors on the robot body. Both methods are defined in the following section.

2.6.1 Discrete – Time Kinematic Equations of Motion

The kinematic equations of motion for the system are defined first. Figure 6 shows the degrees of freedom for the Khepera IV robot. This diagram shows a body-fixed (x, y, z) axis attached to the center of mass of the robot. The robot moves with a velocity $v(t)$ in the (x, y) plane, and rotates in the (x, y) plane with angular velocity $\omega(t)$. $\dot{x}(t)$ is defined as the component of the velocity along the x -axis, and $\dot{y}(t)$ is the component of the velocity along the y -axis. $\theta(t)$ is defined as the angle between the velocity vector and the x -axis. Equations (1), (2), and (3) are the continuous-time kinematic equations of the robot.

$$\dot{x}(t) = v(t) * \cos(\theta(t)) \quad (1)$$

$$\dot{y}(t) = v(t) * \sin(\theta(t)) \quad (2)$$

$$\dot{\theta}(t) = \omega(t) \quad (3)$$

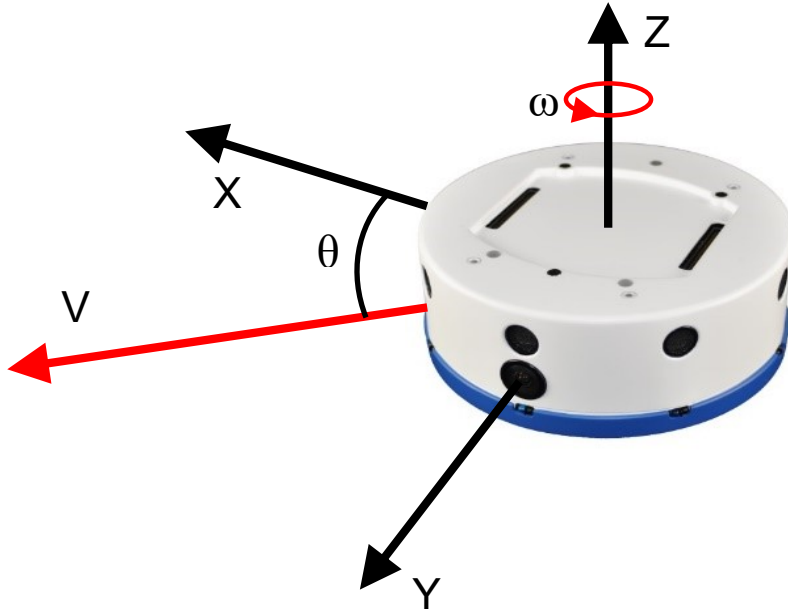


Figure 6: Degrees of Freedom on the Khepera IV Robot [1]. Copyright © 2015 K-Team S.A.

From the continuous-time equations, the approximate discrete-time equations of motion were obtained. Discretization is the process of estimating a continuous-time phenomenon with relation to discrete-time values. This aids in computational procedure by evaluating a set of equations at discrete time instances, thereby allowing a processor to evaluate the state of the robot as an iterative process. Time interval T is defined as the time between iterations. The continuous time t is substituted for a discrete time-step k . Equation 4) shows the relation between t , T , and k , and Equations (5), (6), and (7) are the discrete-time kinematic equations. Time k represents the previous time-step, and time $k + 1$ represents the next time step, approximated from the continuous-time equations.

$$t = k * T \quad (4)$$

$$\dot{x}(t) \approx v(t) * \cos(\theta(t)) \approx \frac{x(k + 1) - x(k)}{T} \quad (5)$$

$$\dot{y}(t) \approx v(t) * \sin(\theta(t)) \approx \frac{y(k+1) - y(k)}{T} \quad (6)$$

$$\dot{\theta}(t) \approx \omega(t) \approx \frac{\theta(k+1) - \theta(k)}{T} \quad (7)$$

The goal of deriving the discrete-time kinematic equations is to determine the position of the robot at the “next” time step; therefore, time $k+1$ is the time of interest in this investigation. Equation (8) approximates the orientation at time t as the orientation at time $k+1$. Equations (5), (6), and (7) are solved at time $k+1$ to find the position and orientation of the robot at time $k+1$ (Equations (9), (10), (11)).

$$\theta(k+1) \approx \theta(t) \quad (8)$$

$$x(k+1) = x(k) + v(k+1) * T * \cos(\theta(k+1)) \quad (9)$$

$$y(k+1) = y(k) + v(k+1) * T * \sin(\theta(k+1)) \quad (10)$$

$$\theta(k+1) = \theta(k) + \omega(k+1) * T \quad (11)$$

At this step, the velocity and angular velocity are unknown quantities in the discrete-time equations. Velocity and angular rate can be determined using the robot’s geometry. The value d is the length between the robot’s two wheels, and v_R , and v_L are the right and left wheel velocities. $v(k+1)$ is defined as the average of the two wheel velocities, and $\omega(k+1)$ is defined as the sum of the wheel velocities divided by d . Next, the translational and angular displacements are defined as the velocities divided by T . Solving for the displacements $\Delta D(k+1)$ and $\Delta \theta(k+1)$ gives Equations (12) and (13).

$$\Delta D(k+1) = v(k+1) * T = \left(\frac{v_R(k+1) + v_L(k+1)}{2} \right) * T \quad (12)$$

$$\Delta \theta(k+1) = \omega(k+1) * T = \left(\frac{v_R(k+1) - v_L(k+1)}{d} \right) * T \quad (13)$$

Finally, Equations (12) and (13) are substituted into Equations (9), (10), and (11), to form the final discrete-time kinematic equations of the robot at time $k+1$ (Equations (14), (15), (16)).

$$x(k+1) = x(k) + \Delta D(k) * \cos(\theta(k+1)) \quad (14)$$

$$y(k+1) = y(k) + \Delta D(k) * \sin(\theta(k+1)) \quad (15)$$

$$\theta(k+1) = \theta(k) + \Delta \theta(k) \quad (16)$$

2.6.2 Dynamic Model Representation

In order to accurately model these equations, the state-space representation is used. The three discrete time kinematic Equations (14), (15), and (16) completely describe the motion of the robot. Therefore, we can use $x(k)$, $y(k)$, and $\theta(k)$ as states in our model. The translational displacement ($\Delta D(k)$, Equation (12)) and angular displacement ($\Delta \theta(k)$, Equation (13)) are functions of the wheel velocities, v_L and v_R ; therefore, Equations (12) and (13) represent the control inputs of the dynamic system. The states and the control inputs are shown in vector form in Equations (17) and (18).

$$\mathbf{X}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} \quad (17)$$

$$u(k) = \begin{bmatrix} \Delta D(k) \\ \Delta \theta(k) \end{bmatrix} \quad (18)$$

The state-space representation can be used to determine the dynamic model of the Khepera IV system. The equations of motion are nonlinear; therefore, a nonlinear dynamic model is defined in Equations (19) and (20), where the next state, $\mathbf{X}(k+1)$, is a function of the current state, $\mathbf{X}(k)$, and the inputs, $u(k)$. The model accounts for process noise $n(k)$ added to the system.

$$\mathbf{X}(k+1) = f(\mathbf{X}(k), u(k)) + n(k) \quad (19)$$

$$f(\mathbf{X}(k), u(k)) = \begin{bmatrix} x(k) + \Delta D(k) * \cos(\theta(k) + \Delta \theta(k)) \\ y(k) + \Delta D(k) * \sin(\theta(k) + \Delta \theta(k)) \\ \theta(k) + \Delta \theta(k) \end{bmatrix} \quad (20)$$

2.6.3 Measurement Model Representation

The measurement model is based on the measurements taken from the three-axis accelerometer and three-axis gyroscope on the robot. Both sensors measure accelerations and angular rates in three coordinate axes. Only three measurements (from both sensors) were necessary for the experiment: acceleration in the x- and y-directions (a_x , a_y) and the angular rate in the xy-plane (ω) (see Figure 6 for a diagram of the body-fixed coordinate axes). It is therefore necessary to convert these measurements into the same form as the states in Equation (11). This is done using a “pseudo-integration” technique, whereby the x- and y- position are divided by T^2 , and the angular rate is divided by T . This “pseudo-integration” is shown in Equations (21), (22), and (23), and in matrix form in Equation (24).

$$a_x(k+1) = \frac{x(k+1)}{T^2} \quad (21)$$

$$a_y(k+1) = \frac{y(k+1)}{T^2} \quad (22)$$

$$\omega(k+1) = \frac{\theta(k+1)}{T} \quad (23)$$

$$h(\mathbf{X}(k+1)) = \begin{bmatrix} a_x(k+1) \\ a_y(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} T^{-2} & 0 & 0 \\ 0 & T^{-2} & 0 \\ 0 & 0 & T^{-1} \end{bmatrix} \begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} \quad (24)$$

The nonlinear measurement model is shown in Equation (25). The nonlinear measurement model contains the function of the states at the next time step, denoted as $h(\mathbf{X}(k+1|k))$, which is a nonlinear function. The measurement model also accounts for noise, denoted as $w(k+1)$.

$$z(k+1) = h(\mathbf{X}(k+1|k)) + w(k+1) \quad (25)$$

2.6.4 Linearization

The next step in obtaining the final state estimate of the robot is to linearize the nonlinear functions in the dynamic model and the measurement model. Linearization involves the use of Jacobian matrices, which calculate the partial derivatives of the function with respect to each variable. In this section, three linearization are made: the state matrix, the input matrix, and the measurement matrix, in that order. Equations (26) and (27) describe the Jacobian of the state matrix, and Equation (28) is the linearized state matrix.

$$\nabla_{x,y,\theta} f(\mathbf{X}(k)) = \begin{bmatrix} \frac{\partial f(k)}{\partial x} & \frac{\partial f(k)}{\partial y} & \frac{\partial f(k)}{\partial \theta} \end{bmatrix} \quad (26)$$

$$\nabla_{x,y,\theta} f(\mathbf{X}(k)) = \begin{bmatrix} \frac{\partial x(k)}{\partial x} & \frac{\partial x(k)}{\partial y} & \frac{\partial x(k)}{\partial \theta} \\ \frac{\partial y(k)}{\partial x} & \frac{\partial y(k)}{\partial y} & \frac{\partial y(k)}{\partial \theta} \\ \frac{\partial \theta(k)}{\partial x} & \frac{\partial \theta(k)}{\partial y} & \frac{\partial \theta(k)}{\partial \theta} \end{bmatrix} \quad (27)$$

$$\nabla_{x,y,\theta} f(\mathbf{X}(k)) = \begin{bmatrix} 1 & 0 & -\Delta D * \sin(\theta(k) + \Delta\theta(k)) \\ 0 & 1 & \Delta D * \cos(\theta(k) + \Delta\theta(k)) \\ 0 & 0 & 1 \end{bmatrix} \quad (28)$$

Next, Equations (29) and (30) describe the Jacobian of the input matrix, and Equation (31) is the linearized state matrix.

$$\nabla_{\Delta\theta, \Delta D} f(\mathbf{X}(k), u(k)) = \begin{bmatrix} \frac{\partial f(k)}{\partial \Delta\theta} & \frac{\partial f(k)}{\partial \Delta D} \end{bmatrix} \quad (29)$$

$$\nabla_{\Delta\theta, \Delta D} f(\mathbf{X}(k), u(k)) = \begin{bmatrix} \frac{\partial x(k)}{\partial \Delta\theta} & \frac{\partial x(k)}{\partial \Delta D} \\ \frac{\partial y(k)}{\partial \Delta\theta} & \frac{\partial y(k)}{\partial \Delta D} \\ \frac{\partial \theta(k)}{\partial \Delta\theta} & \frac{\partial \theta(k)}{\partial \Delta D} \end{bmatrix} \quad (30)$$

$$\nabla_{\Delta\theta, \Delta D} f(\mathbf{X}(k), u(k)) = \begin{bmatrix} -\Delta D * \sin(\theta(k) + \Delta\theta(k)) & \cos(\theta(k) + \Delta\theta(k)) \\ \Delta D * \cos(\theta(k) + \Delta\theta(k)) & \sin(\theta(k) + \Delta\theta(k)) \\ 1 & 0 \end{bmatrix} \quad (31)$$

Finally, Equations (32) and (33) describe the Jacobian of the measurement matrix, and Equation (34) is the linearized state matrix.

$$\nabla_{x,y,\theta} h(\mathbf{X}(k+1|k)) = \begin{bmatrix} \frac{\partial h(k+1)}{\partial x} & \frac{\partial h(k+1)}{\partial y} & \frac{\partial h(k+1)}{\partial \theta} \end{bmatrix} \quad (32)$$

$$\nabla_{x,y,\theta} h(\mathbf{X}(k+1|k)) = \begin{bmatrix} \frac{\partial x(k+1)}{\partial x} & \frac{\partial x(k+1)}{\partial y} & \frac{\partial x(k+1)}{\partial \theta} \\ \frac{\partial y(k+1)}{\partial x} & \frac{\partial y(k+1)}{\partial y} & \frac{\partial y(k+1)}{\partial \theta} \\ \frac{\partial \theta(k+1)}{\partial x} & \frac{\partial \theta(k+1)}{\partial y} & \frac{\partial \theta(k+1)}{\partial \theta} \end{bmatrix} \quad (33)$$

$$\nabla_{x,y,\theta} h(\mathbf{X}(k+1|k)) = \begin{bmatrix} 1/T^2 & 0 & 0 \\ 0 & 1/T^2 & 0 \\ 0 & 0 & 1/T \end{bmatrix} \quad (34)$$

Using the relations in Equations (28), (31), and (34), the linearized equations are redefined, and the dynamic model and measurement model are linearized. The linearized dynamic model in Equation (38) is a function of the states, the state matrix (Equation (35)), the inputs, the input matrix (Equation (36)), and the process noise. The linearized measurement model in Equation (39) is made up of the states, the measurement matrix (Equation (37)), and the measurement noise.

$$\nabla_{x,y,\theta} f(\mathbf{X}(k)) = \frac{df(k)}{d\mathbf{X}}|_{\mathbf{X}(k)} \equiv A(k) \quad (35)$$

$$\nabla_{\Delta\theta, \Delta D} f(\mathbf{X}(k), u(k)) = \frac{df(k)}{du}|_{\mathbf{X}(k), u(k)} \equiv B(k) \quad (36)$$

$$\nabla_{x,y,\theta} h(\mathbf{X}(k+1|k)) = \frac{dh(k+1)}{d\mathbf{X}}|_{\mathbf{X}(k+1)} \equiv C \quad (37)$$

$$\mathbf{X}(k+1) = A(k)\mathbf{X}(k) + B(k)u(k) + n(k) \quad (38)$$

$$z(k + 1) = CX(k + 1|k) + w(k + 1) \quad (39)$$

2.7 The Extended Kalman Filter

In order to autonomously drive any robot (including the Khepera IV), it must be able to accurately determine its position within an inertial frame of reference, a practice commonly referred to as navigation [8]. To accomplish this, the robot must be able to estimate its position and velocity while reducing noise. Noise is an inherent part of any dynamic system. Noise is a stochastic process, a term which refers to a collection of random variables. In this analysis, the case where noise is a Gaussian zero-mean variable is considered. This noise is a detriment to our system, because it adds uncertainty to our measurements over time, leading to errors in determining the state of the system.

A filter is an algorithm that removes a certain element from a signal in order to analyze another element of the signal. The Kalman Filter is a type of filter used to remove noise from linear time invariant (LTI) dynamic systems. The Kalman Filter is an optimal estimation filter which compares the dynamic model of a system with measurement inputs to determine a “best guess” estimation of the system state. In the case of a mobile robot moving on a level plain, three states are defined: x-position, y-position, and orientation (θ). By continually updating these states to reflect the current position and orientation of the robot, it will be able to navigate autonomously [8]. In order to use a Kalman Filter with the Khepera IV, the dynamic system must be defined in terms of the motion of the robot [4].

The motion of this robot cannot be described as a linear system. Therefore, an Extended Kalman Filter (EKF) must be used to estimate the system states. The Extended Kalman Filter expands upon the Kalman Filter with an added step to linearize the predicted state estimates. Due to approximations in linearization, the EKF is not an optimal estimation scheme, rather an approximation of one. Upon determining the linear dynamic model and linear measurement model, Equations (40)-(45) are used to determine the estimated state of the robot [4].

$$P(k + 1|k) = A(k)P(k|k)A^T(k) + Q(k) \quad (40)$$

$$Q(k) = B(k) \begin{bmatrix} \sigma_{\Delta\theta}^2 & 0 \\ 0 & \sigma_{\Delta D}^2 \end{bmatrix} B^T(k) \quad (41)$$

$$K(k + 1) = P(k + 1|k)C[CP(k + 1|k)C^T + R]^{-1} \quad (42)$$

$$R = \begin{bmatrix} \sigma_{ax}^2 & 0 & 0 \\ 0 & \sigma_{ax}^2 & 0 \\ 0 & 0 & \sigma_g^2 \end{bmatrix} \quad (43)$$

$$P(k + 1|k + 1) = \quad (44)$$

$$P(k + 1|k) - K(k + 1)(CP(k + 1|k)C^T + R)K(k + 1)^T$$

$$\begin{aligned} \hat{X}(k + 1|k + 1) \\ = \hat{X}(k + 1|k) + K(k + 1)[z(k + 1) - C\hat{X}(k + 1|k)] \end{aligned} \quad (45)$$

The Equations (40)-(44) form the covariance matrices, which describe how much “confidence” is placed in the dynamic model and the measurement model. Equation (40) is the Process Error Covariance Matrix, which determines the “confidence” dynamic model, based on the Process Noise Covariance Matrix in Equation (41). The Kalman Gain in Equation (42) is the overall “confidence” in the measurement value of this system, using Equations (40) and (43). Equation (43) is the Measurement Noise Covariance. The final state estimate is presented in Equation (45) [4].

2.8 Control

A control scheme for the robot was laid out by the study conducted by Anglin et. al [3]. This scheme takes input x and y point-mass velocity inputs from the plume algorithm to direct the robot to the area of highest gas concentration. These x and y velocities are transformed into a translational velocity (defined as v) and angular velocity (defined as ω), which are then sent to the robot wirelessly to create the left and right wheel velocities. The total v and ω are then input to the Kalman Filter (computed on the base station), which outputs the actual x and y positions, as well as orientation. The orientation angle is then fed back to the v and ω calculation, as a means of error correction (Anglin, Hunt, and Myles, 2015). See Figure 7 for a block diagram of the control scheme [3].

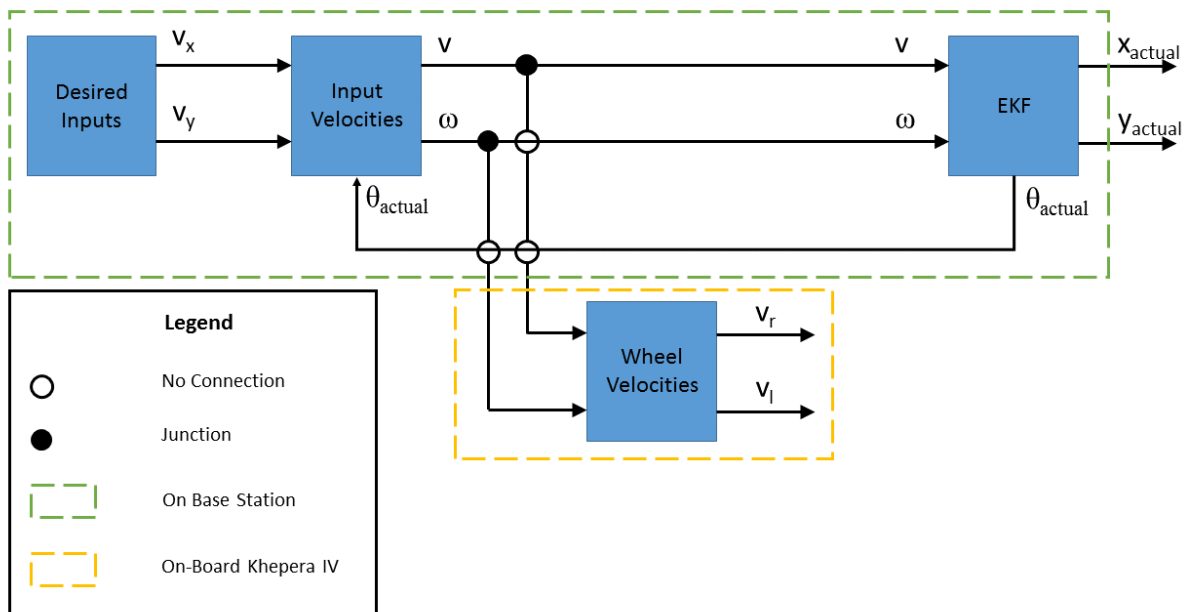


Figure 7 System Block Diagram for Plume Detection Experiment

3 Goals, Objectives, & Approach

3.1 New Exterior Chassis Proposal

One objective was to reevaluate the mount design concluded upon in by Anglin, et. al. [3]. One design restriction is that COZIR sensors, if along the same axis, need to be separated by two feet for proper data acquisition and minimal data interference. With this specification included within the design, the mount can be realized as a thin cross mounted on the Khepera IV. The mount needs to be elevated from the top of the robot by one inch to allow room for the Kore IO extension board (Figure 2). With all the design specifications of the mount listed above, the three mount designs are proposed. Figure 8 shows the experimental cross mount designed by Anglin, et. al. [3].

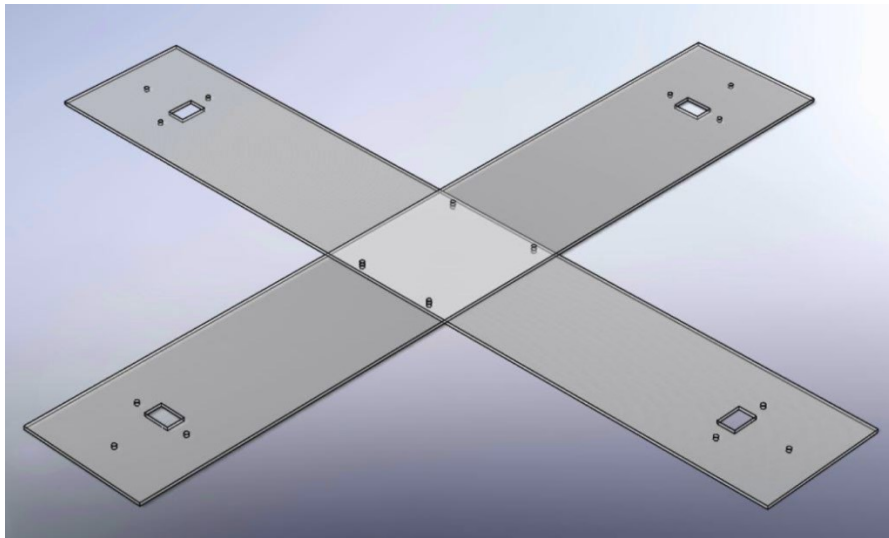


Figure 8. Previous Project's Acrylic Mount Design [3]

3.2 Transmitting Sensor Data to the Base Station

A critical step in the plume estimation process is the ability to transmit recorded data to the base station via wireless communication. The gas sensing robot needs to be able to conduct the data acquisition process in an open environment without being tethered to the base station. One option is to use the built in Wi-Fi and Bluetooth capabilities of the robot to transmit the data to the base station. This option allows for less of a physical load on the robot and a simple wiring

configuration for the sensors. Another option is to implement an intermediary device which will allow the transmission over Wi-Fi. This device needs to be properly integrated into the circuitry and not hinder the robots operations. The device would operate as a system separate from the robot that would collect all the data from the sensors, organize the outputs, and send the data to the base station. The system would comprise of a microcontroller to process all the data from the sensors and a wireless transceiver to send the data once the data has been organized. The microcontroller considered for this task is the Arduino Uno Rev 3 due to it being inexpensive, easy to program, and intuitive design [9].

3.3 Calibration of Sensors

To ensure accurate readings being sent to the base station, the sensors need to be properly calibrated before testing can begin. The Rev P Wind Sensor is calibrated by using a known wind velocity and correcting the constants in equations within its code to reflect the known velocity [10]. To calibrate the COZIR CO₂ sensor, the sensor is exposed to a known gas source, multiple readings are taken, and an average is calculated [7]. The difference between the new reading and the original reading when the sensor was originally calibrated at the factory is stored in the sensors memory. This offset value is then automatically added or subtracted to any subsequent readings taken by the sensor during use. The method of performing the calibration by exposing the sensor to single gas in a sealed environment is the most accurate form of calibration. This is due to knowing how much gas the COZIR sensor is exposed to within the defined area. Another method is to expose the sensor to the outdoor environment, which then assumes the concentration to be 400 parts per million (ppm). The COZIR sensor is previously programmed to 400ppm for outdoor ambient air for easier calibrations. This method is not as accurate as the previously described method but is cost effective due to the process of isolating the single gas properly [7].

3.4 State Estimation

In order to accurately navigate in the test environment, the Khepera IV must keep track of its state, (in this case, position and orientation). Therefore, one of our major goals is to implement

a state estimation algorithm for our Khepera IV. There are five objectives in estimating the state of the Khepera IV robot:

- Determining the kinematic equations of motion.
- Discretizing the kinematic equations.
- Creating the state-space dynamic model and measurement model of the discrete equations.
- Linearizing the dynamic model and measurement model.
- Building a Kalman Filter on top of the linearized equations.

Upon receiving a state estimate, we next then calculate the desired input wheel velocities for the robot. This forms the plant, or controller, of our system, in which the Khepera IV robot tries to match the velocities given from state estimation with actual wheel velocities. The next goal is to determine the correct controller equations to convert state estimates to wheel velocities.

3.5 State Estimate Simulations

3.5.1 Numerical Simulation in MATLAB

Once the state estimate and the controller were developed, the functionality of these constructs were tested. The first test was a numerical simulation, which was be done in MATLAB. The numerical simulation operated by defining a path for the simulated robot to follow. This path represented the true state of the system, theoretically unknown to the system. In order to create the system measurements, the true states were used to estimate the x- and y- accelerations and angular rate. This “measured” data was then corrupted with zero mean normal distribution random noise. The estimated states and measured values were then input to a Kalman Filter function, which output the true state of the system, as per Section 2.6. Finally, the MATLAB script plotted the true states against the estimated states output from the Extended Kalman Filter, thereby testing the accuracy of the filter.

3.5.2 Path Following Navigation on the Khepera IV

Once the numerical simulation is successful, we will test the state estimation procedure on the Khepera IV robot. This is done using a similar procedure, in which a path is defined for the

robot to follow, and the EKF must filter out the measurement and process noise that is present in the actual system. We will film robot from above as it travels along this path, and transpose the actual path (the true states) over the estimated path (the estimated states) and compare the two for accuracy. Having completed this physical simulation, our goal of determining the true states of the robot in a real system will be complete.

4 Results

4.1 Experimental Sensor Mount Design

As reported in Section 3.1, one objective was to reevaluate the mount design specified in the report by Anglin, et. al [3]. Three materials were evaluated: Aluminum 6061-T6, Acrylic, and Acrylonitrile Butadiene Styrene (ABS). Each material was analyzed in terms of weight, flexibility, and difficulty of manufacturing. As noted in Section 2.1, Khepera IV can only support a payload of 2000 grams and retain full motion capabilities [1]. An important quality in choosing the structure material is therefore total weight. After the weight consideration, flexibility was analyzed for its influence on sensor readings. As the robot moves from point A to point B, the sensor mount will be subjected to forces that would vibrate the mount, thereby adding unwanted noise to each sensor. The mount material was selected to be as inflexible as possible in order to reduce the forces while it moves, in turn reducing the noise in each sensor.

4.1.1 Sensor Mount Design

The proposed mount from the plume detection experiment is shown in Figure 9. The part was designed to be mounted on the Khepera IV mounting holes; the mounting holes on the part were dimensioned accordingly. Similarly, the gas sensors were to be placed at a distance of 10 cm apart from each other. The mounting holes for the gas sensors were located from the technical drawings from the COZIR manufacturer. The mount was designed to simplify the manufacturing process as much as possible. The mount was created from aluminum 6061-T6 alloy 0.125in thick sheet stock. The sheet was machined to the major dimensions, and then drilled the holes at the proper locations. See Appendix E for a technical drawing of the mount.

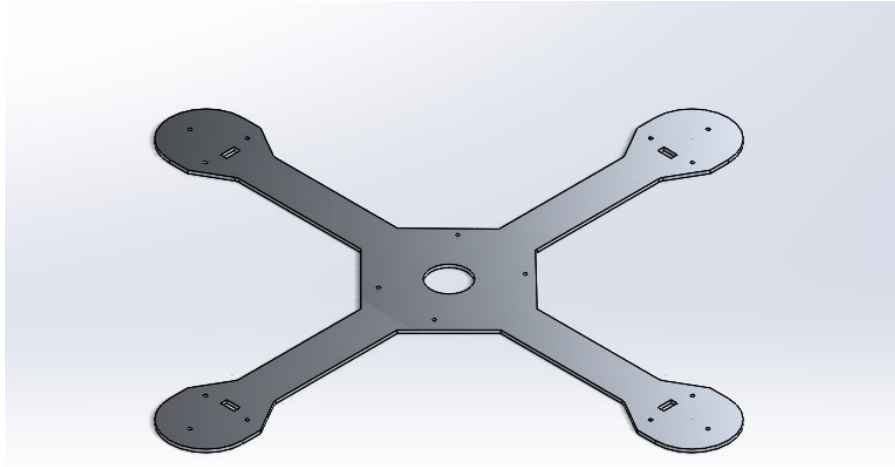


Figure 9: Aluminum 6061 Alloy Flat Sheet Design.

Another proposed mount design is presented in Figure 10. This design attempts to combine both functionality and ascetics to prevent interference within the sensors. The cylindrical tube would hide the wires from the COZIR gas sensors. This mount design has the highest rigidity of the proposed mounts but is the most difficult to manufacture. The mount design requires welding the plates and bars together. Welding with aluminum is extremely difficult at the scale being considered. The thickness of walls being welded range between 2mm and 7mm, which would melt the aluminum before a strong weld may be created. For this design to be feasible, an experienced welder would need to be hired to aid in this manufacturing process.

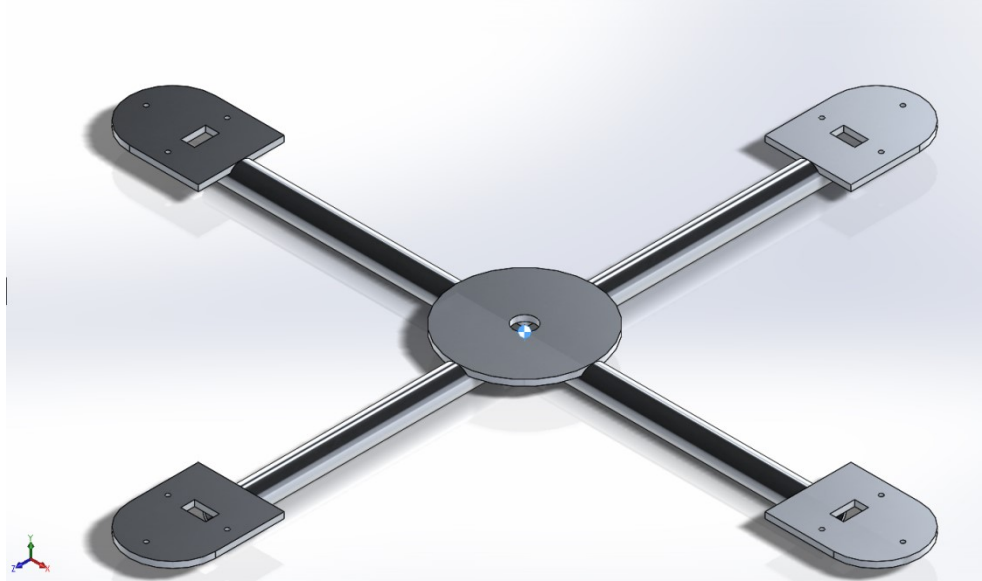


Figure 10: Aluminum 6061 Alloy Weld Design.

4.1.2 Raised Structural Support

An important parameter of the plume estimation experiment is that the gas concentration readings are measured from a set distance above the ground in order to account for disturbances in flow from the floor. Therefore, one of the requirements for the experimental mount was that the CO₂ sensors need to be a set distance from the floor. With this in mind, three mount options were considered: a mount raised one meter (1 m) from the floor, a mount raised 0.75 m from the floor, and a mount raised 0.5 m from the floor. To comply with these requirements, the cross fixture needed to be supported by extending rods. The extending rods (Figure 11) are made of aluminum 6061-T6 and were developed modularly. The rods are threaded and tapped so they may be screwed into the Khepera IV mounting holes, to another rod, or to a screw attached to the cross mount. For the heights of 75cm to 100cm, the rods require a connector stabilizer (Figure 11) that connect each rod to each other to be placed in between each rod. The stabilizer is used to increase the rigidity of the rods and whole unit as the height increases. This modular design is beneficial for the experiment because it allows plume detection to be tested at various heights which influences how the plume diffuses. However, the design does have drawbacks in overall stability. As the height increases, the unit becomes more prone to tipping over due to a higher center of mass. All technical drawings are included in Appendix E.

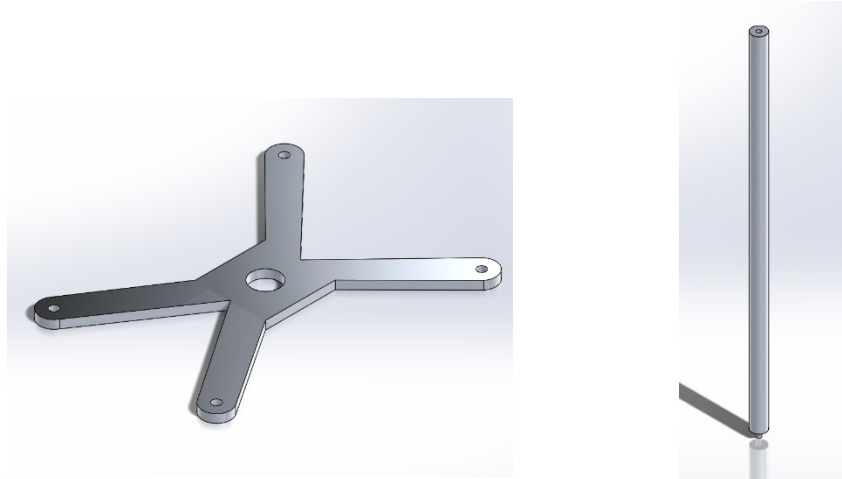


Figure 11: Rod Stabilizer (left), 0.25 m Rod (right).

4.1.3 Component Holding Shelf

Still required for the mount design is a holding unit for the various components used in the experiment. A “shelf” design was proposed to hold the Arduino Uno and sensor power supplies. The component shelf is not designed to be a part of the structure; therefore, material strength is not a consideration for material selection. The shelf is an auxiliary structure, and as such, low mass is important to avoid the weight limit of the Khepera IV. The shelf cannot be produced via CNC machining due to the complexity of the fixture. The only viable option is to produce the shelf by rapid prototyping. With these design considerations in mind, the shelf was made from ABS plastic due to a low material density (as compared to aluminum 6061-T6). The first proposed shelf unit has three levels that hold each component separately. Each level has a large hole in the center for wire accessibility from level to level, as well as holes to accommodate the support rods (above). Placing the rods through the component shelf will add structural rigidity to the experimental mount. The weight of the component shelf is 185.52 grams; the shelf can be seen in Figure 12.

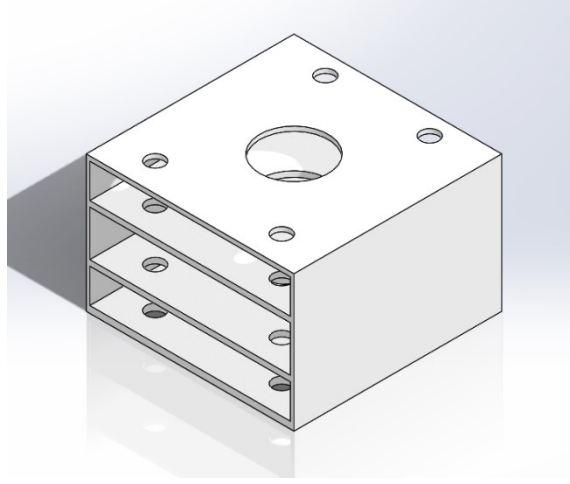


Figure 12: Electronic Shelf with Rod Accessible Mounting Holes.

4.2 Symbolic Block Diagram

The symbolic block diagram presented in this section correlates to the experimental setup presented in Section 3, Goals, Objectives, and Approach. In this configuration, the state estimation is done on board the Khepera IV processor. IMU data is sent to the Extended Kalman Filter algorithm internally. The state estimate is sent to the base station wirelessly, along with the gas sensor data, to be used by the plume estimation. The base station then transmits the desired robot velocities to the controller. The state estimate also is output to the controller to calculate wheel velocities, which are then sent to the wheel motors to move the robot. The system block diagram is presented in Figure 13.

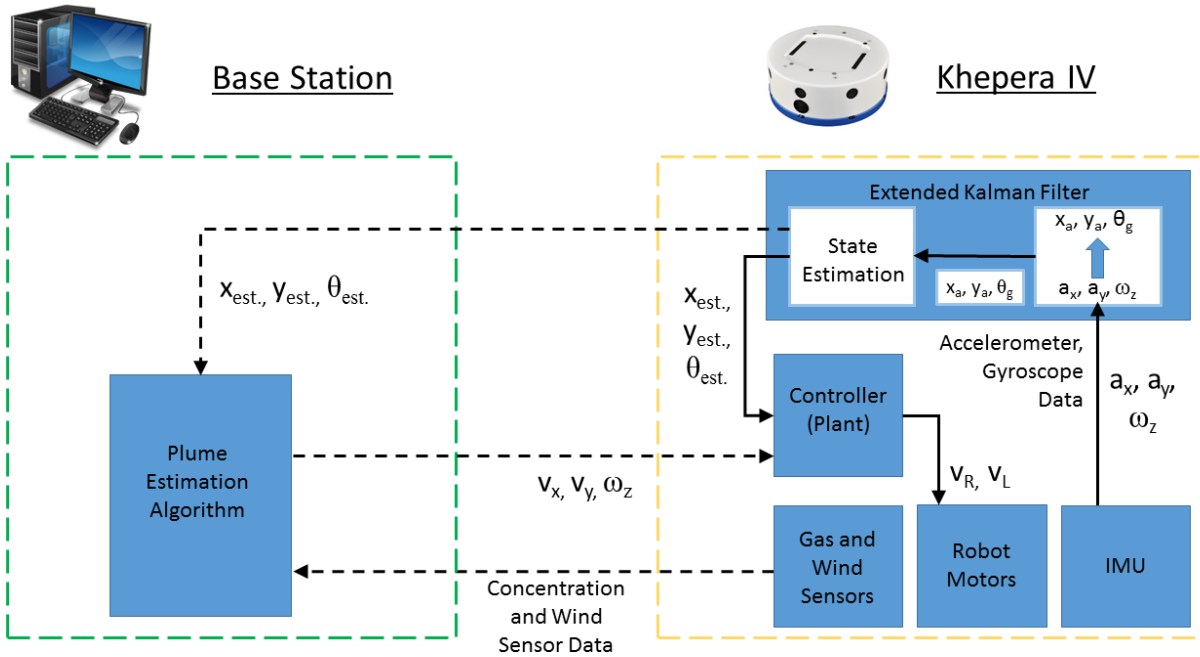


Figure 13: Symbolic Block Diagram for Experimental Setup with State Estimation on the Khepera IV.

5 Experiment Outcomes

5.1 Experimental Sensor Mount Assembly

Once all the parts were created and analyzed in SOLIDWORKS, the Aluminum 6061 alloy flat sheet design and the rod accessible shelf were chosen for the final assembly. Reviewing the designs the cross mount, the Aluminum 6061 Alloy Weld Design was not feasible due to its complexity. The weld mount would require a skilled welder to build this design due to the thin features on the model and the difficulty with welding aluminum. The acrylic mount was already developed but needed to be improved upon for accurate data acquisition. Acrylic is receptive to vibrations and forces at that thickness, which would cause the acrylic arms to bend and shake. This shaking would cause noise in the sensors and, in turn, cause inaccurate sensor readings. The flat plate design was chosen above the two other candidates due to the rigidity and ease of manufacturing. The flat plate is much less flexible when compared to the acrylic mount and was manufactured in a CNC machine which has the benefit of being repeatable. The component shelf was manufactured using rapid prototyping. The shelf adds support to the extender rods by acting as trusses between each rod. Finally, the center of mass of the unit is closer to the ground with the rod shelf compared to the wind sensor shelf. The wind sensor shelf would have been surmounted on top on the cross mount, which is 50cm to 100cm off the ground. The wind sensor shelf would make the center of mass closer to the top of the assembly, which causes greater instability within the Khepera IV as it travels. The rod shelf is closer to the ground, lowering the center of gravity and allow for better motion of the Khepera IV. The extender rods and rod stabilizers were designed and manufactured with modular height configurations so it may allow experiments to be conducted at different heights. Figure 14 shoes the assembled model of the experimental test mount.

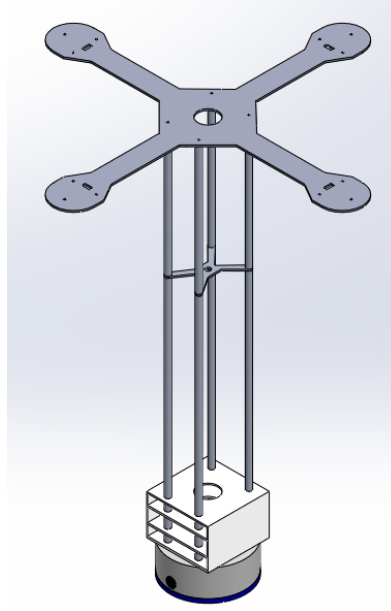


Figure 14: Final Assembly

5.2 Extended Kalman Filter Testing

The tests specified in Section 3.5 were performed on the robot. The first test, a numerical simulation in MATLAB, was performed as a proof of concept that the Extended Kalman Filter (EKF) equations developed in Sections 2.6 and 2.7 were effective in eliminating inherent sensor noise in a dynamic system. This experiment is detailed in Section 3.5.1. The test involved inputting the EKF equations and kinematic equations in MATLAB. The a parametric equation for a figure-eight path was defined for one thousand points from $-\pi$ to π , as specified in Equations (46) and (47):

$$x(t) = 2000 * \cos(t) + 2000 [mm] \quad (46)$$

$$y(t) = 2000 * \sin(2 * t) [mm] \quad (47)$$

These parametric equations represented the true state of the system; in a real system, the true states are unknown. Equations (46) and (47) also represented the desired states of the system, as they were used to calculate the wheel velocity inputs. The accelerations and angular rate data was generated from Equations (46) and (47). Approximate integration was performed by multiplying the true x-position and true y-position by $2/T^2$, and the true orientation (theta) by

1/T. Then, the acceleration and angular rate data was corrupted with noise according to a zero-mean normal distribution with standard deviations specified by the accelerometer manufacturer (Section 2.1). With the true states, desired states, and measured states developed, the estimated states were found by running the desired states and measured states through the EKF Equations (40) – (45). The simulation output a plot of the true path against the estimated path, as in Figure 15:

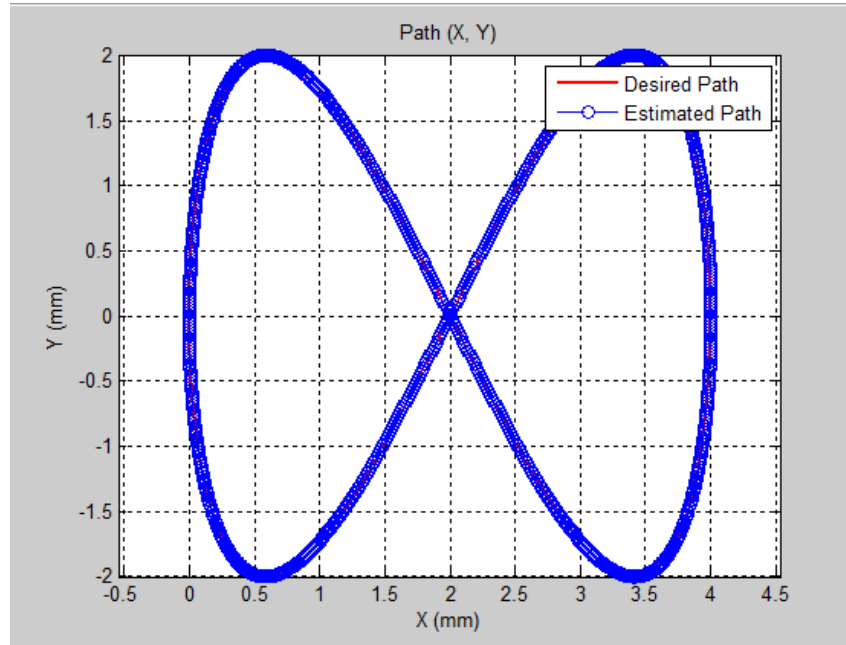


Figure 15: True Path vs Estimated Path, Extended Kalman Filter Simulation

The next task completed was to download the simulation equations generated in MATLAB to the robot processor to test on the actual system, as explained in Section 3.5.2. As the robot is programmed using C executable code, the MATLAB Coder Toolbox was used to transfer the complex matrix equations for the EKF to C. Once completed, the converted EKF code was downloaded to the Khepera IV processor. The path for this simulation was a simple 1 meter straight line. The desired inputs were input to the EKF function, which output the estimated states and used to calculate the noise-corrected wheel velocity inputs. These inputs were then given to the wheel velocity function (from K-Team) to drive the robot. This process was repeated recursively until the robot had traveled 1 meter. The test was performed successfully.

6 Recommendations

This section details recommendations for further research into the area of plume estimation using a gas sensing mobile robot. First, it is recommended that the group familiarize themselves with the Khepera IV commands, programming, and the base station. The Khepera IV comes with executable commands developed by the manufacturer, which are needed to control the many features of the Khepera IV. Programming the Khepera IV along with the sensors and base station will take a large amount of work due to the complexity of the program needed for proper code execution and coordination between each system. The base station uses the Ubuntu distribution of the Linux operating system. The base station is tasked to run MATLAB and various other programs for the localization of the plume to occur. Familiarity with MATLAB is also critical due to the fact that the Extended Kalman Filter developed with for this project was generated from a MATLAB script. Learning C or C++ is highly recommended, as this is what the Khepera IV is coded in and the Arduino uses a simplified C programming language. By familiarizing the group with these programming languages and systems before the beginning their research, future study groups should be well-equipped to handle the various functions of this experiment.

Second, the connection between each sensor, microcontroller and power supply is needed for the sensors to operate. The Arduino Uno board has the ability receive and organize the data from each sensor and output the data to a given communications device. However, the Arduino cannot supply power to all of the sensors; an external power source is required for this function. This power source or (sources) must power all the sensors, Arduino, and the Wi-Fi transceiver while remaining compact to fit on the exterior sensor unit.

Finally, future project groups should start working on the experiment portion of the project and be well informed about the plume generation procedures developed in parallel to this portion of the experiment. Once the information regarding the Khepera IV and the other systems are understood, the group can focus on developing tests in which demonstrate the robot's capabilities. If there is a plume generation group, meeting frequently with the counterpart group will keep both teams focused on the overall team goal as well as provide insightful information. As the project proceeds, the groups involved with plume localization with a gas sensing robot will eventually merge as the Khepera utilized its COZIR sensor to travel. The more time that is spent experimenting with the setup and trying new things, the more likely the experiment will be successful.

In summary, the recommendations for a further study in this research area are as follows:

1. Familiarize oneself with the programming languages of C, MATLAB and Khepera IV's onboard commands.
2. Develop code and wire connections for sensor, microcontroller, and power supply.
3. Start experiments as soon as possible.
4. Meet frequently with the plume generation research group.

7 Conclusions

This report covers the initial experimental trials conducted with the gathered information from the Anglin, et. al. report [3]. Upon reviewing the information, the group conducted research into experimentation for the Khepera IV Robot, a new design for the exterior sensing unit, and a wind sensor that complies with our design specifications and budget. The redesign for the exterior sensing unit is recommended due to satisfying the experimental parameters as well as the Wind Sensor Rev. P chosen be utilized. It was also determined that an external microcontroller should receive the reading from the gas and wind sensors. The Extended Kalman Filter worked as expected in two simple experiments, as detailed in Sections 5.2 and 5.3. The experimental sensor mount was manufactured to the design specifications detailed in Section 4.1.

Unfortunately, the experiment to integrate the sensors to the Khepera IV for gas localization could not be performed. The sensor configuration could not be determined due to the complexity in regulating the power supplies for each sensor. The program to localize the gas plume was not finalized before the end of the experiment, also attributing to the plume detection experiment not being conducted. For the experiment to run, the plume detection program needs to be fully functional, along with the sensors in proper configuration. Despite these drawbacks, much progress was made toward developing the plume detection experiment using a gas sensing mobile robot.

8 References

- [1] Tharin, J., Lambercy, F., & Carron, T. (2015). Khepera IV User Manual (Vol. 1.1, pp. 26-28). Vallorbe, Switzerland.
- [2] Dudek, G., & Jenkin, M. (2010). Computational Principles of Mobile Robotics (Second ed., pp. 40-43). New York, New York: Cambridge University Press.
- [3] Myles, M., Anglin, M., & Hunt, M. (2015). Gas Source Localization with a Mobile Sensing Ground Vehicle. Major Qualifying Project Report, (MAD 1501).
- [4] Otahal, T., & Tanner, H. (2008). An Extended Kalman Filter Implementation for the Khepera II Mobile Robot. The University of New Mexico, (ME-TR-08-001), 23.
- [5] Hellstrom, T. (2011). Kinematics Equations for Differential Drive and Articulated Steering. Department of Computing Science, UMINF-11.19 (ISSN-0348-0542).
- [6] Real World Interface. (2015) ATRV-Jr Mobile Robot Tech Sheet.
- [7] CO2meter.com: CO2 Measurement Specialists. (2015) COZIR Ambient 10K CO2 Sensor.
- [8] Cowlagi, R. V. (2015). Course Notes for AE 4733 Guidance, Navigation, and Communication. Worcester Polytechnic Institute.
- [9] Arduino.cc. (2015). Arduino Uno.
- [10] ModernDevice.com. (2015). Wind Sensor Rev P.
- [11] ST-LSM330DLC Accelerometer. (2015). Datasheet for the accelerometer on the Khepera IV.
- [12] Spectral Noise Density: A Useful Way to Look at Data Converter Performance for Software Defined Systems. (2014). Information from Analog Devices on Noise Density.

Appendix A: Chosen Khepera Mount Stress and Strain Tests

The generated stress and strain distributions data was produced using SOLIDWORKS simulations, primarily stress analysis simulation. This data is to analyze the displacement of the cross fixture when forces are applied to the top. In the examples in Figure 16, Figure 17, and Figure 18, only gravitational force is applied to the cross mount.

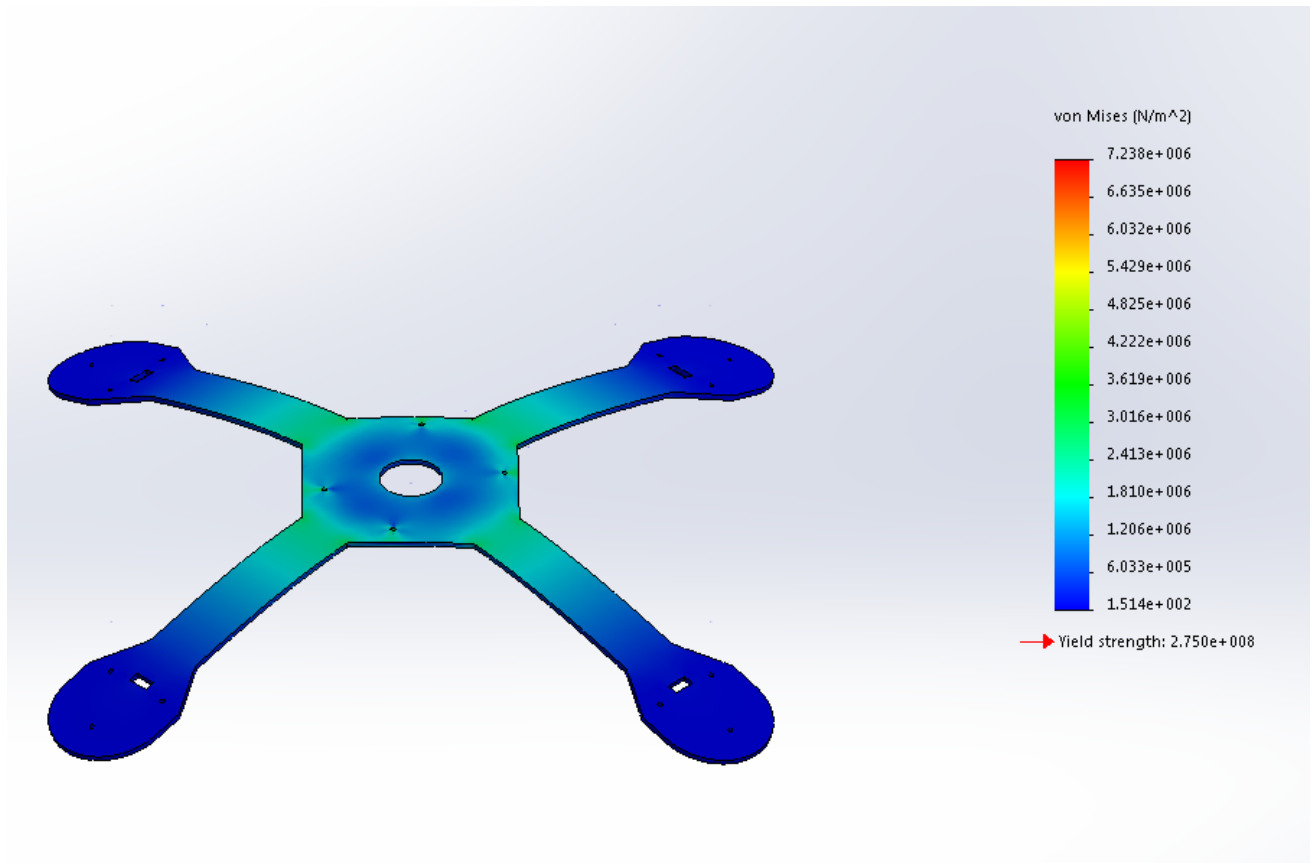


Figure 16: Cross Mount, Stress Test

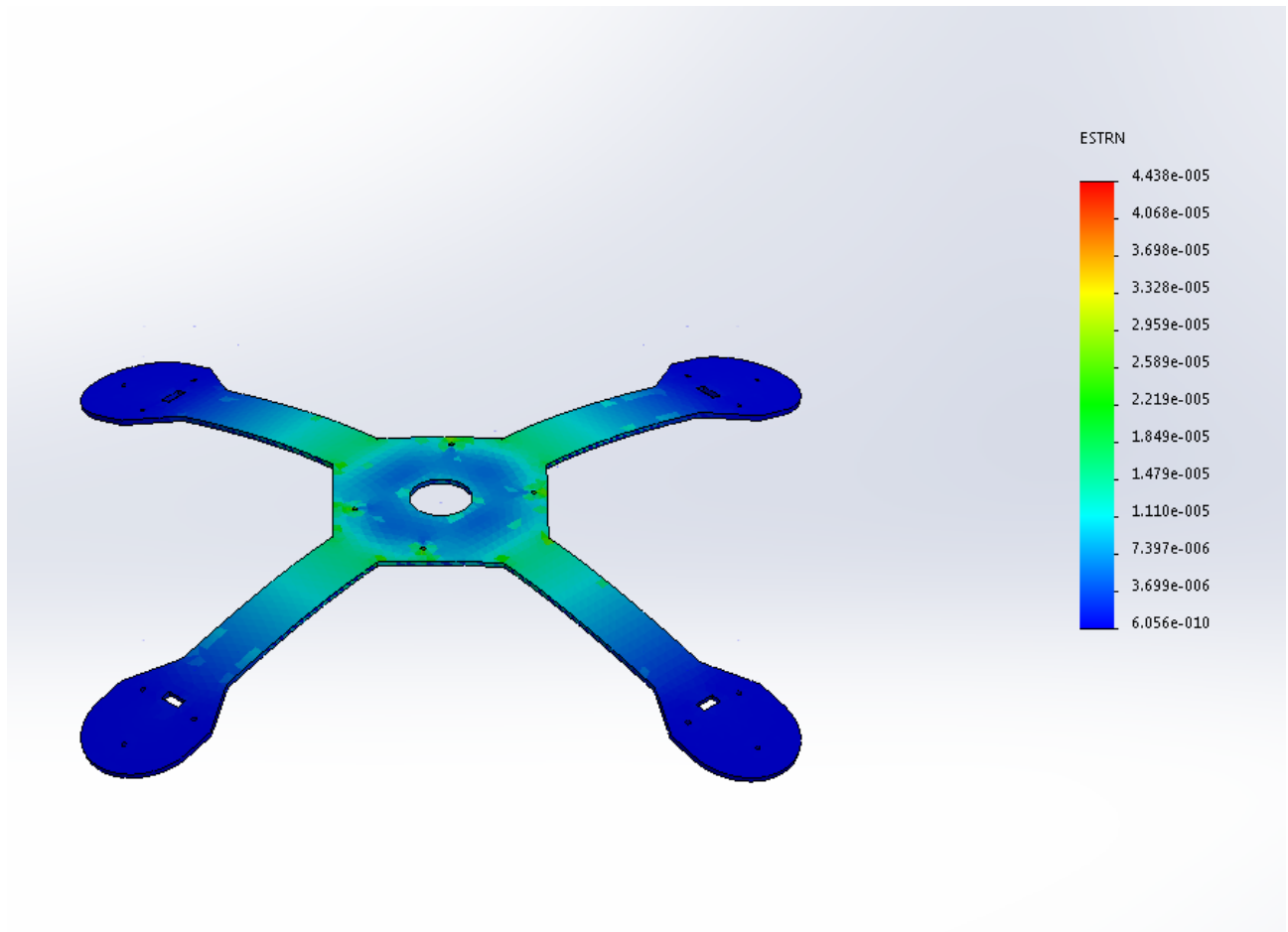


Figure 17: Cross Mount, Strain Test

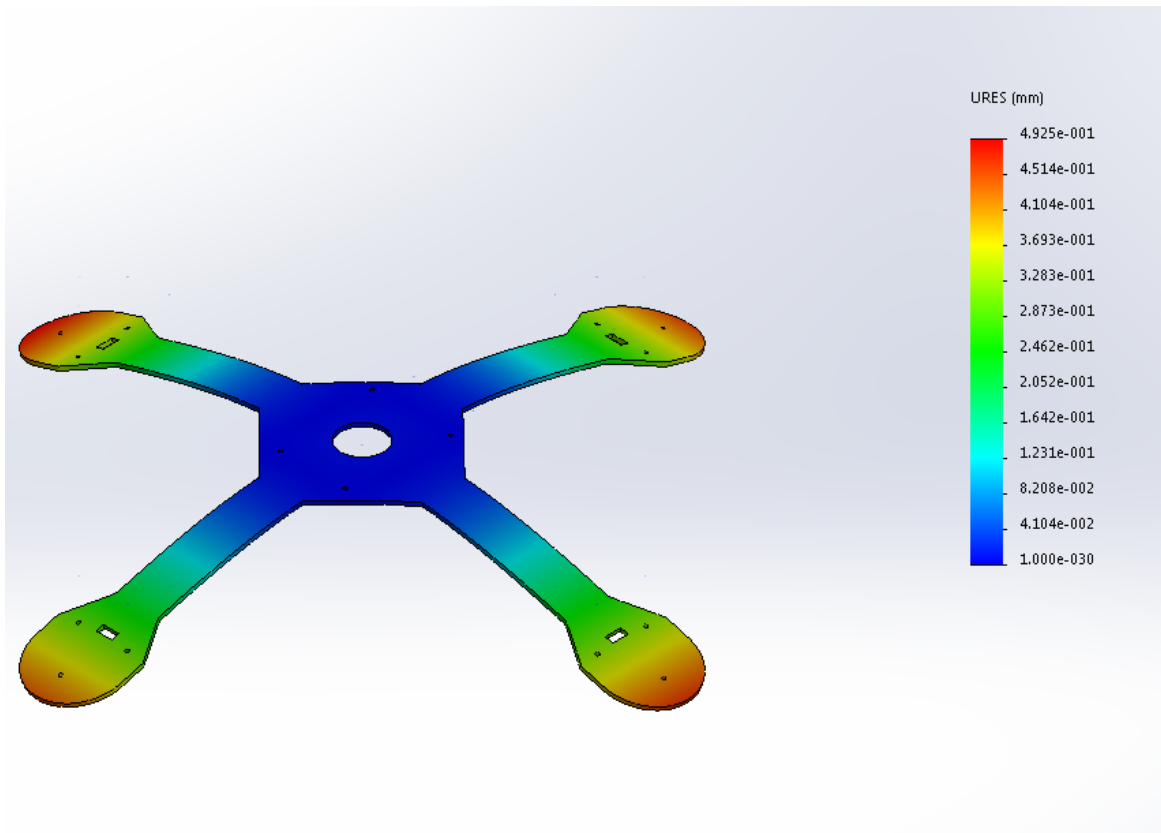


Figure 18: Cross Mount, Displacement Test

Appendix B: Weight of Chosen Mount with Different Materials

The data generated by SOLIDWORKS mass properties function. The data was used to determine the material for the cross mount which will hold the CO₂ sensors.

Mass properties of MQP Khepera Mount 1
Configuration: Default
Coordinate system: -- default --

Density = 0.00 grams per cubic millimeter

Mass = 181.06 grams

Volume = 177508.80 cubic millimeters

Surface area = 120420.70 square millimeters

Center of mass: (millimeters)
X = 0.00
Y = 1.59
Z = 0.09

Figure 19: Cross Mount Mass Properties, ABS

Mass properties of MQP Khepera Mount 1
Configuration: Default
Coordinate system: -- default --

Density = 0.00 grams per cubic millimeter

Mass = 213.01 grams

Volume = 177508.80 cubic millimeters

Surface area = 120420.70 square millimeters

Center of mass: (millimeters)
X = 0.00
Y = 1.59
Z = 0.09

Figure 20: Cross Mount Mass Properties, Acrylic

Mass properties of MQP Khepera Mount 1
Configuration: Default
Coordinate system: -- default --

Density = 0.00 grams per cubic millimeter

Mass = 479.27 grams

Volume = 177508.80 cubic millimeters

Surface area = 120420.70 square millimeters

Center of mass: (millimeters)
X = 0.00
Y = 1.59
Z = 0.09

Figure 21 Cross Mount Mass Properties, Aluminum 6061 – T6

Appendix C: Frequency Analysis Data for Experimental Mount Assembly

This section contains data generated by SOLIDWORKS frequency analysis used to determine a suitable height for the CO₂ sensors and length of the extender rods, while balancing stability of the assembly.

Table 2: Frequency Analysis of Experimental Mount Assembly, Mode 1

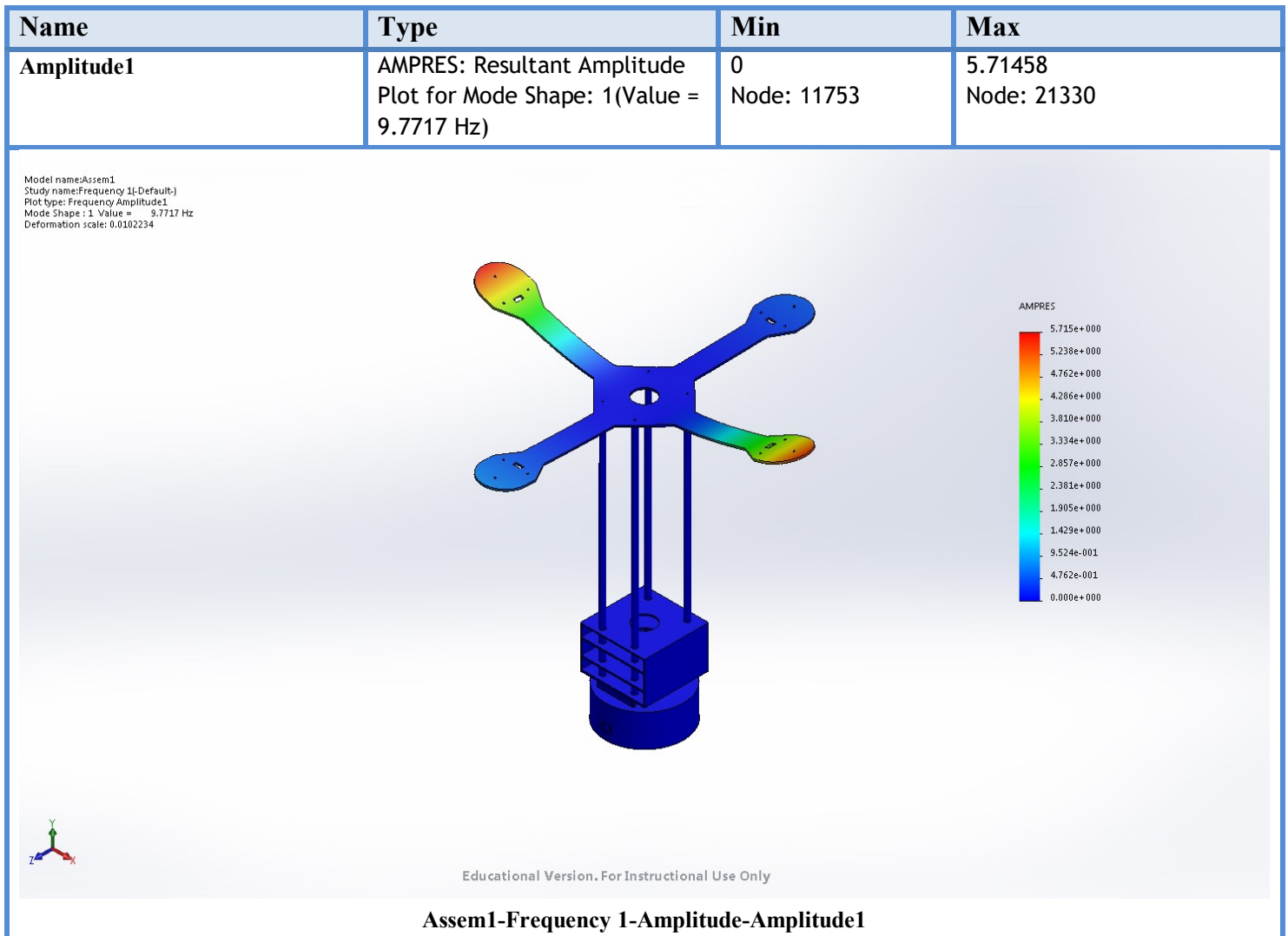


Table 3: Frequency Analysis of Experimental Mount Assembly, Mode 2

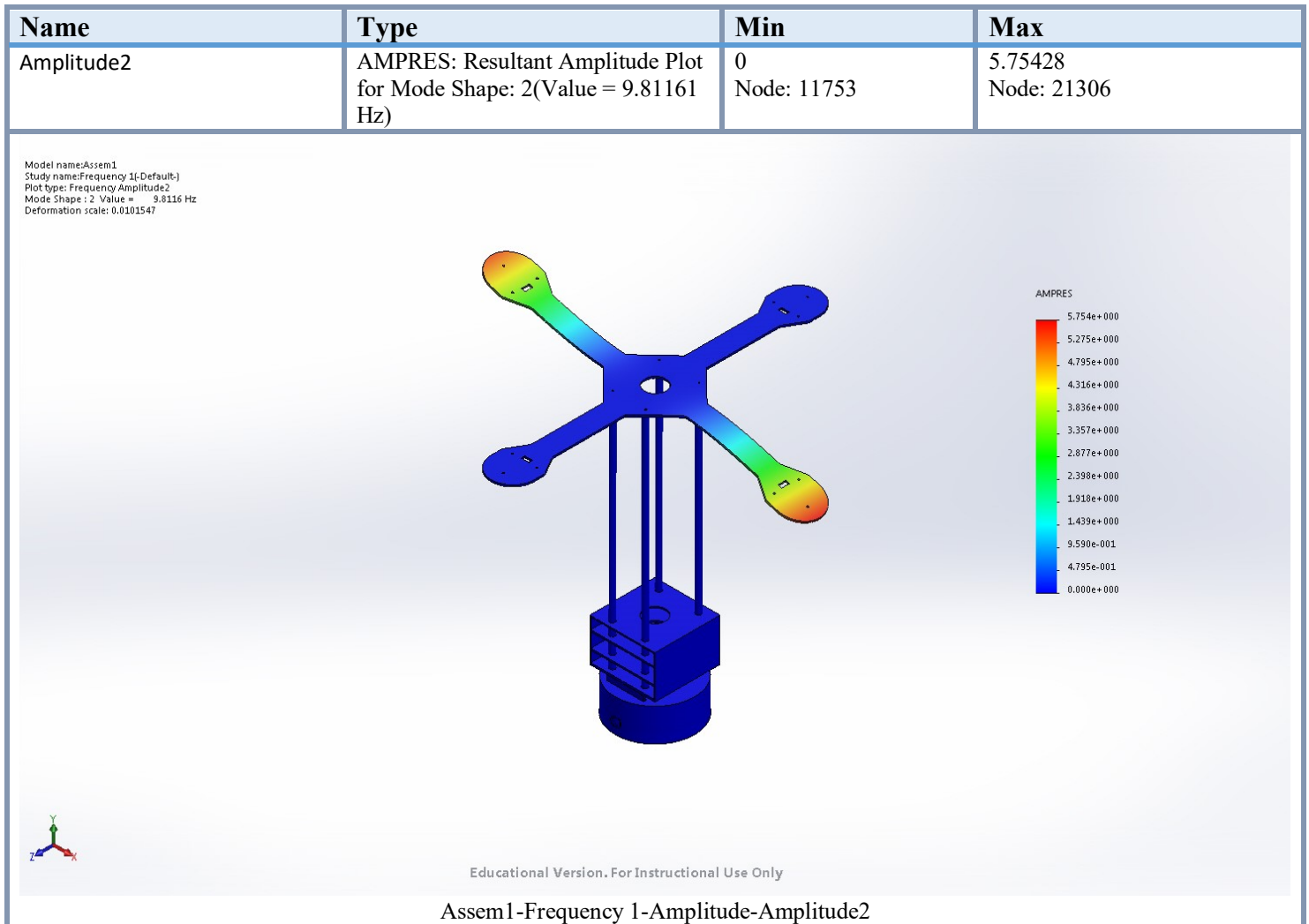


Table 4: Frequency Analysis of Experimental Mount Assembly, Mode 3

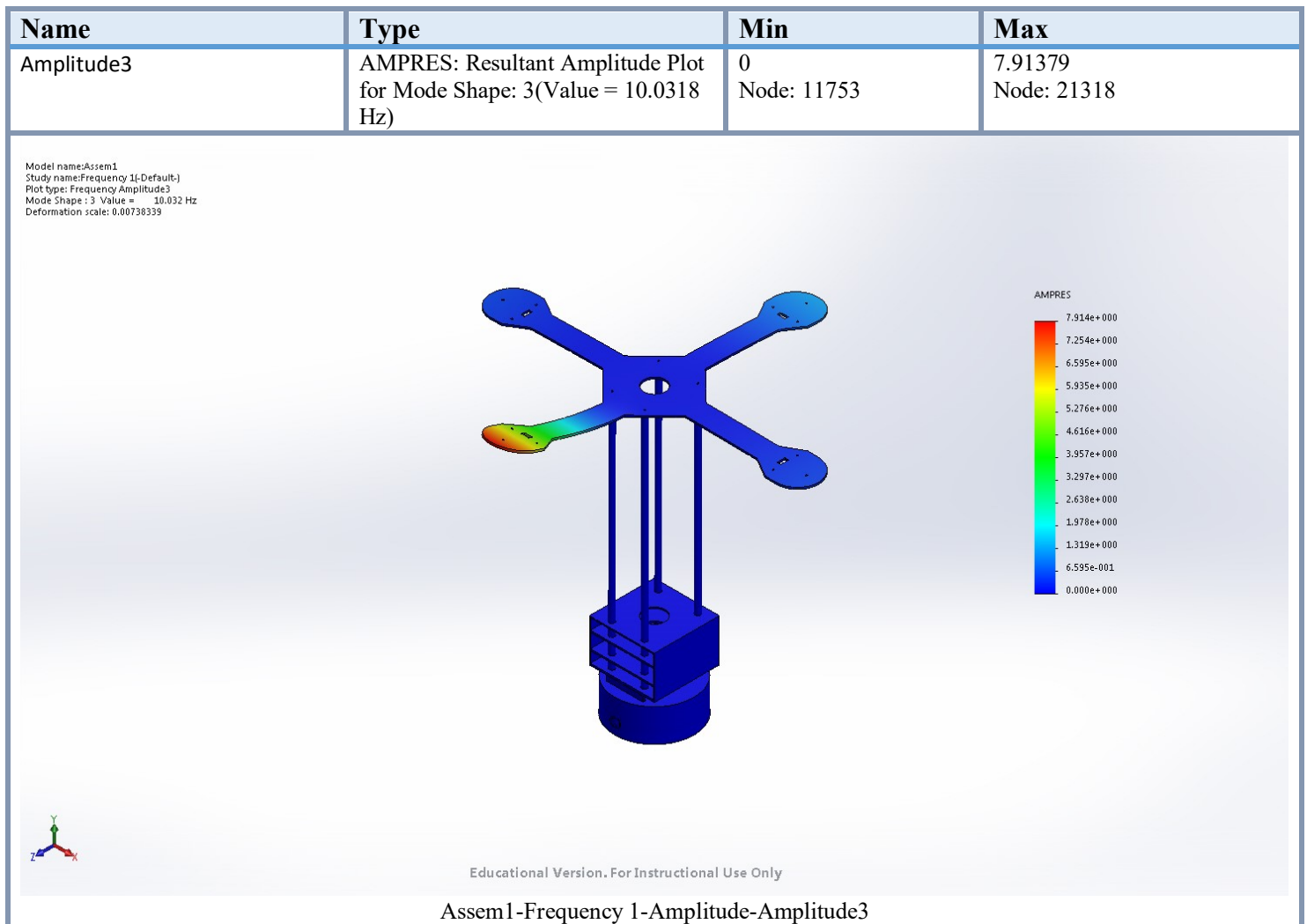


Table 5: Frequency Analysis of Experimental Mount Assembly, Mode 4

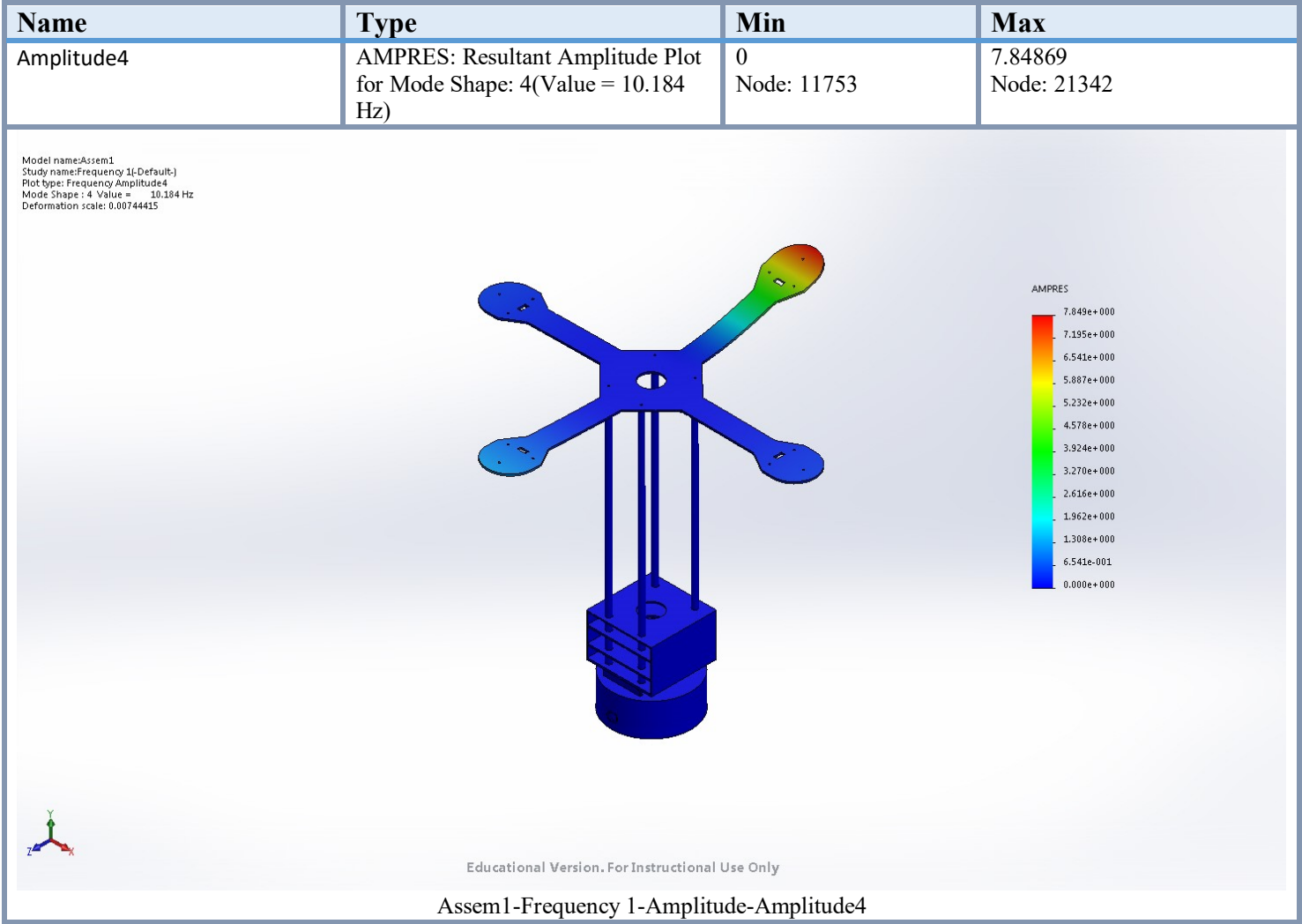


Table 6: Frequency Analysis of Experimental Mount Assembly, Mode 5

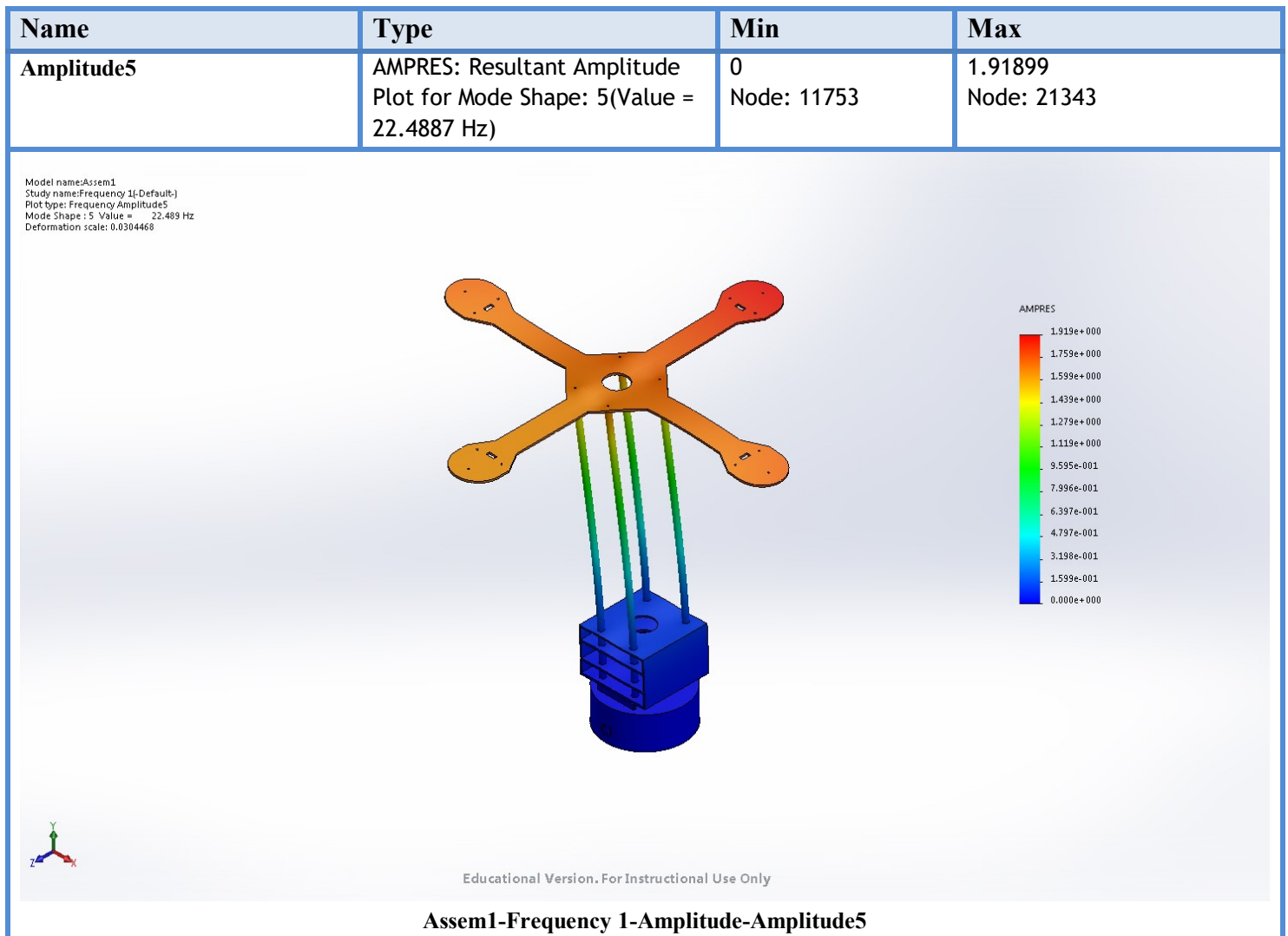


Table 7: Measurements for each Frequency Number of the Experimental Mount Assembly

Frequency Number	Rad/sec	Hertz	Seconds
1	61.397	9.7717	0.10234
2	61.648	9.8116	0.10192
3	63.032	10.032	0.099683
4	63.988	10.184	0.098193
5	141.3	22.489	0.044467

Table 8: Frequency and Displacement Values for Each Mode of the Experimental Mount Assembly

Mode Number	Frequency(Hertz)	X direction	Y direction	Z direction
1	9.7717	3.3976e-007	0.049709	1.3576e-005
2	9.8116	0.00042602	3.8586e-005	3.3108e-008
3	10.032	1.7014e-009	0.0093603	0.00049711
4	10.184	1.492e-007	0.021026	0.00020501
5	22.489	0.26825	8.0458e-009	5.9381e-005
		Sum X = 0.26867	Sum Y = 0.080134	Sum Z = 0.00077511

Appendix D: Bill of Materials for Project

A description of each material used or purchased. Each material is described by their part number, weight, unit cost. A total cost and weight budget are provided at the bottom of the table. The table was made in Microsoft Excel.

Table 9: MAD 1601 Bill of Materials

Part #	Part Name	Weight(g)	Description	Qty	Unit Cost	Cost	Weight Total
GC-0012	COZIR Ambient 10k CO2 Sensor	20	High preformance CO2 sensors	4	\$ 109.00	\$ 436.00	80
	Rev P Wind Sensor	2	Lightweight wind sensor	2	\$ 24.00	\$ 48.00	4
MAD1601P1	Cross Sensor Platform	479.27	Aluminum Platform for Sensors and Electronics	1	\$ 41.90	\$ 41.90	479.27
MAD1601P2	Holding Shelf	185.52	ABS plastic shelf for organizing electronics	1	\$ 90.44	\$ 90.44	185.52
A000066	Arduino Uno Rev 3	25	A microcontroller board for reading the sensor I/O	1	\$ 19.95	\$ 19.95	25
MAD1601P3	43.8cm Rods	92.86	Extender rods for cross mount	4	\$ 9.00	\$ 36.00	371.44
MAD1601P4	25cm Rods	52.28	Extender rods for cross mount	8	\$ 5.00	\$ 40.00	418.24
MAD1601P5	Rod Stabilizers	18.59	Support stabilizers for rods	2	1	\$ 2.00	37.18
MAD1601P6	Wind Sensor Unit	3.37	small holding unit for wind sensor			\$ -	0
MN 1604	12 Volt A23 Duracell Alkaline Batteries	45.1	A battery unit for powering the sensors	1	\$ 7.99	\$ 7.99	45.1

27WK22SLD25	Electronix Express-Hook up Wire Kit (Solid Wire Kit) 22 Gage	0	Connection wire for system function	1	\$ 22.00	\$ 22.00	0
	Khepera IV	530	Mobile testing robot	1	\$3,400	\$ 3,400.00	530
nRF24L01+	Wifi Transcievers	1	Wireless transceiver for Arduino board	1	\$ 5.95	\$ 5.95	1
IB400	Microtivity IB400 400-point Experiment Breadboard	10	Wire connection board for sensors and Arduino	1	\$ 5.15	\$ 5.15	10
702105149218	10 Pcs PC PCB Motherboard Brass Standoff Hexagonal Spacer M3	0	Standoffs to separate the Platform from the Khepera IV	1	\$ 2.97	\$ 2.97	0
B000NHQEQ	Class 4.8 Steel Machine Screw, Zinc Plated Finish, Pan Head, Phillips Drive, Meets DIN 7985, 12mm Length, M3-0.5 Metric Coarse Threads	0	Fasteners for the cross platform and the standoffs	1	\$ 2.67	\$ 2.67	0
N82E16833139027	IOGEAR GBU521 USB Bluetooth 4.0 Micro Adapter	0	enables Bluetooth connection between the base computer and the Khepera IV	3	\$ 12.49	\$ 37.47	0
Total				33		\$ 4,198.49	1656.75

Appendix E: Technical Drawings, Generated Using SOLIDWORKS

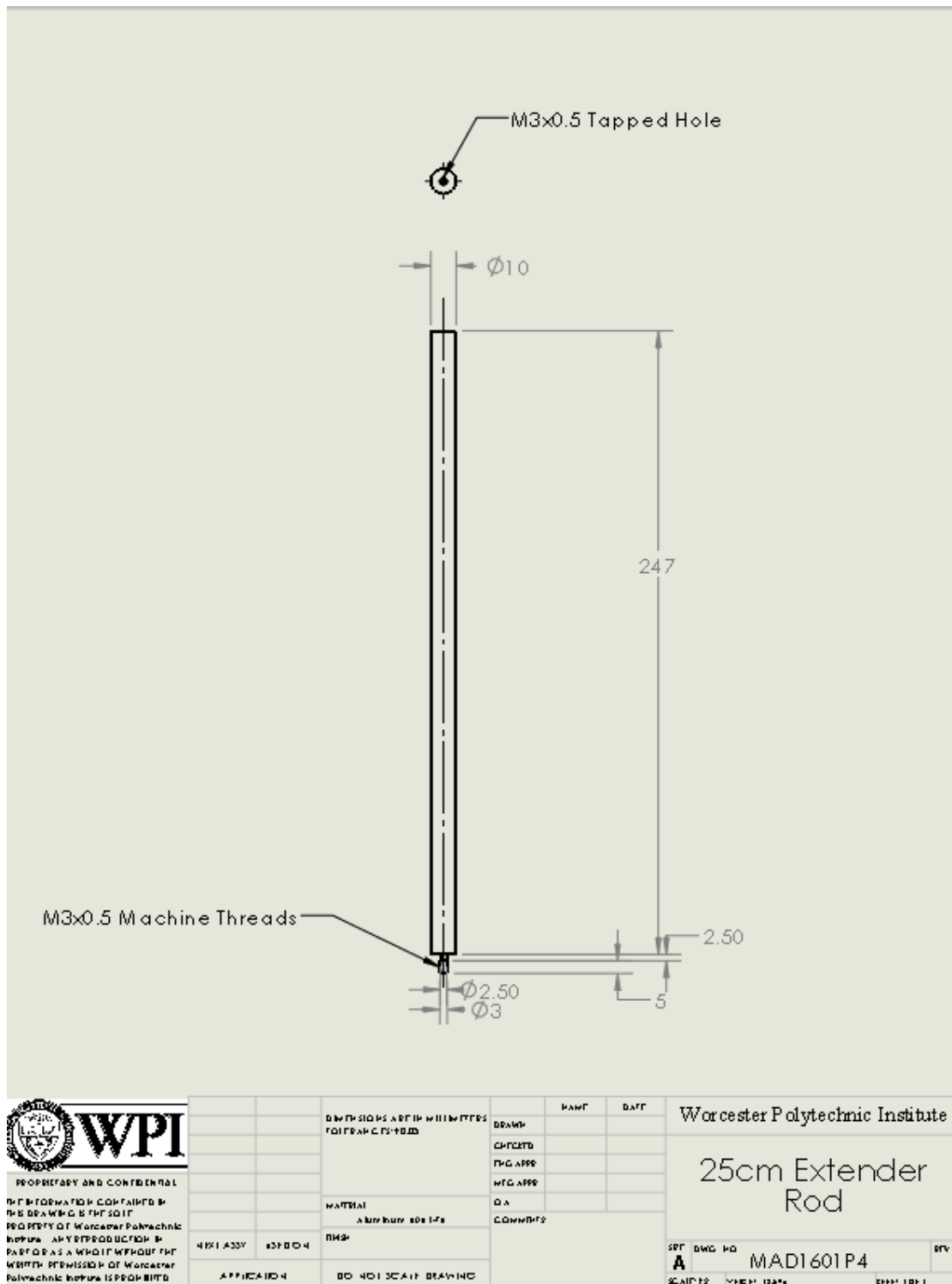


Figure 22: Extender Rod, 48.3 cm

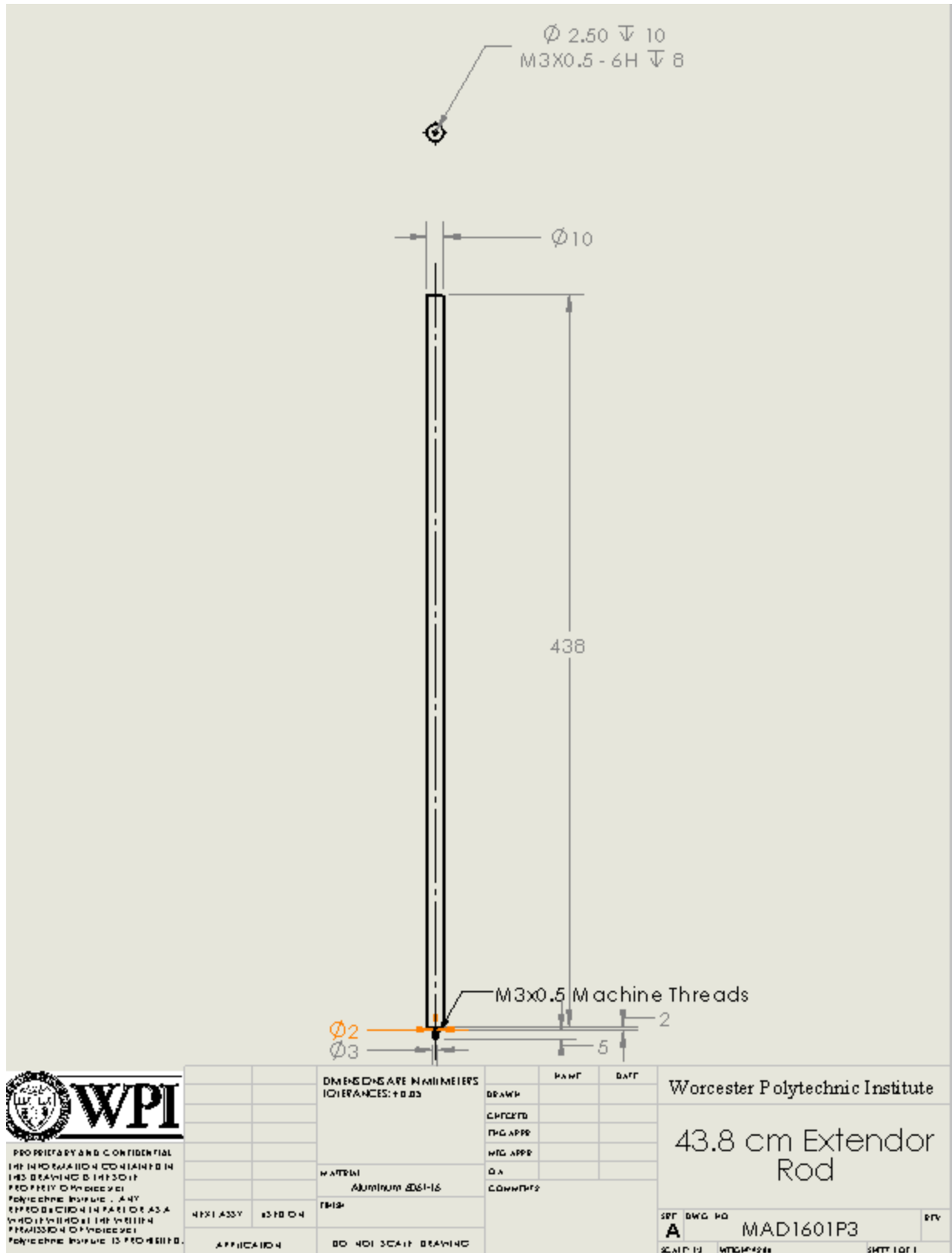


Figure 23: Extensor Rod, 25 cm

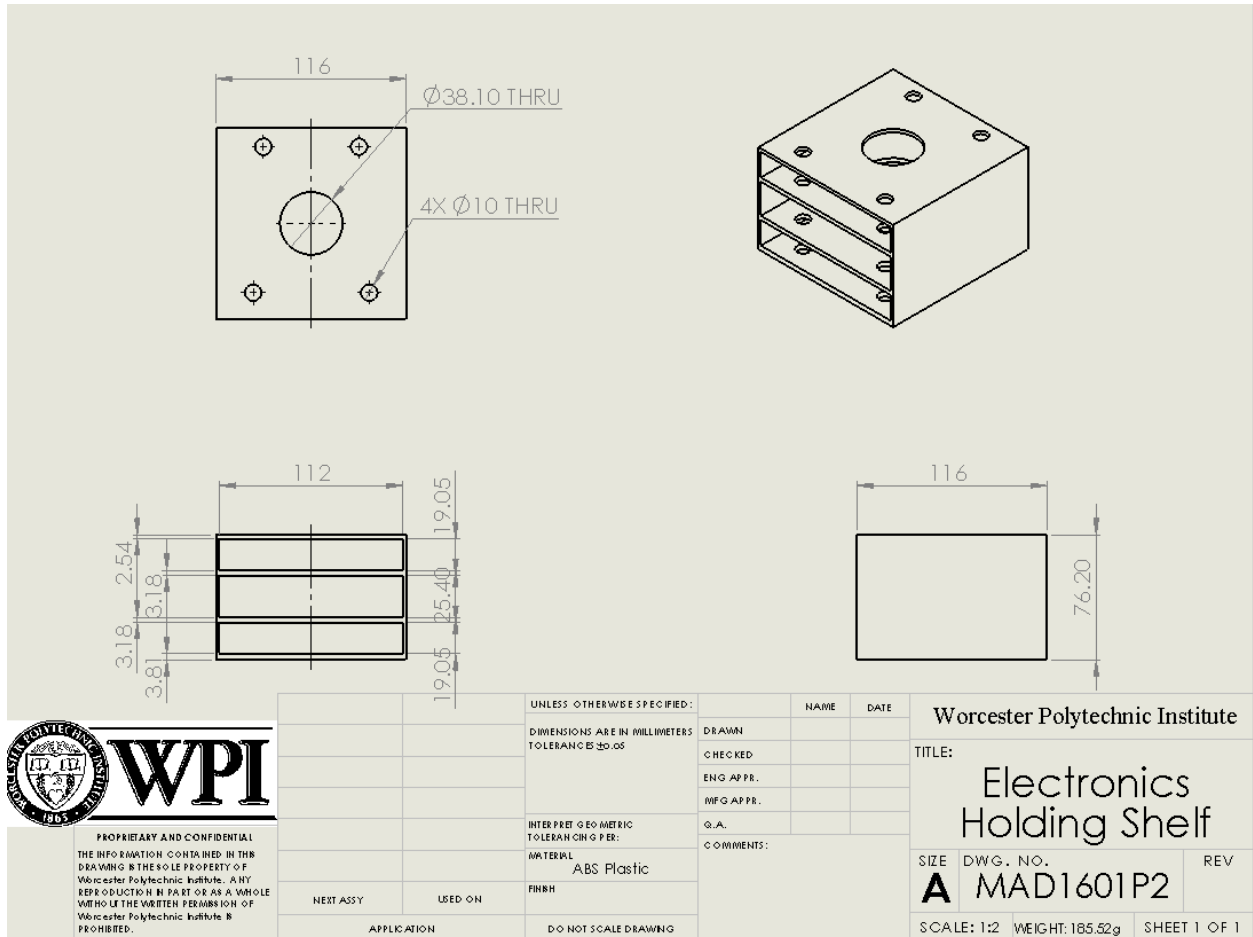


Figure 24: Component Shelf

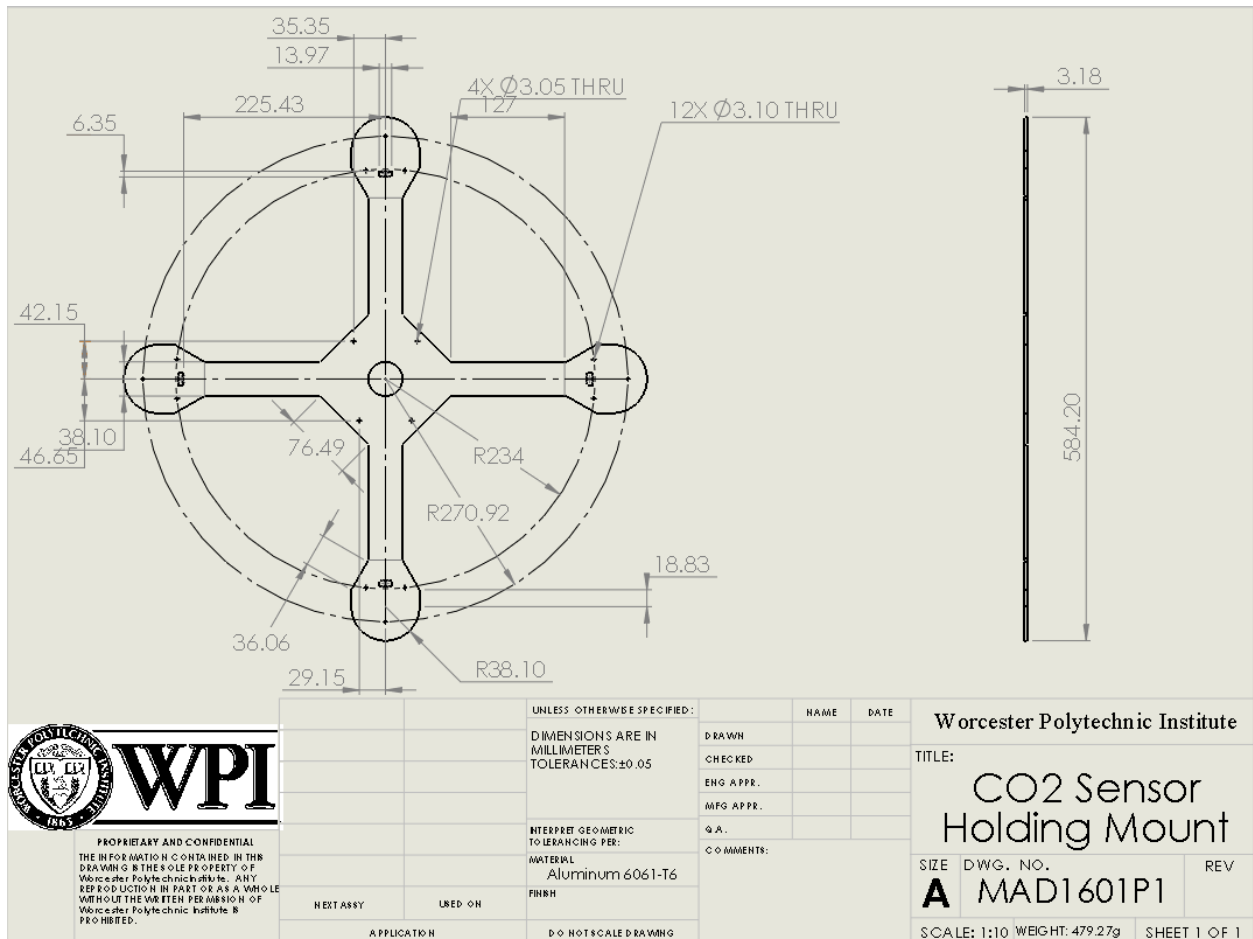


Figure 25: Sensor-Holding Cross Mount

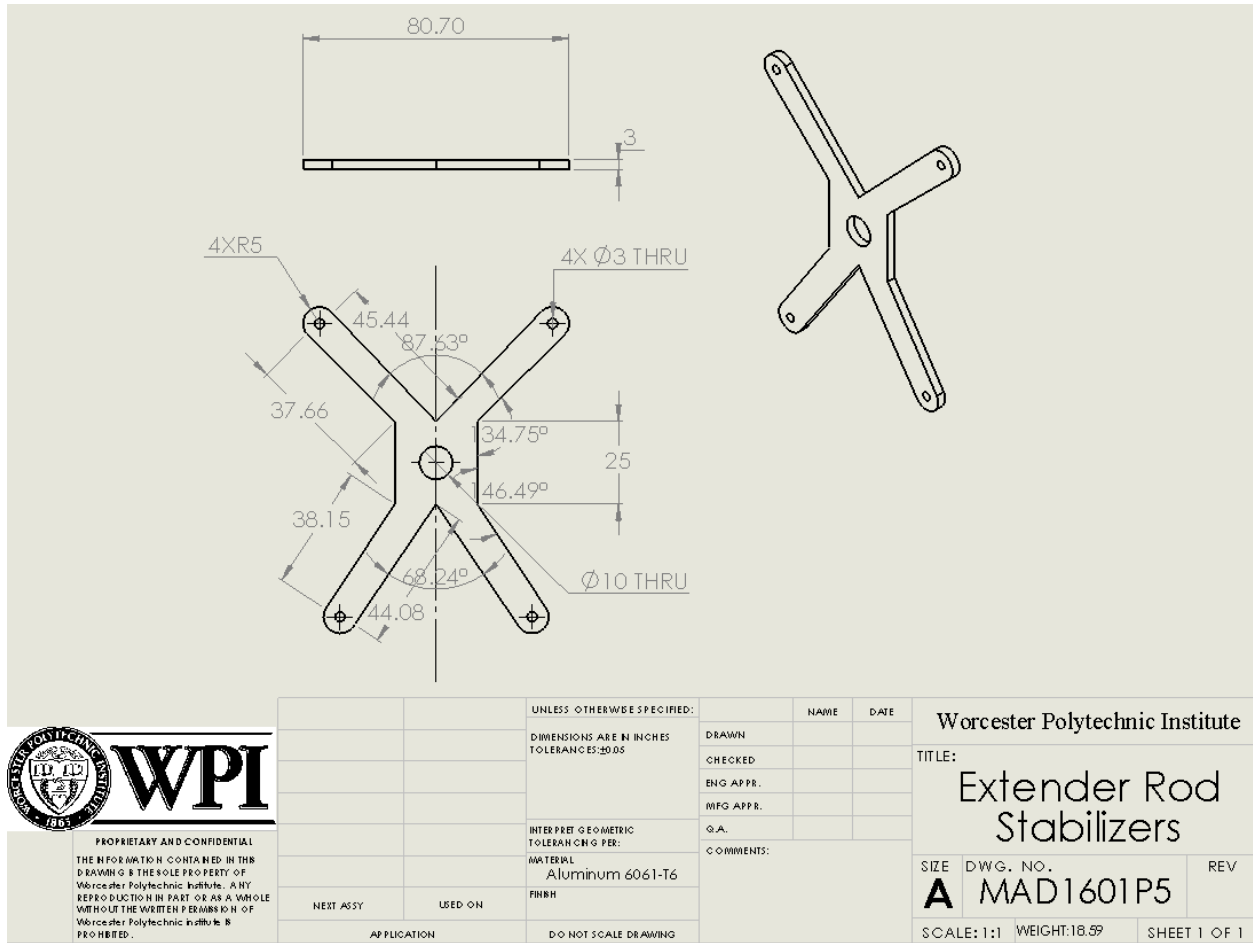


Figure 26: Extender Rod Stabilizer

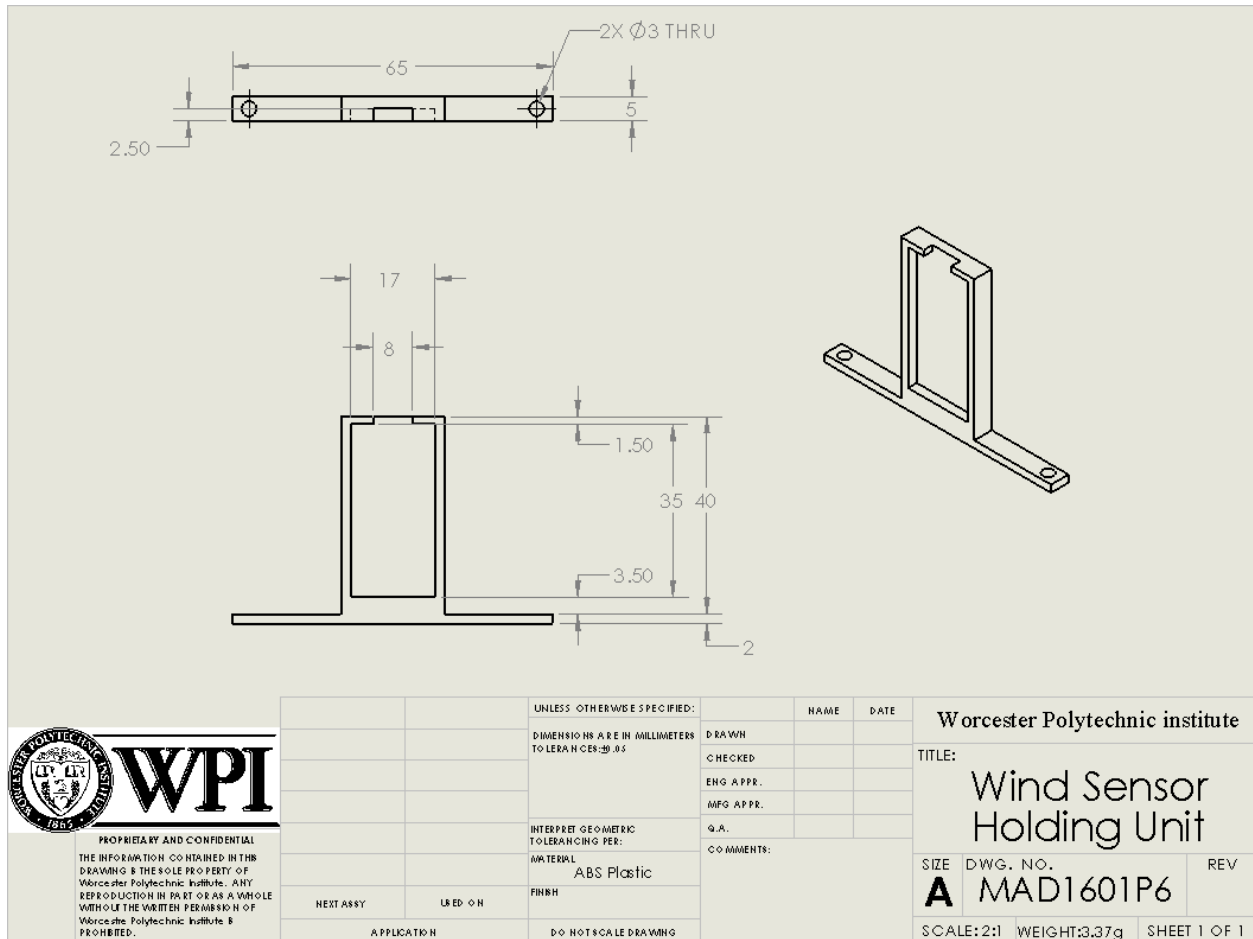


Figure 27: Wind Sensor Holder