Scalable Remote Labs Using Miniature Setups and Partial Streams

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communications Engineering
by Research

by

Animesh Das
20161302
animesh.das@research.iiit.ac.in

Advisor: Dr. Sachin Chaudhari

International Institute of Information Technology Hyderabad
500 032, India

October 2023

International Institute of Information Technology Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that the work presented in this thesis titled ***Scalable Remote Labs Using Miniature Setups and Partial Streams*** by *Animesh Das* has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Dr. Sachin Chaudhari

Acknowledgement

It took me a really long time to write this out, and I would like to thank the numerous people in my journey for helping me get through.

To Professors Jayanthi, PJN, Shatrunjay and Dipti, thank you for that second chance back in mid-2016. To my former advisor, Dr. Suryakanth VG, thank you for taking me in, putting up with my erratic behaviour, and encouraging me in the right direction. From that first day in the Signals and Systems course in Monsoon 2017, to now, we have come a really long way. To Prathima ma'am, thank you for taking up my case with the RSAC and recommending me to Sachin sir. To the IIIT academic and hostel administration, and specially, to late Prof. Govindarajulu, thank you.

To my current advisor, Dr. Sachin Chaudhari, thank you for your guidance and support throughout the completion of this thesis. From bailing me out of the rut of no progress, to providing me with a problem statement to work on, you have always been there for me, even when I was feeling lost or overwhelmed. You have also been a great mentor to me, teaching me not only about research but also about life. I am truly grateful for your guidance and support.

My time with the Remote Labs team was one of the best times I ever had, with Viswanadh, Rishabh, Akshit and Raj taking care of everything, giving me time to work on the algorithm. I would also like to thank Purvansh and Akshara, the two people I worked with the entire summer, and Nagesh, my successor in Remote Labs, for the crucial work we all were able to do. SPCRC isn't just a research center, it's a family. And I'm grateful to this family for everything. I appreciate my labmates and friends Nikhil, Jigyasu, Ihita, Sumanth, Anjani, Om, Shreyash, Ankit, Nilesh, Nitin, and everybody else for keeping the energy high at the lab. A3-203 isn't just a room number now, but an emotion, thanks to them.

To Arhant, Dipankar, Mugdha, thank you for encouraging me to join Toastmasters. I never would have found my voice without it. To my friends in various Toastmasters clubs around the world, thank you for helping me find confidence, and for your valuable input. To my Raagini bandmates Haala, Arpit, and Rahul, thank you for being the people to look forward to. And for that gig in CYIENT - without it, I never would have found an outlet for myself.

I would like to thank my parents, Jagannath and Swapna Das, Dr. Swati Sood, my cousins Subhajit, Priya, Shubham, Sneha, my family, and my friends and batchmates for the constant support and unconditional love that kept me sane through this whirlwind I call college. I would not have been able to complete this thesis without them. I am eternally grateful for their unwavering support, encouragement, and love throughout my academic journey. Their belief in me has been my driving force and motivation.

It's been a long and bumpy journey to this point. But now, like all good things, this journey must end, for another to start.

Abstract

Remote labs allow students from anywhere in the world to access and conduct experiments without the need to physically be present in a lab at anytime. This is extremely important for students with little to no access to proper science labs because of a lack of infrastructure or a pandemic. A major open problem statement is adding scalability to existing Remote Labs, for several reasons, including the ability to handle growing demand while reducing costs and increasing flexibility. There is a potential for scaling up the process so that queue delay can be removed and people are able to access dedicated experiment setups.

The work presented in this thesis is aimed at scalability of Remote Labs by using miniaturization and partial streams. Miniaturization involves making the experiment apparatus simple, smaller and portable. Partial streams involves using a single camera for creating point-of-view video streams for multiple units of the same experiment apparatus. Both aspects operate in tandem in order to be able to scale up the in-house Remote Labs system.

The proof of concept demonstration of the methodology put forward is conducted using two miniaturized setups of the school-level experiment "Vanishing Rod". These experiments are set up in our Remote Labs at IIIT Hyderabad, India (IIIT-H). The miniaturized setups are 3D printed in the lab. To demonstrate the effectiveness of the proposed approach, a performance comparison in terms of cost, size, and energy consumption is carried out for the proposed architecture (miniaturized + partial streaming) compared to the traditional setup (lab-scale + single-streaming). In both cases, the software framework for the dashboard is developed and implemented at IIIT-H. With the above approach, power cost is reduced to one-sixth and actuation component cost is reduced to one-fifth. In addition, the miniature setups require less space as compared to the lab-scale setups, making it easier to install.

Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Program Interface
AQI	Air Quality Index
CAs	Conversational Agents
CLAHE	Contrast-Limited Adaptive Histogram Equalisation
CoAP	Constrained Application Protocol
COVID-19	Coronavirus Disease - 2019
CV	Computer Vision
DIY	Do-It-Yourself
DL	Deep Learning
DSLR	Digital Single-Lens Reflex
EBTC	European Business and Technology Centre
FPS	Frames Per Second
GB	Gigabyte
GHz	Gigahertz
HTTPS	Hyper Text Transfer Protocol
HVAC	”Heating, Ventilation, and Air Conditioning”
IC	Integrated Circuit
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things.
INR	Indian Rupee
IoT	Internet of Things
IR	Infrared
LMS	Learning Management Systems
LoRa	Long Range
M2M	Machine-to-Machine
mA	Milli-Amperes
MB	Megabyte
MCU	Micro-Controller Units
MEITY	Ministry of Electronics and Information Technology

ML	Machine Learning
MP	Micro-Processor
MQTT	Message Query Telemetry Transport
MSE	Mean Square Error
PCB	Printed Circuit Board
PILAR	Platform Integration of Laboratories based on the Architecture of VISIR
PM10	Particulate Matter with diameter less than 10 micrometers
PM2.5	Particulate Matter with diameter less than 2.5 micrometers
RAM	Random Access Memory
RH	Relative Humidity
RLMS	Remote Laboratory Management System
RTMP	Real-Time Messaging Protocol
SoC	System-on-Chip
TDS	Total Dissolved Solids
UNILabs	University Network of Interactive Labs
UV4L	Universal Video4Linux
V4L2	Video4Linux 2.0
VISIR	Virtual Instrument Systems in Reality
WAN	Wide Area Network
WebRTC	Web Real-Time Communication
Wh	Watt-hour
WHO	World Health Organisation
WiFi	Wireless Fidelity

List of Related Publications

[P1] Animesh Das, K. S. Viswanadh, Rishabh Agrawal, Akshit Gureja, Nitin Nilesh and Sachin Chaudhari, "Using Miniature Setups and Partial Streams for Scalable Remote Labs," in proceedings of The IEEE International Conference on Future Internet of Things and Cloud (FiCloud), Marrakesh, Morocco, August 2023.

Chapter 1

Introduction

1.1 Motivation

Access to basic laboratory facilities is a problem in many developing nations' schools and institutions, particularly in rural areas. This problem worsened during COVID-19, and even educational institutions with excellent experimental sets were unable to access their labs. In these circumstances, Remote Labs, which use the Internet of Things (IoT), can be useful since the students can conduct the experiments remotely on a real experiment setup through the Internet from any location in the world at any time of the day. Each experiment's input parameters are user-configurable, and the dashboard receives the associated outcomes. With smart devices like computers or smartphones, these outputs can be seen using a browser [1, 2]. Plots and tables are used to visualize the results so that you may readily make observations and comprehend the experiment.

To create a remote lab, the experiment apparatus is required to be retrofitted with sensors and actuators as needed, the micro-controllers should be connected to the Internet to be able to access the cloud server, which has to be able to forward the data to the client in a format that the front-end is able to read. Simultaneously, a system should be in place for showing the live experiment apparatus, like a point-of-view live-stream. For multiple experiments, a management system needs to be in place. A dashboard must be created for ease of access. A Remote Laboratory Management System (RLMS) is an essential pre-requisite for interfacing between the micro-controller/micro-processor and the web front-end.

In every step of the way, open problem statements exist. For example, IoT retrofitting for physics experiments seems straightforward as a task, but seems slightly difficult for chemistry experiments, particularly because of electronic board safety hazards involved due to possible fumes or vapours of chemicals or high-energy reactions. Creating RLMS architectures that are lightweight and easy to setup seems to be an open challenge that researchers and industry intermediates in the field have taken up over the years. A major open problem statement is adding scalability to existing Remote Labs, for several reasons, including the ability to handle growing demand while reducing costs and increasing flexibility.

There have been few works on the scalability of Remote Labs in literature [3–6].

The work presented in this thesis is aimed at scalability of Remote Labs by making the experiment apparatus simple, smaller and portable, and using a single camera for creating point-of-view video streams for multiple units of the same experiment apparatus, both aspects operating in tandem in order to be able to scale up the in-house Remote Labs system. Given that these labs are now online, the lab setups can be miniaturized as long as the visualization of the results through the live stream is not affected. This can be easily done as modern-day cameras have good resolution, and advanced image/video processing techniques can be used. There are several benefits of the miniaturization of remote lab setups, including affordability, flexibility, and scalability. Most miniaturized setups can be easily 3D printed at a low cost in the lab. This reduces the cost and energy consumption of the experimental setups as smaller components are often less expensive than their larger counterparts. Miniaturization improves portability as the smaller setups are easier to transport and deploy in remote and hard-to-reach locations. In addition, miniaturized setups take much lesser space, and they can be easily stacked up on a shelf, and then a single camera can be used to do multi-experiment streaming (or partial streaming), reducing the scaling cost significantly. Partial streaming [7], which selectively transmits only parts of the video necessary for display, can significantly reduce the amount of data required to transmit a video stream while maintaining acceptable levels of video quality.

1.2 Contributions

The following are the technical contributions in this thesis:

- The proof of concept demonstration of the methodology put forward is conducted using two miniaturized setups of the school-level experiment “Vanishing Rod”. These experiments are set up in our Remote Labs at IIIT-H [8]. The miniaturized setups are 3D printed in the lab.
- Image processing techniques are used to stream the two experiments simultaneously using only one camera and microprocessor setup. Although demonstrated for two setups, this method can be scaled for more setups depending on the camera and the experiment.
- Costs are further reduced by replacing the expensive microprocessor with a very low-cost microcontroller. This replacement is done only for the two actuation setups while the camera is still operated using one micro-processor unit.
- To demonstrate the effectiveness of the proposed approach, a performance comparison in terms of cost, size, and energy consumption is carried out for the proposed architecture (miniaturized + partial streaming) compared to the traditional setup (lab-scale + single-streaming). In both cases, the software framework for the dashboard is developed and implemented at IIIT-H. With the above approach, power cost is reduced to one-sixth and actuation component cost is reduced to one-fifth.

In addition, the miniature setups require less space as compared to the lab-scale setups, making it easier to install.

1.3 Structure of the thesis

The rest of the thesis is organised as follows:

- **Chapter 2** offers a concise introduction to IoT, including a discussion of its four-layer architecture, followed by a brief on its various applications and the challenges associated with its implementation.
- **Chapter 3** discusses in detail the concept of Remote Labs and how a remote lab is developed component-wise, before going into the literature survey done regarding the various advancements in and enhancements to Remote Labs. The focus of the literature survey is on the scalability of Remote Labs, as a precursor to the contribution of the thesis.
- **Chapter 4** describes the proposed methodology involving miniaturisation and partial streams to minimise component costs while maintaining user experience in Remote Labs. The use-case of Vanishing Rod experiment is taken to demonstrate the approach. The chapter also discusses additional results obtained apart from the related publication authored.
- **Chapter 5** presents the conclusion and future direction of the thesis.

Chapter 2

An Overview of IoT

In this chapter, a brief introduction is given on IoT, and how it works. In this section, the architecture for a general IoT system is given and explained, followed by a discussion on its applications, and the various challenges faced. Note that this is just an overview, and if more details are needed, one can refer to the various books available on the topic of IoT [9–11].

2.1 Definitions

Broadly, IoT is the network of embedded computing devices in commonplace objects (referred to as "things" in IoT) via the Internet, which empowers them to exchange data and perform consequential actions based on the information received. According to the United Nations International Telecommunication Union (UN-ITU), "*IoT is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies*" [12].

2.2 Architecture

Fig. 2.1 shows the general IoT architecture, which is a combination of components such as **sensors**, **actuators**, **processors**, **communication protocols**, and **cloud services** that make up IoT systems. There are two kinds of flow in the IoT architecture – data flow and control flow. While various researchers have defined IoT architecture between four to six layers, this thesis presents a four-layer architecture where sensors and actuators are the same layer. Also shown are the 2 kinds of flows.

1. **Sensors and Actuators**: Many different types of **sensors** exist in IoT devices, each designed to measure different physical properties or environmental conditions at any moment. Some common examples include:

   (a) **Environmental** sensors like temperature, humidity, light, air quality, pressure.

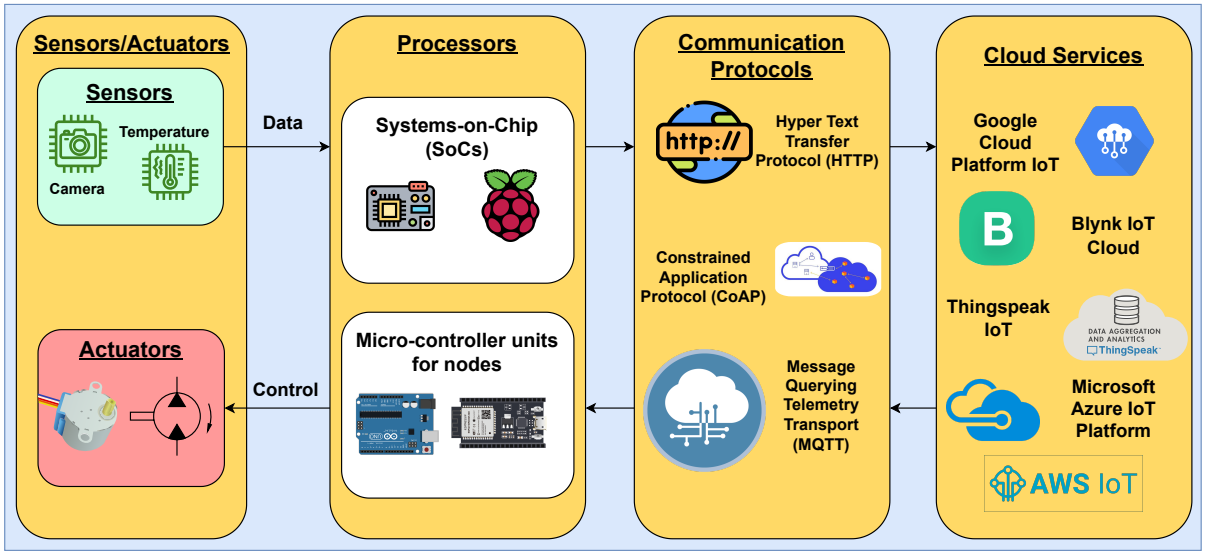


Figure 2.1 Four-layer IoT architecture consisting of **(i)** sensors and actuators, **(ii)** processors, **(iii)** communication protocols, and **(iv)** cloud services.

(b) **Visual** sensors to capture images or video, which can be used for various applications.

(c) **Acoustic** sensors to detect and measure sound waves or vibrations.

(d) **Motion** and **proximity** sensors.

Actuators can range from a simple stepper motor to a peristaltic pump operated by a DC motor. A peristaltic pump can be used to push liquids from one container to another, and can be used in various applications like a low-cost system for performing titration experiments.

2. **Processors**: Depending on the specific requirements of the device and application, a variety of processors exist for IoT. MCUs (Micro-controller units) are small, low-power processors designed for embedded systems, and therefore can be used for simple data processing or control functions. Examples of MCUs include Atmel AVR, ARM Cortex-M and Espressif ESP32. SoCs (Systems-on-chip) are ICs that combine multiple components like microprocessors, memory and I/O interfaces into a single chip and therefore can be used for more advanced processing functions. Raspberry Pi boards are good examples of SoCs.

3. **Communication Protocols**: It is essential for IoT devices to have a communication protocol to be able to exchange data and communicate with other devices and systems. MQTT, HTTP and CoAP are some of the commonly used protocols. MQTT is a lightweight publish-subscribe protocol commonly used for machine-to-machine (M2M) communication in applications such as smart energy. HTTP is a standard protocol used for transmitting data over the internet, commonly used in IoT applications for cloud-based services like remote monitoring and control. CoAP is a lightweight protocol for IoT applications that require low-latency communication.

4. **Cloud Services**: Cloud computing plays a pivotal role in IoT as a fundamental architectural component. The integration of cloud services with IoT systems enables the efficient processing, storage, and analysis of the massive volumes of data generated by IoT devices. By leveraging the cloud, IoT deployments can offload computational tasks, allowing devices with limited processing power to focus on data collection and basic operations. The cloud also facilitates real-time data analysis, enabling actionable insights to be derived from sensor data. This centralized approach enables remote monitoring, management, and control of IoT devices, making it possible to scale IoT applications without significant hardware investments. Additionally, cloud services enhance the flexibility and adaptability of IoT solutions, allowing rapid development and deployment of applications and services across various industries, from healthcare to smart cities. The cloud provides these categories of services:

   (a) Infrastructure as a Service (IaaS) – Amazon Web Service (AWS), Microsoft Azure, ThingSpeak.

   (b) Platform as a Service (PaaS) – Blynk IoT Cloud, Google Cloud IoT Core.

   (c) Software as a Service (SaaS) – Dropbox, Office 365.

2.3 Applications of IoT

IoT has a wide range of applications across different industries, transforming the way people interact with their surroundings and enabling a more connected and efficient world. In addition to Remote Labs, here are some key applications of IoT:

1. **Air pollution monitoring**: Current systems for measurement and analysis of air pollution are large and expensive, and suffer from difficulty in accessibility of information. This prohibits their production and deployment and raises the need for a compact, IoT-enabled low-cost version with easy access to data. As proposed in [13], a set of IoT-enabled low-cost nodes can be created and deployed in a small area to test out deployment before expansion. This deployment can then be optimised using hierarchical clustering-based spatial sampling as has been proposed in [14]. These nodes can also be functionally improved using the method proposed in [15]. Dense deployment has a lot of benefits as has been previously shown in [16], and more kinds of data, specifically CO and NO_2, can be collected to provide a better understanding of pollution, as has been described in [17]. However, limitations of all these in terms of requirement of special gas and PM sensors have been addressed by combining image processing, ML and temperature-RH sensing to estimate AQI in real-time with 82% accuracy [18]. Similarly, use of ML on real-time traffic data instead of time-stamped images is also proposed in [19].

2. **Smart Water Meter**: Difficulties exist in measurement of water levels using analog water meters in large containers, reservoirs, borewells, due to positioning of equipment at hard-to-reach places.

Hence, an IoT-based economic retrofitting setup can be, and has been, created for digitizing the analog water meters to make them smart. This system, presented in [20] and further discussed in [21], employs a combination of ML and IoT in a cost-effective retrofitting approach, enabling analog water meters to collect and transmit real-time data. Additionally, [21] introduces DL algorithms to further enhance the efficiency and accuracy of the IoT-based retrofit model, paving the way for a more precise and sophisticated water consumption monitoring system. These innovations represent a significant step forward in the domain of water management, offering a promising solution for more efficient resource utilization.

3. **Energy Monitoring**: An end-to-end energy monitoring system is proposed in [22]. Real-time data on electric consumption and vital energy parameters is transmitted through LoRaWAN to the oneM2M platform where it undergoes comprehensive processing and analysis.. The results are then presented through intuitive interfaces, empowering users with immediate insights into their energy usage patterns. The integration of LoRaWAN and oneM2M not only enhances scalability but also paves the way for efficient management of large-scale energy networks.

4. **Room Occupancy Monitoring**: In many construction projects, a major portion of energy is spent on HVAC systems. Making them demand-driven depending on human occupancy can help with usage optimization. One way to accurately estimate human occupancy in an enclosure involves the use of multiple heterogeneous sensor nodes and applying ML models on the data collected. For this purpose, as demonstrated in [23], inexpensive and non-intrusive sensors can collect data on the five parameters of illumination, temperature, CO2, sound and motion.

5. **Water Quality Monitoring**: According to [24], a low-cost and robust IoT-based TDS measurement system could be built for smart campuses. The low-cost design guarantees precise and uninterrupted output data. The dynamic reading of data, storage capacity, and calibration errors of sensors are the major challenges for IoT-based TDS measurement systems. These challenges can be taken on using a non-invasive mechanism for data collection, wireless connectivity to the data server, and machine learning calibration of sensor nodes.

6. **Smart Lamppost**: Smart Lamp-Posts can enable real-time collection of city data, such as weather, air quality, temperature, people and/or vehicle flow related information, for city management and the support of various applications of smart city initiatives. They can also provide services such as WiFi hot-spots, electric vehicle charging facilities, information dashboard for maps and directions, real-time traffic updates, and car parking vacancy space information [25].

These applications are just a glimpse of the wide-ranging impact that IoT is having on various industries and aspects of our lives, making them more efficient, convenient, and interconnected. Many of the above applications are a part of the Smart City Living Lab [26], an open-innovation ecosystem set up at IIIT Hyderabad, India, with support from the MEITY, Smart City Mission and Government of Telangana and in collaboration with the technology partners EBTC and Amsterdam Innovation Arena,

and a dynamic research initiative focused on leveraging advanced technologies for urban development. Through interdisciplinary collaboration and real-world testing, the lab pioneers innovative solutions to address complex urban challenges. By engaging with stakeholders, conducting pilot projects, and emphasizing data-driven decision-making, the Living Lab sets a standard for smart city development and serves as a beacon of best practices in urban innovation.

2.4 Challenges

While IoT offers many benefits, there are several challenges that must be addressed to ensure the success and widespread adoption of the technology. Here are some of the key challenges facing IoT:

- **Sensor cost and maintenance**: IoT relies heavily on sensors, which are a hard requisite in collecting data and providing valuable inferences as per the applications. However, cost is a big factor in IoT implementation, since initial costs are high, especially when buying and deploying in bulk. The cost of the sensor can also vary depending on the type, accuracy, and the range of data they can collect. Sensors require regular maintenance, occasional replacement, and periodic calibration takes up a lot of time as well [15, 27].

- **Power optimisation**: Many IoT devices rely on batteries and backup power, and the limit on the stored charge means continuous maintenance. Since IoT devices have to be deployed at hard-to-reach places, it is imperative that power consumption be kept to a minimum. Hence, power optimisation is required for optimizing battery life while maintaining device performance.

- **Performance**: IoT applications involve increasingly many devices and sensors, so the system must be able to deal with and accommodate the additional, expanding load [28]. Ensuring that the system can handle the increasing volume of data and traffic is crucial for maintaining performance benchmarks. Since many IoT devices rely on batteries, their processing power is limited and so are their data transmission capabilities [29]. Hence, power management is required for optimizing battery life while maintaining device performance.

- **Interoperability**: Because IoT devices and systems often come from different vendors, they may use diverse hardware components, protocols, and standards, hence making it difficult to integrate them all into a single system. Additionally, combining and analyzing data from various sources might be challenging due to difference in data formats [30].

- **Privacy and security**: In the effort to compress everything into as small a board as possible, a tradeoff exists between adding security features and keeping the essentials. The majority of IoT devices are, therefore, not completely suitable for deployment in hostile environments in a regulated and remote manner. Additionally, the large amounts of data collected by IoT devices raise concerns about data privacy and the potential misuse of personal information [31].

Chapter 3

Remote Labs - A Literature Survey

This chapter delves into the idea of Remote Labs, covering the evolution of laboratories from classical to virtual to remote, and briefs on what it takes to create a Remote Lab, before diving into the important literature on the existing work on Remote Labs around the world, and finally bringing it back home with a summary on the work done so far by Remote Labs at IIIT-H. It is to be understood that IoT is inherently involved in the building blocks of Remote Labs around the world.

3.1 Evolution of Laboratory

"Laboratories" or "Labs" play a crucial role in clarifying concepts in science education. A laboratory is a defined space (e.g., a room or building), where experiments are done in a controlled environment, away from the field or factory. In schools and colleges, experiments are done in laboratories under the guidance and supervision of mentors. Such experimental teaching and learning has often been regarded with special attention as an essential complement to the abstraction of concepts and of defined methodologies and procedures [32]. In summary, laboratories provide an interactive and dynamic learning environment that complements the classroom teaching.

Fig. 3.1 shows the evolution of laboratory. There are three kinds of labs as shown in Fig. 3.2: *classical hands-on*, *simulated labs*, and *remote labs* [33]. All these types are explained briefly here.

3.1.1 Hands-On Labs

Hands-on labs, as shown in Fig. 3.2(a), are the labs, where all the equipment required to perform the laboratory is physically set up and the students are physically present in the lab. Hands-on labs are

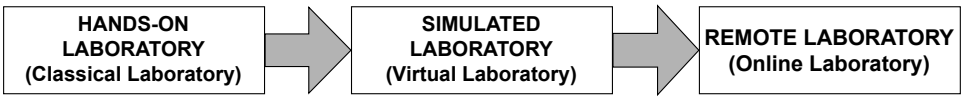
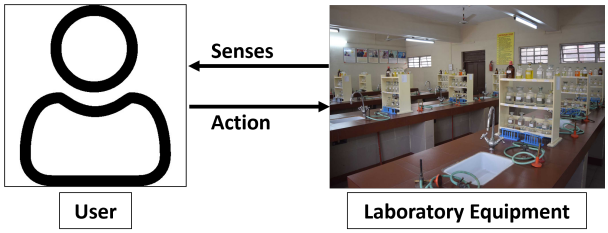
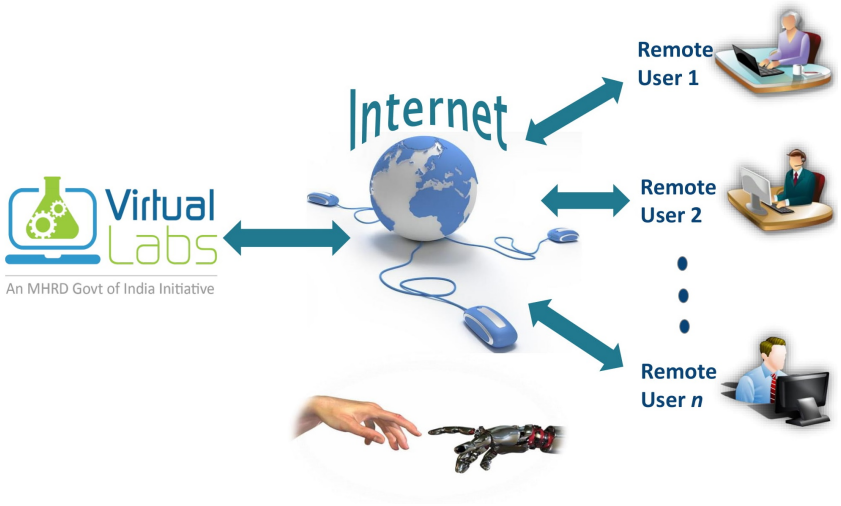


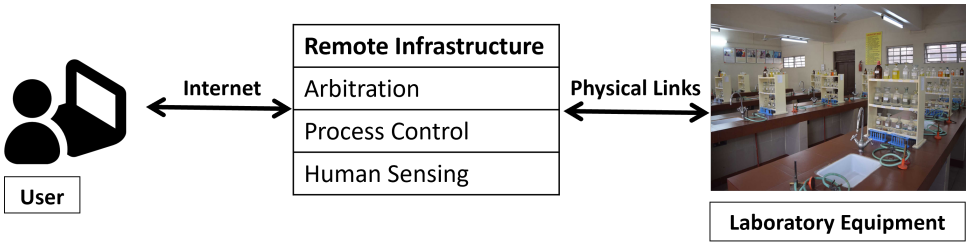
Figure 3.1 Evolution of labs.



(a)



(b)



(c)

Figure 3.2 Three kinds of labs. (a) Classical hands-on laboratory. (b) Virtual simulated laboratory (c) Remote laboratory.

good because they provide the students with real, practical data and help them understand the difference between practical phenomena and their theoretical understanding of said phenomena. At the same time, however, they may also be constrained by factors like equipment availability, safety concerns, and resource limitations. This creates the need for an easily accessible alternative, and simulated labs are well-known to serve that purpose [33].

3.1.2 Simulated Labs

Also known as virtual environments, simulated labs replicate real experiments using computer software. They offer a controlled and safe space for learners to conduct experiments. Simulated labs enable learners to explore scenarios that may be complex, expensive, or even dangerous to recreate in a physical lab. For instance, experiments involving rare or expensive materials, extreme temperatures, or hazardous substances can be safely conducted in a simulated environment. However, they are not guaranteed to be able to capture, or take into account, the complete sensory feedback of hands-on labs. Data from simulated labs are not real and therefore, the students can't learn by trial-and-error. Hence arises the need for remote labs [33].

3.1.3 Remote Labs

Remote labs, shown in Fig. 3.2(c), enable learners to remotely control real laboratory equipment through the internet. This approach combines the benefits of hands-on labs with the convenience of online accessibility, thus offering the best of both worlds. It provides opportunities for experimentation that may not be feasible due to geographical constraints or resource limitations. Remote labs have been gaining recent traction in the post-pandemic era due to their ability to provide affordable real experiment data and their extension of conventional laboratories beyond borders through the power of the Internet. The flexibility of Remote Labs increases the number of times and places a student can perform experiments. Because remote labs allow users to perform experiments and laboratory tasks over the Internet without being physically present near the actual equipment, hence the same interaction takes place at a distance with the assistance of the remote infrastructure. There is a 'new layer' that sits in between the user and the laboratory equipment. It is responsible for conveying user actions and receiving sensory information from the equipment [34].

3.2 Component-wise development of Remote Labs

As shown in Fig. 3.3, in creating Remote Labs for various experiments, it is important to take care of each and every component involved. To create a remote lab, the experiment apparatus is required to be retrofitted with sensors and actuators as needed, the micro-controllers should be connected to the Internet to be able to access the cloud server, which has to be able to forward the data to the client in a format that the front-end is able to read. For multiple experiments, a management system needs to be in place. A dashboard must be created for ease of access.

The RLMS is a system that centralizes access control to laboratory experiments and, additionally, provides functions for the management of laboratory experiments, such as: authentication, authorization, scheduling, lab resources management and authoring tools for the management of the laboratory experiments activities. This is required for interfacing between the micro-controller/micro-processor

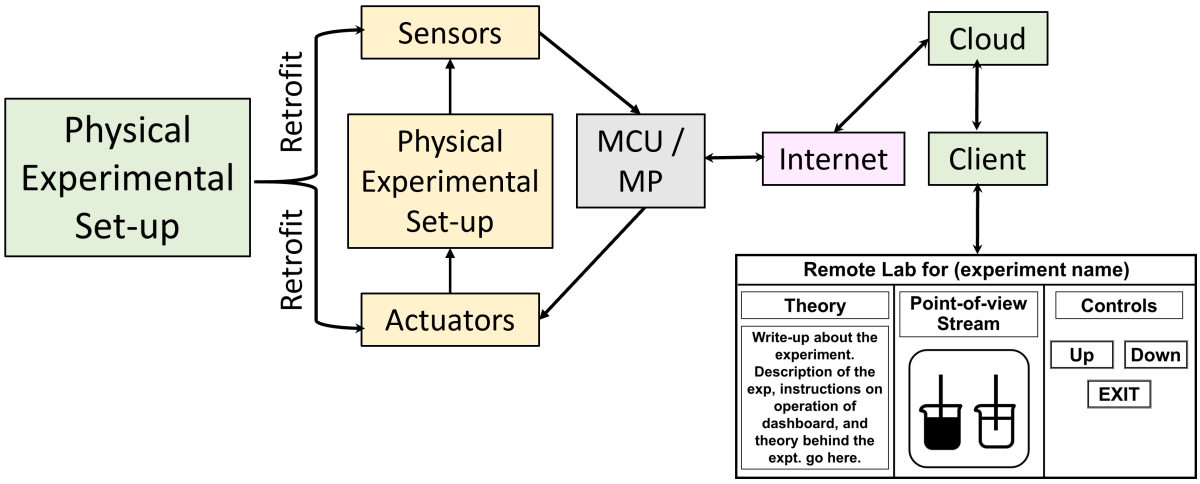


Figure 3.3 Block diagram of a remote lab. It is to be noted that the sensors include video as well. It is to be understood from this figure that IoT is integral to the creation of a remote lab.

and the web front-end. Most of the literature discussed in the next section is focused on the RLMS section of a remote lab, while some propose entirely different workflows for end-to-end Remote Labs.

3.3 Remote Labs Around the World

This section takes a look at some of the important work that has been done on the concept of Remote Labs around the world. The focus of this section is to highlight some of the source material that serves as inspiration for the method proposed in this thesis.

3.3.1 LabsLand Team - DeustoTech

LabsLand is a global network of remote laboratories, that connects institutions to equipment located worldwide [35]. This idea of a sharing economy where multiple providers provide access to their laboratories to each other, was first coined in [36]. The contribution by the LabsLand team in [37] explores, through use cases, various strategies to overcome both technical and organizational challenges associated with implementing remote laboratories in a commercial setting In [38], a visual domain-specific language and web-based authoring platform for intelligent tutors and CAs tailor-made for Remote Labs is proposed. The authors in [4] introduce a flexible and expandable framework designed to enhance remote access to embedded systems laboratories. [39] delineates the Erasmus+ project known as PILAR.

In [3], an open and scalable web-based interactive live-streaming architecture designed for remote laboratories is proposed as shown in Fig. 3.4. The platform is designed to be flexible and customizable, allowing users to create their own remote laboratory environments and adapt the platform to their specific needs. Using [3] as a basis, [5] proposes a remote lab architecture, shown in Fig. 3.5 for embedded

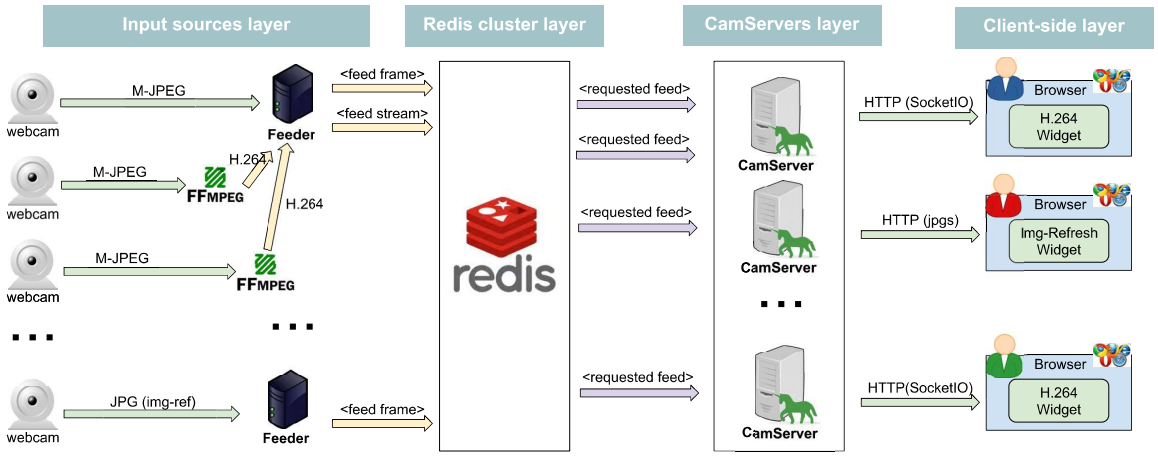


Figure 3.4 Diagram for the software architecture of WILSP as proposed in [3].

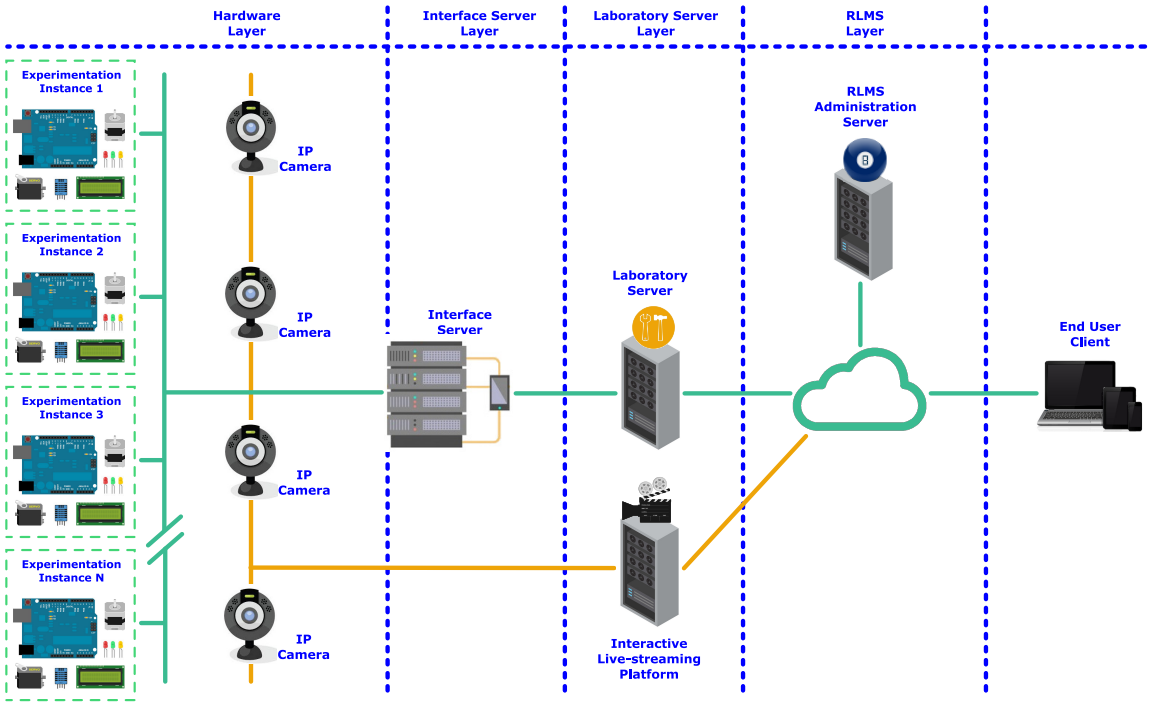


Figure 3.5 Block diagram of the architecture for Arduino Uno Remote Lab as proposed in [5].

systems experiments with the intent to provide scalability through universality and adaptability. The several instances of embedded systems experiments are conducted on Arduino Uno boards, which are connected and controlled by a single Raspberry Pi, which in turn is connected to the laboratory server and the RLMS layer, which is hosted in the cloud.

3.3.2 UNILabs

UNILabs [40, 41] is a University Network of Interactive Labs. It houses a collection of 15 virtual and remote laboratories, comprising 30 web-based labs encompassing simulations and experiments operated remotely, sourced from 8 distinct universities. The web interface is powered by Moodle, an advanced Learning Management System fostering a collaborative environment for students. This integration with Moodle enhances the virtual and Remote Labs, providing a platform for students to engage in discussions regarding experiments and to exchange their lab reports with peers and instructors. Furthermore, Moodle streamlines the dissemination of essential resources required for a comprehensive online experiment. For example, within a Moodle-hosted course linked to a web-lab, students can access descriptions of the studied phenomena, instructional videos, task protocols, questionnaires, and more. Lastly, Moodle enables the creation of open courses, ensuring accessibility to anyone interested, including self-motivated learners and educators seeking demonstrative resources for their classes.

3.3.3 Other universities

In [6], the authors propose a layered architecture that separates the user interface, experiment management, and hardware control aspects of Remote Labs. This modular design allows each layer to be independently scalable and adaptable, making it easier to provide a seamless experience for a wide range of users and experiments. The effectiveness of this architecture is demonstrated through a practical case study, where a remote laboratory is implemented for teaching control theory concepts. This lab incorporates various hardware components and sensors.

The results in [6] demonstrate that the proposed architecture can efficiently support a substantial number of users while maintaining scalability and flexibility. Overall, the "Scalable Remote Experiment Manager" paper provides a valuable framework for designing and implementing remote laboratories, emphasizing the significance of scalability and adaptability to cater to the evolving needs of users in educational and research environments.

3.3.4 Remote Labs @ IIIT-H

In [42], applications of CV are demonstrated for experimentation in Remote Lab, with the use-case of Conservation of Mechanical Energy. The setup is mainly made of a camera, a microprocessor and IR sensors. The CV-based approach employs video-processing techniques to estimate an object's velocity. This method is then compared with another IR sensor-based approach for the same purpose. Linear regression applied to the CV-based implementation results in an optimal MSE, nearly 10 times better than IR-based implementation.

In [43], the case study of the Kirchhoff's Voltage Law experiment was taken up to propose a 3-layer software architecture using WebSocket, and a comprehensive communication pipeline developed from

scratch, to enhance user experience, accelerate input-output communication and implement multi-user multiplexing. This serves as a counter to the inefficiency of existing web-based Remote Labs with 4-layered architectures, mainly existing implementations that use Blynk IoT platform as the middleware.

Chapter 4

Using Miniature Setups and Partial Streams for Scalable Remote Labs

This chapter describes the work done in the thesis. It begins by describing the hardware and software created for the lab-scale experiment setup for the Vanishing Rod Experiment, beginning with the peripheral and the circuitry before going into the dashboard created. Next, the miniature setup for the same experiment is described, before the partial streams methodology and the effect of the same on the software flow is described. Next, the various results, including the power consumption, monetary cost, and the possible bottlenecks are discussed.

4.1 Lab-Scale Experimental Setup and Dashboard

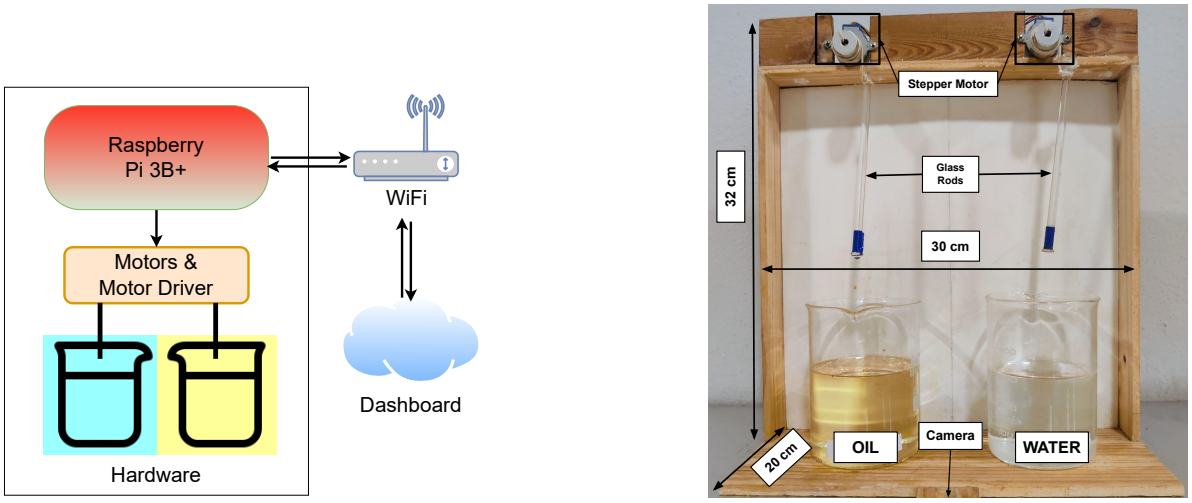
In this section, the physics principles involved in the experiment are presented and the proposed lab-scale experimental setup is described along with its dashboard.

4.1.1 Theory

The vanishing rods experiment [44] demonstrates the principles of refraction of light. When a glass rod is immersed in a beaker of oil with the same refractive index as of the rods, the light passing through the oil does not get refracted at the oil-glass interface, causing no light rays to reflect back to the observer and resulting in the apparent disappearance of the glass rod. In contrast, when the same glass rod is immersed in a beaker of water, the light passing through the water is refracted differently, causing the light rays to reflect off the surface of the glass rod and back to the observer's eye, allowing the observer to see the glass rod. This experiment highlights how the behavior of light is affected by the refractive indices of the media it passes through.

4.1.2 Hardware Setup

Fig. 4.1 shows the hardware description of this setup, consisting of a block diagram and a physical setup. The experiment's frame is made up of wooden sheets and two 500 ml Boro-silicate beakers are placed to hold the desired liquids. The liquids used here are Sunflower oil (refractive index 1.47)



(a) (b)

Figure 4.1 Hardware description of the Vanishing Rod setup.

as its refractive index is very similar to that of glass rods (refractive index 1.5) and the other liquid is tap water (refractive index = 1.33). Glass rods are dipped in and out of these beakers to visualize the concept. These rods are controlled using two 28BYJ-48 stepper motors, one for each rod. Stepper motors are used due to their precise movement in the long run whereas servo motors can have errors that could add up. These motors are controlled using a Raspberry Pi 3B+ via a ULN2003 stepper motor driver. Fig. 4.2 displays the circuit diagram of the experimental setup. The RaspiCam is positioned to capture both beakers in the video feed, which is transmitted to a Raspberry Pi. The experiment is connected to a dashboard that uses Blynk [45] to communicate with the Raspberry Pi and YouTube to live-stream the experiment.

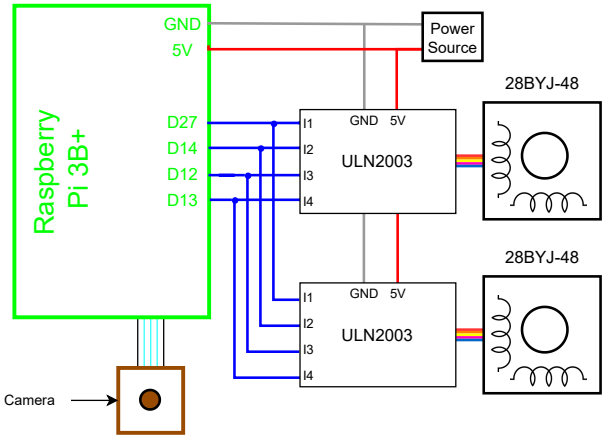


Figure 4.2 Circuit Diagram.

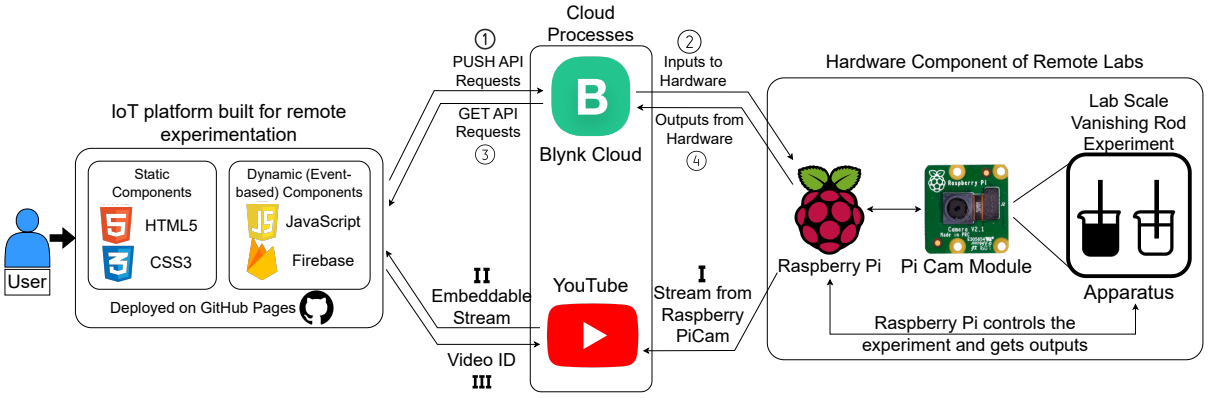


Figure 4.3 Overall flow of the platform designed for the Remote Labs for the Lab-Scale Vanishing Rod Experiment.

4.1.3 Software Framework

Remote Labs is an amalgamation of multiple elements and components which are required to work in perfect sync for a user to conduct remote experimentation. These components must enable the user to provide input, receive outputs, and view the experiment setups in real time. These elements are crucial to the framework, along with the communication between them, which makes it possible for the user to perform an experiment remotely.

As illustrated in Fig. 4.3, Remote Labs at IIIT-H is based on three primary components - an IoT-based platform, cloud services - Blynk and YouTube for communication and streaming respectively, and the hardware nodes. These primary components, along with the unique flow created by their integrated working, are explained in detail below.

4.1.3.1 IoT-based platform

As shown in Fig. 4.3, to perform the experiments, a user must access the IoT-based platform that has been developed. This platform is essentially a webpage as shown in Fig. 4.4, that allows users to access the hardware experiments without requiring any special software to be downloaded; hence experiments can be conducted on any smart device that supports a modern browser. As can be clearly seen, the webpage is comprised of two central parts - (i) controls and (ii) display with the real-time video stream.

The controls form the input channel by permitting the users to actuate the motors and move the glass rods. They act as a medium for users to instantiate communication with the Blynk Cloud. Vanishing Rod Experiment provides two labeled buttons to make the working intuitive for the user:

- **Down** - Dip the glass rods into the beakers

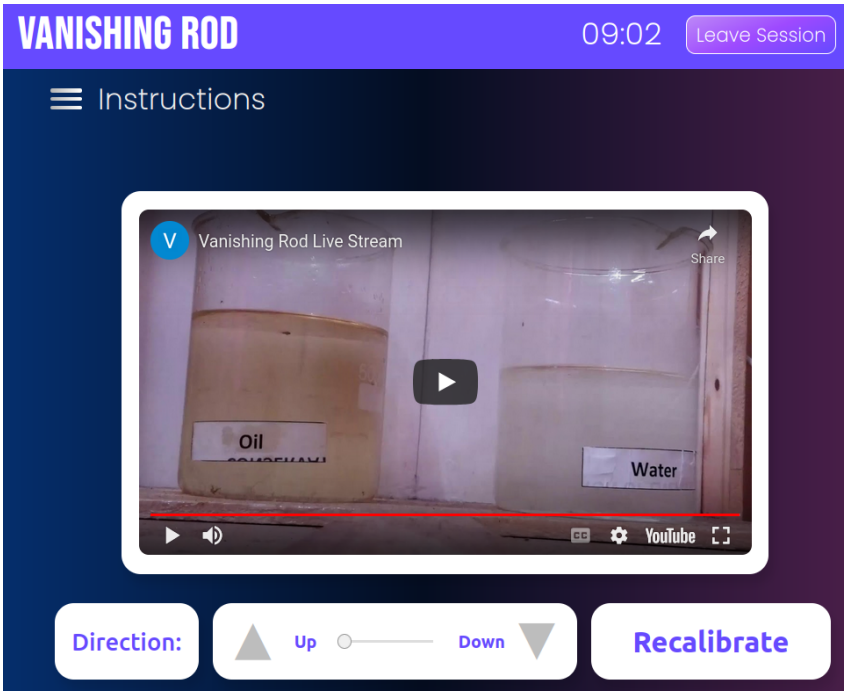


Figure 4.4 User Interface for Vanishing Rod Experiment.

- **Up** - Bring the dipped glass rods back to their original position.

The display is symbolic of the communication between the IoT platform and YouTube. It essentially projects the video stream coming from Raspberry Pi via YouTube on the webpage itself. This ability to see the hardware nodes in real time is elementary to remote experimentation on mechanical setups involving actuation. For a better understanding of the framework, the further process is divided into two parts based on cloud services to first learn the individual flows and then how they combine to produce the learning experience.

4.1.3.2 Blynk IoT Cloud

Remote Labs requires the parameters to be conveyed to the Blynk cloud whenever an input field is modified by the user. When any one of the control buttons is clicked, we proceed to the first step (Step-1) of the Blynk pipeline, in which we communicate the user's desired state of the glass rods to the Blynk IoT Platform using HTTPS API calls [45]; this action updates the parameter on the cloud itself.

The second step (Step-2) involves the transmission of these input values to the Raspberry Pi associated with the hardware experiment using Blynk's proprietary protocol. These inputs are crucial for any actuation and processing that happen on the hardware nodes.

The third step (Step-3) is to update the output values calculated by the hardware nodes on the Blynk cloud directly from the Raspberry Pi devices. The fourth and final step in the pipeline (Step-4) allows these outputs to be available to the platform using Blynk GET API calls. These results are then displayed on the webpage using tables, charts, and multiple other interactive components.

4.1.3.3 YouTube and Live Streaming

The real-time live stream is essential to Remote Labs, especially for experiments involving actuation and visual changes. Unlike the communication with Blynk, this streaming must happen irrespective of the user inputs. YouTube is used to facilitate real-time video streaming. Each Vanishing Rod experiment has a dedicated channel on the YouTube platform with its unique secret key. The real-time video is transmitted from the RaspPi Cam to the experiment's dedicated channel using the RTMP protocol and the secret key which is marked as (I) in Fig. 3.

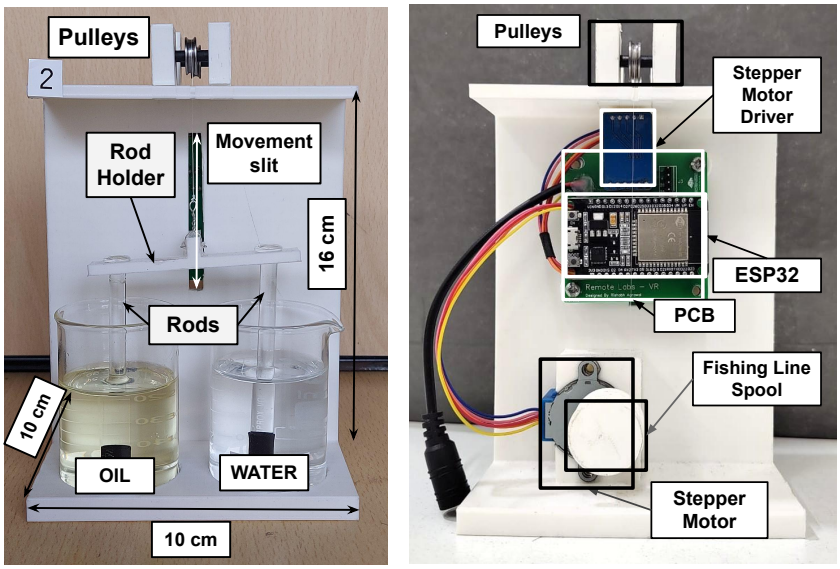
The procedure involved with displaying the live stream on the platform for the users is reflected in (II) and (III). As the video channel for the live stream is pre-decided and known to the platform, YouTube APIs are used to get the video ID corresponding to this live stream. Power shortages or any other adversary which might force the Raspberry Pi to restart would lead the video ID to get updated, and this deems it necessary to use YouTube APIs and get the dynamic video ID for a seamless learning experience.

After the video ID is received from the APIs, an embeddable YouTube video URL is generated by string concatenation. This dynamically generated video link of the live stream has been embedded in the experiment's dedicated webpage (display section of the IoT platform) for the video streaming to be visible during the remote experimentation.

4.2 Miniaturised Experimental Setup

In this section, the details of the proposed miniaturized experimental setup are presented along with their physical comparison with lab-scale setup.

Figs. 4.5 and 4.6 show the experimental setup and the circuit diagram for the miniature experimental setup. The setup primarily consists of an ESP32, single 28BYJ-48 stepper motor, ULN2003 motor driver, two pulleys, two borosilicate glass rods, and two 50ml beakers filled with sunflower oil and water. The exoskeleton of the experiment is completely 3D printed in multiple smaller parts that can be assembled like puzzle pieces within a few minutes. The DIY assembly-based modular nature of the 3D printed setup makes it easy to set up the equipment, thus adding to the scalability of the experiment through ease of installation. Slots are provided on the exoskeleton to fit the beakers, glass rods and



(a) (b)

Figure 4.5 Hardware description of the Miniaturised Vanishing Rod setup.

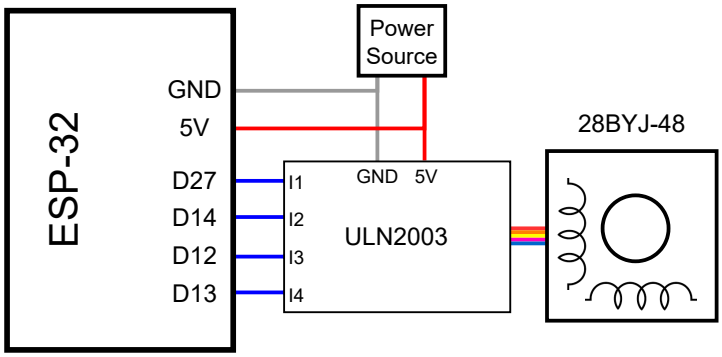


Figure 4.6 Circuit Diagram of the experimental setup.

electronic components that are part of a single PCB as shown in Fig. 4.5(b). The glass rods are tied to a small rectangular platform (rod holder) spooled over a pulley connected to a single motor that rotates clockwise and anticlockwise. This platform is designed to move in a linear path — up and down. It is constrained into a slit which blocks it from rotating and toppling keeping rods stable when moving whereas, in the lab-scale experimental setup, the rods can freely rotate and swing when rods are moving. The slit which constrains this platform is indicated by the arrow in Fig. 4.5(a).

Fig. 4.6 shows the actual experimental setup. The controls of the experiment are only to move the glass rods into the beakers and bring them out. This is carried out by the stepper motor placed at the top. The ESP32 is the micro-controller used, chosen due to its low cost, small form factor, and networking capabilities to integrate the experiment with a dashboard. A single motor ensures the movement of both

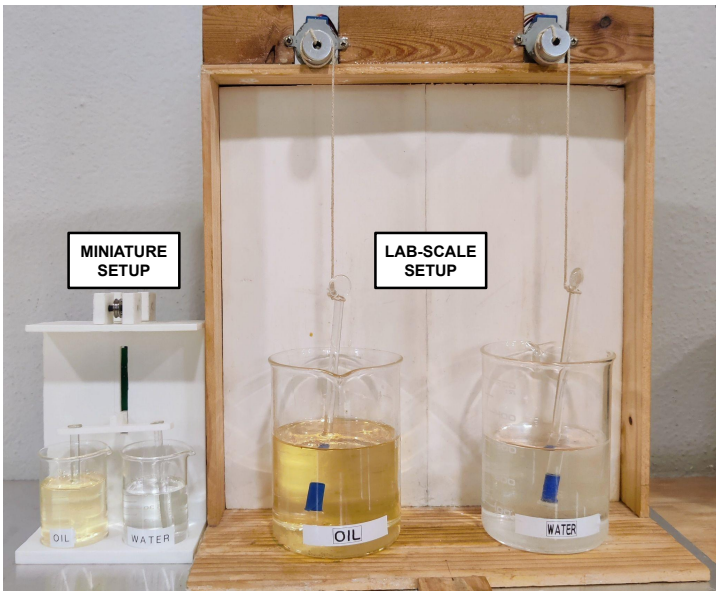


Figure 4.7 Both lab-scale and miniaturized setups together. Miniaturized setups consume much less volume as compared to lab-scale setups.

rods is the same.

Table 4.1 shows the dimensions and weights of both lab-scale and miniature experimental setups. It can be observed that the miniaturized setup is almost 10 times lower in volume and 5 times lighter in weight as seen in Fig. 4.7. This is really helpful for scaling up these experiments. They can be easily stacked up and multiple copies can be replicated easily, given the simple build of the setup.

Setup	Length (in cm)	Width (in cm)	Height (in cm)	Weight (in Kg)
Lab-Scale	30	20	32	1.8
Miniature	10	10	16	0.33

Table 4.1 Comparison of the physical dimensions of the two experimental setups.

4.3 Methodology for Partial Streams

In this section, we present the methodology used for obtaining live feeds of various experiments from a single camera. Fig. 4.9 shows the pipeline used for achieving the partial streams. Two major phases involved are:

- Extraction: Crop the camera feed to the desired number of regions of interest

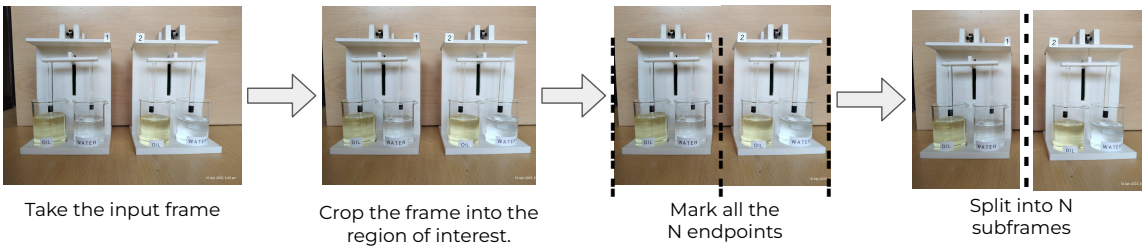


Figure 4.8 Workflow for cropping and splitting an image.

- Publish: Push the cropped frames to respective live channels for streaming the experiments

We will discuss each of the phases below.

4.3.1 Extraction

This phase primarily involves splitting the feed from the single camera into multiple feeds and cropping them appropriately according to the number of experiments. Fig. 4.8 shows the step-by-step breakdown for the base function of the split screen technique being used and Algorithm 1 shows the steps to crop and split the frames. To develop this technique, it was assumed that the camera is stationary and all setups are fixed with known positions and all setups are equidistant from the center in the point-of-view of the RasPi Cam.

Initially, the input frame is taken and the main Region of Interest from (x_1, y_1) to (x_2, y_2) is marked and cropped. N equally-spaced endpoints are marked on the horizontal axis and, accordingly, N Region-of-interests (RoIs) are cropped as (e_1^1, e_2^1), (e_1^2, e_2^2), ..., (e_1^N, e_2^N), where $e_2^N - e_1^1 = x_2 - x_1$. After this, a total of N cropped images, called sub-frames, are available to process.

4.3.2 Publish

In this phase, the individual feeds are assigned to virtual video devices and streamed to respective streaming channels. To enable live streaming, a set of N virtual streaming devices are created as a prerequisite. Additionally, N live streams are created from N distinct accounts, with each account associated with one of the live streams.

After obtaining N sub-frames from an image, each sub-frame is transmitted to its corresponding virtual streaming device, enabling the sub-frame to be streamed to the respective live stream. Subsequently, the link for each individual live stream is embedded and transmitted to the corresponding user dashboard. This method provides an efficient approach for image streaming that can effectively handle the data by partitioning the image into smaller sub-frames and distributing them across multiple virtual streaming devices.

Algorithm 1 Crop and split frame

procedure CROPFRAME(F, x_1, y_1, x_2, y_2, N)
Inputs:
F : single input image, or single input video frame
(x_1, y_1): start point of Region of Interest
(x_2, y_2): end point of Region of Interest
N: Number of fragments needed from the split.
Output:
S: Collection of all output frame fragments after the split.
Method:
C = F[$x_1 : x_2$][$y_1 : y_2$]
L = $x_2 - x_1$
l = $\lceil \frac{L}{N} \rceil$
Start = 0
Endpoints = []
    while Start $<$ L **do**
        Start = Start + l
        Append Start to array Endpoints
    end while
k = 0
start = 0
S = []
    while k $<$ N **do**
        Stop = Endpoints[k]
        NewImage = C[Start:Stop][:]
        Start = Stop
        Append NewImage to array S
        k = k+1
    end while
end procedure

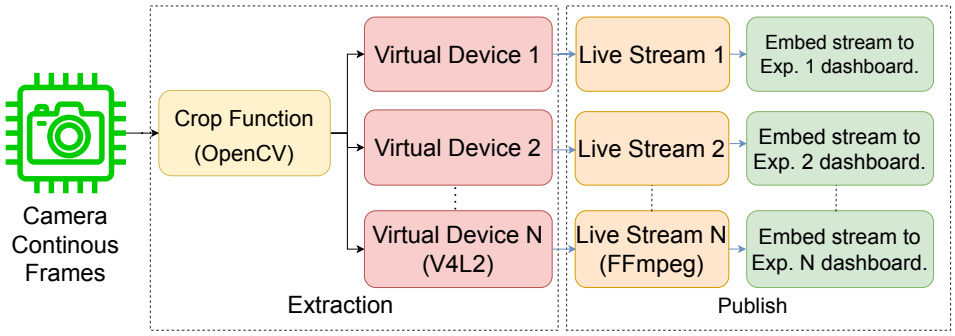


Figure 4.9 Workflow for end-to-end split screen technique.

4.3.3 Implementation Details

The implementation of the stream-splitting system for partial streaming involves several open-source libraries and tools. The *OpenCV Python* library was utilized to access image slicing and video handling capabilities. Additionally, the *V4L2* [46] interface was utilized to create virtual video devices, which is a powerful and reliable interface for creating loopback devices. The *FFmpeg* [47] package was mandatory for playback and handling of the virtual video feeds and finally pushing these virtual feeds to YouTube for live streaming which is available to the users on the dashboard.

The above-mentioned libraries are open-source and are available for Linux systems. We have configured them on a Raspberry Pi 3B+ with 1GB of RAM for the purpose of streaming multiple experiments. Two experiments ($N = 2$) are streamed from a single Raspi Cam with FPS=25 and 480p resolution. Fig. 4.10 shows the algorithmic split for the streams obtained from the Raspberry Pi before sending it to 2 different channels where it is viewed by 2 different users while experimenting on 2 different setups. Fig. 4.11 shows the demonstration of the two experiments on a real user dashboard. The lower resolution was sufficient as the setups are relatively smaller in size.

To test the hard limit, we have increased the value of N. It was observed that the Raspberry Pi supported $N = 3$ comfortably without any loss in quality or frames. From $N = 4$ onwards, although the experiment videos were getting streamed, there were losses in the frames transmitted, which is undesirable. However, it can be noted that the above-mentioned pipeline for streaming multiple experiments can be used on any Linux system. The same pipeline was tested on a Desktop with an Intel i5-7400 CPU 3GHz quad-core processor and 8GB RAM. This system has supported N=8 streams without any compromise on the quality of the video streams.

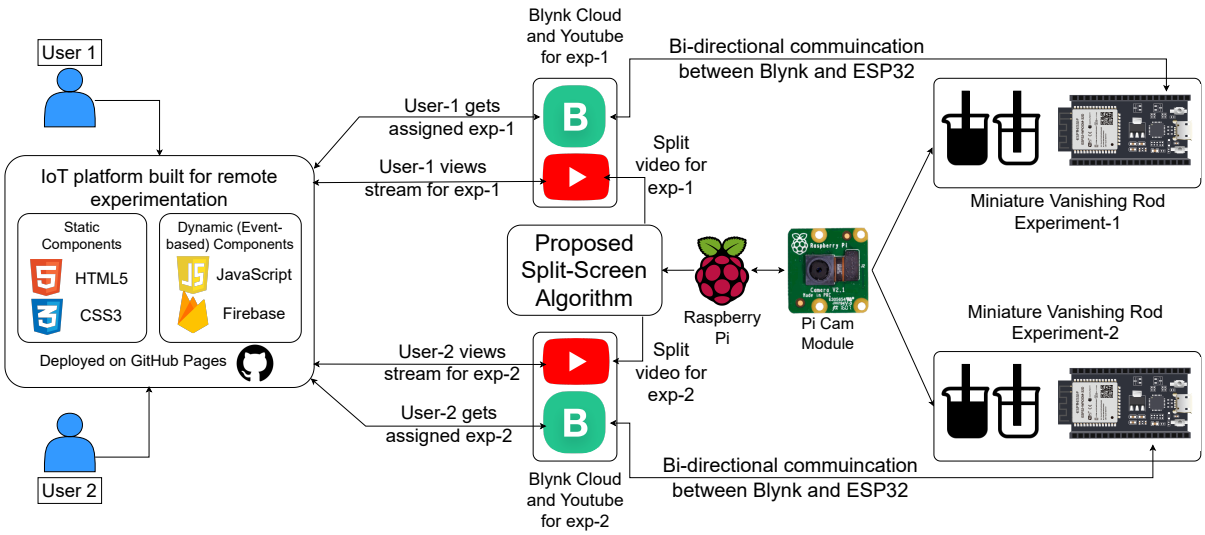


Figure 4.10 Proposed flow of the platform for the Remote Labs for Miniature Vanishing Rod Experiments with N=2.

4.4 Results

4.4.1 Current Consumption

To compare the current consumption of both setups without any cameras attached, we calculated the average current drawn in 2 states - “idle” and “control” - over 5 cycles of alternation between the two. We report the current consumption values in Table 4.2 and find that for the lab-scale setup, which is built on a Raspberry Pi 3B+, the idle-state current consumption averages 424.5 mA, while that of the miniature setup stays around 70.2 mA. The control-state peak for the lab-scale setup is higher than that of the miniature setup. An additional observation is that the Raspberry Pi setup takes a lot of power while booting up - its current consumption is as high as 1.5-1.8 A in that period.

Fig. 4.12 shows the current consumption plots for both setups for 5 cycles. The control state window for the lab-scale setup is 13.8 seconds while that of the Miniature setup is from 4.9 to 5.4 seconds, averaging at 5.1 seconds. This difference in control window length is because of the time it takes for the rod to go into the dip position from the lift-up position, which is anyways higher in the case of a full-scale setup. The larger relative bump in current during the control state is because of the additional power required to handle the weight of the rods in the full-scale setup as compared to the miniature setup, where the rods are lighter. On average, assuming 100 usage instances per day, the full-scale setup consumes 52.093 Wh (watt-hour) while the miniature setup uses 8.77 Wh, which is one-sixth that of the full-scale setup.

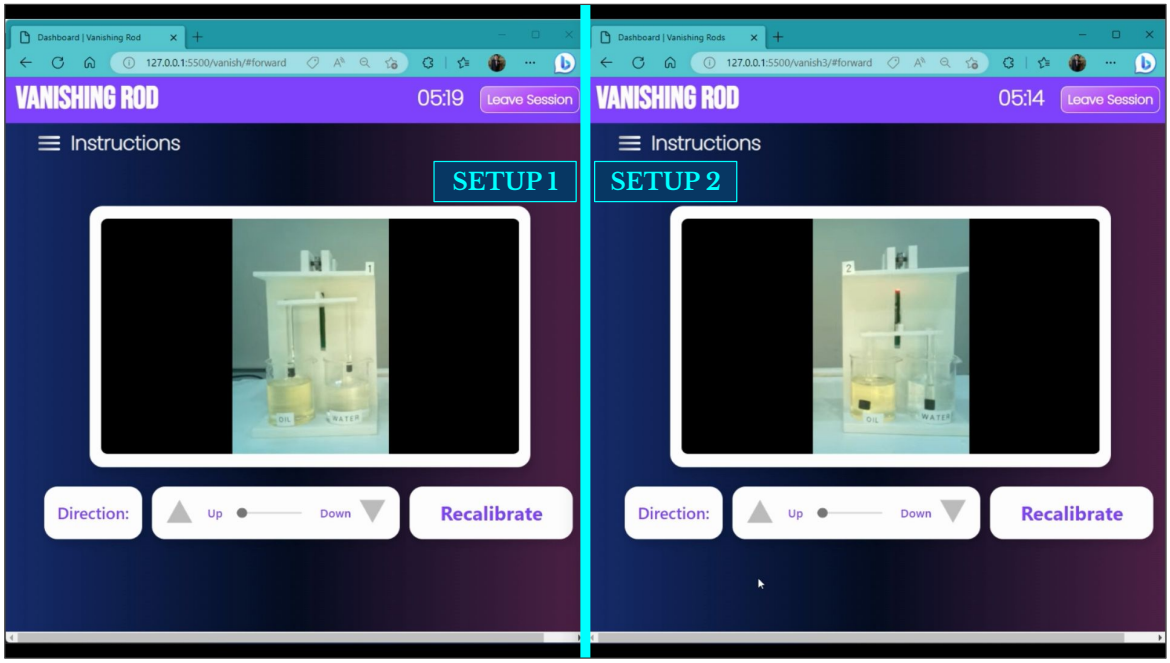


Figure 4.11 Demonstration of two miniaturized experimental setups in use at the same time.

Table 4.2 Current Consumption (mA).

Setup	Idle state	Control state
Full-scale	424.5	725.2
Miniature	70.2	315.4

4.4.2 Component cost

Table 4.3 shows the total cost comparison for building the lab-scale and miniature setups. These costs are exclusive of the cameras required for streaming the experiment. It is observed, when deploying the miniature setups and using the partial streaming technique, that the component cost without considering streaming functions, as observed from Table 4.4, is reduced to at least one-fifth when shifting from the lab-scale setup to the miniature setup. Note that the biggest drop is due to shifting from Raspberry Pi 3B+ to ESP32 for actuation of the motors.

In reality, because one Raspberry Pi can handle up to 3 partial live-streams of miniature setups, it is imperative that this be taken into account when calculating the total cost for multiple units of both setups. It is observed that as the number of setups to be installed increases, the cost growth is steeper in case of the lab-scale setup than in the case of the miniature setup, as is seen in the cost growth plot in Fig. 4.13.

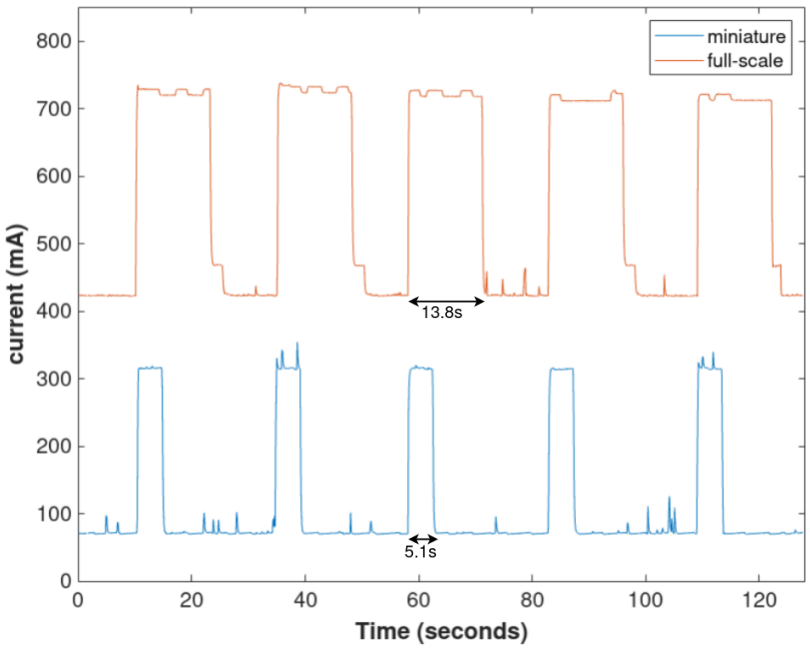


Figure 4.12 Current consumption pattern for both the Lab-Scale Experimental setup (orange, labeled "full-scale") and Miniaturised Experimental setup (blue, labeled "miniature").

Table 4.3 Comparison of the cost of two units of each experimental setup with the cameras(as of April 2023).

Component	Lab Scale Setup	Miniature Setup
Raspberry Pi 3B+	20,000	10,000
Pi Camera	5,000	2,500
ESP32	N/A	800
Stepper Motor	320	160
Motor Driver	200	100
Frame	1,000	2,000
Beakers	800	200
Glass Rods	100	60
Misc	1,000	1,000
Total (in INR)	**28,420**	**16,820**

4.5 Practical Effect of Partial Streams

It has already been established that the proposed methodology is useful in terms of current and energy consumption and monetary expenditure. While the proposed method proves to be a cost-saver in these aspects, there are a few other practical limitations in the implementation of this method. This section discusses the major limitations in streaming latency, computing resource consumption, frame rate and

Table 4.4 Comparison of the cost of two units of each experimental setup without streaming function (as of April 2023).

Component	Lab Scale Setup	Miniature Setup
Raspberry Pi 3B+	20,000	N/A
ESP32	N/A	800
Stepper Motor	320	160
Motor Driver	200	100
Frame	1,000	2,000
Beakers	800	200
Glass Rods	100	60
Misc	1,000	1,000
Total (in INR)	**23,420**	**4,320**

image resolution. These limitations need to be kept in mind when implementing the method as shown in Fig. 4.10 in its current state.

4.5.1 Latency

It was observed that while partial streaming reduced components, it took the latency of the stream from an already high 7 seconds to an even higher 15 seconds. The reason for the latency being 7 seconds without partial streaming is because of Youtube adding 5 seconds for queueing the frames. An additional 0.7 second comes from using V4L2 in Raspberry Pi, and upto 0.4 seconds per frame from the partial streaming application, and the ffmpeg command for live-streaming in Raspberry Pi consuming a large chunk of memory. This latency increases as the frame resolution is increased from 360p to 480p to 720p.

Table 4.5 Latency comparison and breakdown (in seconds).

Component	Latency with split	Latency w/o split
Frame loop	0.2	0
Temp copy by ffmpeg	5.01	1.93
Youtube	10	5
Total	**15.2**	**6.93**

4.5.2 Computing Resource consumption

Resource consumption, in the context of our system, refers to the amount of memory a process takes up, and the CPU usage for the process. It is observed that in the workflow, the ffmpeg command occupies the most memory out of all the components. It is also observed that the memory consumed

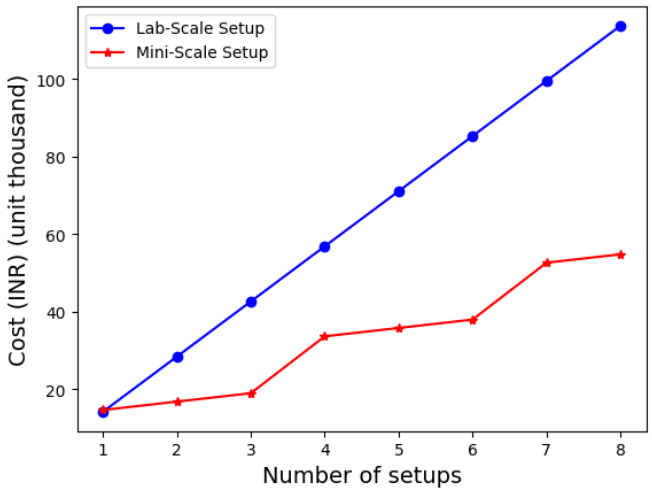


Figure 4.13 Cost growth plot for both lab-scale and miniaturised setups. Lab-scale setup costs are denoted by the blue plot, while miniature setup costs are denoted by the red plot.

increases when the partial streaming is applied.

If the task manager is to be logged, ffmpeg as a process used a reserved set memory of 125 MB, and shows a CPU usage of 37% to 41% when partial streaming is applied. While the memory usage remains the same, the CPU usage is logged at 58% to 61% when partial streaming is not applied. This is due to an additional 66 MB usage by the Python3 process that runs the code for partial streaming, which logs a CPU usage of 19%.

The reason for such high consumption can be seen in the process thread log generated by the command "top". Let us bear in mind that the Raspberry Pi, as close to a computer board as it can possibly be, is still a constrained device. It is observed, as is also shown in Fig. 4.14 that the process opens 10 threads at a time, with CPU usage for threads going as high as 75.2%.

4.5.3 Frame Rate (FPS)

The frame rate of the stream is the number of frames displayed per second and is measured in FPS. It is observed that the intended 30 FPS is obtained when the partial streaming is not applied and the feed is taken directly from the camera instead of a V4L2 device. When partial streaming is applied, not only does the frame rate go below half the max rate, going as low as between 5-9 FPS, but some frames also get dropped.

Some of this can be attributed to live streaming being a heavy process in Raspberry Pi, a computer intended for lightweight applications, V4L2 still having a slight hand in that because of the slow copy,

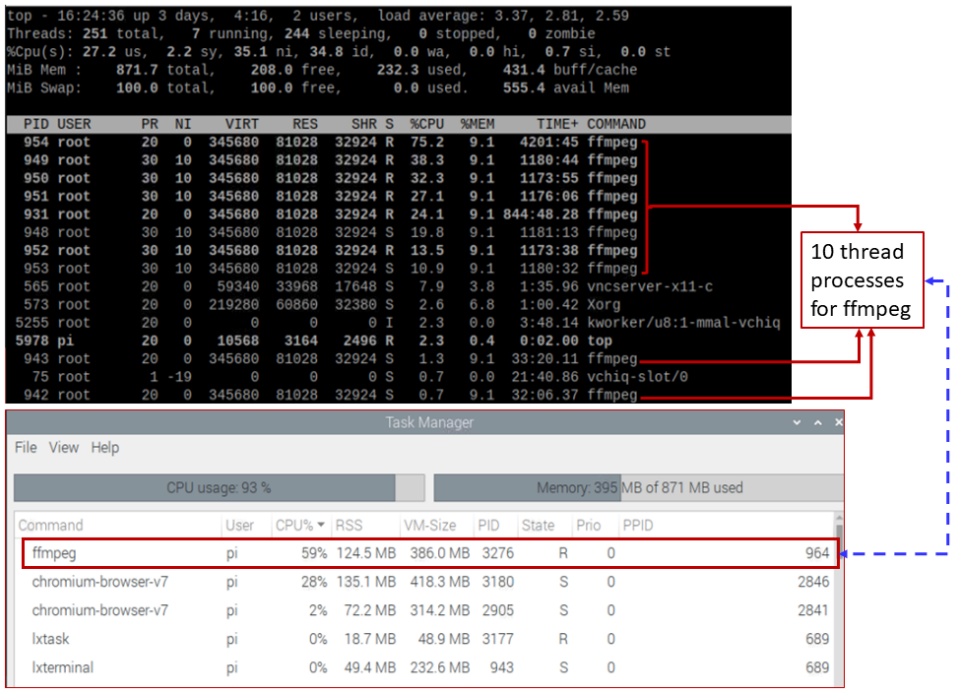


Figure 4.14 Screen captures showing output of "top" command in threads mode, and the task manager log. Observe that ffmpeg alone has 10 entries in the "top" log. Note also that the major consumption of CPU resources comes from 8 of the 10 threads. This leads to major CPU resource consumption by the ffmpeg command.

while it is also possible that due to the naive status of the partial streaming application and the heavier compiler, the operation itself uses more memory than is intended for a Python code that uses the OpenCV library.

4.5.4 Image Resolution

It was observed that in trying to create a full HD (1080p) livestream and stretching the Raspberry Pi Camera video mode to its limit, the Raspberry Pi system freezes and almost stops working, hence the maximum image resolution affordable for streaming from a Raspberry Pi is 720p, even though the Pi Camera Rev 1.3 can take pictures upto 1080p [48]. While the memory consumption of the ffmpeg command alone is 58%, it can increase to beyond 90% with multiple 720p streams. With multiple processes running in a Raspberry Pi, the overall memory required to be consumed could go beyond what's available and cause the system to freeze. This can be kept in mind when looking for a suitable microcomputer board to run a live-stream.

Chapter 5

Conclusion and Future Work

The concept of Remote Labs and its future application is expected to have a profound impact on education and research by democratizing access to practical learning experiences. They break down geographical barriers, allowing students and researchers from around the world to engage in experiments and gain hands-on skills. This accessibility not only enhances educational opportunities but also fosters collaboration and knowledge-sharing on a global scale.

This thesis began with a literature survey on IoT and Remote Labs, where the component-wise development of remote laboratories was discussed, followed by the various ways that RLMS had evolved. Some of the work mentioned in the literature survey served as inspiration for what is proposed in the thesis.

A cost-effective approach is put forward for scaling up Remote Labs through the miniaturization of setups, the use of image processing techniques, and partial streaming. This technique is evaluated through the Vanishing Rod experiment, where an end-to-end remote lab has been developed to demonstrate its efficacy in comparison to the traditional technique of using lab-scale setups and a single camera per experiment setup. The miniaturization of experimental setups reduced installation costs, and this was verified from the cost table. The 3D printing of the I-beam, the rod holder, and the pulley holder helped to reduce manufacturing expenses. Further cost and component reduction involved a single camera for multiple experiments, and this is where the concept of partial streaming was applied. The results demonstrate that this process not only reduces power consumption by around one-sixth but also reduces actuation costs by approximately one-fifth and total costs by more than half.

Future work can focus on optimizing the proposed approach by exploring the use of cloud-based image processing techniques to make the pipeline more efficient, potentially reducing latency and further enhancing the overall user experience. Instead of streaming to Youtube, which the portal used to demonstrate the proposed approach does, a better, sub-second latency alternative like WebRTC can be used. Image quality can be enhanced using various other image-processing algorithms, or a successive

combination of the existing algorithms. Since the Raspberry Pi is a constrained device running a heavy application like V4L2, a lighter alternative like UV4L can be used to generate virtual video-capture streams. While the use-case of Vanishing Rod experiment was taken for development and trial, the approach can be extended to many other experiments like the "focal length" experiment, where the lateral view of the bench for multiple experiment setups can be provided using a single camera while keeping the lens point-of-view stream specific to the setup.

Bibliography

[1] "UNILabs," https://unilabs.dia.uned.es/, accessed: 28- Mar- 2022.

[2] "RT Labs - NITK," http://rtlabs.nitk.ac.in/?q=page/rt-lab, accessed: 28- Mar- 2022.

[3] L. Rodríguez-Gil et al., "An Open and Scalable Web-Based Interactive Live-Streaming architecture: The WILSP Platform," *IEEE Access*, vol. 5, pp. 9842–9856, 2017.

[4] I. Angulo et al., "Scaling up the Lab: An Adaptable and Scalable Architecture for Embedded Systems Remote Labs," *IEEE Access*, vol. 6, pp. 16 887–16 900, 2018.

[5] A. Villar-Martínez et al., "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.

[6] M. Rábek and K. Žáková, "Scalable Remote Experiment Manager," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17 246–17 251, 2020, 21st IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320324022

[7] K. Ratakonda, "Efficient partial streaming of compressed video," in *Proceedings. International Conference on Image Processing*, vol. 3, 2002, pp. III–III.

[8] "Remote Labs - IIIT Hyderabad," https://remote-labs-iiith.github.io/, accessed: 10 Apr, 2023.

[9] S. Misra et al., *Introduction to IoT*. Cambridge University Press, 2021.

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

[11] 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

[12] "Overview of the Internet of things," https://handle.itu.int/11.1002/1000/11559, accessed 30-Sep-2023.

[13] C. R. Reddy et al., "Improving Spatio-Temporal Understanding of Particulate Matter using Low-Cost IoT Sensors," in *IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–7.

[14] C. R. Reddy and S. Chaudhari, "Hierarchical Clustering based Spatial Sampling of Particulate Matter Nodes in IoT Network," in *8th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2021, pp. 198–203.

[15] C. R. Reddy et al., "Maximum Frequency based Adaptive Sensing for Particulate Matter Nodes in IoT Network," in *IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 482–487.

[16] A. P. et al., "Development of End-to-End Low-Cost IoT System for Densely Deployed PM Monitoring Network: An Indian Case Study," 2022.

[17] S. Sara et al., "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.

[18] N. Nilesh et al., "IoT-based AQI Estimation using Image Processing and Learning Methods," in *IEEE 8th World Forum on Internet of Things (WF-IoT)*, 2022, pp. 1–5.

[19] N. N. et al., "IoT and ML-based AQI Estimation using Real-time Traffic Data," in *IEEE 8th World Forum on Internet of Things (WF-IoT)*, 2022, pp. 1–6.

[20] A. Kumar Lall et al., "Making Analog Water Meter Smart using ML and IoT-based Low-Cost Retrofitting," in *8th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2021, pp. 157–162.

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

[22] S. Mante et al., "Energy Monitoring Using LoRaWAN-based Smart Meters and oneM2M Platform," in *IEEE Sensors*, 2021, pp. 1–4.

[23] A. P. Singh et al., "Machine Learning-Based Occupancy Estimation Using Multivariate Sensor Nodes," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.

[24] S. U. N. Goparaju et al., "Design of an IoT System for Machine Learning Calibrated TDS Measurement in Smart Campus," in *IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 877–882.

[25] "Smart Lamppost," https://smartcityresearch.iiit.ac.in/research/focus_area/lamppost/, accessed 31-Aug-2023.

[26] "Smart City Research Center," https://smartcityresearch.iiit.ac.in/, accessed 06-Oct-2023.

[27] I. Patwardhan et al., "Comparative Evaluation of New Low-Cost Particulate Matter Sensors," in *8th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2021, pp. 192–197.

[28] A. P. Singh and S. Chaudhari, "Embedded Machine Learning-Based Data Reduction in Application-Specific Constrained IoT Networks," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, ser. SAC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 747–753. [Online]. Available: https://doi.org/10.1145/3341105.3373967

[29] A. Shastri et al., "Improving Accuracy of the Shewhart-based Data-Reduction in IoT Nodes using Piggybacking," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 943–948.

[30] G. Ihita et al., "Security for oneM2M-Based Smart City Network: An OM2M Implementation," in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2023, pp. 808–813.

[31] ——, "Security Analysis of Large Scale IoT Network for Pollution Monitoring in Urban India," in *IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 283–288.

[32] X. Chen et al., "Virtual and Remote Laboratory Development: A Review," *Earth and Space 2010: Engineering, Science, Construction, and Operations in Challenging Environments*, pp. 3843–3852, 2010. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/41096%28366%29368

[33] J. Ma and J. V. Nickerson, "Hands-on, Simulated, and Remote Laboratories: A Comparative Literature Review," *ACM Comput. Surv.*, vol. 38, no. 3, p. 7–es, sep 2006. [Online]. Available: https://doi.org/10.1145/1132960.1132961

[34] "What are Remote Laboratories?" https://remotelaboratory.com/remote-laboratories/what-are-remote-laboratories/, accessed 21-Aug-2023.

[35] "LabsLand," https://labsland.org/en, accessed 06-Oct-2023.

[36] P. Orduña et al., "LabsLand: A sharing economy platform to promote educational remote laboratories maintainability, sustainability and adoption," in *IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–6.

[37] P. O. et al., "Addressing technical and organizational pitfalls of using remote laboratories in a commercial environment," in *IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–7.

[38] L. Rodríguez-Gil et al., "New Approach for Conversational Agent Definition by Non-Programmers: A Visual Domain-Specific Language," *IEEE Access*, vol. 7, pp. 5262–5276, 2019.

[39] F. Garcia-Loro et al., "PILAR: a Federation of VISIR Remote Laboratory Systems for Educational Open Activities," in *IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2018, pp. 134–141.

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

[41] J. Saenz et al., "A virtual and remote lab of the two electric coupled drives system in the University Network of Interactive Laboratories," in *American Control Conference (ACC)*, 2015, pp. 5623–5628.

[42] K. S. Viswanadh et al., "CV and IoT-based Remote Triggered Labs: Use Case of Conservation of Mechanical Energy," in *9th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2022, pp. 100–106.

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

[44] "Disappearing Glass Rods - NYU Physics Demos," https://physics.nyu.edu/~physlab/Demos/updatedEquipment/light/immersionoil.html.

[45] "Blynk APIs," https://docs.blynk.io/en/blynk.cloud/https-api-overview, accessed: 31- Mar- 2022.

[46] "V4L2 - Video for Linux API," https://www.kernel.org/doc/html/v4.9/media/uapi/v4l/v4l2.html, accessed 10-Jan-2023.

[47] "FFmpeg - A complete, cross-platform solution to record, convert and stream audio and video." https://ffmpeg.org/, accessed 16-Jan-2023.

[48] "Raspberry Pi Camera Documentation," https://www.raspberrypi.com/documentation/accessories/camera.html#hardware-specification, accessed: 30- Mar- 2022.