Design of Low-Cost Remote Labs using IoT-based Retrofitting

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communication Engineering
by Research

by

Kandala Savitha Viswanadh
2019112011
savitha.viswanadh@research.iiit.ac.in

Advisor: Dr. Sachin Chaudhari

International Institute of Information Technology
Hyderabad - 500 032, India

May 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

This is to certify that work presented in this thesis proposal titled ***Design of Low-Cost Remote Labs using IoT-based Retrofitting*** by *Kandala Savitha Viswanadh* has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Advisor: Dr. Sachin Chaudhari

To

My Family and Friends

Acknowledgement

My journey at IIIT Hyderabad has been nothing short of extraordinary, marking a profound shift for a young individual from the city of destiny, Vizag, to the vibrant metropolis of Hyderabad and the enchanting campus of IIIT. The memories and lessons learned here will forever shape the trajectory of my life.

My deepest gratitude is reserved for my research advisor, Dr. Sachin Chaudhari. His unwavering motivation, guidance, and provision of resources have propelled my research to new heights. His trust and belief in my capabilities transformed me into a leader, adept at managing research groups and handling resources—qualities indispensable for life beyond academia. I thank Dr. Venkatesh Choppella, Dr. Prabhakar Bhimalapuram, Dr. Harikumar Kandath and Dr. Aftab Hussain for their invaluable contributions to our research project. A special acknowledgement goes to Dr. Karthik Vaidhyanathan for his steadfast support and guidance. My gratitude extends to the IT office, R&D office, Hostel Manager, and the Academics office for their indispensable support during my tenure at IIIT.

I am grateful to my project mates Rishabh, Akshit, and Animesh. Special thanks to my friends at SPCRC—David, Om, Nagesh, Shreyas, and Ritik—for the joyous moments we shared. My deepest thanks go to my lab seniors Ayush Dwivedi, Rajasekhar Reddy, Spanddhana, Ihita, Usha, Nitin, Rishi, and Nilesh, with whom I established excellent knowledge connects. I extend my best wishes to the new researchers—Nagesh, Abhinav, Vedant, Krishna, and Sannidhya.

A heartfelt acknowledgement goes to my batchmates—Sasanka U (Heartbeat), Niteesh (Companion), Sasanka GRS (Battle Sage), Avadhani (Catalyst), and Siva (Zen Master)—whom I consider my extended family. Without them, the journey would have taken a markedly different course. Special shoutouts to Harshitanjani and Manaswini for being guiding lights during challenging times. Thanks also to my old buddies Vikram (Maestro) and Pavan (Lifeline) and my friends groups- “529” and “MMVJ”. To everyone else who supported me, please forgive my limited memory.

My parents, Srinivas and Lalitha Kandala, and my brother Viswakanth Kandala will hold a special place throughout this life. None of this would have been possible without their love, encouragement, and blessings. I will cherish their belief in me forever. Special thanks to my cousins Sai, Vangmi, Ragini, Gayatri, Vasudha, Bharadwaj, Anirudh, Madhu, Keervani, and Gopal, along with other family members, for the constant love and affection they showered upon me throughout this transformative journey.

Abstract

Remote labs are a groundbreaking development in the education industry, providing students with access to laboratory education anytime, anywhere in the world. This has proved to be particularly important during the recent COVID-19 pandemic, where remote access has become essential. However, most remote labs are costly and need more flexibility for institutions to replicate experiments. This becomes a significant concern when trying to scale up to accommodate more students in developing countries.

The work presented in this thesis primarily revolves around building hardware for a proposed end-to-end remote lab system (RLabs). This is presented in two parts: one is retrofitting Internet of Things (IoT) components to laboratory experiments, and the second is adopting Computer-Vision (CV) techniques in a remote lab experiment. The RLabs platform includes two use case experiments: *Vanishing Rod* and *Focal Length*. The hardware experiments are built at low cost by retrofitting IoT components. The proposed solution is qualitatively evaluated against seven non-functional attributes - affordability, portability, scalability, compatibility, maintainability, usability, and universality. Finally, user feedback was collected from a group of students, and the scores indicate a positive response to the student's learning and the platform's usability.

Additionally, CV techniques are explored and used for a use case experiment of *Conservation of Mechanical Energy* to improve the accuracy and reduce the dependence on sensors used for remote experimentation. These CV techniques track the velocity of an object in the experiment, which is otherwise found out from infrared (IR) sensors. Later, velocities calculated from the CV technique are compared with the traditional IR sensors. The calculated velocities are further made accurate by employing linear regression.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
CSI	Camera Serial Interface
CV	Computer Vision
DC	Direct Current
FL	Focal Length experiment
FoV	Field of Vision
FPS	Frames Per Second
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IIIT-H	International Institute of Information Technology - Hyderabad
IoT	Internet of Things
IR	Infrared
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MSE	Mean Square Error
NFA	Non-functional attribute
P2P	Peer-to-Peer
RaspiCam	Raspberry Pi Camera
RFID	Radio Frequency Identification
RL	Remote Labs
RLabs	Remote Labs at IIIT Hyderabad
RoI	Region of Interest
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
UI	User Interface
USB	Universal Serial Bus
VR	Vanishing Rod experiment
WiFi	Wireless Fidelity

Symbols

μ	refractive index
u	object distance
f	focal length
g	acceleration due to gravity
E_T	mechanical energy of an object
E_k	kinetic energy
E_p	potential energy
m	mass of an object
v	image distance/ instantaneous velocity
h	height of the object relative to an arbitrary reference level
v_i	velocity at an intermediate point
h_i	height at an intermediate point
h_1	height of a topmost point
Δh	height with respect to a topmost point

List of Related Publications

[P1] K. S. Viswanadh, A. Gureja, R. Agrawal, W. Nagesh, S. Sinha, S. Chaudhari, K. Vaidhyanathan, V. Choppella, P. Bhimalapuram, H. Kandath, and A. Hussain, **"Engineering End-to-End Remote Labs using IoT-based Retrofitting"**, Submitted to ACM Transactions on IoT, December 2023

[P2] K. S. Viswanadh, O. Kathalkar, P. Vinzey, N. Nilesh, S. Chaudhari and V. Choppella, **"CV and IoT-based Remote Triggered Labs: Use Case of Conservation of Mechanical Energy"**, *IEEE International Conference on Future Internet of Things and Cloud (FiCloud)*, 2022.

Related co-author publications:

[P3] Animesh Das, K. S. Viswanadh, Rishabh Agrawal, Akshit Gureja, Nitin Nilesh and Sachin Chaudhari, **"Using Miniature Setups and Partial Streams for Scalable Remote Labs",** IEEE International Conference on Future Internet of Things and Cloud (FiCloud), 2023.

Related Patents Filed:

[Pt 1] K. S. Viswanadh, Nitin Nilesh, Om Kathalkar, Sachin Chaudhari, Venkatesh Choppella, **"System and Method for Implementing an Experiment Remotely Using a Computer Vision Technique"**, US Patent US20240078641A1, September, 2022

[Pt 2] K. S. Viswanadh, Rishabh Agrawal, Sachin Chaudhari, **"REFRACTION DETECTING ROD"**, India Design Patent Appl. Num. 389763-001, July, 2023

[Pt 3] K. S. Viswanadh, Rishabh Agrawal, Sachin Chaudhari, **"Design of Focal Length"**, India Design Patent, 2023

Other co-author publications:

[P4] G. V. Ihita, K. S. Viswanadh, Y. Sudhansh, S. Chaudhari and S. Gaur, **"Security Analysis of Large Scale IoT Network for Pollution Monitoring in Urban India"**, *IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021.

Chapter 1

Introduction

1.1 Motivation

The Internet of Things, or IoT, is a pioneer in transforming the current landscape by regularly innovating new solutions. Presently, approximately 15 billion devices are connected to the internet, a number expected to double to around 29 billion by 2030 [1]. We are approaching a time when virtually all items in our surroundings will be internet-enabled. The widespread connectivity of the internet, facilitating data transfer globally, has given rise to advancements in remote control capabilities and real-time data analytics. Soon, the ability to control various objects from any location will become a reality, including motor cars, drones, and other devices. This would have applications in many domains like healthcare, education and smart cities. This transformation is further accelerated by the ongoing evolution of 5G and beyond networks, bringing about low-latency and high-bandwidth connectivity. As the IoT ecosystem progresses, the integration of global connectivity, remote control, and real-time data exchange will reshape how we interact with our environment.

Out of many applications, IoT also has the potential to revolutionise traditional learning methodologies. Presently, IoT is utilised to enhance existing pedagogical approaches, aiming to improve learning outcomes through innovative methods by introducing smart classrooms. One interesting avenue in this would be the exploration of the confluence of IoT in laboratory education. Laboratory-based learning holds significant importance in the educational journey, providing students with practical applications of theoretical knowledge. The hands-on nature of laboratory work fosters essential skills such as critical thinking, problem-solving, and scientific inquiry. However, replicating the experiential nature of laboratory education through the internet poses challenges. Recent global events, such as the COVID-19 pandemic, have further exacerbated this issue, depriving many students of traditional laboratory experiences. In response, educators have sought innovative alternatives to ensure the continuity of education, with traditional classroom settings transitioning to online platforms. Despite these efforts, the accessibility of laboratory education has remained limited. Virtual Labs emerged as a partial solution, offering online simulations of experiments conducted in a physical laboratory. However, these simulations need the authenticity of real-life errors and nuances.

An emerging solution to bridge this gap is the implementation of Remote Labs. In this approach, laboratory experiments are modified to be controlled remotely through the Internet anytime, anywhere. This allows students to experience an authentic and immersive laboratory experience, overcoming the limitations of traditional online simulations. However, most remote labs are costly and need more flexibility for institutions or individuals to create and replicate the experiments. This issue becomes particularly significant when attempting to scale up to accommodate a larger number of students, especially in densely populated developing countries like India.

Taking this as the motivation, an effort is made in this thesis to present a hardware perspective of building an end-to-end remote lab solution where the remote experiments are built by retrofitting low-cost IoT components to the traditional laboratory equipment.

1.2 Summary of contributions

The main contributions of this thesis are split into two chapters: Chapter 4 and Chapter 5

- **Chapter 4**
  - An end-to-end remote labs solution, known as **RLabs** (short for **R**emote **Labs**), has been implemented and deployed on the campus of the International Institute of Information Technology - Hyderabad (IIIT-H), India [1].
  - Two use case experiments, Vanishing Rods and Focal Length, based on high-school physics, are designed by retrofitting IoT components to the traditional laboratory equipment.
  - Additionally, a modular version of the Focal Length use case experiment is proposed and built to make the setups portable.
  - User feedback from forty-seven students is collected and reviewed, containing questions to rate their usability and learning outcomes after using the built RLabs system.
- **Chapter 5**
  - A CV and IoT-based remote labs implementation is proposed. The proof of concept is demonstrated for the physics experiment of *Conservation of Mechanical Energy*.
  - In this experiment, CV is used to estimate the velocity of a moving object along different points on the track, which are then shown to be close to the theoretical values found based on the principle of conservation of energy.
  - To improve the performance of CV-based velocity estimation of the moving object at a given point, linear regression is used to find the line of best fit for all the points on a straight track. Also, the implementation parameters are changed to observe their effect on estimated velocities.

[1] `https://www.iiit.ac.in/`

  - A performance comparison of CV-based implementation is carried out with IR–based velocity estimation. It is also shown that the CV-based implementation can estimate velocity at any location on the track without the need for a sensor.

1.3 Organisation of the Thesis

This thesis is structured as follows:

- **Chapter 2:** This chapter provides an overview of the IoT, outlining an architecture model with four components that are generalised for broader applicability. It also explores various applications of IoT and discusses the challenges encountered in this domain.
- **Chapter 3:** An introduction to the concept of remote labs is provided in this chapter, along with a comprehensive literature survey on this concept.
- **Chapter 4:** This chapter details developing a remote lab system at the IIIT-H campus. It thoroughly describes designing and creating cost-effective, retrofit hardware for two experimental use cases.
- **Chapter 5:** A novel application of CV techniques in remote labs is introduced in this chapter, proposing an alternative to traditional sensor-based methodologies.
- **Chapter 6:** This chapter discusses the various challenges encountered during the experimental setup phase. It also includes survey results reflecting the usage and impact of the developed remote labs platform.
- **Chapter 7:** The final chapter concludes the thesis and presents insights into the future directions of this research.

Note

For this thesis, the author was responsible for conceptualising the ideas, building and programming the hardware components in [P1], implementing the CV algorithms in [P2], and writing the manuscripts. In [P1], Akshit Gureja developed and designed the software platform for the RLabs system, and Nagesh created the automated testing pipeline for the RLabs platform. In [P2], Om and Piyusha built the hardware setup while Nitin guided the implementation of the CV algorithms. Prof. Sachin Chaudhari contributed to all the publications by reviewing the technical results and providing feedback on the manuscripts. The other co-authors helped provide insightful feedback and guidance.

Chapter 2

An Overview on IoT

This chapter presents a concise overview of IoT, covering its key applications and inherent challenges. It introduces an IoT system architecture consisting of essential components. This framework aims to clarify the interaction and integration of different layers in IoT deployment. Additionally, the chapter discusses the major challenges faced in the IoT domain. Elaborate discussions on IoT can be found in these textbooks [2–4].

2.1 Introduction

IoT is often described differently in the literature due to the diverse purposes for which IoT is used. IoT is about making devices "smart" by providing internet capabilities that enable networking and thereby expanding their overall functionality. Britannica [5] defines it as "the vast array of physical objects equipped with sensors and software that enable them to interact with minimal human intervention by collecting and exchanging data via a network." However, the objectives for data collection and exchange differ across various use cases.

IoT can involve objects that range from everyday gadgets like watches, fans, and TVs to complex industrial machinery and healthcare devices. Take a sound speaker and an Alexa Echo[1] for example. A sound speaker plays the music you tell it to from a computer. However, Alexa, a virtual assistant by Amazon[2], works with an Echo speaker connected to the internet and can play music using voice commands without a computer. The transformative power of IoT extends across a broad spectrum of our daily lives and industries worldwide, continually evolving and diversifying in its applications.

2.2 Components of IoT

IoT represents a complex and dynamic ecosystem composed of various interconnected components. These components work in unison to create effective and responsive IoT systems. For clarity, this thesis will explore a four-layer model of the IoT ecosystem [2], focusing on devices, cloud and data processing,

[1] `https://www.amazon.in/amazon-echo/b?ie=UTF8&node=14156834031`
[2] `https://www.amazon.com/`

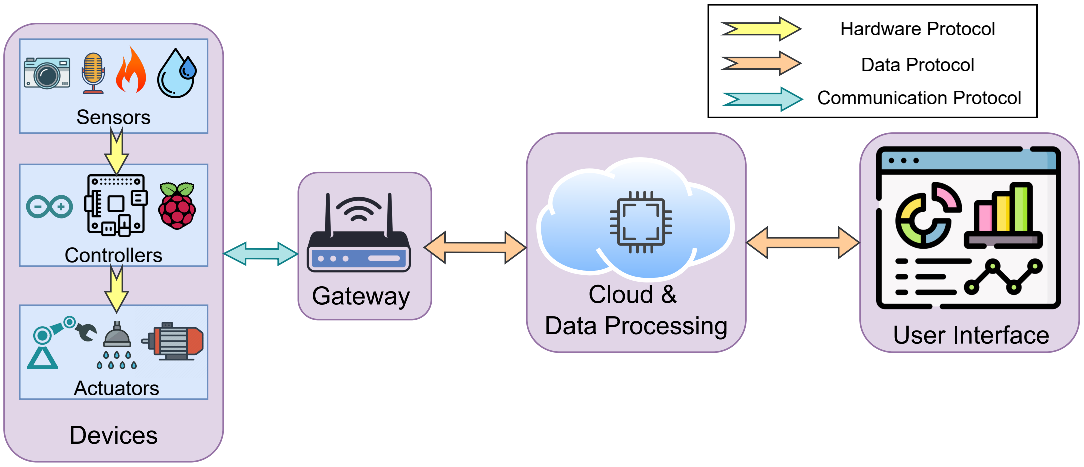


Figure 2.1 Overview of a four-layer IoT architecture

user interface, and connectivity. Fig. 2.1 shows the interactions between each layer. Each layer is crucial in ensuring seamless interaction and data flow within an IoT system and is explained next.

2.2.1 Devices

Devices in IoT are the tangible touchpoints where the physical world meets the digital. Their role involves sensing, processing, and acting upon various environmental inputs. These devices, often small and unobtrusive, consist of sensors, actuators, and controllers:

1. **Sensors**: Sensors are the eyes and ears of IoT devices, collecting vital data from their surroundings and converting it into digital form. This data can range from simple temperature measurements to complex visual inputs captured by advanced imaging sensors. The diversity in sensor technology is vast, with specific sensors designed to capture particular types of data. For instance, physical sensors measure tangible attributes like pressure or movement, while environmental sensors are tuned to detect changes in ambient conditions like humidity or air quality. The capability of these sensors to convert various physical inputs into quantifiable data forms the basis of IoT's interaction with the real world.

2. **Actuators:** Actuators are the counterparts to sensors and are responsible for converting digital commands into physical actions. They are the muscle of IoT devices, responding to commands based on sensor inputs or remote instructions. Actuators vary widely in their nature and application. For example, mechanical actuators might control the movement of a robotic arm, while fluidic actuators could regulate the flow in an industrial piping system. The diversity of actuators allows IoT systems to interact with and manipulate their environment in numerous ways.

3. **Controllers:** Controllers are the brains within IoT devices, orchestrating the operation of sensors and actuators. They process sensor data, make decisions based on programmed logic or remote commands, and direct actuators accordingly. Controllers range from simple microcontrollers,

adequate for basic tasks, to more powerful microprocessors capable of handling complex computations and edge computing. The choice of controller depends on the use case being dealt with, which considers factors like processing power, energy consumption, and cost. Examples include:

(a) Microcontrollers: They are compact circuits integrating a single processor core with memory, suitable for embedded control applications (e.g., Atmel AVR and Intel MCS-51).

(b) System on a Chip (SoC): A System on Chip (SoC) integrates all key computer components on a single chip, including the processor, memory, and peripherals, suitable for more computationally heavy tasks (e.g., Raspberry Pi, BeagleBone).

2.2.2 Cloud and Data Processing

In IoT, data processing transforms raw data from devices into actionable insights. With the vast volume of data generated by IoT sensors and controllers, effective data processing is essential. This processing can occur directly on the devices (edge computing) or be offloaded to cloud services, which offer more substantial computational resources. In the cloud, data can be further analysed using machine learning algorithms for predictive insights and enhanced service offerings. Cloud services are categorised into:

1. **Infrastructure as a Service (IaaS)**: Provides fundamental computing resources like servers and storage (e.g., DigitalOcean, Amazon Web Services).

2. **Platform as a Service (PaaS):** Offers a computing platform and solution stack for application development (e.g., Heroku, Microsoft Azure).

3. **Software as a Service (SaaS):** Delivers software applications over the internet (e.g., Adobe Creative Cloud, GitHub).

2.2.3 User Interface

The User Interface (UI) in IoT serves as the crucial point of interaction between the technology and its users. An effective IoT UI encapsulates the complexity of IoT systems into an intuitive and user-friendly format, enabling users to easily monitor, control, and understand the vast array of data and functions within the IoT ecosystem. Key attributes of a well-designed IoT UI include intuitiveness, allowing users to navigate and control systems with minimal training; responsiveness, ensuring compatibility and usability across various devices and screen sizes; real-time data presentation, which is vital for accurately reflecting the current state of IoT systems; and customisability, offering users the flexibility to tailor the interface to their specific needs and preferences.

A central feature of the IoT UI is the dashboard, which acts as a centre for monitoring and controlling the IoT environment. Dashboards typically display a wide range of information, from real-time data feeds to historical analytics, using an array of visual tools like graphs, gauges, and alarms. These tools help in distilling complex data into digestible and actionable insights. Additionally, control panels form a critical part of the UI, allowing users to interact with and manipulate IoT devices and systems directly. These controls can range from simple toggles to intricate settings, enabling users to adjust and optimise the performance of their IoT ecosystem according to their requirements. Overall, the UI in IoT not only bridges the gap between humans and machines but also enhances the usability and accessibility of the IoT system, making technology more adaptable and beneficial for a diverse range of users.

2.2.4 Connectivity and IoT Gateway

Connectivity in the IoT ecosystem is a multifaceted domain, comprising various methods and protocols that facilitate communication within IoT systems. This section explores the mechanisms through which devices and systems interact, ensuring the seamless transmission of data from sensors to actuators, controllers, and the cloud. Central to this communication framework is the IoT gateway, serving as a pivotal hub in IoT networks. It not only translates and routes data between devices operating on different protocols but also provides processing and storage capabilities. Additionally, IoT gateways enhance network security by managing data encryption and secure device authentication. Data transfer in IoT environments is carried out using a diverse range of protocols, each with a specific function:

1. **Hardware Protocols:** These protocols are concerned with the physical and electrical standards for data transmission within IoT devices. Examples include UART, SPI, and I2C. UART (Universal Asynchronous Receiver/Transmitter) enables serial communication, SPI (Serial Peripheral Interface) offers a fast data transfer interface, and I2C (Inter-Integrated Circuit) provides a method for multiple devices to communicate with each other over a single bus [6]. These protocols define how devices physically connect and interact at a hardware level.

2. **Communication Protocols:** These protocols enable connectivity between IoT devices and internet gateways. Examples include various technologies like WiFi (IEEE 802.11), Bluetooth Low Energy (BLE) (IEEE 802.15.1), and Zigbee (IEEE 802.15.4). These protocols enable wireless communication, offering various technologies with different characteristics. WiFi provides high-speed connectivity, BLE is energy-efficient for low-power devices, and Zigbee is suitable for low-data-rate, short-range communication.

3. **Data Protocols:** These protocols are associated with the application layer of the OSI model [7] and play a crucial role in the transmission of data between IoT devices, gateways, and cloud services. Prominent among these are HTTP (Hypertext Transfer Protocol), MQTT (Message Queuing Telemetry Transport), and CoAP (Constrained Application Protocol). HTTP is widely used in web-based communication and enables standard internet data exchange. MQTT, optimised for low-bandwidth and lightweight messaging, is particularly effective in environments with constrained connectivity. Similarly, CoAP is tailored for constrained devices, catering to their specific needs. Both MQTT and CoAP are known for their performance efficiency in scenarios with limited connectivity [8]. The strategic selection of these application layer protocols

is of great importance as it profoundly influences the efficiency, reliability, and scalability of the IoT system.

Together, these elaborations provide a comprehensive view of the integral components of IoT, highlighting their unique roles and interdependencies within the broader ecosystem.

2.3 Applications of IoT

IoT has revolutionised how we interact with the world around us, offering vast applications in numerous fields, primarily due to its seamless integration with other advanced technologies like Artificial Intelligence (AI), Machine Learning (ML) and Computer Vision (CV). IoT's ability to connect everyday devices to the internet and each other has opened up a world of possibilities, allowing for more efficient, automated, and personalised experiences across various sectors.

1. **Smart Cities:** IoT plays a crucial role in the advancement of smart cities where various aspects are parameters and analysed to make better decisions. These technologies enable proactive measures to combat pollution, enhancing the environmental health and sustainability of urban areas that benefit and improve citizens' living. Various aspects can be analysed, and this includes:

   (a) **Air Quality:** Air quality poses a significant global concern, with many countries grappling with severe pollution issues. Monitoring stations are in place to detect concentrations of various air pollutants, including PM2.5, NO_x, and SO_x. In India, the Central Pollution Control Board[3] (CPCB) monitors these parameters in different regions of the country. However, these monitoring stations are limited in number and may not provide real-time updates. To address this, multiple IoT devices can be deployed. These devices can be stationary [9, 10] or mobile, such as those attached to vehicles [11], to enable real-time monitoring of air quality parameters. Additionally, indoor air pollution can be monitored [12] to enhance the planning of living spaces for improved health outcomes.

   (b) **Water Management:** IoT devices can be used in water management systems to assess multiple parameters that are critical to efficient water resource utilisation. These devices can monitor various aspects such as the quality of drinking water, water levels in borewells [13], and readings from water meters [14–16]. The information gathered from these measurements can be used for strategic water distribution planning, which can help in the identification of potential water wastages or leakages. This proactive approach facilitates judicious water use, ensuring this vital resource's sustainable and optimal utilisation.

   (c) **Vehicular Traffic**: The number of vehicles is increasing daily, and monitoring this traffic is essential. This planning and management help provide better navigation around cities and conserve fuel and other resources. IoT can be leveraged for better management and

[3] `https://cpcb.nic.in/`

planning, like implementing intelligent traffic signals that automatically set timers based on vehicular traffic and predictive vehicular maintenance. These measures can help prevent vehicle breakdowns and accidents [17].

2. **Smart Healthcare and COVID-19 Management**: IoT in healthcare, particularly during the COVID-19 pandemic, has shown immense potential. Wearable devices equipped with sensors can monitor vital signs like temperature, heart rate, and oxygen levels, providing real-time data to healthcare providers [18]. This technology has been crucial for remote patient monitoring, reducing the need for hospital visits during the pandemic. IoT devices have also been used in contact tracing and ensuring adherence to quarantine guidelines, thereby playing a pivotal role in managing and controlling the spread of the virus [19, 20].

3. **Agriculture and Farming:** In agriculture, farmers can now monitor and manage their crops and livestock more effectively with the use of IoT. Sensors placed in fields measure various environmental factors such as soil moisture, temperature, and nutrient levels, enabling farmers to make informed decisions about irrigation, fertilisation, and pest control. IoT technology also supports the efficient use of resources, reducing waste and increasing productivity in the agricultural sector [21].

4. **Retail Shopping**: In the retail sector, IoT is being used to transform the customer shopping experience. Smart shelves equipped with weight sensors and RFID tags can track inventory levels in real-time, reducing stockouts and overstock situations [22]. IoT devices also enable personalised shopping experiences through targeted promotions and recommendations based on customer behaviour and preferences. Furthermore, IoT technology helps supply chain management [23], ensuring smoother operations from warehouse to delivery.

2.4 Challenges

Although IoT brings numerous advancements, it also faces several significant challenges that need addressing to ensure its effective and safe implementation.

1. **Power Consumption**: IoT devices are typically designed for continuous operation. The diversification of standards and protocols, while offering benefits, can lead to increased power usage, further reducing battery longevity. The situation worsens, particularly for applications where battery replacement or recharging is not feasible. Usually, energy is saved by using low-duty cycles and by implementing sleep and idle modes for an operating device.

2. **Cost and Maintenance**: IoT deployments often consist of sensors that collect data. The sensors used would vary in the deployments and would typically be chosen depending on their type, working range, accuracy and cost. The cost of some sensors can be expensive due to various

reasons like availability in specific regions and the global silicon shortage that has occurred in recent times, particularly in 2023 [24].

3. **Scalability**: As IoT networks grow in size and complexity, the ability to efficiently manage and support an increasing number of connected devices becomes a concern. This involves not just handling the sheer volume of devices but also ensuring the network infrastructure can support increased data traffic and processing needs. Scalability issues can lead to system slowdowns, reduced reliability, and increased costs, impacting the overall effectiveness of IoT solutions [25].

4. **Security and Privacy**: The rapid development and adoption of IoT-enabled technologies has introduced a range of significant security challenges. IoT devices are designed to collect and transmit extensive amounts of data, some of which may be highly sensitive. This data undergoes a journey from a sensor through a controller, network gateway, and cloud server. Vulnerabilities in any of these intermediary steps can lead to cyberattacks, data breaches, and unauthorized access. For instance, a security compromise on implementing MQTT protocol for transferring data of an IoT deployment can reveal the security keys of a cloud platform to which data is being transmitted [26]. Often, IoT devices lack robust security features due to their constrained computing environments. Additionally, deployed devices are not regularly updated with their latest firmware, turning them even more vulnerable to security threats.

5. **Interoperability and Standards**: Interoperability refers to the collaboration of two or more systems or components for the purpose of data exchange. With a surge in global manufacturers producing diverse IoT devices, ensuring effective communication and collaboration is challenging due to differing protocols and standards. For instance, hardware sensors may use protocols like UART, I2C, or SPI, requiring the microcontroller to be compatible with these diverse protocols for smooth operation. Developing interoperable standards also raises the bar of security required for IoT applications [27].

Chapter 3

Remote Labs - An Overview

In this chapter, we begin with an overview of remote labs. Subsequently, we conduct an in-depth literature review, exploring various research groups and projects globally that have developed remote lab solutions along with IIIT Hyderabad. A common ground is established to qualitatively compare various literature works along with the proposed work of this thesis.

3.1 Overview of Remote Labs

In remote labs, laboratory experiments situated at a given geographical location can be accessed from anywhere in the world using the internet over a browser. Fig. 3.1 shows the overview of a generic IoT-based remote lab, which consists of two blocks connected by the internet. The first block is the internet platform at the user's side. The second block consists of hardware at a remote location and includes the physical setup of the experiment, sensors, actuators, cameras, and controllers. In such setups, cameras are used to stream the live feed of the experiment to the user. The user can control the experiment parameters using the internet platform. Experiment results can be presented in various formats, including tables, interactive plots, and diagrams on the internet platform.

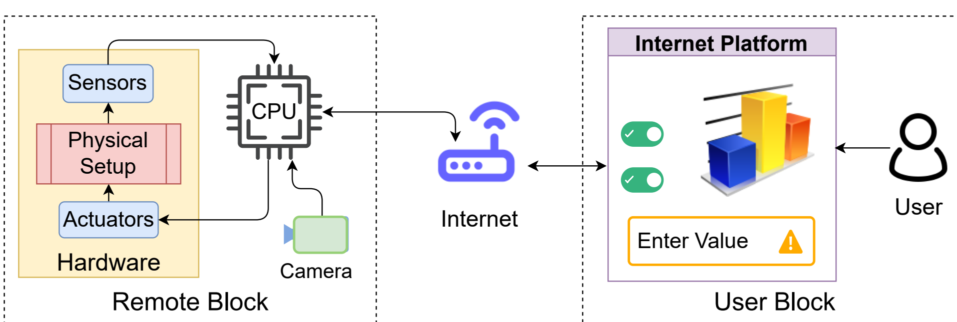
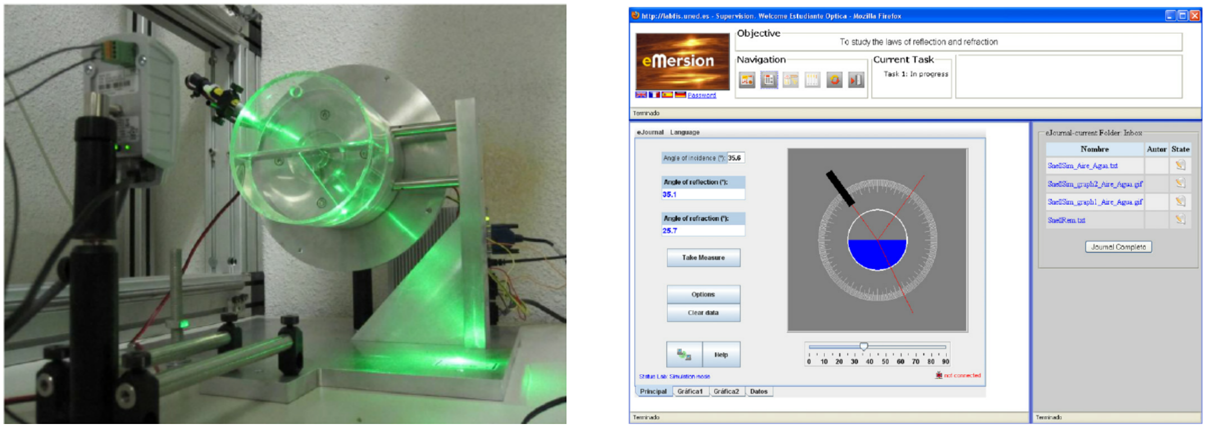


Figure 3.1 Overview of an IoT-based remote lab



(a) (b)

Figure 3.2 (a) Didactical setup of the Snell's law experiment. (b) User interface of Snell's law. Adopted from [43]

3.2 Related Work

3.2.1 Remote Labs Implementations by Different Research Groups

Numerous universities and research groups worldwide have implemented IoT-based remote lab solutions [28–42]. Prominent examples include WebLab-Deusto and its spin-off, LabsLand, both originating from the University of Deusto [28–32]. Multiple architectures were proposed to accommodate multiple features in their platform. Their design of remote labs, primarily in different science and engineering domains, comes with extensive features that primarily focus on building low-cost [28]. The experiments are made reliable [29] by maximising their availability through fault-detection and replication [31]. The fault detection included thorough checks on both the hardware experiments and software dashboard side, including a unique automation testing tool that automatically checks the mechanical components' working in experiments by experimenting with a computer program using CV techniques. In [32], a scalable architecture is designed to support a wide range of embedded systems so that laboratories for new or upgraded ones can be implemented and deployed quickly. Additionally, [30] presents the requirements for an interactive live-streaming platform firstly, and later, it proposes an architecture based on open technologies to satisfy those requirements that rely on Redis to achieve high scalability.

UNILabs[1] [43–45] is a network of web-based laboratories with several Spanish universities participating and hosting experiments that are shared over a Learning Management System [45]. UNILabs offers more than a dozen virtual and remote labs on different fields and topics, such as Control Engineering or Optics [43], and includes experiments based on concepts of spring elasticity and the laws of reflection and refraction as shown in Fig. 3.2. The labs were also integrated with Moodle[2] (previously integrated with eMersion and later migrated to Moodle), a Learning Management System. Several fea-

[1] https://unilabs.dia.uned.es/
[2] https://moodle.org/

tures were integrated to enrich the labs so that students could work on the hosted experiments. These included wikis, glossaries, lessons, questionnaires, workshops, demo videos, and saving the experimental data. The support for collaborative work among students (i.e., student-student interaction) and instructors (i.e., instructor-student interaction) is improved using asynchronous collaborative tools such as forums, chat, email, group discussions and public forums. Also, the remote experiments are available 24 by 7, and a slot-booking feature is also implemented to reserve beforehand to ensure access to the experimentation.

Additionally, the Internet School Experimental System (iSES) [3], which has built modular hardware and software systems, offers over 15 experiments in physics and allied domains. The experiments are built using the iSES Remote Lab SDK, comprising a library of over 20 customisable JavaScript widgets. These highly configurable widgets provide many thoroughly documented options, allowing even non-programmers to build a user interface for measuring and controlling the experimental parameters with data and video transfer facilities. This open modular SDK can communicate with different measurement platforms like Arduino, CMA CoachLab and Vernier, and standard universal measurement devices with a COM or USB port [33].

3.2.2 Remote Labs in Different Domains

Specialised iterations of remote labs have emerged to cater to specific domains of science and engineering, including electronics and embedded systems [28,34]. In [28], a multi-instance microcontroller-based embedded device remote laboratory is developed, surpassing state-of-the-art architectures in terms of scalability on low-cost setups. Furthermore, [34] introduces an open-source interface equipped with interchangeable circuit boards (plugs), providing educators with the flexibility to adapt circuits swiftly. Remote labs also teach programming languages like Python [35,36], enabling users to program sensors and actuators, such as temperature sensors and LCDs.

Several implementations of remote labs in the control engineering domain are discussed in [37–41]. In [37], the authors introduce a low-cost, fully open-source remote laboratory for teaching automatic control systems. It includes complete instructions for building and replicating the remote laboratory's hardware and software components. In [38], a low-cost remote laboratory for control engineering experiments is developed using an inexpensive BeagleBone Black development board[4], with experiments hosted on a dedicated Java application. In [39, 40], authors discuss the use of remote laboratories in control engineering education experiments, highlighting the importance of active learning pedagogy while providing limited insight into the implementation details. In [41], the results of a feedback study involving a remote laboratory used in the context of control engineering are presented to illustrate the impact of the remote laboratory on examination results. A new software architecture is proposed in [42] to create a novel wiki-based remote laboratory platform by combining both Wiki technology and re-

[3] `https://www.ises.info/index.php/en`
[4] `https://www.beagleboard.org/boards/beaglebone-black`

mote laboratory technology and was used to teach Mechanical Engineering courses at the University of Houston.

Commercial organisations such as LabsLand and Emona Tims[5] offer services that typically feature a platform or software enabling users to remotely access and conduct experiments utilising authentic laboratory equipment over the internet.

3.3 RLabs @ IIIT Hyderabad

Remote labs at IIIT Hyderabad, called RLabs, consist of several experiments built and deployed on the campus. The work presented in this thesis is a part of this deployment. In [46], a three-layer software architecture is proposed to showcase the feature of software multiplexing where multiple users are accommodated on a single experimental instance to work upon simultaneously. This feature works for experiments with low response time, typically in the range of milliseconds, and the authors have demonstrated it for the use case experiment of Kirchhoff's Voltage Law. The software architecture was built with the help of Blynk as a middleware.

In [47], a novel method to scale remote lab solutions was designed. This included designing miniaturised experimental setups, which are low-cost and miniature in dimensions compared to lab-scale experiments, although the functionalities would remain the same. Later, partial streams are implemented on a Raspberry Pi, where a single camera feed is split to stream multiple experiments simultaneously.

3.4 Attributes of Remote Labs

As mentioned previously, the literature presents numerous remote lab solutions (systems), each with its features and capabilities. To establish the suitability of these systems, the non-functional attributes (NFAs), also referred to as non-functional requirements [48], are considered. There are several NFAs, and they define how a system will work by laying out its requirements and limitations. It is crucial to establish a common ground, and to this end, seven crucial NFAs are identified, and relevant features are classified under each NFA. These NFAs and features serve as critical comparison criteria for remote laboratory solutions. The identified NFAs and their features are as follows:

1. **Affordability**: It describes the cost-effectiveness of a remote labs system, primarily focusing on *low-cost designs*. The cost factor is particularly crucial for making labs affordable and accessible to a large number of students in developing countries like India.

2. **Portability**: It describes the ease with which hardware experiments can be transported from one location to another. This can be achieved by designing *modular setups*.

3. **Scalability**: It describes the system's capacity to grow and adapt to increasing demands. This can be achieved in multiple ways:

[5] `https://www.emona-tims.com/`

- *Adding hardware (HW) instances to the platform* - denotes the capability of adding multiple instances of the same experiment to the remote lab's platform.
- *Miniaturised HW setups* - refers to designing remote experiments that are more compact than traditional laboratory experiments.
- *Peer-to-Peer (P2P) live video streaming* - refers to a live video streaming architecture that does not require a centralised streaming server.

4. **Compatibility**: It describes the system's ability to integrate and operate with a diverse range of configurations. This includes several features of the system:

    - Compatibility with *different hardware boards* for controlling the actuators and sensors and connect to the internet for data transmission (like ESP32[6] and Raspberry Pi[7]).
    - Compatibility with *multiple IoT platforms* for receiving data from different hardware boards and making the data accessible online (like Blynk[8] and Thingspeak[9]).
    - Compatibility with *various types of cameras* for live video streaming of the experiments. Cameras can include standalone cameras like IP cameras and non-standalone cameras like USB cameras.

5. **Maintainability**: It describes the efforts required to keep the system running normally. *Automated testing* showcases the system's ability to automatically report any errors affecting its functionality.

6. **Usability**: It describes how user-friendly the system is and is best known from *user feedback* collected from individuals (users) who have used the system.

7. **Universality**: It refers to the system's ability to be accessed across a variety of devices (mobiles, desktops), operating systems (Windows, iOS, Android), and web browsers (Firefox, Edge, Chrome), maintaining a consistent and adaptable user experience.

These NFAs offer a comprehensive basis for evaluating and qualitatively comparing various remote lab solutions, providing valuable insights into different solutions' strengths and weaknesses. Recent works (≤ eight years) that provide proper implementation details of the hardware and software components have been considered to evaluate prominent and relevant remote lab solutions in the literature. The focus is on labs operating within web browsers without external plugins rather than dedicated applications. According to the above criteria, the chosen remote lab implementations are: LabsLand [28–30], RaspyLab [36], RaspyControl Lab [37] and iSES [33]. Table 3.1 compares the chosen remote labs among the identified NFAs and the proposed solution in this paper.

[6] `https://www.espressif.com/en/products/socs/esp32`
[7] `https://www.raspberrypi.com/products/`
[8] `https://blynk.io/`
[9] `https://thingspeak.com/`

Table 3.1 Feature comparison of various Remote Labs

Attribute	Features	LabsLand [28–30]	RaspyLab [36]	RaspyControl [37]	iSES [33]	**RLabs** (Proposed)
Affordability	Low-cost setups	✓	✓	✓	×	✓
Portability	Modular hardware setups	×	×	×	×	✓
Scalability	Adding HW instances to the platform	✓	✓	×	✓	✓
	Miniaturised HW setups	×	×	×	×	✓
	Peer-to-Peer live video streaming	×	×	×	×	✓
Compatibility	Tested different HW boards	✓	×	×	✓	✓
	Tested multiple IoT platforms	×	×	×	×	✓
	Tested different cameras	✓	×	×	×	✓
Maintainability	Automated testing	✓	×	×	×	✓
Usability	User feedback	✓	✓	×	✓	✓
Universality	Device-friendly	✓	✓	✓	✓	✓

✓Implemented × Not Implemented

The proposed RLabs system is novel concerning the relevant recent remote lab implementations as shown in Table 3.1. The existing implementations lack features like modular and miniaturised hardware experimental setups, and P2P live video streaming and did not test different IoT platforms, leading to portability, scalability and compatibility limitations. Additionally, the systems in [33,36,37] did not test different cameras for live streaming and lack an automation testing suite that limits the compatibility and maintainability of the systems. In [33], the setups are not low-cost as they use personal computers for connecting the experiments to the internet, affecting the affordability of the systems. In [37], different hardware instances of an experiment cannot be added, and a user feedback is not presented that impacts the scalability and questions the system's usability. In [36], different hardware boards are not tested to design the experiments that limit the system's scalability.

Chapter 4

Engineering End-to-End Remote Labs using IoT-based Retrofitting: A Hardware Perspective

This chapter introduces the development of the RLabs system at the IIIT-H campus. A description of the experiments designed as part of this system is presented. The structure of the chapter is outlined as follows. Section 4.1 briefly presents the theory of two use case experiments - Vanishing Rod and Focal Length. Section 4.2 provides detailed insights into the hardware components used for retrofitting the experiments and creating their modular versions. Section 4.3 briefly discusses the software platform designed for hosting and managing these experiments. In Section 4.4, results are articulated systematically.

4.1 Use case Experiments

In this section, two considered use case experiments based on fundamental high-school physics concepts are explained in detailed, including their aim, theory, methodology, and results.

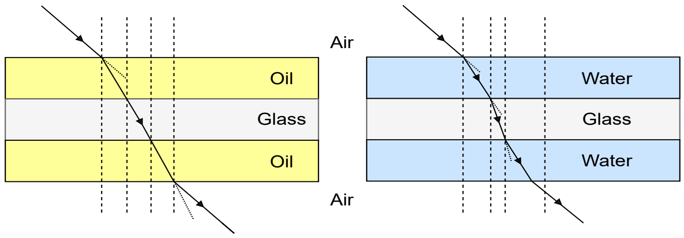
4.1.1 Experiment 1: The Vanishing Rod Experiment

4.1.1.1 Aim:

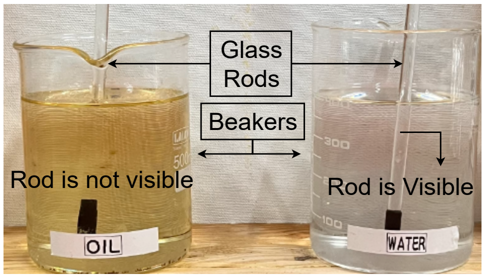
To observe the change in visibility of a glass rod when immersed in oil and in water media

4.1.1.2 Theory:

The concept of refractive index is explored in this experiment. While light travels from one medium (e.g. air) to another (e.g. water), the bending of light can be observed. This happens due to the slowing down of the speed of light when the medium changes. Every material is associated with a refractive index that quantifies the refraction of light. The higher the refractive index, the higher the deviation of light in the entering medium. However, suppose another medium with a numerically closer refractive index surrounds an object. Then, the object appears to have disappeared in the medium, as there will be no reflection and refraction of the light while passing through the object, as illustrated in Fig. 4.1(a).

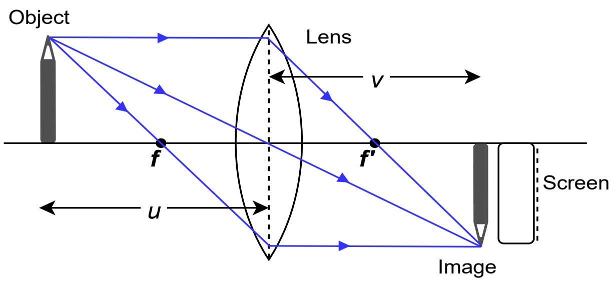


(a)

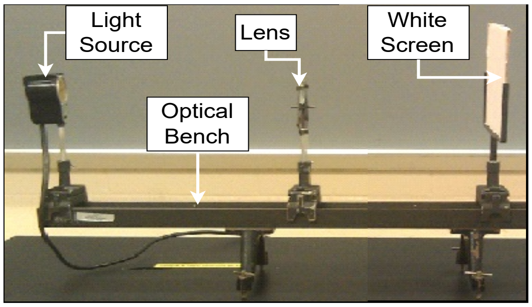


(b)

Figure 4.1 (a) Ray diagram of light entering different surfaces. (b) Visibility of glass rods when dipped in beakers containing water and oil separately



(a)



(b)

Figure 4.2 (a) Ray diagram of a thin biconvex lens. (b) Experimental setup of the focal length experiment (source: `hirophysics.com`)

4.1.1.3 Methodology:

Fig. 4.1(b) shows the apparatus used in the setup which includes borosilicate glass rods (Refractive index μ = 1.5), sunflower oil (μ = 1.47), drinking water (μ = 1.36), borosilicate glass beakers (μ = 1.5). One beaker is filled with sunflower oil and the other beaker is filled with water. The experiment is performed by placing the glass rods into the beakers and observing the visibility of glass rods in the beakers.

4.1.1.4 Results:

Fig. 4.1(b) depicts the visibility of glass rods when immersed in glass beakers. It can be observed that the glass rod dipped in the oil beaker tends to disappear while the other glass rod remains clearly visible. This is attributed to the fact that the glass rod and sunflower oil have very similar refractive index values that make the glass rod vanish in the oil medium. However, the glass rod and water have different refractive index values that make the glass rod visible in the water medium.

4.1.2 Experiment 2: Focal Length Experiment

4.1.2.1 Aim:

To determine the focal length of a biconvex lens by forming a sharp image of an object on the screen.

4.1.2.2 Theory:

The experiment focuses on the field of optics and aims to determine the focal length of a biconvex lens [49]. The focal length is a measure that defines the distance between the lens and the point where light rays converge to form a sharp image. It determines the lens's viewing angle and the magnification of the image produced. To determine the focal length of a given thin biconvex lens, an experiment is set to get a sharp image of an object on a screen by adjusting the distances of the screen and object from the pole of the lens as shown in Fig. 4.2(a). Once a sharp image is formed on the screen, the object distance (u) and the image distance (v) are used to calculate the focal length (f) of the used lens using the lens formula with proper sign conventions. Any measurement towards the direction of the incident ray is considered positive, and distances opposite to the direction of the incident ray are negative. In the case of a thin biconvex lens, as shown in Fig. 4.2(a), the object is on the left side of the lens and the real image will form on the right side if the object distance is greater than f. So according to the sign convention, u will always be negative and v will always be positive for real images. Then the modified lens formula would be:

$$\frac{1}{f} = \frac{1}{v} + \frac{1}{u}, \tag{4.1}$$

where u, v and f would be the absolute values.

4.1.2.3 Methodology

Fig. 4.2(b) shows the apparatus used in the setup which includes a white screen, a light source, a biconvex glass lens and an optical bench upon which the experiment is performed. Firstly, the biconvex glass lens is placed on the lens stand. The light source and the white screen are placed on the either sides of the lens such that the object, lens and the screen lie on a straight line. The experiment is performed by adjusting the positions of both the object and the screen platforms and observing the image formation on the screen.

4.1.2.4 Results

Table 4.1 displays a few pairs of u and v values that yield a sharp image during the experimentation. It is notable that multiple sets of values can result in a clear and sharp image. Irrespective of these different values observed, the lens's focal length remains the same, given that the lens used is the same in every trial.

Table 4.1 Focal Length calculations for different values of u,v for which sharp images are obtained

S.No	u (cm)	v (cm)	f (cm)	% error
1	20.59	20.38	10.24	2.4
2	29.5	15.89	10.33	3.3
3	42.65	13.95	10.51	5.1

Table 4.2 Comparison of different stepper motors used in the experiments

Motor Type	**28BYJ-48**	**Nema 17**
Step Angle	0.0875°	1.8°
Rated Voltage (V)	5	12-48
Rated Current (A)	0.4	1.68
Holding Torque (kg-cm)	0.34	4.2
Step Accuracy	Low	High
Size (mm^3)	34×18×10	40×42×42
Weight (g)	~50	~300

4.2 Hardware: Design and Implementation

In this section, firstly, various design strategies that were part of the proposed hardware are discussed. Secondly, designs of retrofitted lab-scale remote experiments are presented. Finally, more compact versions of the lab-scale versions are also presented, emphasising their modular designs.

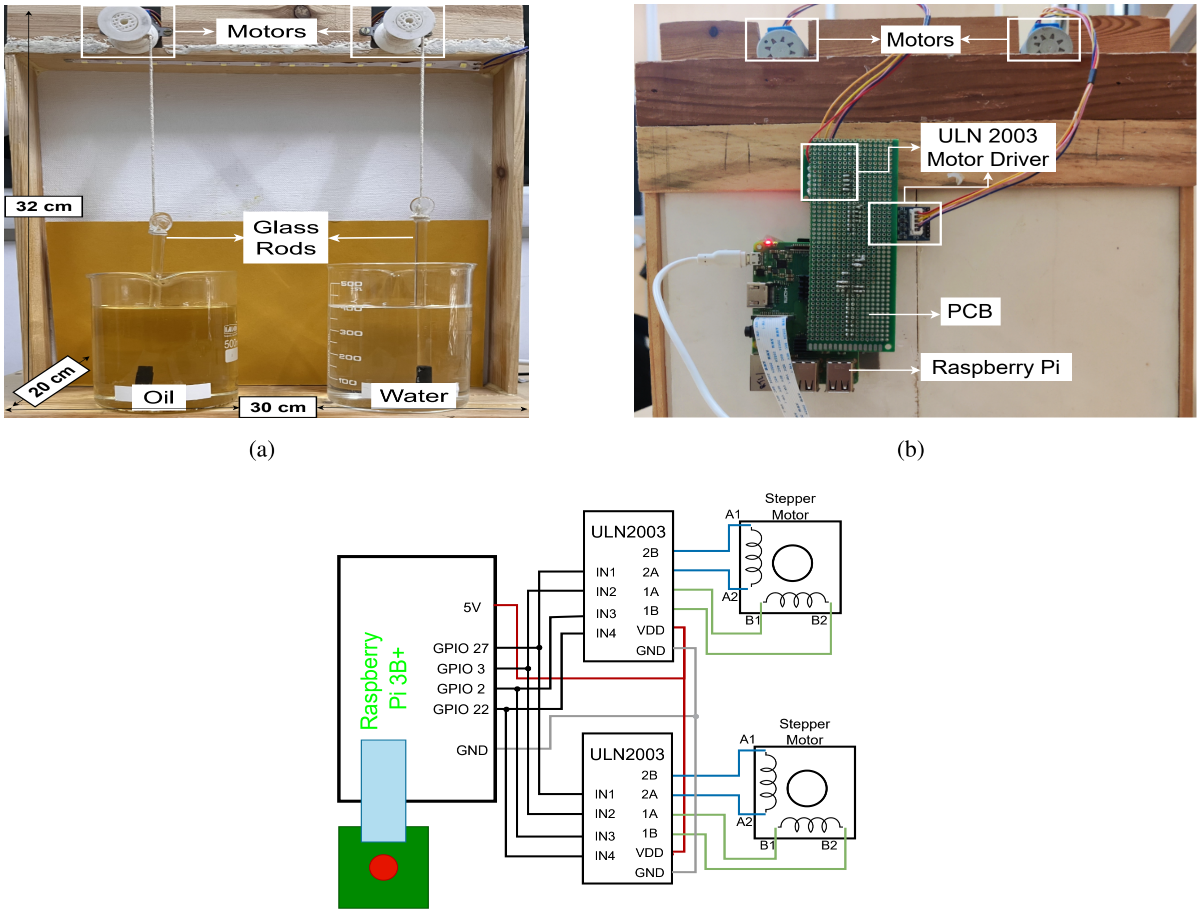
4.2.1 Design Choices

The experiments are designed for remote operation using two different strategies. First, lab-scale setups are created by retrofitting IoT components into the existing experimental equipment used in traditional laboratories, offering users a visual experience similar to traditional lab experiments. Second, modular setups are developed, utilising 3D printed components and PCBs to achieve modularity and portability. Despite the difference in the build, both setups offer the same functionalities.

The hardware used in the experiments comprises of embedded processing boards, that can be categorised into two main types: microcontroller-based boards (MCUs), such as the ESP32[1], and microprocessor-based boards (MPs), like the Raspberry Pi[2]. These boards control the actual experimental setups, interfacing with sensors and actuators using hardware communication protocols like UART, I2C, or SPI. Typically, these boards establish connections with a software platform, usually through WiFi or Ethernet, to relay all the relevant information to the user. Additionally, the experimental setups often complemented with live-streaming capabilities to capture real-time visual data from the ongoing experiments.

The Raspberry Pi 3B+[3] is chosen as the primary board for designing and building lab-scale experiments in this paper for several reasons. Firstly, it boasts multiple GPIO pins, enabling interaction with

[1] `https://www.espressif.com/en/products/socs/esp32`
[2] `https://www.raspberrypi.com/products/`
[3] `https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/`



(a) (b)

(c)

Figure 4.3 Hardware description of the experimental setup for the vanishing rod experiment. (a) Front-view of the experimental setup. (b) Back-view of the experimental setup. (c) Circuit Diagram of the setup

numerous sensors and actuators simultaneously. Secondly, its compatibility with a Linux operating system facilitates programming in a variety of languages. Thirdly, it supports various cameras like the Raspberry Pi camera and the USB cameras. Specifically, the Raspberry Pi Camera (RaspiCam), which is used for live-streaming the experiments, can be connected via the Camera Serial Interface (CSI). The CSI is advantageous due to its high data throughput, ensuring real-time image and video processing [50]. Lastly, the Raspberry Pi 3B+ provides versatile internet connectivity options, either via Wi-Fi or an Ethernet port.

The experiments use various actuators, primarily comprising of motors for moving the experimental apparatus. Multiple motor types can be used for this purpose, such as servo motors and stepper motors [51]. Each motor serves a specific purpose, varying in terms of accuracy, cost, and delivered torque. In this work, stepper motors are primarily utilised due to their cost-effectiveness, aligning with the requirements of the presented experiments. However, different stepper motors (Table. 4.2) are used in the use case experiments based on their specific torque requirements.

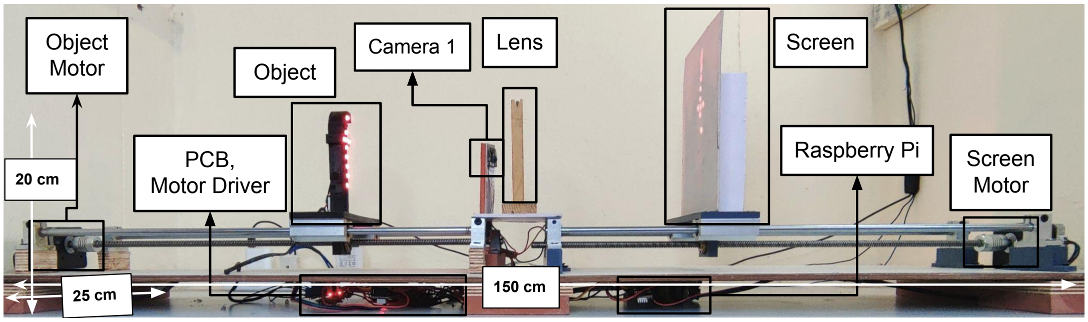


Figure 4.4 Side-view of the experimental setup for focal length experiment

4.2.2 Lab-Scale Experimental Setups

4.2.2.1 Experiment 1: Vanishing Rod

The apparatus required for the traditional experiment is already discussed in Section 4.1. Additional apparatus required to retrofit the experiment for remote accessibility includes plywood, 28BYJ-48 stepper motors (Table 4.2), Raspberry Pi 3B+ and a RaspiCam.

Figs. 4.3(a), 4.3(b) and 4.3(c) show the physical setup and circuit design of the Vanishing Rod experiment. One beaker is filled with sunflower oil and the other beaker is filled with water. They are then placed near the base, which is prepared from plywood. The glass rods are attached with strings to the stepper motors fixed above the beakers. The RaspiCam is positioned to capture both beakers within its field of view in the video feed. This feed is sent to Raspberry Pi 3B+, which also controls the motors.

The experiment is performed by moving the glass rods into the beakers placed and observing the visibility of glass rods in the beakers. Both the glass rods are moved up or down vertically from the beakers simultaneously by the stepper motors connected to the Raspberry Pi. User can control the glass rods' movement where the glass rods can either be dipped into the beakers or not.

4.2.2.2 Experiment 2: Focal Length

The apparatus required for the traditional experiment is already discussed in Section 4.1. Additional apparatus required to retrofit the experiment for remote accessibility includes two NEMA 17 stepper motors (Table. 4.2), two A4988 micro-stepping driver, two limit switches (end-stop switches), a light source, a Raspberry Pi 3B+, a RaspiCam and an USB camera.

Figs. 4.4 and 4.5 show the physical setup and circuit design of the experiment. Stepper motors are used to move the object and screen, while the position of the lens is fixed. The stepper motors used for this experiment are different from the motors used in Vanishing Rod experiment as more accuracy and torque are required for this experiment. The distance moved by the object and screen is linearly proportional to the number of steps/ degrees rotated by the stepper motor, that moves in precise and repeatable increments, which allows for consistent and accurate movement of the the object and the screen. To improve the movement of the screen and object, a micro-stepping driver (A4988) is used

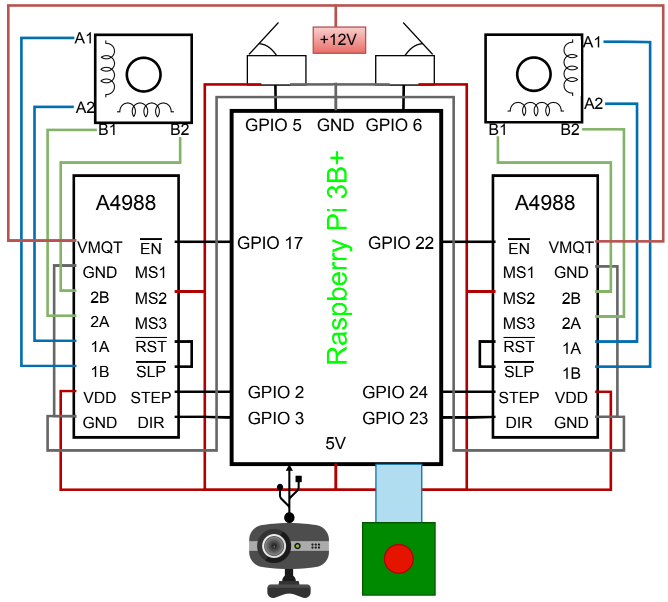


Figure 4.5 Circuit diagram of the Focal Length setup

to control the motors. This micro-stepping driver allows the motor to take 400 steps per revolution. Here, 1 step is equivalent to 10 µm and all the movements are based on this relation. The motors are attached to a screw shaft on which the screen/object is mounted, which converts the rotational motion to linear motion. Limit switches are placed to re-calibrate the motors which control the object and screen movements. The light source is used as an object to observe the inversion in the image formed by the lens. The full-sized optical bench allows the user to explore different types of image formations through the biconvex lens. RaspiCam is positioned to capture the white screen on which the image is formed, while the USB camera captures the side-view of the entire experimental setup. A USB camera is used instead of another RaspiCam as a single Raspberry Pi 3B+ can only handle one RaspiCam interfaced using CSI. The captured feed from both the cameras is sent to the Raspberry Pi which also controls the motors and receives signals from the limit switches.

The experiment involves adjusting the positions of both the object and the screen platforms to attain a clear and sharp image of the object on the screen. Individual sliders facilitate the independent movement of the object and screen. The focal length of the lens is determined using the distances both the screen and object have been moved from the lens once a sharp image is acquired, applying the Equation 4.1. A sharp image can be obtained at multiple pairs of u and v values, each resulting in a different magnification of the image displayed on the screen.

4.2.3 Modular Experimental Setups

Modular setups are smaller replicas of the lab-scale setups, primarily constructed using 3D printed components and commonly available materials that require no welding, adhesive bonding, or specialised mechanical tools typically used to build lab-scale setups. The designs are intentionally crafted for easy

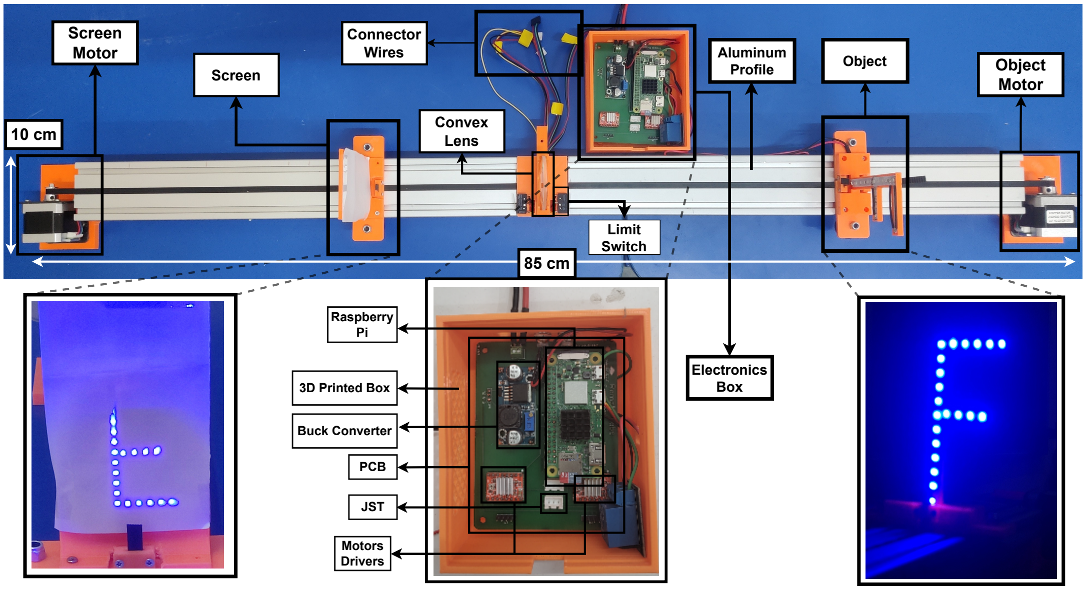


Figure 4.6 Hardware description of the modular Focal Length setup

assembly, allowing users to follow a manual for straightforward construction. Minimal tools, such as a screwdriver and soldering iron, are sufficient for assembly.

4.2.3.1 Experiment 2: Focal Length:

Fig. 4.6 shows the modular Focal Length experiment. The modular setup apparatus employs a design similar to that of 3D printers, comprising two A4988 drivers, NEMA 17 stepper motors, limit switches, two Raspberry Pi Zero 2 W[4] and two RaspiCams. The body is constructed from V-slot aluminium profiles, which serve as a versatile base for attaching various components. Two such profiles are used, one each for the object and screen. Using these aluminium profiles as a base, the rest of the structure is built, which includes the mechanism for moving the platform upon which the object/ screen is placed, space for motor operation and slots for the electronics. The other structural components, like the screen and object platforms and supports for the motors, are 3D-printed. These 3D-printed parts are designed to have slots and holes for screws appropriately, which can be fixed to the aluminium extrusion profiles using sliding nuts (slide and lock mechanism), eliminating the need for drilling holes in the profiles. In contrast to the traditional linear screw actuation, this setup utilises a belt drive mechanism to manoeuvre the object and screen platforms. This modification reduces the number of components and decreases the overall weight of the apparatus. A timing belt facilitates the movement of the screen, and object looped over a pulley and the stepper motor. Once the two profiles are built with the motors joined at the ends, the profiles are joined with a joint 3D-printed base upon which the lens stand is mounted. This 3D-printed base has connections between the motors and the illuminated object. These connections are made using JST and DuPont connector wires, which enable plug-and-play links to the electronics

[4]https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/

box containing the experiment's circuitry, as depicted in Fig. 4.6. The PCB inside the electronics box provides slots for connecting the Raspberry Pi, buck converter, and motor drivers. Additionally, a power supply not included in the setup can be directly connected through the power socket on the box's side. A Raspberry Pi and a RaspiCam are placed to stream the side view of the experiment. A single Raspberry Pi could not handle the streaming from both a RaspiCam and a USB camera; hence, two Raspberry Pis were used.

Raspberry Pi Zero 2 W is a more compact version of the Raspberry Pi 3B+ with a CSI port for video streaming and Wi-Fi connectivity. However, it has slightly lower processing capabilities and lacks an Ethernet port. It is important to note that the streaming pipeline is decoupled from the experiment controls in specific setups to showcase the compatibility of various hardware boards and cameras. This is discussed in detail in Section 4.4.

4.3 RLabs Software Platform

A software platform is required to operate the above designed hardware experiments. This platform should have various capabilities as follows:

1. Host multiple experiments and multiple instances of the same experiment.
2. Live video streaming capability to stream the remote experiments with low delays ($\leq$ 1sec).
3. User management to control multiple users simultaneously working on different experiments.
4. Slot booking and queues to manage the users efficiently.

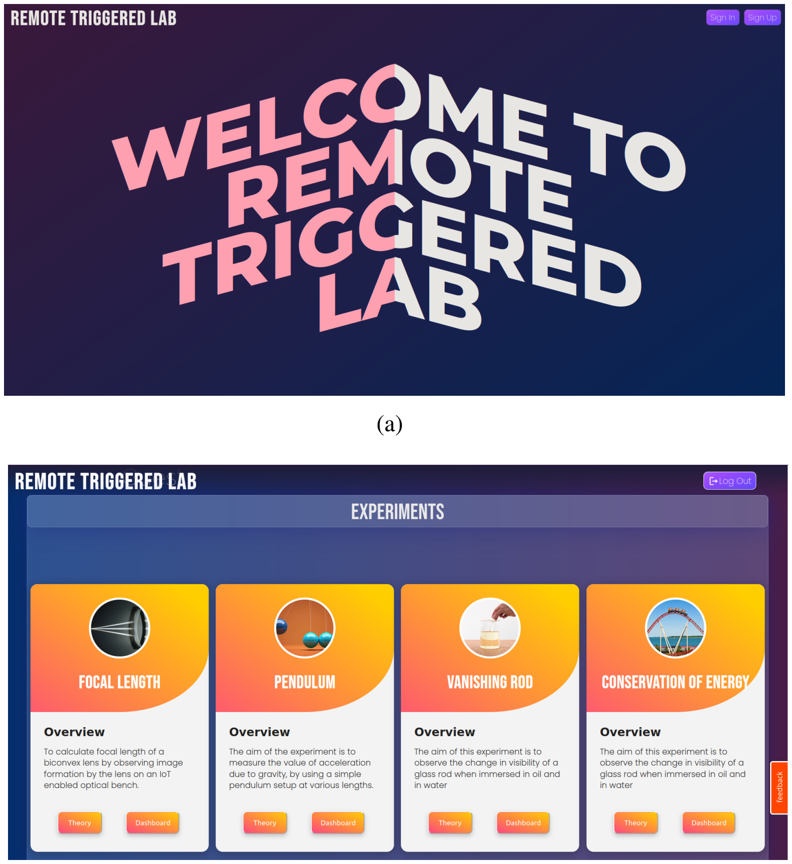
To this extent, the RLabs platform is implemented with the above features by the Remote Labs team at IIIT Hyderabad hosted at `remote-labs.in/`. Fig. 4.7 shows the earlier version of the designed RLabs dashboard. This platform is complemented with many other features, including automation testing, a suite of tools that automatically conducts an experiment to observe the working of the experiments through the camera feed and evaluate the working using computer-vision techniques. The Remote Labs team at IIIT Hyderabad is developing this.

4.4 Results

This sections primarily presents three results: cost tables for building the experiments to show that the setups are affordable, assembly steps for the modular setups to show the portability and compatibility of the system with various configurations of the setup

4.4.1 Low-cost hardware setups

Tables 4.3 and 4.4 show the total material costs for building the lab scale setups for the two use case experiments in INR. It can be observed that the total cost of the lab scale vanishing rod and focal length



(a)

(b)

Figure 4.7 RLabs software dashboard. (a) Welcome screen. (b) Experiments selection screen

experiments are 7700 INR (92 USD) and 14500 INR (174 USD[5])), respectively. Firstly, the setups are low-cost and can be attributed to using single-board computers, which has brought down the costs as individual servers or PCs are not required for hosting the experiments. Raspberry Pi 3B+ (5000 INR) is the costliest item (33% and 66%) in the lab scale setups. This cost can be reduced by replacing it with cheaper boards like Raspberry Pi Zero 2 W (1650 INR), significantly reducing the costs.

4.4.2 Portable Hardware Setups

A step-by-step assembly of the experimental setup is presented to showcase the portability of the built Focal Length experiment. Fig. 4.8 shows the step-by-step assembly of the Focal Length experiment from its components in a modular fashion. The setup is designed using various techniques to make the assembly easy. The parts have different shapes that fit together, making it intuitive for the user to join them. This also eliminates the use of screws and bolts to some extent. For example, sliding nuts and the aluminium profile create a slide and lock mechanism for the focal length setup. The sliding nuts easily

[5] 1 USD is approximately 83 INR as of December 2023

Table 4.3 Costs for lab scale vanishing rod setup

Lab Scale VR		
Item	**Qty**	**Cost**
Raspberry Pi 3B+	1	5000
28BYJ-48	2	160
ULN2003	2	80
Glass Rod	2	60
Beaker	2	400
Frame	1	500
Camera	1	500
Misc	1	1000
Total		**7700 INR 92 USD**

Table 4.4 Costs for lab scale focal-length setup

Lab Scale FL		
Item	**Qty**	**Cost**
Raspberry Pi 3B+	1	5000
NEMA 17	2	1700
A4988	2	300
USB cam	1	1200
Camera	1	500
8mm Axle	4	1200
Axle support	8	400
Screw Rod + Nut	2	650
Shaft coupler	2	150
Bearing	2	200
Slider	4	800
Wooden Planks	1	600
3D printed parts	1	300
Misc	1	1500
Total		**14500 INR 174 USD**

slide into the profile slots and can be tightened using a bolt that holds the centre base and other parts against the profile. This system overcomes the need to drill holes into the profile, making it easier for an individual to assemble the setup. The setup is entirely made of off-the-shelf parts along with 3D-printed modules that are usually available in the local hardware and electronics stores. This renders the setup perfectly portable, similar to IKEA[6] products, enabling users to assemble and disassemble them easily. This portability facilitates the shipment of setups to various locations worldwide, including remote rural areas, thereby extending their accessibility.

4.4.3 Compatibility with Different Hardware Boards and Cameras

In order to ensure broader compatibility, the RLabs' software architecture is designed to focus on supporting a wide range of hardware boards. This is shown using different boards (such as Raspberry Pi and ESP32 [47]) to implement use case experiments. The boards used must be able to connect online using any IoT platform that facilitates data access via APIs. This allows for the accommodation of various hardware boards. However, there can be boards that do not support internet connectivity. Those boards can then be coupled with an ESP8266 or ESP32, which provides smooth data transfer from the experiment to the dashboard at an economical price point, starting from as low as 400 INR (~5 USD).

If a board does not support live-streaming from a camera, a Raspberry Pi Zero 2 W[7], which costs around 1600 INR (~20 USD), can be used alongside standard RaspiCams solely for streaming purposes.

[6] https://www.ikea.com/
[7] https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/

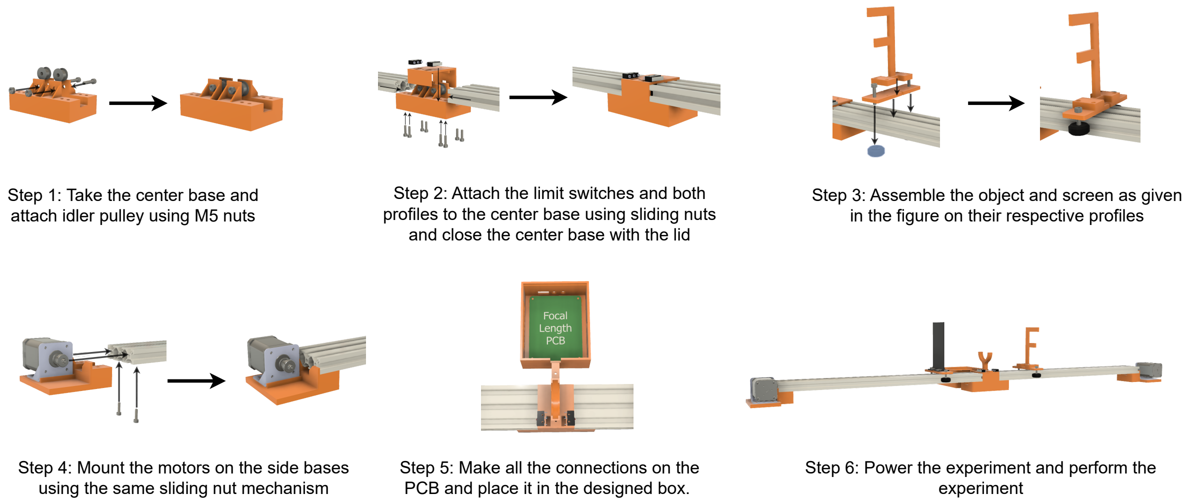


Figure 4.8 Assembly of the modular Focal Length Setup

Table 4.5 Latency in live-streaming of video across various cameras

Camera	Delay (in sec)
RaspiCam + Raspberry Pi 3B+	0.35
RaspiCam + Raspberry Pi Zero 2 W	0.86
TP-Link Tapo C100 IP cam	2.01

For experiments that use a Raspberry Pi, any IP camera or USB camera compatible with and recognised by a Raspberry Pi can be used for streaming. Standalone IP cameras can be easily set up by users with minor configurations, including setting up a static IP for the camera and port forwarding to grant global access. Table 4.5 shows the observed delay in live streams across different devices. The WebRTC facilitates the streaming in Raspberry Pis, while the stream from the TP-Link Tapo C100 IP Camera[8] is accessed using Real Time Streaming Protocol (RTSP). The Raspberry Pi Zero W[9] was evaluated with the streaming script as well. However, it delivered a poor-quality stream and exhibited significant lag, with delays exceeding 3 seconds. Alternatively, the ESP32 Cam module offers streaming capabilities, requiring minor tweaks to support WebRTC [52]. However, it is essential to note that the authors have not officially tested this configuration.

[8] `https://www.tapo.com/in/product/smart-camera/tapo-c100/`
[9] `https://www.raspberrypi.com/products/raspberry-pi-zero-w/`

Chapter 5

CV and IoT-based Remote Labs: Use Case of Conservation of Mechanical Energy

5.1 Introduction

RL can host a variety of experiments from different fields, including Engineering, Physics, Chemistry and Biology. Sensors are often used in building these experiments to collect data. However, there are several disadvantages of using sensors where data collection can be limited by physical constraints like orientation, type and number of sensors that can be placed. CV algorithms can be used to overcome the limitations faced by measuring devices like sensors.

Experimental setups utilising CV-based techniques for obtaining outputs are used in [53–56]. In [53], a motion analysis system was developed for physical experimental education using CV. The force acting on a moving object is visualised as a vector overlapped on the object's trajectory. In [54] and [55], simple setups consisting of a camera and a computer were used to analyse a falling object and determine the acceleration due to gravity g. [55] mentions that the results can be published on a web page later. In [56], Raspberry Pi 4 with a camera module was used to set up an experiment that detects central elastic collisions of two plastic balls, where OpenCV was used to detect collisions. However, all these experiments cannot be considered RL experiments as they should be performed manually. Also, publishing the results and resetting the experiment are not automated or controlled remotely over the internet. To this extent, this chapter considers deploying a CV-based remote lab solution by considering a use case experiment that does not require any manual intervention and has the potential to replace sensors for certain types of experiments.

The rest of the chapter is structured as follows: Section 5.2 describes the physics of the use case experiment, followed by the description of the hardware designed in Section 5.3. Section 5.4 presents the methodology of the CV-based approach for estimating the object's velocity. Section 5.5 shows the IR-based approach for estimating the velocity, and finally, the results are presented in Section 5.6

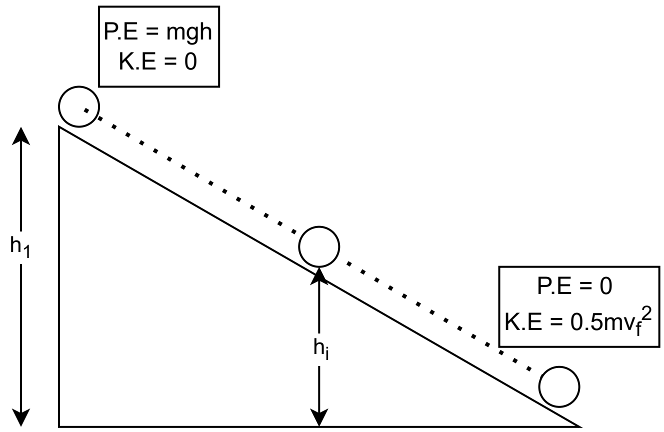


Figure 5.1 An example of Conservation of Mechanical Energy

5.2 Experiment

In this section, the principle of *Conservation of Mechanical Energy* is discussed and later illustrated with an example.

5.2.1 Theorem

The principle of conservation of mechanical energy: This states that the mechanical energy of a moving body at any point remains constant throughout its motion.

According to the theorem, under the assumption that there are no resistive forces like friction, the mechanical energy of an object, E_T, which is the sum of its kinetic energy E_k and potential energy E_p, is always a constant, i.e.,

$$E_T = E_p + E_k = \text{constant}. \tag{5.1}$$

Here the kinetic energy E_k is the energy of motion and is defined as

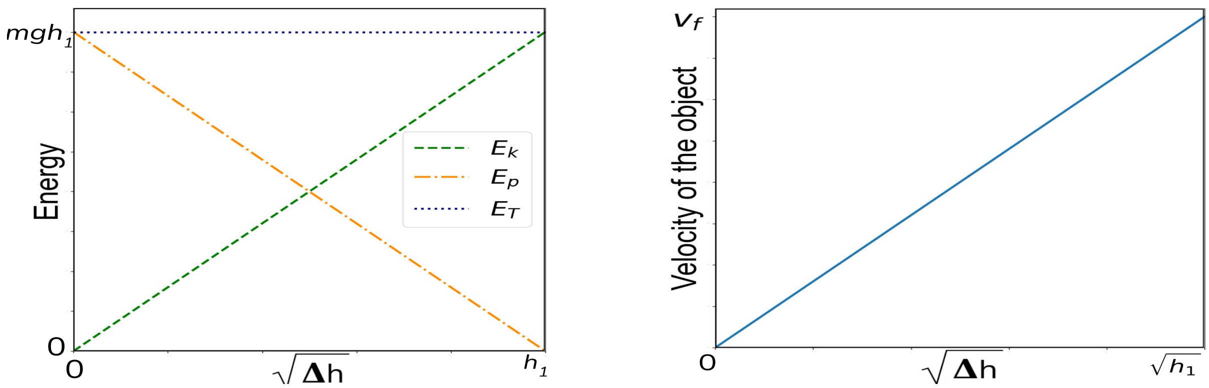
$$E_k = \frac{1}{2}mv^2, \tag{5.2}$$

where m is the mass of the point object and v is the instantaneous velocity of the moving object. Also, gravitational potential energy E_p is the energy of the object due to its position relative to the earth and is defined as

$$E_p = mgh, \tag{5.3}$$

where g is the acceleration due to gravity, and h is the height of the object relative to an arbitrary reference level.

To keep the experiment simple and intuitive, only two forms of energies, gravitational potential energy E_p and translational kinetic energy E_k are considered throughout the experiment. All the dis-



(a) (b)

Figure 5.2 Plots of change in energy and velocity wrt $\sqrt{\Delta h}$ (a) Change in various forms of energy vs $\sqrt{\Delta h}$ (b) Plot of velocity of the object vs $\sqrt{\Delta h}$

cussions and experiments are restricted to point mass objects, i.e., when the object of interest covers a much greater distance than its dimensions, it can be considered a point object.

5.2.2 Example

Consider Fig. 5.1, where a ball is released from rest on a frictionless triangular wedge of vertical height h_1 and fixed to the ground as shown. As the ball moves down the wedge, E_p decreases while E_k increases as the magnitude of the velocity of the ball increases. Consider an intermediate point where the ball is at a height of h_i and has a velocity v_i. Using (5.1), the total energy at the height h_i is given by

$$E_T = \frac{1}{2}mv_i^2 + mgh_i. \tag{5.4}$$

On the other hand, the total energy at the initial rest point is entirely because of potential energy so that

$$E_T = mgh_1. \tag{5.5}$$

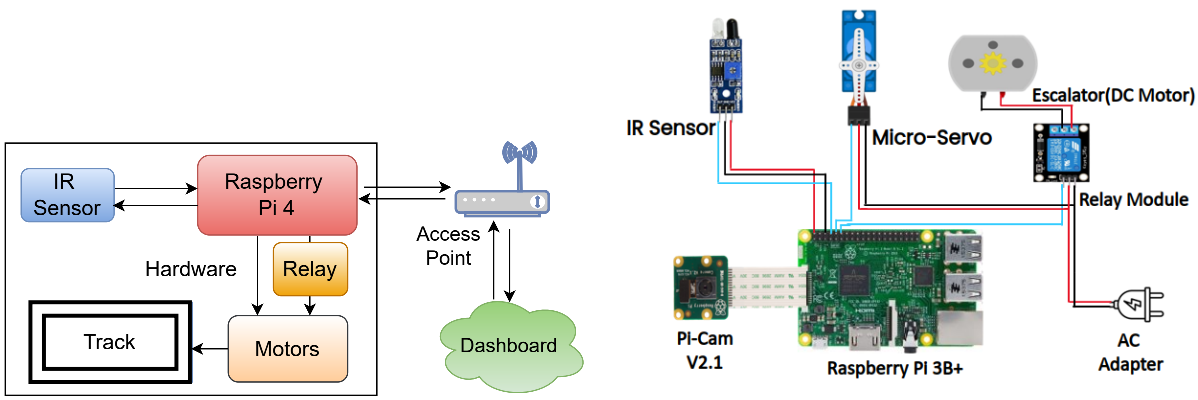
Using (5.4) and (5.5), we get

$$\begin{aligned} mgh_1 &= \frac{1}{2}mv_i^2 + mgh_i, \\ \therefore \quad v_i^2 &= \frac{2mg(h_1 - h_i)}{m}, \\ \therefore \quad v_i &= \sqrt{2g\Delta h}, \end{aligned} \tag{5.6}$$

where $\Delta h = h_1 - h_i$ is the height of the intermediate point wrt topmost point (h_1).

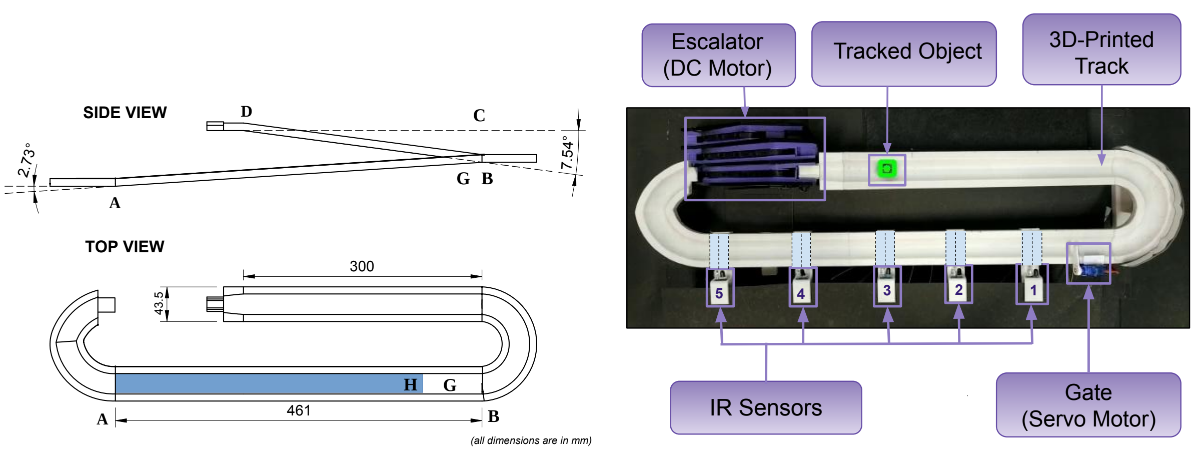
Fig. 5.2(a) shows the change in E_T, E_p and E_k as the ball falls down the wedge. It can be observed that the mechanical energy E_T is constant everywhere while E_p and E_k change as the height of the object changes. From (5.6) and Fig. 5.2(b), it can be observed that $v_i \propto \sqrt{\Delta h}$, indicating that when an object falls down from a starting point (h_1), its velocity increases linearly with the square-root of difference of height wrt starting point. Effectively, (5.1) becomes equivalent to (5.6) and this will be used to show the conservation of mechanical energy in further sections.

5.3 Hardware Setup



(a) (b)

Figure 5.3 Hardware description of the experimental setup (a) Block architecture of the RL setup (b) Circuit Diagram of the RL setup



(a) (b)

Figure 5.4 Setup of the experiment (a) 3D track used for the experiment (b) Actual Setup

Figs. 5.3(a) and 5.3(b) show the block architecture and circuit design of the RL setup designed to implement the experiment. The setup essentially consists of a 3D modelled track, IR sensors [57], an active high relay module, a micro-servo motor, a Direct Current (DC) motor, Raspberry Pi 3B+[1] and a Raspberry Pi V2.1 camera module[2].

Raspberry Pi 3B+ is the microprocessor used in the RL setup that interfaces the dashboard with all the sensors and actuators. This microprocessor is connected to a Wireless Fidelity (WiFi) network to host the experiment online. A 3D track is designed using Fusion 360[3] and printed with white poly-lactic acid (PLA) material. All the dimensions and coordinates of the points on the track are predetermined as shown in Fig. 5.4(a). The structure consists of a continuous track of two gradually varying slopes along which IR sensors are placed and an escalator. The object, in this case, a stainless steel ball of diameter 18.5 mm and mass 500 g, is set to roll on the track. The escalator, operated by a DC motor, forms a closed path in the track by raising the object to the top position. The servo motor acts as a gate (G), as shown in 5.4(a), to bring the object to a halt and release it from rest later. This gate can be controlled by a user remotely via the dashboard. Point H in Fig. 5.4(a) denotes the point where the velocity estimation starts (discussed in the later sections).

The Raspberry Pi camera, a widely-used camera with Raspberry Pi, is used to record at 30 Frames Per Second (FPS) and is fixed vertically above the track. The camera is fixed such that the whole track is captured while recording. Fig. 5.4(b) shows the actual setup captured from the camera. All the hardware components are mostly hidden and covered with the background to avoid their visual disturbances during recordings that might affect the velocity estimations. It supports up to 40 FPS with full Field of Vision (FoV) while a maximum of 90 FPS [58] is supported with limited FoV which is undesirable as the whole track can not be captured. So, a mobile camera[4] is used to record the track at higher frame rates of 60, 240 and 480 FPS to analyse the effect of change in FPS on the results. This mobile camera can be replaced with a USB camera that supports higher FPS and can be attached to the experimental setup.

5.4 Computer Vision Based Setup

To compute the object's velocity, in our case, the steel ball, a CV-based approach has been used. Using the setup explained in section 5.3, the experiment is recorded while the object is moving on the track as shown in Fig. 5.4(b). CV-based algorithms, including background subtraction, morphological transformation, image filtering, thresholding, and contour detection, are used to track the object. Fig. 5.5 shows the algorithmic pipeline of the implementation.

5.4.1 Object localization and tracking

In order to localize and track the object from the video captured, the following steps are taken:

[1] `https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/`
[2] `https://www.raspberrypi.com/products/camera-module-v2/`
[3] `https://www.autodesk.com/products/fusion-360/`
[4] `https://www.oneplus.in/8t`

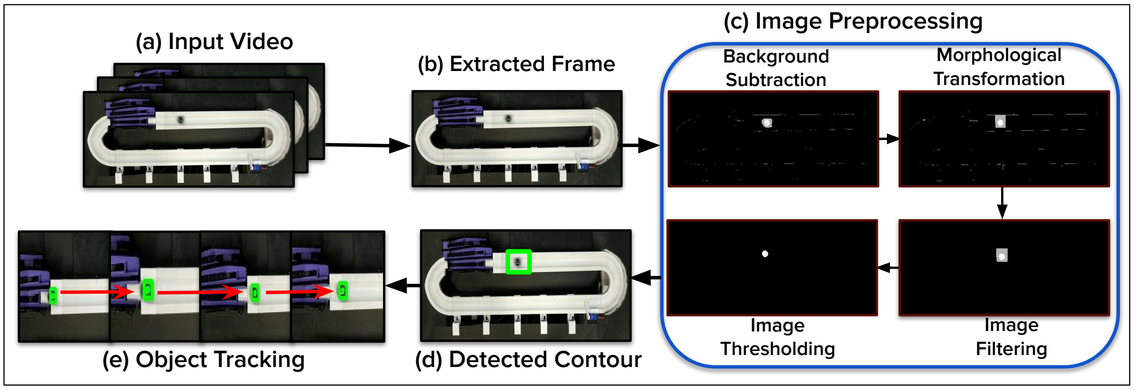


Figure 5.5 Algorithmic pipeline of the CV-based setup for object tracking. The arrow in (e) object tracking shows the moving steel ball being tracked in consecutive frames.(Best viewed in color)

5.4.1.1 Frame Acquisition

The first step is to extract the frames from the video sequentially. The video is recorded from the top view of the whole setup. The extracted frames are stored for further processing.

5.4.1.2 Image Preprocessing

The image preprocessing is a 4-step process which is as follows: i) background subtraction, ii) morphological operation, iii) image filtering, iv) image thresholding.

Our task is to localize the moving steel ball, which is a part of the image foreground. The background is subtracted from the image to get the foreground which mainly contains the steel ball due to its motion. After getting the foreground image, which can still contain noise, morphological operations are used to remove these noises. This method helps to close small holes inside the foreground objects in the frame. The closing morphology method consists of repeated steps of dilation followed by erosion. After this, image filtering processes the edges using a median filter to remove the noise. Finally, the image thresholding process is applied after converting the image into grayscale. As a result, frames are converted into a binary image, and a closed curve is formed around the steel ball. For this process, median thresholding (threshold value set to 127) is used. This process is done for the system to create blobs for the later stage of inspection. Fig. 5.5(c) shows the output of the frame after the image has been preprocessed.

5.4.1.3 Contour Detection

The preprocessed frame is used to localise the desired object. The closed curves formed around the object are used to determine the location of the steel ball. In any image, *Contours* can be defined simply as a curve joining all the consecutive points along the object's boundary. As in our case, the curves are formed around the steel ball (similar can be seen in Fig. 5.5(d)), finding the contours will provide the steel ball's location. As the contour's location is determined in coordinates, a bounding box is drawn to highlight the region of interest, i.e., the localised object and the centroid of the bounding box is stored.

5.4.1.4 Tracking

After determining the object location in one particular frame, the same process mentioned above is applied to all the consecutive frames. In each frame, the object location is determined and tracked further. Fig. 5.5(e) shows the bounding box (green) around the moving object tracked in each frame.

5.4.2 Calculations

Algorithm 1 presents our approach to estimate the velocity of the object between 2 frames. Its inputs include the frames captured from the camera, coordinates of the object's centroid for each frame tracked earlier, a conversion factor (d_w) to convert pixel distance to real-life distance (specific for a given track) and the frame rate of the input video (lines 2~6). Every two consecutive frames are considered and the Euclidean distance between their centroids is calculated (lines 10~11). Then the velocity of the moving object is estimated (lines 12~14) using the time taken to travel between two consecutive frames.

Algorithm 1 Estimate Velocity (V_{est})

1: **procedure** ESTVEL(F, C, d_w, FPS)
2: **Inputs:**
3: F: array of captured frames
4: C: array of x and y coordinates of the centroids (framewise)
5: d_w: conversion factor (pixels to meters)
6: FPS: frame rate of the recorded video
7: **Output:**
8: V_{est}: Estimated velocities between every two frames
9: **Method:**
10: index = 0
11:     **while** index $\leq |F| - 1$ **do**
12:         a = Euclidean distance between centroids of index and index+1 frames (in pixels)
13:         $d_p = d_w \times a$
14:         $t_p = \frac{1}{\text{FPS}}$
15:         $V_{est} = \frac{d_p}{t_p}$
16:         index += 1
17:     **end while**
18: **end procedure**

5.5 IR Sensor based Setup

In the non-CV based setup, sensors are required to measure velocity. In this paper, Infrared (IR) modules are used to measure the velocity of the moving object. HW-201 Infrared Obstacle Avoidance Sensor Module [57] is used for this purpose. Table 5.1 shows the specifications of these sensors. IR sensors are placed in rectangular 3D printed cases that are fitted to the track's edges, as shown in Fig. 5.4(b), enabling the sensors to remain in a fixed position. The modules are placed perpendicular to the track, and their effective range is set to 3 cm to ensure accurate detection of the object.

Table 5.1 Specifications of HW-201 IR sensor [57]

Power Supply	Distance Range	Sampling Rate	Sensitivity Range	Frequency Range
3-5V DC	2-30 cm	1KHz	800-1100 nm	35-41 KHz

5.5.1 Calculations

An IR sensor is used to calculate the velocity in a Region of Interest (RoI), where the average velocity is approximated as the object's instantaneous velocity. The RoI is a small rectangular region (<8 cm^2) in front of the IR sensor (marked with dotted lines in blue regions in Fig. 5.4(b)), where the sensor can detect the presence of an object. A timer is started when the object passes through an RoI and is incremented till the object leaves the RoI. Let Δt be the total increments of the timer. The estimated velocity of the ball using IR setup (V_{IR}) is

$$V_{IR} = \frac{0.0185}{\Delta t}, \tag{5.7}$$

where the distance covered by the object will be equal to the diameter of the ball, i.e. 0.0185 m.

5.6 Results

In this section, the performance of CV-based implementation is first presented, including the effects of the number of data points considered N and the effects of FPS of the video. Later, the performance comparison between the CV and IR based implementations is presented. For all the experiments in this section, we are using the longest segment of the track, i.e., segment GA of length 420 mm, where the IR sensors are also deployed. Although the ball is released from point G, velocities are only measured from point H which is 50 mm apart from G. This is done to ensure that the object tracker is not disturbed by the movement of the gate G. The segment HA of length 370 mm (shown by the blue region in Fig. 5.4(b)) is divided uniformly into N data points and the object's velocities are estimated for these points. Heights h_i are measured with reference to point A (i.e., height of point A = 0). Initial point G is at a height h_1 of 20 mm. The conversion factor d_w is the ratio of length of the segment GA (420 mm) and Euclidean

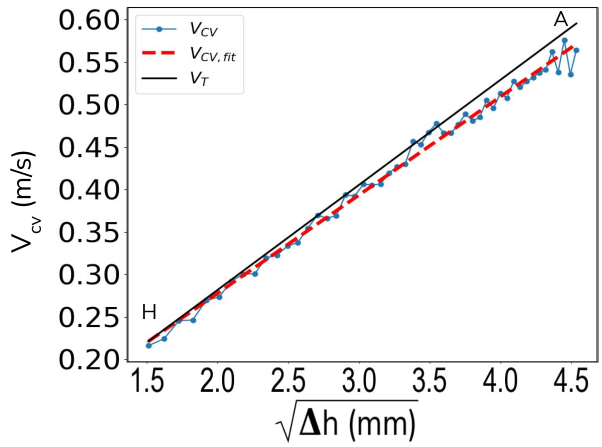


Figure 5.6 Plots of $V_{CV}, V_{CV,fit}$ and V_T vs $\sqrt{\Delta h}$ for N = 50 and FPS = 480

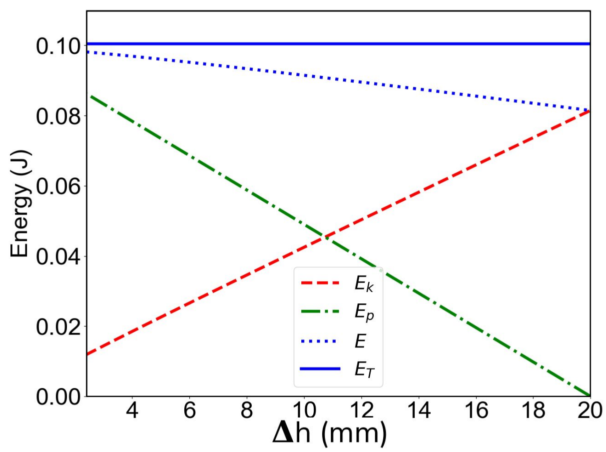


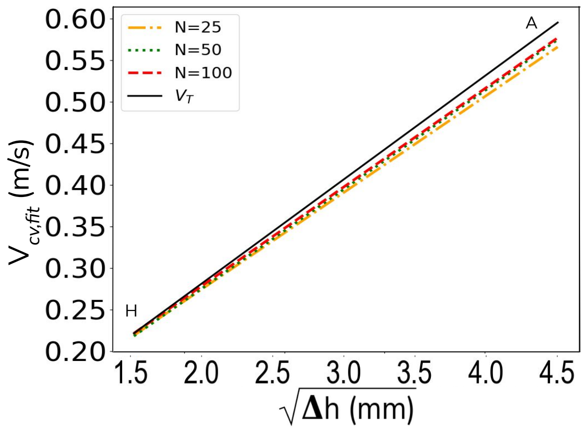
Figure 5.7 Plot of E_k, E_p, E_T and E vs Δh for N = 50 and FPS = 480

distance of the segment GA (in pixels). In general, students perform an RL experiment for about 5-10 times in a session. Therefore, the number of experimental runs N_r over which the results are averaged has been set to 10. Unless specified, default values of g, N and frame rate of the video in FPS are 9.8 m/s^2, 50 and 480, respectively. For the CV-based implementation, *OpenCV*[5] is used, which is a well-known open-source library that is extensively used in the domain of CV for image segmentation, object detection and object tracking. It has several built-in functions specifically designed for the purpose used directly in this work.

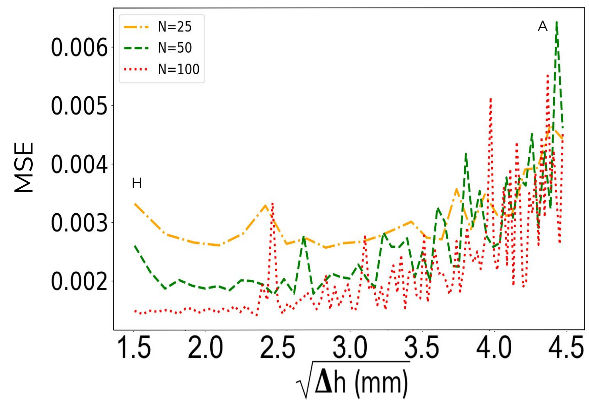
5.6.1 Effects of line fitting

Fig. 5.6 shows the theoretical velocities V_T and estimated velocities using CV (V_{CV} and $V_{CV,fit}$) at different points on the track as a function of the square root of difference of height of the object wrt

[5] `https://opencv.org/`



(a)



(b)

Figure 5.8 Plots for different number of regions (N) (a) Plot of $V_{CV,fit}$ vs $\sqrt{\Delta h}$ for FPS = 480 (b) Plot of MSE vs $\sqrt{\Delta h}$ for FPS = 480

point G (i.e., $\sqrt{\Delta h}$). From the figure, it can be observed that the curves are very close and the velocities increase as Δh increases. The second observation is that the error at various data points fluctuates. From (5.6), we have $v_i \propto \sqrt{\Delta h}$. Therefore, to improve the estimated values at individual data points, we used the line of best fit using linear regression $V_{CV,fit}$, which can be seen to have better performance as seen in Fig. 5.6. It can be observed that the estimated velocities are deviating from V_T as Δh increases. This can be attributed to measurement errors as well as friction along the track.

Fig. 5.7 shows the corresponding plot of E_k, E_p, E_T and the calculated mechanical energy E vs Δh. Here, E is the sum of E_k and E_p, where E_k is estimated using $V_{CV,fit}$. It can be noted that E_k is non-zero at point H as velocity is not estimated from the release point of the object, i.e, point G. Secondly, E deviates slightly from E_T and this happens due to friction and measurement errors.

5.6.2 Effect of the number of data points (N)

Figs. 5.8(a) and 5.8(b) show the line of best fits and Mean Square Error (MSE) with respect to $\sqrt{\Delta h}$ for N = 25, 50 and 100. From Fig. 5.8(a), it can be observed that the line of best fit comes closer to the theoretical value as N increases. However, the change from N = 50 to N = 100 is very small indicating that the line of best fit obtained is nearly optimal and the remaining bias is mainly due to friction. These claims are supported by Fig. 5.8(b), where MSE for N = 50 and N = 100 are very similar and lower compared to N = 25. A similar result can be observed from Fig. 5.9, which shows the MSE averaged over the segment HA as a function of N.

5.6.3 Effect of FPS of the video

Fig. 5.10 shows the plot of MSE for V_{CV} wrt V_T for N = 10 and FPS = 30, 60, 240 and 480. The low value of N is chosen to gather sufficient frames when FPS = 30. It can be observed that the MSE

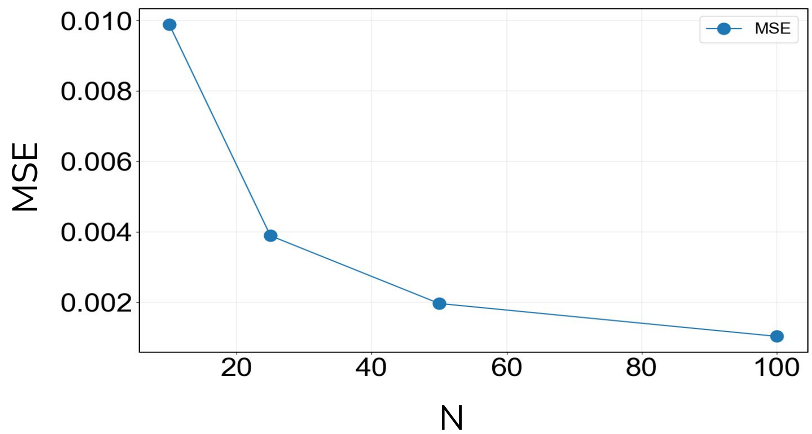


Figure 5.9 Plot of MSE vs N for FPS = 480

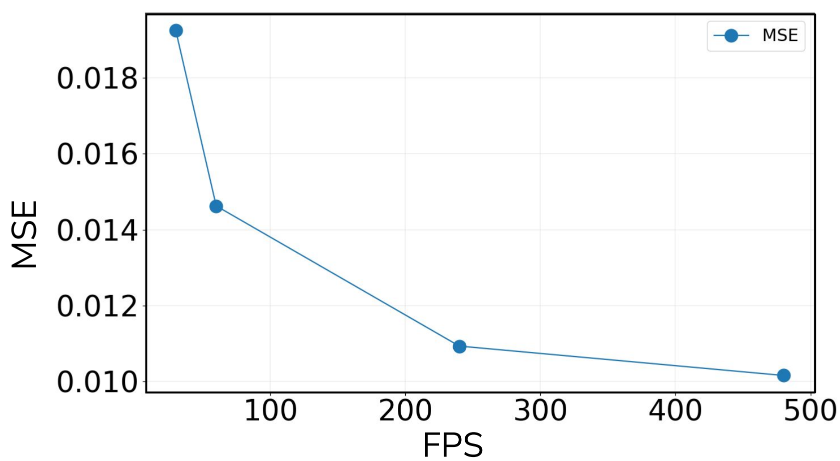


Figure 5.10 Plot of MSE vs FPS for N = 10

reduces as the FPS values of the recorded video increase. Secondly, it can be observed that the drop in MSE decreases as FPS increases from 30 to 480 FPS. There is no significant reduction in MSE for videos recorded above 240 FPS.

5.6.4 Comparison of CV and IR-based implementations

Table 5.2 shows the MSE for velocities estimated using CV (i.e., V_{CV} and $V_{CV,fit}$) and sensor-based (i.e., V_{IR} and $V_{IR,fit}$) implementations wrt V_T where $V_{IR,fit}$ is the line of best fit for velocities estimated using the IR-based implementation. Here, $N_{disc} = 5$ discontinuous regions are chosen for the IR-based implementation that is marked with dotted lines in Fig. 5.4(b). While using the CV-based implementation, velocities are estimated for $N = 50$ and velocities of all the data points lying in each of the N_{disc} regions are averaged for comparison purposes. It can be observed that the MSE increases with height due to obvious reasons in both cases. Also, line fitting improves the MSE for both

Table 5.2 Comparison of velocities obtained using both implementations for N_{disc} = 5, N = 50 and FPS = 480

Region	V_T (m/s)	MSE for V_{IR}	MSE for $V_{IR,fit}$	MSE for V_{CV}	MSE for $V_{CV,fit}$
1	0.25	0.0002	0.0001	0.0004	**0.00005**
2	0.32	0.0008	0.0007	0.0007	**0.00008**
3	0.40	0.0027	0.0013	0.0017	**0.00015**
4	0.48	0.0030	0.0019	0.0022	**0.00023**
5	0.59	0.0034	0.0024	0.0031	**0.00031**

implementations. However, the best MSE values were obtained for $V_{CV,fit}$ that are almost 10 times better showing that the CV implementation can be used instead of IR-based implementation. Also, we can increase the number of data points as we increase the FPS of the recorded video, allowing us to monitor changes happening at multiple points along the track. However, this is not possible with sensor-based implementations as the number of sensors that can be placed is constrained by the length of the track (a maximum of 12 IR sensors can be placed for this track).

Chapter 6

Survey & Challenges

This chapter first presents a summary of a user survey conducted with individuals who have used remote labs, followed by a discussion of various challenges encountered in the process of building remote labs from scratch.

6.1 User-feedback

Designing systems is essential, but usability and user appeal are equally important factors. Evaluating these aspects is best accomplished through user surveys. User feedback was collected from 47 undergraduate students at IIIT Hyderabad in December 2022, resulting in 90 combined responses for both experiments. The survey had seven questions that primarily questioned the system aspects apart from the conceptual understanding. This included questions that asked users to rate various aspects, such as the build quality of hardware experiments, live-streaming delay, and the accessibility of the software platform, among others [41]. Responses were provided using a Likert scale, where 1 represented complete dissatisfaction, and 5 indicated complete satisfaction. Table 6.1 displays the specific questions posed and the average scores for each.

The user feedback on the remote lab solution indicates a generally positive user experience, with high scores in usability, clarity of streamed results, and content quality, suggesting that the platform is both user-friendly and educationally effective. The scores ranged from 3.31 to 4.51 on a 5-point scale, with video latency being the primary area for improvement. Some users have reported their suggestions

Table 6.1 Average Scores and Responses of User Feedback on Remote Labs

Questions	Average Score / Response
How visually appealing was the dashboard?	4.41
How easy was it to navigate through different sections on the platform?	4.51
How accessible and responsive were the buttons and other controls?	4.48
What is the level of clarity of the experiments being streamed?	4.14
How is the latency of the video stream? (1 for high lag and 5 for low lag)	3.31
How satisfactory was the experiment in terms of conceptual understanding?	4.43
How well was the experiment described theoretically?	4.41

Figure 6.1 The summary of the comments provided in the feedback session is visualised in a word cloud.

at the end of their experimentation, and Fig. 6.1 shows the word cloud of the comments received. Some of the comments received were to introduce more experiments, reduce the video streaming delay and introduce new features like including 3D rendering of the experiment and demo of the experiment. Most users also found the experiment parameters and feedback form sufficient, reinforcing the solution's comprehensiveness and effectiveness. The results overall suggest that the remote lab solution successfully meets user expectations in various vital aspects, although minor improvements could enhance the experience further. These improvements are being made and are expected to be released in the upcoming versions.

6.2 Challenges

Creating remote labs is a significant task, and maintaining them is an equally major undertaking. The process from the initial ideation to the final implementation of these labs involves several challenges, especially in areas related to hardware and networking. These challenges are critical as they greatly influence how practical and durable these labs will be. Some of the challenges encountered are:

1. Unavailability of Specific Components: This issue is often due to supply chain disruptions, regional availability, or cost constraints. This was evident when the global silicon shortage [24] during the 2020s affected the availability of Raspberry Pis. Similarly, this can affect other mechanical parts too. Emphasising open-source designs and leveraging 3D printing technology can be an effective workaround. This approach reduces dependency on specific suppliers and encourages innovation and customisation in lab design.

2. Wear and Tear of Components: In remote labs, the frequent use of components like motors leads to wear and tear issues such as jamming and skipping. This necessitates regular maintenance and the availability of replacement parts. Implementing predictive maintenance using IoT sensors can help monitor the condition of these components and schedule maintenance before failures occur, thus minimising downtime.

3. Bulky Setups: The bulkiness of lab scale setups presented a logistical challenge, particularly when moving, servicing, or reconfiguring the setups. This issue was later mitigated by designing

modular setups, allowing easier disassembly and reassembly. Using lightweight materials and adopting a minimalist design approach can also reduce the bulk and enhance the portability of these setups.

4. Firewall Restrictions: The firewall restrictions often restrict the usage of essential applications required for remote access and control of lab equipment. This issue was particularly faced with the implementation of data protocols for data transmission from the controllers to platforms like Thingpseak that supported MQTT and MQTTS, which worked on 1883 and 8883—overcoming this involved configuring network settings of the campus network to allow specific traffic (for instance, allowing the UDP traffic, opening specific ports). However, later HTTP was used for data transmission.

Chapter 7

Concluding Remarks

7.1 Conclusions

The research carried out in this thesis mainly focuses on developing an end-to-end Remote Labs system. This is presented in two chapters: developing hardware for a low-cost IoT-based Remote Labs system and integrating CV techniques in a remote lab experiment.

Chapter 4 presented the development of a remote lab system named RLabs that included the development of two use case experiments along with a software platform. The proposed system is qualitatively evaluated against seven NFAs - affordability, portability, scalability, compatibility, maintainability, usability, and universality (in Chapter 3). The experiments were built by retrofitting IoT components on traditional laboratory equipment. Modular version of the same experiments are also built using 3D-printed components. The retrofitted setups of both the experiments costed less than 15k INR. The system is scalable as many experiments can be built at low cost and with fewer materials. The compatibility of the system is shown by connecting different hardware boards (like Raspberry Pi, ESP32), and cameras (RaspiCam, USB camera and IP Camera).

In chapter 5, the usage of CV on a remote labs system has been proposed and successfully demonstrated the same for the use case of *Conservation of Mechanical Energy*. In this experiment, CV is used to estimate the velocities of a moving object on different points along the track, which are observed to be close to the theoretical velocity values obtained from the principle of mechanical energy conservation. A comparison has been made between the CV and IR sensors-based implementations to compute the velocity of the moving object. After the detailed comparison, the CV-based implementation combined with linear regression outperforms the IR sensors-based setup while computing the velocities.

Finally, the challenges faced while implementing the labs are discussed in Chapter 6, especially when dealing with the hardware, which included wear and tear of components, bulky setups, and unavailability of the components, along with firewall restrictions. The usability survey, filled out by a group of 47 engineering students, showed an average score of 4.24, indicating a positive learning experience and good system usability.

7.2 Future Scope

In the future, our goal is to design new retrofitting approaches that will use “hardware blocks” similar to LEGO[1] blocks. These blocks will be easy to fit onto laboratory experiments and connect seamlessly. It is planned to integrate large language models (LLMs) into the platform to enhance the user experience by introducing chatbots to answer experiment-related queries. We plan to improve our CV-based detection algorithm to work with various types of experiments that involve visual changes. Our algorithms will also be optimised to work in real-time on controllers like Raspberry Pi by upgrading them to leverage parallel programming. Additionally, for the Conservation of Mechanical Energy experiment, we intend to study the effects of friction and estimate the coefficient of friction from the estimated velocity plots.

[1] https://www.lego.com/en-in

Bibliography

[1] "IoT connected devices worldwide 2019-2030 — Statista," 7 2023. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] S. Misra, A. Mukherjee, and A. Roy, *Introduction to IoT*. Cambridge University Press, 2021. [Online]. Available: https://books.google.co.in/books?id=zioNEAAAQBAJ

[3] P. Lea, *Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security*. Packt Publishing, 2018. [Online]. Available: https://books.google.co.in/books?id=3NRJDwAAQBAJ

[4] R. Kamal, *Internet of Things*. McGraw Hill Education, 2017. [Online]. Available: https://books.google.co.in/books?id=uS1HDwAAQBAJ

[5] S. Greengard. (2023) Internet of Things. Encyclopedia Britannica, 20 Nov. 2023. [Online]. Available: https://www.britannica.com/science/Internet-of-Things

[6] R. Kamal, *Internet of Things: Architecture and Design Principles*. Mc Graw Hill India, 2017. [Online]. Available: https://books.google.co.in/books?id=KuOizQEACAAJ

[7] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. USA: Prentice Hall Press, 2010.

[8] C. Bayılmış, M. A. Ebleme, Ünal Çavuşoğlu, K. Küçük, and A. Sevin, "A survey on communication protocols and performance evaluations for Internet of Things," *Digital Communications and Networks*, vol. 8, no. 6, pp. 1094–1104, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864822000347

[9] C. R. Reddy, T. Mukku, A. Dwivedi, A. Rout, S. Chaudhari, K. Vemuri, K. S. Rajan, and A. M. Hussain, "Improving Spatio-Temporal Understanding of Particulate Matter using Low-Cost IoT Sensors," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–7.

[10] A. Parmar, S. Sara, A. K. Dwivedi, C. R. Reddy, I. Patwardhan, S. D. Bijjam, S. Chaudhari, K. S. Rajan, and K. Vemuri, "Development of end-to-end low-cost iot system for densely deployed

pm monitoring network: an indian case study," *Frontiers in the Internet of Things*, vol. 3, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/friot.2024.1332322

[11] S. Sara, A. Rebeiro-Hargrave, A. Parmar, P. L. Fung, I. Patwardhan, S. Varjonen, C. R. Reddy, S. Chaudhari, and S. Tarkoma, "The Application of Mobile Sensing to Detect CO and NO2 Emission Spikes in Polluted Cities," *IEEE Access*, vol. 11, pp. 79 624–79 635, 2023.

[12] R. Bose, A. Parmar, H. Narla, and S. Chaudhari, "Comparative Evaluation of Low-Cost CO2 Sensors for Indoor Air Pollution Monitoring," in *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, 2022, pp. 1–6.

[13] T. Bhatt *et al.*, "String tension based borewell water level monitoring using IoT," 2023, (in press).

[14] A. K. Lall, A. Khandelwal, R. Bose, N. Bawankar, N. Nilesh, A. Dwivedi, and S. Chaudhari, "Making Analog Water Meter Smart using ML and IoT-based Low-Cost Retrofitting," in *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2021, pp. 157–162.

[15] A. K. Lall, A. Khandelwal, N. Nilesh, and S. Chaudhari, "Improving IoT-based Smart Retrofit Model for Analog Water Meters using DL based Algorithm," in *2022 9th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2022, pp. 207–212.

[16] N. Bawankar, A. Kriti, S. S. Chouhan, and S. Chaudhari, "Iot-enabled water monitoring in smart cities with retrofit and solar-based energy harvesting," *IEEE Access*, vol. 12, pp. 58 222–58 238, 2024.

[17] A. Celesti, A. Galletta, L. Carnevale, M. Fazio, A. Ĺay Ekuakille, and M. Villari, "An IoT Cloud System for Traffic Monitoring and Vehicular Accidents Prevention Based on Mobile Sensor Data Processing," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 4795–4802, 2018.

[18] M. M. Ali, S. Haxha, M. M. Alam, C. Nwibor, and M. Sakel, "Design of Internet of Things (IoT) and Android Based Low Cost Health Monitoring Embedded System Wearable Sensor for Measuring SpO2, Heart Rate and Body Temperature Simultaneously," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2449–2463, 2020.

[19] V. Jahmunah, V. K. Sudarshan, S. L. Oh, R. Gururajan, X. Zhou, X. Tao, O. Faust, E. J. Ciaccio, K. H. Ng, and U. R. Acharya, *Future IoT tools for COVID-19 contact tracing and prediction: A review of the state-of-the-science*. Wiley Online Library, 2021.

[20] S. S. Vedaei, A. Fotovvat, M. R. Mohebbian, G. M. E. Rahman, K. A. Wahid, P. Babyn, H. R. Marateb, M. Mansourian, and R. Sami, "COVID-SAFE: An IoT-Based System for Automated Health Monitoring and Surveillance in Post-Pandemic Life," *IEEE Access*, vol. 8, pp. 188 538–188 551, 2020.

[21] E. Navarro, N. Costa, and A. Pereira, “A Systematic Review of IoT Solutions for Smart Farming,” *Sensors*, vol. 20, no. 15, p. 4231, 2020.

[22] A. Dolgui and J. M. Proth, “RFID TECHNOLOGY IN SUPPLY CHAIN MANAGEMENT: STATE OF THE ART AND PERSPECTIVES,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 4464–4475, 2008, 17th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016396483

[23] “The Internet of Things (IoT) in retail: Bridging supply and demand,” *Business Horizons*, vol. 62, no. 1, pp. 47–54, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000768131830137X

[24] “Understanding the global chip shortage ,” 10 2023. [Online]. Available: https://www.bcs.org/articles-opinion-and-research/understanding-the-global-chip-shortage/

[25] A. R. Biswas and R. Giaffreda, “IoT and cloud convergence: Opportunities and challenges,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 375–376.

[26] G. Ihita, K. Viswanadh, Y. Sudhansh, S. Chaudhari, and S. Gaur, “Security Analysis of Large Scale IoT Network for Pollution Monitoring in Urban India,” in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 283–288.

[27] G. Ihita, V. Acharya, L. Kanigolla, S. Chaudhari, and T. Monteil, “Security for oneM2M-Based Smart City Network: An OM2M Implementation,” in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2023, pp. 808–813.

[28] A. Villar-Martínez, L. Rodríguez-Gil, I. Angulo, P. Orduña, J. García-Zubía, and D. López-De-Ipiña, “Improving the Scalability and Replicability of Embedded Systems Remote Laboratories Through a Cost-Effective Architecture,” *IEEE Access*, vol. 7, pp. 164 164–164 185, 2019.

[29] A. Villar-Martínez, J. García-Zubía, I. Angulo, and L. Rodríguez-Gil, “Towards Reliable Remote Laboratory Experiences: A Model for Maximizing Availability Through Fault-Detection and Replication,” *IEEE Access*, vol. 9, pp. 45 032–45 054, 2021.

[30] L. Rodríguez-Gil, J. García-Zubia, P. Orduña, and D. Lopez-de Ipiña, “An Open and Scalable Web-Based Interactive Live-Streaming architecture: The WILSP Platform,” *IEEE Access*, vol. 5, pp. 9842–9856, 2017.

[31] V.-M. Aitor, J. García-Zubía, I. Angulo, and L. Rodríguez-Gil, “Toward Widespread Remote Laboratories: Evaluating the Effectiveness of a Replication-Based Architecture for Real-World Multi-institutional Usage,” *IEEE Access*, vol. 10, pp. 86 298–86 317, 2022.

[32] I. Angulo, L. Rodrìguez-Gil, and J. Garcìa-Zubìa, “Scaling up the Lab: An Adaptable and Scalable Architecture for Embedded Systems Remote Labs,” *IEEE Access*, vol. 6, pp. 16 887–16 900, 2018.

[33] F. Lustig, J. Dvorak, P. Kuriscak, and P. Brom, "Open Modular Hardware and Software Kit for Creations of Remote Experiments Accessible from PC and Mobile Devices," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 12, no. 07, p. pp. 30–36, Jul. 2016. [Online]. Available: https://online-journals.org/index.php/i-joe/article/view/5833

[34] C. Lavayssière, B. Larroque, and F. Luthon, "Laborem Box: A scalable and open source platform to design remote lab experiments in electronics," *HardwareX*, vol. 11, p. e00301, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468067222000463

[35] J. B. d. Silva, G. d. Oliveira, I. N. d. Silva, P. M. Mafra, and S. M. S. Bilessimo, "Block.ino: Remote Lab for Programming Teaching and Learning," *International Journal of Advanced Engineering Research and Science*, vol. 7, no. 1, pp. 41–47, 2020.

[36] J. A. Ariza and S. G. Gil, "RaspyLab: A Low-Cost Remote Laboratory to Learn Programming and Physical Computing Through Python and Raspberry Pi," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 17, no. 2, pp. 140–149, 2022.

[37] J. Álvarez Ariza and C. Nomesqui Galvis, "RaspyControl Lab: A fully open-source and real-time remote laboratory for education in automatic control systems using Raspberry Pi and Python," *HardwareX*, vol. 13, p. e00396, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468067223000032

[38] J. Sáenz *et al.*, "Open and Low-Cost Virtual and Remote Labs on Control Engineering," *IEEE Access*, vol. 3, pp. 805–814, 2015.

[39] D. Reid, J. Burridge, D. Lowe, and T. Drysdale, "Open-source remote laboratory experiments for controls engineering education," *International Journal of Mechanical Engineering Education*, vol. 50, no. 4, pp. 828–848, 2022. [Online]. Available: https://doi.org/10.1177/03064190221081451

[40] A. V. den Beemt, S. Groothuijsen, L. Ozkan, and W. Hendrix, "Remote Labs in Higher Engineering Education: Engaging Students with Active Learning Pedagogy," *Journal of Computing in Higher Education*, vol. 35, no. 2, pp. 320–340, August 01, 2023 2023. [Online]. Available: https://doi.org/10.1007/s12528-022-09331-4

[41] A. Chevalier, C. Copot, C. Ionescu, and R. De Keyser, "A three-year feedback study of a remote laboratory used in control engineering studies," *IEEE Transactions on Education*, vol. 60, no. 2, pp. 127–133, 2017.

[42] N. Wang, X. Chen, Q. Lan, G. Song, H. R. Parsaei, and S.-C. Ho, "A Novel Wiki-Based Remote Laboratory Platform for Engineering Education," *IEEE Transactions on Learning Technologies*, vol. 10, no. 3, pp. 331–341, 2017.

[43] L. de la Torre *et al.*, “Two web-based laboratories of the FisL@bs network: Hooke’s and Snell’s laws,” *European Journal of Physics*, vol. 32, no. 2, p. 571, feb 2011. [Online]. Available: https://dx.doi.org/10.1088/0143-0807/32/2/027

[44] S. Dormido *et al.*, “Physics Experiments at the UNEDLabs Portal,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 8, no. S1, p. pp. 26–27, Jan. 2012. [Online]. Available: https://online-journals.org/index.php/i-joe/article/view/1945

[45] H. Vargas *et al.*, “A Network of Automatic Control Web-Based Laboratories,” *IEEE Transactions on Learning Technologies*, vol. 4, no. 3, pp. 197–208, 2011.

[46] A. Gureja *et al.*, “Software Architecture for Multi-User Multiplexing to Enhance Scalability in IoT-Based Remote Labs,” 2023, (in press).

[47] A. Das, K. S. Viswanadh, R. Agrawal, A. Gureja, N. Nilesh, and S. Chaudhari, “Using miniature setups and partial streams for scalable remote labs,” in *2023 10th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2023, pp. 256–263.

[48] A. Satpathy, “What are Non-Functional Requirements and How to Build Them,” https://www.modernrequirements.com/blogs/sub-topic/what-are-non-functional-requirements-and-how-to-build-them, 2023, accessed: 2023-12-05.

[49] S. J. Ling, J. Sanny, and W. Moebs, *University Physics*. OpenStax, Sep 2016, vol. 3.

[50] P. Kumar, “MIPI camera vs USB camera – a detailed comparison,” 2021. [Online]. Available: https://www.e-consystems.com/blog/camera/technology/mipi-camera-vs-usb-camera-a-detailed-comparison/

[51] M. Scarpino, *Motors for Makers: A Guide to Steppers, Servos, and Other Electrical Machines*. QUE Publishing, 2015.

[52] Sepfy, “GitHub - sepfy/libpeer: WebRTC Library for IoT/Embedded Device using C,” 2021. [Online]. Available: https://github.com/sepfy/libpeer/tree/main

[53] G. Nakagawa *et al.*, “A Learning Material for Physics Experiment with High-Accuracy Using Computer Vision Technique,” in *5th Intl Conf on Applied Computing and Information Technology (ACIT)*, 2017, pp. 86–89.

[54] K. Pattanashetty *et al.*, “Web-based physics experiments in dynamics using image processing,” in *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, 2016, pp. 1–5.

[55] G. Illeperuma *et al.*, “Computer vision based object tracking as a teaching aid for high school physics experiments,” in *4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017, pp. 1–6.

[56] M. Hajba *et al.*, "Elastic collisions visualization using OpenCV object motion tracking," in *43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2020, pp. 1573–1578.

[57] "HW-201 IR sensor," https://5.imimg.com/data5/YT/KV/MY-1833510/arduino-ir-infrared-obstacle-avoidance-sensor-module.pdf, accessed: 26- Apr- 2024.

[58] "Pi Camera specifications," https://picamera.readthedocs.io/en/release-1.13/fov.html, accessed: 31- Mar- 2022.