A 2-D MODEL OF OPTICAL SATELLITE COMMUNICATION SYSTEM

By

Xuanyi JIN

Yuxuan LI

Zhijun ZHAO

Jun ZHENG

Final Report for ECE 445, Senior Design, Spring 2025

TA: Yue Yu

Advisor: Pavel Loskot

13 May 2025

Project No. 33

Abstract

The growing demand for stable, multifunctional, and efficient satellite communication systems has called for advancements in aerospace and communication technologies. Low Earth Orbit (LEO) satellites, such as those in the Starlink constellation, offer key advantages—low latency, low signal loss, and cost-effectiveness, making them ideal for applications like global mobile network coverage, climate monitoring, and geographical mapping. However, due to their low orbits, their rotational velocities are high and the coverage of Earth surface per satellite is limited, so that a large constellation is required to ensure all-time ground-to-satellite communication, introducing challenges in maintaining reliable communication links due to alignment issues.

To better observe these challenges, this project presents a 2-D Model of Optical Satellite Communication System designed to simulate LEO satellite dynamics and ground-station interactions. The model employs rotating disks to represent Earth and satellites, with a laser transmitter and photoreceptor emulating optical communication. Adjustable motor-driven rotation, configurable transmitted messages, and a software interface enable users to analyze factors affecting signal efficiency, misalignment losses, and system performance under varying orbital conditions. The results provide valuable insights for optimizing future satellite communication systems.

Contents

1. Introduction

With the rapid advancement of aerospace and communication technologies, there is an increasing demand for more stable, multifunctional, and efficient satellite communication systems. In particular, Low Earth Orbit (LEO) satellites have attracted huge attention due to their advantages of low latency, high signal recall, and cost-effectiveness compared to traditional geostationary satellites. As exemplified by commercial systems like Starlink, LEO satellites have demonstrated remarkable potential in various fields, including climate monitoring, geographical mapping, and internet coverage to even the most remote areas. However, the inherent characteristics of LEO satellites also bring new challenges — their relatively low altitude and fast orbital velocities result in limited coverage of ground stations per satellite, necessitating large-scale satellite constellations to ensure continuous global communication. This dynamic environment introduces complexities in maintaining reliable communication between satellites and ground stations, especially when using optical communication methods that are highly directional and sensitive to alignment and synchronization.

To visualize and study these challenges, we designed and developed a 2-D Model of Optical Satellite Communication Systems. This model simulates the orbital motion of satellites and the rotational movement of Earth in a controlled environment, using rotating disks to represent the Earth and satellites in LEO. Optical communication will be emulated using a laser transmitter mounted on the Earth disk and a photoreceptor on the satellite disk. Through adjustable motor-driven rotation, configurable laser transmission frequencies, and a software monitoring interface, this model will allow users to investigate the factors influencing optical signal transmission efficiency in LEO communication scenarios. Compared to previous conceptual models or purely software-based simulations, this solution provides an intuitive and interactive hardware-software hybrid platform that physically demonstrates real-world communication dynamics, signal loss due to misalignment, and system performance under various orbital conditions.

1.1 Visual Aid

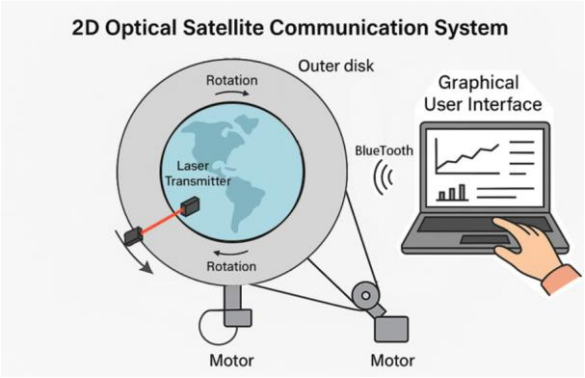

Figure 1 An Illustration of the model

This visual aid as shown in Figure 1 illustrates the overall workflow of our 2-D optical satellite communication system. It includes:

1. An inner disk that represents the 2-D projection of Earth. A laser transmitter is installed on the edge of the disk to represent a ground station on Earth's surface.
2. The outer rotating disk represents a LEO. A photoreceptor is loaded on the edge of this disk to simulate a satellite.
3. Two motors working independently at different rotational speeds that power the movement of the two disks.
4. Belts and pulleys that the motors leverage to drive the disks to rotate. Each disk is directly attached to one pulley, and each motor has a pre-installed pulley. Two rubber belts transmit the kinetic energy in each motor-disk pair.
5. A software interface connected wirelessly (via Bluetooth) to control disk rotation speeds and laser emission messages.
6. The user interacts with the software interface to adjust system parameters and view performance metrics, such as received signal strength and transmission efficiency.

1.2 High-level Requirements

1. The system must physically simulate the relative motion between LEO satellites and Earth by rotating two concentric disks with independently controllable speeds to mimic orbital dynamics.
2. The optical communication subsystem must successfully transmit and receive binary-encoded signals using a laser transmitter and photoreceptor, with proper alignment constraints to reflect real-world scattering angle limitations.
3. The software control and monitoring system must allow users to configure key simulation parameters, including disk rotation speeds and signal transmission frequency, while visualizing the transmission efficiency and received data in real time.
4. The system must decode the received optical signals into human-readable messages, demonstrating the feasibility of optical satellite communication and enabling the evaluation of transmission accuracy under different conditions.

1.3 Functionality

Overall, we designed and built a 2-D simulation of satellite communication system. As shown in Figure 3, we have an iron pole in the middle of the acrylic base as the axis of rotation for the two disks. The smaller black disk represents Earth, and the larger wood disk represents the orbit of a satellite. The two disks are driven by two motors, each running independently at different user-defined speeds and directions. Their real-time speeds are recorded by a Hall encoder then displayed in a graphical user interface. A laser transmitter is placed inside a box mounted beneath the top disk, shooting laser beams according to the commands from an Arduino UNO board that is also fixed beneath the disk. Our photoreceptor PCB is placed on top of the bottom disk, capturing the message sent from the laser transmitter. Users can type a string of a maximum of 15 characters in the graphical user interface, which will be sent to the Arduino board on the top disk via Bluetooth. The received message will be sent from the Arduino board on the bottom disk via Bluetooth back to the graphical user interface. The time used for receiving each character and the entire string message are recorded for efficiency calculations.

2. Design

We designed and constructed a 2-D model of a ground-to-satellite communication system using laser beams as the information medium. This optical communication-based simulation system integrates a graphical user interface (GUI) subsystem, a motion control system that we call the orbit simulation subsystem, a power subsystem, a data processing and control subsystem, and an optical communication subsystem. Our system allows a user to control and monitor the rotation speed of a two-disk system that simulates the movements of a satellite and a communication tower on the surface of the Earth and the data transmitted via an optical communication link. An Arduino-powered microcontroller is implemented to distribute motor control, signal processing, and efficiency analysis. Power is supplied via a 24 V Li-ion battery, and data transmission will be conducted through an optical communication system containing a laser transmitter and a laser receiver. Figure 2 shows the layout of our simulation system.

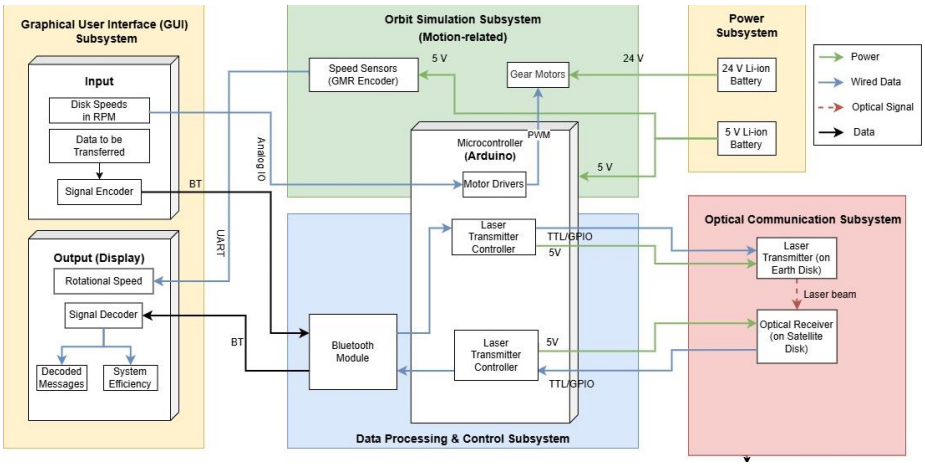

Figure 2 Complete System Flowchart

The physical design involves a thick acrylic board as the foundation. The two motors are placed in parallel. A steel rod is fixed on the board as the rotational axis for the two disks, simulating the Earth and the orbit of a satellite. Each disk is attached to a gear. The gears are connected to the motors through two belts. The laser transmitter and laser receiver are mounted on the edge of the two disks. Figure 3 shows concisely how our physical model is assembled. This report will explain in detail the functionalities and components of each subsystem in the remainder of this section.

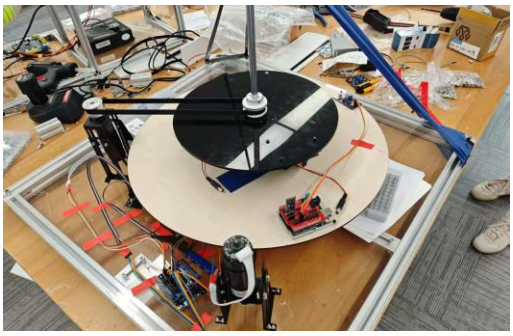

Figure 3 Sketch of Physical Model

2.1 Graphical User Interface (GUI) Subsystem

The Graphical User Interface subsystem, developed with Python, serves as the monitoring and control hub of the laser communication simulation (Orbit Simulation Subsystem and Optical Communication Subsystem.) It allows user interactions with the physical model by allowing configuration of system parameters such as motor speeds and laser modulation parameters and providing visualized real-time feedback from the embedded hardware in the Orbit Simulation Subsystem and the Optical Communication Subsystem. As shown in the block diagram (Figure 2,) the GUI has an Input submodule and an Output (Display) submodule.

The GUI subsystem receives data such as encoded digital readings from the photoreceptor and real-time motor RPM (Revolutions Per Minute) via serial communication at a baud rate of 38,400 bps. The computer running the GUI application must pair with the HC-05 Bluetooth module, which emulates a virtual serial port that typically appears as "/dev/cu.usbmodemXXXX" on MacOS or "COMx" on Windows. The GUI establishes a non-blocking serial connection, enabling continuous data listening in a background thread without freezing the user interface.

On user input, commands for disk speeds and modulation frequency are formatted and sent to the Data Processing and Control Subsystem. The data streams returned from the Data Processing and Control Subsystem data will be decoded and visualized by the GUI. Specifically, the GUI enables closed-loop PID (Proportional-Integral-Derivative) control by continuously adjusting motor RPM based on sensor readings.

2.1.1 GUI Schematics

Since this subsystem is purely software, a schematic will not be presented here. However, a screenshot of the two parts of our GUI is included in section 5.1.

2.1.2 Input

The GUI allows the user to adjust disk rotational speed, laser pulse frequency and message to be communicated using interactive elements like sliders. Additionally, users can trigger data transfer commands to initiate communication and signal sampling.

2.1.3 Signal Encoder

The signal encoder is not an actual component, but a function integrated into the GUI. The encoder converts the input character string into a two-byte long binary signal, then sends this data to the Data Processing and Control Subsystem where it will be made ready for optical communication.

We chose to convert character to binary format because the laser transmitter only has two states, "ON" and "OFF." To enable effective communication between the laser transmitter and the receiver within the short timeframe when they are aligned, we tried to maintain the length of the binary encoding of each character concise but informative. Each character in the original message will be represented by 16 bits (2 bytes.) The first four bits are always 0. The next five bits encode the actual character. We used a modified ASCII table, where 0 represents "a," 1 represents "b," and decimal 25 represents letter "z." We also support space, period, and the next line character ("\n.") The complete dictionary for the character-encoding pair is in Appendix A. Bit 6 is always 0. The next four bits represent the index of the current

character in the original message. Bit 11 and 12 are always 01. Figure 4 shows an example of how we encode our message.

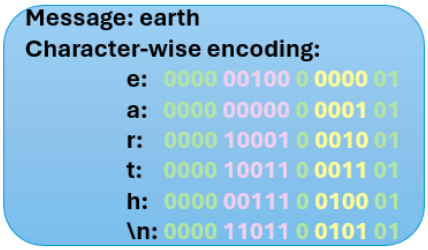

Figure 4 An Example of Message Encoding

When a signal (taking the test string "earth" as an example) enters the encoder, the encoder will first disassemble each character by mapping it to a 5-bit code and taking its index in binary format. The presence of the first four zeros is to pad the encoding to a multiple of byte, a format friendly to Bluetooth signal sending. The middle zero bit is to make sure that the sequence "11111" does not appear within each 16-bit encoding (think about a "p" at index 12,) for it is a starter sequence that marks the start of each character which will be appended by the Data Processing and Control Subsystem before the laser transmitter shoots out the beams representing this character. The encoder also puts an "\n" symbol with index "5" to the end of the user message, for decoder to know where the message ends. Table 1 provides all the characters we support and their encodings.

Table 1 Encoding of each character

Code	a	b	c	d	e	f
Encoding	00000	00001	00010	00011	00100	00101
Code	g	h	i	j	k	L
Encoding	00110	00111	01000	01001	01010	01011
Code	m	n	o	p	q	r
Encoding	01100	01101	01110	01111	10000	10001
Code	s	t	u	v	w	x
Encoding	10010	10011	10100	10101	10110	10111
Code	y	z	(Space)	\n		
Encoding	11000	11001	11010	11011		

2.1.4 Signal Decoder

The signal decoder does everything the encoder does but in reverse. It is responsible for recompiling the received binary signals into the user input message. Like the encoder, it is not an actual component but rather a function implemented in the GUI. The decoder has an internal buffer. Whenever the decoder finishes decoding a character, it fills it into the buffer at the decoded index. When the next line symbol has been decoded and every empty position before the next line symbol has been filled, the decoder outputs the decoded message to the user and then clears the buffer.

2.1.5 Output

Real-time feedback is provided through embedded plots and status panels. Key outputs include:

1. Rotational Speed: Visualized in RPM using a dynamic matplotlib plot.
2. Decoded Messages: Display of received and reconstructed digital bitstreams.
3. System Efficiency: Future versions aim to display Bit Error Rate (BER) and transmission success metrics.

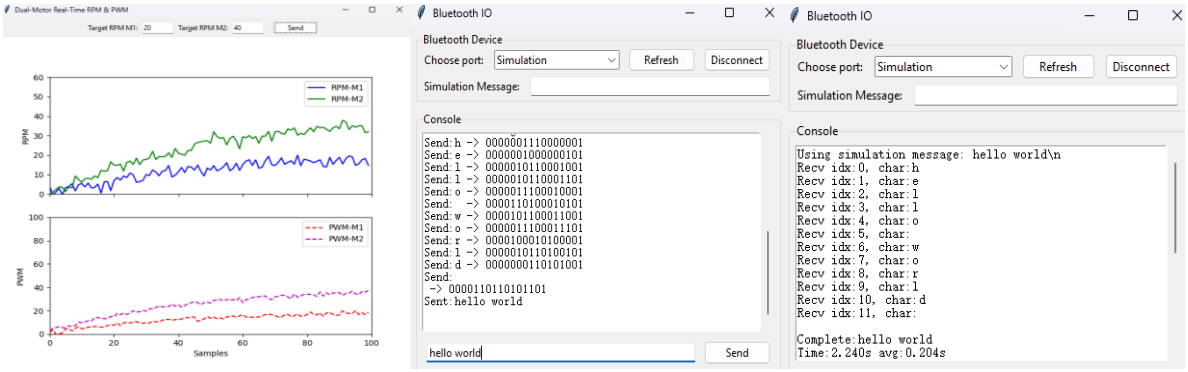

Figure 5 GUI for Motor Control GUI and Text Transmission GUI

2.1.6 GUI Design Alternative

During the development of the GUI subsystem, multiple design alternatives were considered in terms of both architecture and implementation framework. Below, we outline and evaluate key alternatives that were explored:

Table 2 GUI Design Alternatives

Framework	Advantages	Disadvantages	Suitability
PyQt5 / PySide2	Rich set of widgets, modern UI appearance, strong multi-threading support, integrated plotting with PyQtGraph	Steeper learning curve, larger binary size, GPL/commercial licensing, more complex packaging	Suitable for large-scale, professional GUI; offers best styling flexibility
Tkinter (chosen)	Built-in to Python, easy to learn and deploy, low overhead, sufficient for simple control panels	Limited aesthetics, less advanced layout management, primitive widget styling	Suitable for fast prototyping and minimal GUIs like control dashboards
MATLAB-Based GUI	Robust serial communication capabilities and built-in plotting tools. Suitable for data processing.	Requires MATLAB license. Less flexible for custom data format and Bluetooth. Slower response time for frequent data.	Ideal for engineering analysis, less suited for a responsive interface for real-time control
Web-based (Flask + React)	Platform-independent, responsive UIs, can run in browser, easy remote access	Requires web development knowledge, latency due to HTTP loop, complex deployment	Ideal for remote monitoring or user access from multiple devices

After testing several frameworks, Tkinter was selected due to its speed of implementation, ease of Integration with Serial, and no need for additional installations.

2.2 Orbit Simulation Subsystem

The Orbit Simulation Subsystem consists of speed sensors and gear motors. The speed sensors monitor the rotation speed of the motor and send the data to the user interface. The gear motors are controlled by the Microcontroller Subsystem. This system simulates the orbit system by controlling the speed of the rotation. Figure 6 shows the motor and the motor driver.

2.2.1 Orbit Simulation Subsystem Schematics

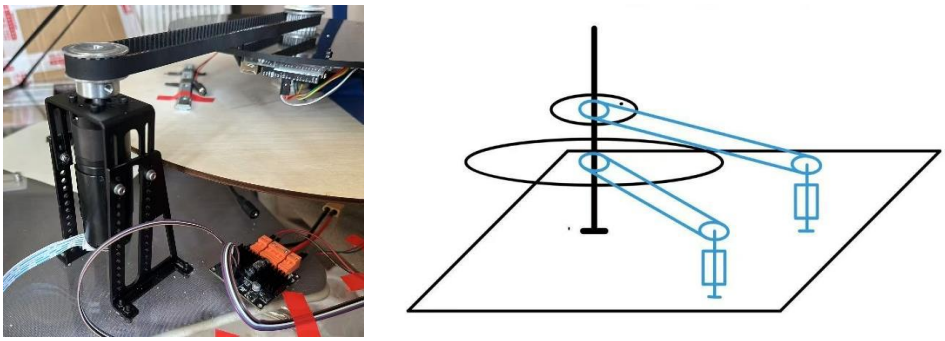

Figure 6 Orbit Simulation Subsystem

2.2.2 Speed Sensor

We use a Hall encoder to measure the rotational speeds of the motors in the Orbit Simulation Subsystem. It generates a digital pulse signal proportional to rotation velocity, which is then read by the microcontroller through a digital input pin. This type of encoder was selected for its high sensitivity, compact size, and durability compared to optical alternatives. It operates at 5 V and provides 360 pulses per revolution, allowing precise speed monitoring. This information is used to simulate orbital motion and evaluate system efficiency in real time.

2.2.3 Gear Motor

The gear motor set drives the disks in the Orbit Simulation Subsystem, simulating Earth and satellite movements. The motors operate at 24 V and feature a 1:30 gear ratio, providing a rated speed of 260 ± 12% rpm and a rated torque of 10 kg · cm. The motor is driven by a PWM signal from the microcontroller, enabling dynamic speed control. Its high torque and stable output are critical for maintaining consistent motion during optical signal transmission. The motor draws a rated current of 2.9 A and was selected for its balance of power (approximately 4 W), reliability, and compatibility with the system's voltage and torque requirements.

We use two motors with pulleys and belts to pull the two disks representing the surface of Earth and orbits of satellites, respectfully. Two master pulleys are mounted on top of the two motors, which are fixed by two racks on the base. Two slave pulleys are attached to the disks. Two belts connect the two master-slave pulley pairs respectfully.

Figure 7 MD36L Motor

We also use two gears and belts to transmit the torque and speed. The gear on the motor has 20 gear teeth and the gear on the disks has 22 gear teeth. Then we can calculate the gear ratio:

$$R = \frac{N_d}{N_m} = \frac{22}{20} = 1.1 \tag{2.1}$$

According to this, we can develop a relationship between speed and toque (T_d) and power (P:)

$$\omega_d = \frac{\omega_m}{R} = \frac{\omega_m}{1.1} \tag{2.2}$$

$$T_d = R \cdot T_m = 1.1 \cdot T_m \tag{2.3}$$

$$P = T_m \cdot \omega_m = T_d \cdot \omega_d \tag{2.4}$$

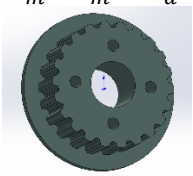

Figure 8 CAD Model of Gear

2.2.4 Motor Driver

The motor driver subsystem is responsible for controlling the speed and direction of the two gear motors used in the orbit simulation subsystem. We selected a dual-channel MOSFET motor driver module, capable of driving two DC motors independently through Pulse Width Modulation (PWM) and digital direction signals.

By modulating the duty cycle of the PWM input, the microcontroller adjusts the average voltage across the motor terminals, thus regulating the motor speed. Direction is controlled by setting the logic level of the IN1/IN2 pair—one HIGH and one LOW to define forward/reverse motion. A logic analyzer and oscilloscope were used during development to verify that correct pulse sequences and switching logic were applied to each channel. This subsystem ensures stable, responsive, and precise control of motor speed, enabling dynamic simulation of satellite and Earth disk motion.

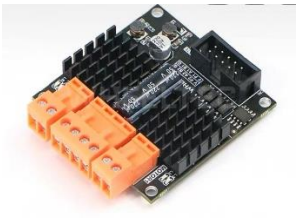

Figure 9 Motor Driver Module

2.2.5 Orbit Simulation Design Alternative

In the initial design of the orbit simulation subsystem, we considered using two motors mounted at different vertical levels: one at the base to rotate the Earth disk and another suspended above on a frame to drive the satellite disk independently. While this configuration allowed for clear mechanical separation, it introduced significant structural and alignment challenges. Supporting a motor overhead requires a stable and rigid frame, increased weight and size, and complicated wiring and maintenance. To address these issues, we revised the design to place both motors at the base and use a combination of gears and belts to drive the concentric disks. This new approach greatly simplified the mechanical structure while maintaining independent control of rotation.

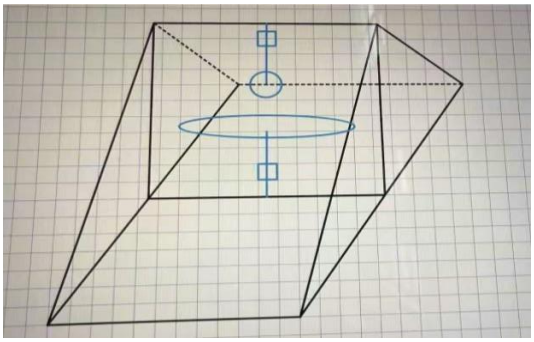

Figure 10 Sketch of Old Disk Holder Design

2.3 Data Processing and Control Subsystem

The Data Processing and Control Subsystem is the control center of this project. The microcontroller consists of three Arduino UNO boards that orchestrate motion control, optical signal processing, and user I/O, coordinating the asynchronous operations of every other subsystem. One Arduino board controls the powertrain of our model, taking user input motor speeds in RPM and using PID control to constantly adjust actual motor speeds to target values. One Arduino board is connected to the laser transmitter in the Optical Communication Subsystem. It stores the user message coming from an HC-05 Bluetooth chip in a buffer and outputs the binary signal to the laser transmitter. The other Arduino board is connected to the optical receiver. It keeps track of the voltage highs and lows on the receiver module output pin and then normalizes them to binary signals and sends them back to users through a JDY-31 Bluetooth chip.

2.3.1 Data Processing and Control Subsystem Schematic

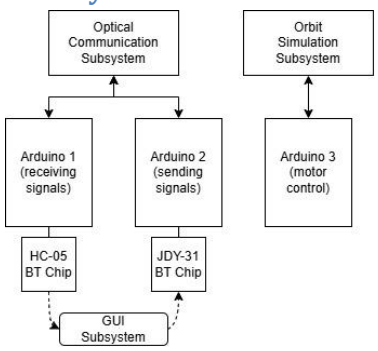

Figure 11 Data Processing and Control Subsystem Schematic

2.3.2 Bluetooth Module

The Bluetooth module is the core component ensuring wireless data transmission. There are two Bluetooth modules in our communication system, one of them is an HC-05 module responsible for sending the incoming signals from the user to the laser transmitter; the other one is a JDY-31 module responsible for sending the signals received from the optical receiver back to the GUI system to process and for users to view and crosscheck.

The two Bluetooth modules are directly connected to the microcontroller through wires and the user's computer through Bluetooth protocols, linking external user devices and our Optical Communication Subsystem.

2.3.3 Arduino (Motor Driver, Transmitter Control, Optical Receiver Control)

The microcontroller consisted of three Arduino UNO R3 boards [1], handle signal sending and receiving, and motor controlling. One board receives target motor speed commands through serial peripheral interface (SPI) port from the user's computer. This board translates the command speed in RPM unit to PWM signals that can be processed by a PID controller on the board. It also receives signals from the built-in Hall sensors on the motors to decode real-time motor speeds. One receives byte data from the Bluetooth unit and translates it to HIGH/LOW signals. It then sends that signal to the laser transmitter. The other board is connected with the signal receiver and can translate the optical signal into 0/1 based binary data and send the data through Bluetooth to the user's computer. Figure 11 shows the schematic of of the Data Processing and Control Subsystem and how the Arduino board interact with other subsystems.

2.3.4 Data Processing and Control Design Alternative

To enhance microcontroller performance, we might replace the three independent Arduino UNO R3 boards and two external Bluetooth modules with one high-performance, Wi-Fi/BLE-enabled microcontroller board—for example an ESP32-S3 module or an STM32WB55 Nucleo, which has a built-in running operating system (FreeRTOS or Zephyr.) All motion control, optical modulation, and optical reception tasks can be executed as separate RTOS threads on the same chip, sharing memory and hardware timers while timing information can be accessed.

2.4 Power Subsystem

The Power Subsystem supplies and regulates electrical energy to all components in the system, including the microcontroller, motors, sensors, and communication modules. The figures below show how batteries are placed in our design.

2.4.1 Power Subsystem Schematics

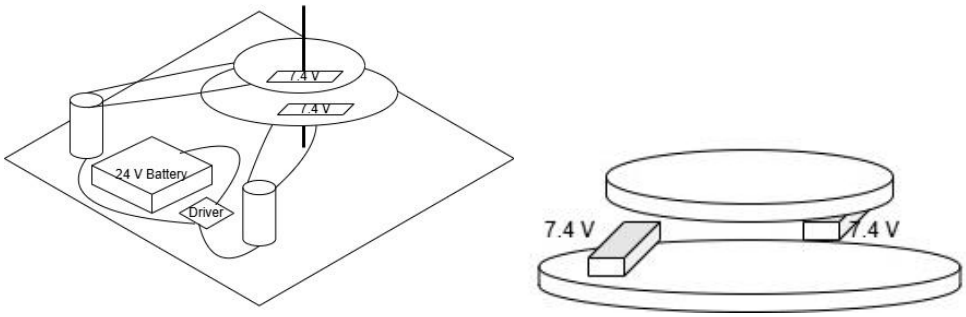

Figure 12 Power Subsystem Schematics

2.4.2 24 V Li-ion Battery

The system is powered by a 24 V 12000 mAh Li-ion phosphate battery. The battery outputs 22.4 V nominal voltage, with a full charge voltage of 25.55 V and a cut-off at 16.5 V. It supports a continuous discharge current of 15 A, ensuring stable and safe delivery under all load conditions. A voltage regulator is used to step down from the 24 V to 12 V and 5 V for different subsystems. The battery features built-in protection against short-circuit, over-current, over-charge, and over-discharge, providing reliable operation across a wide temperature range (−20 °C to 60 °C.) This component was bought, and we do not have its circuit schematic. Since the motors run at 24 V, there are no design alternatives for this module.

Figure 5 24 V Li-ion Battery

2.4.3 7.4 V Li-ion Battery

We have two 7.4 V Li-ion battery modules to power the Arduino boards on disks. These modules utilize two 3.7 V 18650 Li-ion batteries each. The 18650 battery has a normal capacity of 2500 mAh. This component was bought, and we do not have its circuit schematic. Since this module is portable and provides enough voltage to run Arduino boards, there are no design alternatives.

Figure 6 7.4 V Li-ion Battery Module

2.4.4 Power Design Alternative

No component alternatives can be found for this subsystem, we believe we are using the optimized approach. The 24 V battery is necessary to run the 24 V motors. The 7.4 V battery module is portable and chargeable, and it can power the Arduino boards well. However, there might be better ways to fix the 7.4 V battery module on the disks. We used rubber tape to tie the battery on the disk, and it was not very stable. A good alternative is to use screws to nail the module on the disk.

2.5 Optical Communication Subsystem

The Optical Communication Subsystem enables optical data transmission between the rotating Earth disk and the satellite disk using visible laser beams. It consists of a laser transmitter mounted on the Earth disk and an optical receiver on the satellite disk. The system operates at 5 volts and uses TTL/GPIO control from the microcontroller to receive and send data. The below figure shows how the laser transmitter and the photoreceptor module are placed on the disks.

2.5.1 Optical Communication Subsystem Schematics

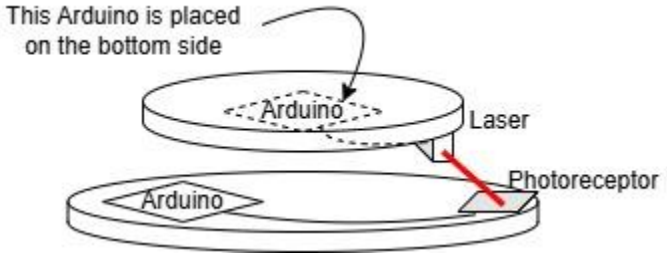

Figure 7 Optical Communication Subsystem Schematic

2.5.2 Laser Transmitter

The laser transmitter is powered by a 200 mW 660 nm laser beam emitter. It has a maximum 2,000 meters projection range indoor, and an adjustable light spot diameter large enough to cover the photoreceptor. We tuned the light spot diameter to be about 1 cm. Since the distance between the two disk edges is 10 cm in our case, this gives us an angle of about 6 degrees that the laser can cover. The laser transmitter works best at 2.8-3.2 V voltage. This component was bought, and we do not have its circuit schematic. The figure below shows what our laser transmitter looks like and how it is placed in our physical model.

Figure 8 Laser Transmitter

2.5.3 Optical Receiver

The optical receiver uses a photoreceptor module to detect incoming laser signals from the Earth disk. It integrates a light-sensitive transistor with an onboard relay and comparator (LM393,) enabling digital signal conversion based on light intensity. The module operates at 3.3 to 5 V. We used the digital output

(DO) pin of our photoreceptor module V, so that it generates output when the received light exceeds a pre-defined threshold. This receiver is critical for establishing line-of-sight optical communication and relaying signals to the decoding subsystem. This module is fulfilled by a printed circuit board (PCB) customized by us. The following two figures show the schematic of this PCB and the actual PCB.

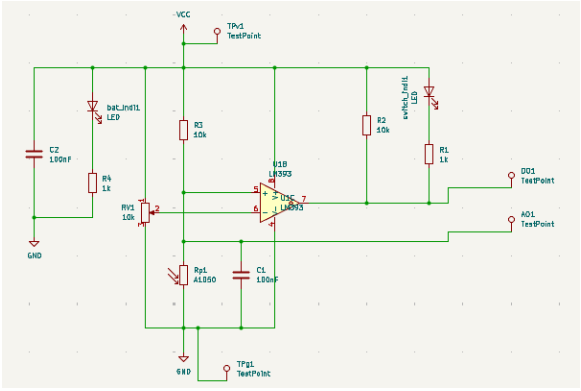

Figure 9 PCB Schematic

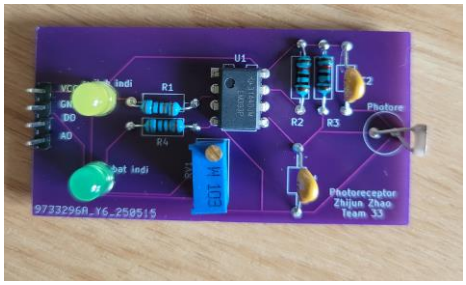

Figure 10 Photoreceptor PCB

2.5.4 Optical Communication Design Alternative

The alternative design is to use photodiodes instead of photoreceptors. They are more sensitive to light change, and they only let current pass when light is shone, creating more clear digital signals than photoreceptors. We can also use surface mount resistors and capacitors to make the PCB more concise and cleaner. In our current design, we use vertical potentiometer, whose resistance is hard to adjust. We could switch to horizontal potentiometers.

3. Design Verification

3.1 Graphical User Interface (GUI) Subsystem

The verification of the GUI subsystem can be divided into two steps, verifying input reliability and output reliability.

3.1.1 Input

Table 3 Verification of GUI Input

Requirements	Verification	Verification Status
1. It should have two vertical sliders that allow users to modify the two motors' rotational speeds from 0–60 RPM.	1. Verification for Requirement 1: a. The two sliders show clearly on the GUI window and can be pushed up and down by users. b. The motors can run at any RPM between 0 and 60.	Confirmed. Motor RPMs accurately follow GUI settings with < ±1 RPM deviation.
2. It can control laser pulse frequency between 100 Hz to 1 kHz.	2. Verification for Requirement 2: a. It should have a knob or a text box in which the user can adjust laser frequency. b. The laser transmitter blinks at any frequency between 100 Hz to 1 kHz.	Confirmed. Laser modulation frequencies match input with < 1% error.

3.1.2 Signal Encoder

Table 4 Verification of Signal Encoder

Requirements	Verification	Verification Status
1. It must be able to transform a message consisting of fewer than 16 characters into the desired 0/1 signal.	2. Verification for Requirement 1: a. Feed "hello" to the encoder over UART. b. Probe the encoder output with a logic analyzer. c. Check if each frame = **11111 (start) +0 + 5-bit letter index +0 + 4-bit position** and match the expected values for h-e-l-l-o.	Confirmed. All 16-bit encodings match specification with no bit errors.

3.1.3 Signal Decoder/Buffer

Table 5 Verification of Signal Encoder

Requirements	Verification	Verification Status
1. It must be able to decode a designed 16-bit signal into the correct character.	2. Verification for Requirement 1: a. Inject five known 16-bit sequences of each letter of "hello" into the decoder. b. After reception, query the decoder buffer, which should hold **h-e-l-l-o** with matching 0-4 index tag.	Confirmed. All 5 characters successfully decoded and reassembled in correct order.

3.1.4 Output

Table 6 Verification of GUI Output

Requirements	Verification	Verification Status
1. It should have a motor RPM plot. The	1. Verification for Requirement 1: a. Timer measured GUI update interval at 100 ± 3 ms.	Confirmed. Plot refresh rate is 102 ms, within target

plot refreshes every 100 ms.	b. PM plot reflects real-time input and updates on motor change within 0.1 s.	range. Data matches live motor speed with error < ±3 RPM.
2. It should have a decoded text plot. The plot clearly shows the decoded text message.	2. Verification for Requirement 2: a. Send the message "hello" using the GUI input field. b. Confirm that the decoded characters appear in the correct order with no missing or repeated characters.	Confirmed. "hello" is fully displayed after transmission. Characters appear clearly in the correct sequence.

3.2 Orbit Simulation Subsystem

3.2.1 Speed Sensor

In the verification process, the GMR speed sensor was mounted on the motor shaft and connected to an Arduino microcontroller. The microcontroller counted pulse signals to calculate real-time RPM. These values were wirelessly transmitted to the GUI via a Bluetooth module and compared against tachometer readings for accuracy verification. The motors can run across the full 10 to 250 RPM range and the sampling rate at 350 RPM is approximately 2.1 kHz. The requirements are verified.

Table 7 Verification of the Speed Sensor

Requirements	Verification	Verification Status
1. The range of the speed detected should be within 10 to 250 RPM.	1. Let the motor run at any speed between 10 to 250 RPM, and the sensor can pick up the speed with ≤ ±3 RPM error.	Confirmed. At 10 RPM → 8.6 RPM (error: 1.4 RPM) At 60RPM → 61.3 RPM (error: 1.3 RPM) At 200 RPM → 197.4 RPM (error: 2.6 RPM)
2. The Sampling rate should be greater than 1 kHz.	2. Measure the actual sampling rate	Measured sampling interval = 930 µs Sampling rate = 1075 Hz (> 1 kHz).

3.2.2 Gear Motor

In the verification of gear motor, a 24 V DC bench power supply was used to drive the gear motor. A torque arm with suspended weights applied a 10 kg · cm load to simulate real operating conditions. A digital tachometer measured the motor's rotational speed, while a multimeter monitored the current drawn to ensure it remained within the 0.4 A limit. We detected a 23.97 V voltage and a 0.37 A current. The motor could maintain the desired rotation speed when a 10 kg · cm load is enforced. All requirements were verified.

Table 8 Verification of the Gear Motor

Requirements	Verification	Verification Status
1. The motors should operate at the voltage of 24 V.	1. Measures the voltage reading when the motor is working and check if it is within ±0.5 V.	Confirmed. Measured voltage: 23.97 V, within range.

2. The rated current draw should be less than 0.4 A.	2. Measure the current when the motor is working and make sure it is less than 0.4 A.	Confirmed. Measured current: 0.37 A, meets requirement.
3. The output torque should be greater than 10 kg · cm.	3. Put a 10 kg · cm load on the motor and see if it still rotates normally.	Confirmed. Motor rotated under 10 kg·cm load.

3.2.3 Motor Driver

Table 9 Verification of the Gear Motor

Requirements	Verification	Verification Status
1. The driver must support 3.3 V to 5 V logic signals.	1. Connect Arduino digital output pins (3.3 V and 5 V) to motor driver control inputs (IN1, IN2, ENA). 2. Observe whether motor direction and speed are properly controlled at both logic levels.	Confirmed. 3.3 V/ 5.0 V logic: Motor direction and enable worked reliably. Direction toggled successfully under both voltage levels.
2. The driver must accept PWM input and modulate motor speed accordingly.	3. Varied PWM duty cycle from 0% to 100%; motor speed responded linearly and smoothly.	Confirmed. At 0% → 0 RPM At 25% → 65 RPM At 50% → 125 RPM At 75% → 184 RPM At 100% → 247 RPM

3.3 Data Processing and Control Subsystem

3.3.1 Bluetooth Module

Table 10 Verification Bluetooth Module

Requirements	Verification	Verification Status
1. It must work under a nominal voltage of 3.3 V and be able to send the signal through Bluetooth and the device's serial.	1. Verification for Requirement 1: b. Power the HC-05 and JDY-31 modules with a regulated 3.3 V supply. c. Pair the module with a laptop/phone and send the ASCII test string “hello” from the microcontroller over its UART pins at 9600 bps. d. Confirm the same string is received on the paired device’s Bluetooth serial monitor, then loop back a response and verify it on the microcontroller serial console.	Confirmed. Supply Voltage: 3.28 V (±0.02 V.) Test message “hello” received on laptop < 20 ms after transmission. Loopback received within 18 ms. Round-trip delay < 40 ms. Communication was successful for both HC-05 and JDY-31.
2. It must be able to correctly update the received data while setting the bit duration to 2ms.	2. Verification for Requirement 2: a. Configure the remote paired device to transmit a continuous 500 bps (2 ms per bit) square-wave test pattern. b. Capture the HC-05’s and JDY-31’s UART outputs with a logic analyzer; measure	Confirmed. Measured bit durations within ±2% of 2 ms. Full test string reconstructed with 4 errors across 100 frames.

	individual bit widths to ensure they are 2 ms ± 10 %. c. Verify that the entire test message is reconstructed without framing on the device's serial monitor.	

3.4 Power Subsystem

3.4.1 24 V Li-ion Battery

Table 11 Verification of 24 V Li-ion Battery

Requirements	Verification	Verification Status
1. It must supply a nominal voltage of 22.4 V and a full charge voltage of up to 25.55 V.	1. Verification for Requirement 1: e. Fully charge the battery using the specified charger. f. Measure voltage using a multimeter. g. Confirm reading is within 22.4 V to 25.55 V.	Measured voltage after full charge: 25.4 V, within specified 22.4 V–25.55 V range.
2. It must provide at least 15 A continuous discharge current.	2. Verification for Requirement 2: d. Connect the battery to an electronic load. e. Gradually increase current draw up to 15 A. f. Confirm that the voltage remains stable and no protection is triggered.	No undervoltage, thermal, or current protection triggered. Voltage drop < 3.0 V from full charge under 15 A load.

3.5 Optical Communication Subsystem

3.5.1 Laser Transmitter

Table 12 Verification of Laser Transmitter

Requirements	Verification	Verification Status
1. It must operate at 5 V DC and draw no more than 300 mA current.	1. Verification for Requirement 1: a. Connect the transmitter to a 5 V power supply b. Measure current using a multimeter c. Confirm current ≤ 300 mA	Confirmed. Measured current: during modulation: 261 mA < 300 mA.
2. It must achieve an indoor transmission distance of at least 5 meters at 10 mW.	2. Verification for Requirement 2: a. Set transmitter to 10 mW output b. Place the receiver 5 meters away c. Confirm stable signal detection at receiver	Distance: 0.50m Received signal voltage: 2.47 Vpp BER = 0% over 10,00 bits. Signal stable over 5 min.

3.5.2 Optical Receiver

Table 13 Verification of Optical Receiver

Requirements	Verification	Verification Status
1. It must stably operate at 12 V Direct Current (DC.)	1. Verification for Requirement 1: a. Connect the module to a 12 V DC power supply	Confirmed. Measured voltage: 5.01 V ± 0.05 V

	b. Use a multimeter to confirm stable voltage across VCC and GND pins	No brownout observed under continuous operation.
2. It must detect red laser light (650 nm) at ≥1 m indoors	2. Verification for Requirement 2: a. Place the receiver module 0.3 m away from the laser transmitter b. Align laser beams directly at the sensor Modulate signal at 1.25 kHz square wave. c. Monitor DO pin output on oscilloscope	Distance: 0.30 m, Laser wavelength: 650 nm, Frequency: 1.25 kHz Oscilloscope shows clean square wave at receiver DO pin: Vpp = 3.46 V Response delay < 0.1ms

4. Costs

There are two parts of costs from our project, parts that we need to assemble our model, and the human labor costs. Below includes a detailed breakdown of the parts costs and the labor costs.

4.1 Parts

Table 14 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
24 V Li-ion battery	WheelTec	23.25	10.32	23.25
Arduino UNO x 3	Zave	13.99	6.99	9.99
Dual H-bridge motor driver	WheelTec	12.00	10.50	12.00
35 W DC motors × 2	WheelTec	40.33	26.40	31.00
5 mW laser	Laser Factory	14.00	10.00	13.00
200 mW laser	Zhonglai	29.99	29.99	29.99
HC-05 and JDY-31 Bluetooth Module	DZQJ	5.29	3.50	4.25
PP disks × 2	Dongxing Plastic	20.15	13.50	18.50
Customized PCB	JiaLiChuang EDA	10.70	4.19	7.80
spindles & enclosure	Qianzhilai	0.99	0.99	0.99
3D printed gear × 4	N/A	0.00	0.00	0.00
PVC base	Dongxing Plastic	15.20	9.00	15.20
Wood disk	Zhixing	5.82	5.82	5.82
Total				**171.79**

4.2 Labor

According to [4], the median salary for a Grainger graduate is $85,000. This translates to an hourly stipend of $40.87. Let's take the floor and use $40 as an estimate of how much we will earn per hour after we graduate from the University of Illinois Urbana-Champaign. Assuming everyone spends 10 hours on this project per week. Given that this project started on February 24 and will be presumed to end on May 26, it lasts approximately 14 weeks. Use C to denote the total labor cost, S to denote the estimated hourly stipend for a Grainger graduate, and T to denote the time each team member spends on the project every week. Since we are a team of four, the total labor cost of our team would be:

$$C = 4 * 14 * S * T \tag{4.1}$$

Substitute S with the estimated hourly stipend number and T with 10, an estimated total labor cost for this project would be $22,400.

4.3 Schedule

The table below shows the detailed weekly schedule of what each member of our team should do.

Table 15 Detailed schedule by week and member

Week (Mon--Sun)	Xuanyi Jin	Yuxuan Li	Zhijun Zhao	Jun Zheng
3/17 – 3/23	Finalize CAD for inner and outer disks	Order motor driver	Select laser and photoreceptor	Sketch GUI frame, set up PySerial project
3/24 – 3/30	Mount disks and motor, test disk spinning at 30 RPM	Write PWM displaying program	Build laser driver and measure beam divergence	Implement live motor speed plot
3/31 – 4/6	Mount optical encoder wheels, check safety issues	Write PID control for motors	Design PCB for photoreceptor	Design motor speed control logic
4/7 – 4/13	Prepare mechanical drawings for final 3-D printed gears	Test PID control for motors and adjust parameters	Test photoreceptor PCB	Design motor speed reading CSV output
4/14 – 4/20	Mount disks on gears and placed on base	Test motor with disks mounted	Mount photoreceptor PCB	Design decoded message displaying window
4/21 – 4/27	System-level fit check, check Li-ion battery safety	Set motor autotune	Test photoreceptor module sensitivity	Design message encoding formats
4/28 – 5/4	Test disk rotation under external vibration	Final optimization for motor drivers	Test photoreceptor and laser transmitter working in pair	Design message receiving efficiency report
5/5 – 5/11	Prepare assembling guide and risk-mitigation documents	Review motor driver codes	Finalize photoreceptor PCB adjustment	Finalize GUI appearance
5/12 – 5/18	Rehearse final demo	Prepare driver release package	Finalize the encoder and decoder program	Integrate signal encoder/decoder to GUI
5/19 – 5/25	Final function test before demo, demo, and final presentation	Final function test before demo, demo, and final presentation	Final function test before demo, demo, and final presentation	Final function test before demo, demo, and final presentation
5/26 – 5/31	Work on report, prepare for exhibition	Work on report, prepare for exhibition	Work on report, prepare for exhibition	Work on report, prepare for exhibition

5. Conclusion

5.1 Accomplishments

Through this project, we successfully designed and implemented a 2-D Optical Satellite Communication System that physically simulates Earth-to-LEO satellite communication using rotating disks and laser-based optical links. The system integrates five functional subsystems: orbit simulation, optical communication, power management, data processing and control, and a user interface. We achieved precise motor control with speed feedback from GMR sensors, reliable optical transmission and reception using a 660 nm laser and photosensitive diode, and real-time monitoring through a GUI.

For the signal transmission, it can transmit signal “hello world” in 2.24s. It is also able to transmit any message within 15 characters in a short time.

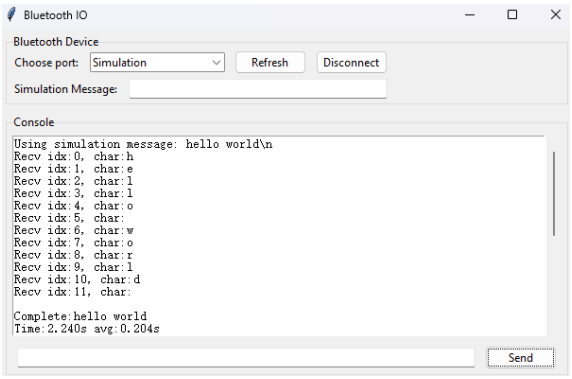

Figure 11 Successful Message Transmission

Our GUI successfully interact with users so that users can input the target rotation speed for the system and adjust the speed. Besides, the real time ration speed will also be displayed to the users in GUI.

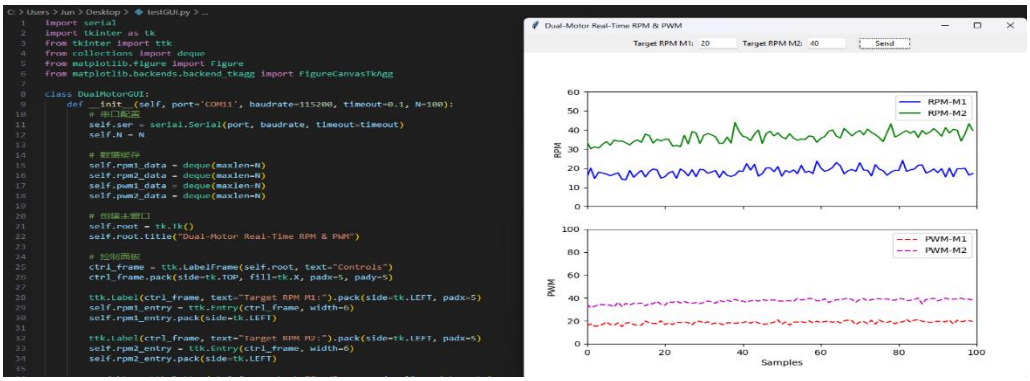

Figure 20 Rotation Speed Display GUI

Our model effectively demonstrates key challenges in satellite communication, such as alignment sensitivity and orbital motion, while remaining cost-effective and intuitive for educational or research purposes.

5.2 Uncertainties

While the prototype performs reliably under controlled conditions, several uncertainties remain that may affect system robustness and scalability. First, the optical communication subsystem is highly sensitive to mechanical misalignment and vibration. Due to the rotating structure, the satellite disk can experience vertical oscillation (up-and-down motion), which occasionally causes the laser to miss the target receiver, resulting in failed transmissions. Additionally, because we employ only a single transmitter-receiver pair and rely on a brief angular time window for alignment, the timing of successful transmission is not consistent. In some cases, data is received within seconds; in others, it may take over a minute, especially as rotational speed increases and the alignment window becomes even narrower.

Moreover, the use of low-cost sensors and components introduces signal inconsistencies at higher speeds. The mechanical instability of the disks under rapid rotation further amplifies these challenges, affecting both the accuracy of speed measurements and the alignment precision of the optical path. Although Bluetooth wireless communication functioned reliably in testing, it may face limitations in high-interference environments or when deployed at larger physical scales.

Finally, the long-term performance and stability of the system under extended operation have not yet been fully evaluated. These uncertainties highlight the need for further testing across a broader range of environmental and dynamic conditions, as well as possible hardware refinements to improve mechanical stability and communication reliability.

5.3 Ethical and Safety Considerations

5.3.1 Ethical Considerations

Our project mainly involves a controlled, simplified 2-D simulation of Optical Satellite Communication System which has little risk of ethical violations. It does not involve personal data handling, privacy risks, or environmental influence. We have reviewed our responsibilities thoroughly according to IEEE [5] and ACM [2] Codes of Ethics. Besides, because our model is used purely for educational and demonstrative purposes, concerns like conflicts of interest or discrimination are negligible. Our team fully understands the IEEE and ACM Codes of Ethics and will follow them throughout the project's lifecycle.

5.3.2 Mechanical Safety

Our model features two rotating disks driven by gear motors, which simulate the relative motion of Earth and orbiting satellites. To reduce mechanical risks, we implemented physical safety guards around the rotating parts, clearly labeled all moving components, and maintained conservative speed limits during testing. These precautions ensured that no physical harm or equipment damage occurred during operation.

5.3.3 Laser Safety

The system uses a 200 mW 660 nm red laser to emulate optical communication. This is a Class 3b laser, which users, according to ANSI [3], should avoid direct beam contact.

5.3.4 User Safety

All development activities followed the University's laboratory safety policies, including mandatory training, appropriate personal protective equipment, and documented safety protocols. A risk assessment was conducted before testing began, and all procedures were performed in accordance with lab guidelines. Since the system is primarily for demonstration and education, it poses little risk to end users.

5.4 Future Work

Future iterations of this system can focus on increasing precision and stability. Mechanically, improvements could include more rigid mounting systems, laser focus control, and vibration reduce. From a software standpoint, expanding the GUI to include real-time error display and automated alignment feedback would enhance usability. Additionally, replacing the Bluetooth module with a more robust wireless protocol such as Wi-Fi could improve communication reliability. Finally, the system could be adapted into a full 3D model or integrated into a networked satellite simulation platform to more accurately mimic real inter-satellite links.

References

[1] Arduino, “UNO R3,” Arduino Docs. Accessed: Apr. 14, 2025. [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3/.

[2] Association for Computing Machinery, “ACM Code of Ethics and Professional Conduct”, ACM. Accessed: April 5, 2025. [Online]. Available: https://www.acm.org/code-of-ethics

[3] B. Kelechava, “Laser Safety: Class 1, 1C, 1M, 2, 2M, 3R, 3B, and 4,” The ANSI Blog, Sep. 14, 2018. Accessed: May 25, 2025. Available: https://blog.ansi.org/2018/09/laser-class-safety-1-1c-1m-2-2m-3r-3b-4/

[4] Illini Success, “2023–2024 Annual Report,” University of Illinois Urbana-Champaign. Accessed: April 5, 2025. [Online]. Available: https://illinisuccess.illinois.edu/23-24-annual-report

[5] Institute of Electrical and Electronics Engineers, “IEEE Code of Ethics”, IEEE. Accessed: April 5, 2025. [Online]. Available: https://www. ieee.org/about/corporate/governance/p7-8.html