

Full Paper

Introduction

Remote laboratories have been developed in engineering education as a method to offer engineering students a new way to be presented with practical learning materials, differing from other methods of presenting students with practical sessions or engineering laboratories (Feisel and Rosa 2005). They are often utilized by engineering disciplines to act as either a replacement, or preferably in compliment of (Cooper and Ferreira 2009) the practical sessions required in the engineering studies. They allow students to access laboratory data via an internet connection from almost any location, including those not on the university campus. This allows for a much greater flexibility in the way students can approach education (Sousa, Alves et al. 2010), and it has been shown that this greater level of flexibility in completing practical laboratories is something students prefer (Lindsay, Liu et al. 2007).

By allowing student to remotely access their laboratory sessions, the way that students approach education, and therefore the way that educational material can be presented to the students can be drastically different to how more traditional laboratories are set up (Corter, Nickerson et al. 2007). Removing the time requirement from university opening hours to when the laboratory equipment is physically available (Lowe and Orou 2012), allowing shared laboratory equipment between universities (Lowe, Machet et al. 2012) or institutions (Lowe, Conlon et al. 2011) and linking students from around the world to the same laboratory equipment (Yazidi, Henao et al. 2011).

Remote laboratories are often separated into one of two categories, those being batch or interactive laboratories (Lowe, Murray et al. 2009). Batch laboratories are controlled by a user inputting set parameters into the experiment and then leaving the experiment apparatus to automatically complete and then return the results of the experiment, logging back into the system once these parameters have been tested to view the result. Opposite this is the interactive setup, were the student views the experiment as it is happening, being able to control the parameters of the experiment in real time to change the input values of the system. They type of data being generated largely determines the method of control given to the user, understandably not all remote laboratories should be set up to be batch or interactive, as this depends on the specific laboratory.

This paper will cover a novel approach to the design of a remote laboratory allowing students to view live data being generated by an active renewable energy system, as well as gain access to the logged data generated. The remote laboratory works by acting as a bridge between the Deakin University logging equipment and the inverter of the renewable energy systems. The data generated is presented to students and accessed through the Deakin University VPN. An important thing to note is that the data presented to the students was all generated from an active renewable energy system. It is important to consider the effects of real data compared to simulated data, as using real data has been seen to increase student motivation (Lowe, Murray et al. 2008), and remote education compared to simulations have been shown to have an effect on the way students perceive the laboratory (Sauter, Uttal et al. 2013).

With a system such as this in place, it is possible to show students the real world data for the expected inputs of a renewable energy system. The laboratory therefore allows students to covert the joules generated to watts, and to design a renewable energy system that would be capable of powering a specific system from the power generated by the outputs of the renewable energy system.

Aims and Challenges of the Remote Laboratory

In terms of remote laboratories, there are general requirements that need to be solved before the system can be brought online. It is possible to download remote laboratory frameworks such as the SAHARA LABS framework (Lowe, Machet et al. 2012), or to create the server architecture in house. There are benefits to both systems, especially systems such as SAHARA, as it has inbuilt error logging, student queue system, client-server architecture separated into a web interface, rig client ECT, and other remote laboratory features that would not need to be developed by the designer of the laboratory. However, creating all laboratory architecture allowed for a much greater degree of control over all aspects of the system, and for a simpler lab layout, it does not require all of the features that SAHARA LABS supplies. When creating any remote laboratory, specific challenges must first be addressed. A brief overview of some of these challenges is below:

For the remote laboratory to be accessible to students, the required hardware needed to be accessible to students from many different locations, and using multiple different hardware solutions. This generates a unique problem for the design, as a system homogenous solution needed to be created. In addition to this, accessing the remote laboratory becomes an issue when you consider that the laboratory is able to be run at all times of the day, even if solar power is not being generated during the low light hours. In addition to this, the remote laboratory needed to be set up such that minimal installation files were required for the end user. Ideally they would be able to log onto the internet from any computer and easily be able to access the remote laboratory through a web page. In addition to this, accessing the remote laboratory through multiple different web browsers and even mobile platforms becomes a benefit.

Once these issues have been addressed, issues with creating the specific remote laboratory to gather the data from the renewable energy system needed to be addressed. An important criteria of this remote laboratory is that it was able to interact with a real world working system. The solar panels and inverter were logging data for other projects, and therefore any hardware needed to be able to interact with these systems without interrupting any of the data transmissions. It therefore needed to be able to interact with any data being transmitted, not interfere with this data and still run scripts to gather data for the remote laboratory.

Overview of the Remote Laboratory Architecture

The overall server architecture could be split into two sections, those being inverter communication architecture and server side architecture. By design, the inverter architecture dealt with all data collection, while the server side architecture received all data and converted it into a format that the students were able to access. The laboratory was set up in such a way that communication specific to the renewable energy requirements of this laboratory were all handled separately to all other communication protocols, meaning the architecture of the laboratory can easily be modified to suit other remote logging applications. This will be important in future while developing other logging architecture with unique systems.

The server side architecture was primarily programmed in python, with all web interface programmed in lamp 14.04 (Linux, apache, MySQL, PHP). The inverter communication architecture was handled with a raspberry pi communicating with the inverter and Deakin University monitoring hardware via the RS232 connection between the two devices. A brief overview of the system is shown in the diagram below:

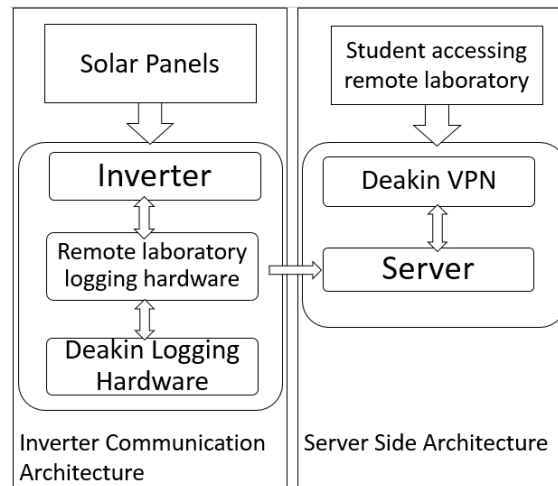


Figure 1: Shows a brief overview of the remote laboratory structure

Inverter communication Architecture

The hardware for the rig client of the remote laboratory involved using a raspberry pi to enable communication between the already established hardware systems. In an attempt to allow communication to the inverter, the RS232 logic converter was disconnected and connected to the raspberry pi (both plugs were converted to USB input for this step).

The raspberry pi was selected due to its ease of programming for an application such as this. It allowed for remote access and control from the server which was also using a Linux environment, meaning that any small changes required could be done remotely, and programming the device could also be done remotely. The Raspberry Pi could also be set up to automatically run the required python scripts, so in the event of a power reset, establishing communication between the Raspberry Pi and the server, as well as communication between Deakin University's logging hardware and the inverter can all be set up to happen automatically, meaning that server downtime in the event of a restart is minimal. Since both RS232 connections were converted to USB outputs and connected to the Raspberry Pi, it was required to configure the device to allow automatic communication between the two USB ports. With this system in place, the Raspberry Pi was therefore able to wait for a break in communication and send its own commands to the inverter. Since a single inverter is used to handle both the solar panel and wind turbine output, only one Raspberry Pi was required

.The data collected from the inverter was the joules generated in the last ten seconds, therefore the raspberry pi only needed to take one reading every ten seconds to be able to gather all required data from the inverter. The method of controlling the Raspberry Pi used threading, having one thread responsible for communicating with Deakin University logging equipment, one thread responsible for communicating with the inverter, and the final thread controlling logging data once communication is allowed. Using threading is advantageous as it allows for all three threads to be executed simultaneously.

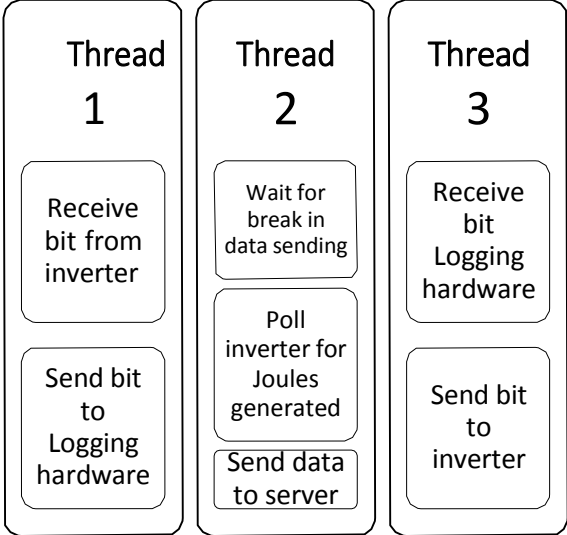


Figure 2: Shows brief communication overview of the Raspberry Pi

Server Side Architecture

The server side architecture was developed in a way that involve splitting all behaviours into two sections, the required program to communicate with the Raspberry Pi, and the programming to create the web interface to correctly communicate with students completing the remote laboratory.

To communicate with the Raspberry Pi, a UDP (user datagram protocol) protocol was used. UDP has advantages in that it has no handshaking procedure, meaning that if either the rig client or the server architecture reset, communication was much more easily be re-established. Data received from the raspberry pi was automatically stored in a MySQL database. The server side was responsible for all analysis of data, meaning it would also input the time logged into the database. This led to an issue with the data being logged, in that the data stored would be logged at roughly 2-3 seconds after it was initially read by the Raspberry Pi, however since the data collected was joules generated over ten seconds, this delay was considered acceptable.

Once the data generated from the Raspberry Pi was being properly stored, it then became important to generate a web page that would properly be able to give this information to students attempting to access the laboratory. For this, Lamps 14.04 was used. LAMPS is a set of open source software packages that allows for the creation of web pages that communicate with data stored in the database. The advantage of using LAMPS is that is allowed for coding web pages in HTML and PHP, therefore most web browsers would be able to access and operate in the page. Not requiring a separate program to be installed allowed for a wide range of computers to easily access the laboratory without needing students to download lab specific installation files, making lab access significantly easier. Since the web page was hosted using one of Deakin University's desktop computers as a server, it requires logging into the Deakin University VPN. This is easily done using JUNOS PULSE, with the Deakin University website hosting instructions on how to properly set up and access the VPN. This allows students to easily access the web server. In addition to this, the VPN is already used to allow remote students' access to Deakin software, meaning students should either already have the VPN configured, or have previous exposure to it, and that technical support is readily available from university staff.

When students access the web server, they are given the option of viewing data as it is generated live, such that a graph will be created updating the data points as they are read from the solar inverter in real time, or by viewing the last two hours of data generated on a static graph. The graph showing live data is set up such that it will poll the MySQL database every ten seconds to access the last stored data point. This data point is added then added to the graph, updating on the students web browser. This method allows the student to see the live data output, and to get a better understanding of the erratic nature of renewable energy power outputs.

The web page also links to CSV files for the students to access logged data from the inverter. This allows students to do analysis of the data over a much longer period of time, not only while they are physically viewing the data generated. The CSV file contains the data logged as well as the time logged, the date and the ID number of the data logged.

An example of the expected output for the solar inverter is shown in Figure 3 below. The data was taken at mid-day, and shows roughly 30k joules being generated every 10 seconds.

	A	B	C	D	E	F	G	H
1	id	Joules	year	month	day	hour	minute	second
2	341351	29279	2015	8	17	12	28	40
3	341352	30460	2015	8	17	12	28	51
4	341353	30492	2015	8	17	12	29	2
5	341354	30744	2015	8	17	12	29	12

Figure 3: Shows the output CSV file of solar data generated by the remote laboratory

Web Layout

When a student logs on to the web server, they are immediately shown a graph with the live data being displayed. An example of this is Figure 4. Note, this graph is the result of two minutes of logging data.

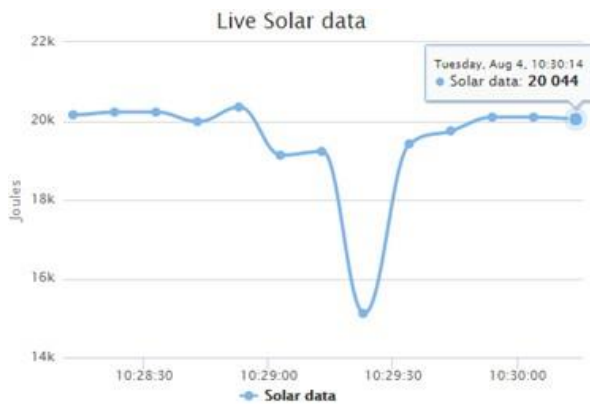


Figure 4: Shows the output graph of live data being generated

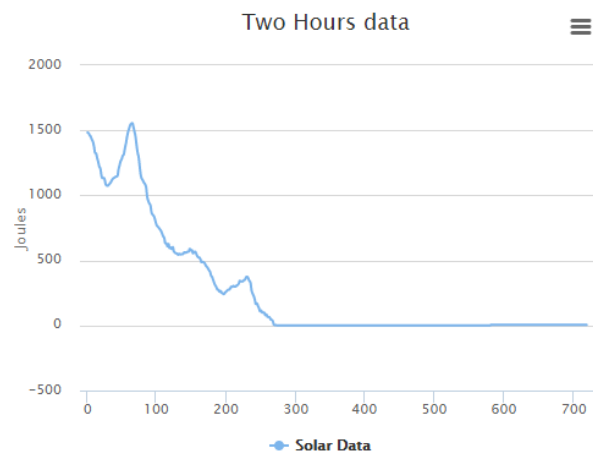


Figure 5: Shows the output graph of two hours' worth of data

While this graph is informative, if a student is looking to quickly see a large amount of data, it is possible to show the last two hours of data. This is presented to the students as shown in the Figure 5. While it would be possible to wait for two hours data to be logged to see the same graph, students can instead log in and immediately see the last two hours' worth of data. This allows for the user to see recent trends with the data output. The above data, for example, was taken at 7.30PM, showing the large drop off of output joules. If the students wish to see a larger amount of data, they will be required to access the log files. This involves downloading the CSV file for a specific data range, and manually graphing the data. Note, there is currently a bug with the output where the x-axis does not show time, but instead the number of data points collected. This will be fixed shortly, but was still an issue at the current time.



Figure 6: Shows the mobile device interface for this remote laboratory



Figure 7: Shows the zoomed in interface for mobile device

An extra feature of this laboratory is that while connected to the Deakin University wireless network on mobile devices, the lab can be accessed in much the same way. A screenshot of the lab while accessed from a mobile device is shown in the Figures 6 and 7.

While the above Figure 6 may be hard to view, it is possible to use the zoom function on the mobile device to zoom into interesting parts of the graph to allow for a much easier viewing experience for the end user. An example of a zoomed view is shown in Figure 7. Note that reading exact values from the graph is now made significantly easier. By allowing the students to access the remote laboratory from a mobile device, the usability of the lab increases significantly.

Initial Student Feedback

At this point feedback is purely anecdotal, with students expressing an interest in completing the laboratory. It has been observed that initial motivation to complete the laboratory seems to be high, but if this level of motivation is maintained is yet to be established. The remote laboratory being accessible from outside university is also a concept that some of the students did not consider possible, and as such they see remote laboratories as a way to have a 'hands on' play with new educational equipment and technology. In this way students see the remote laboratory as something new that they can use to learn on, which in itself could be a factor in maintaining student motivation. Further testing and analysis will be required to make any further claims than this however.

Future additions and Improvements

Currently, the remote laboratory is missing live outputs from the wind turbines. It would be possible to add these outputs to the graph in the same way as the solar data, however further testing to assure that the communication between Deakin's logging hardware and the inverter is not disrupted would be first required.

Another add on that would be beneficial is the inclusion of a live feed from a web camera showing the solar panels. This would act as a reinforcement to prove to the students that the data is real, as well as having a visual indicator for the light levels compared to the solar output. The current design for a web camera is to stream a single image that updates every ten seconds as to not go through too much of the users' bandwidth.

Currently there are plans to migrate the server from a location hosted at Deakin University to a server hosted in a different location. The reason behind this is to enable users to log in to the web server without requiring to connect to the Deakin University VPN, and mobile devices will be able to connect to the server while not connected to the Deakin University wireless network, as well as allowing students access to the laboratory without requiring JUNOS PULSE be configured.

Implications of this Remote Laboratory and Conclusion

This paper covered the design of a remote laboratory that allows students to see the outputs of a renewable energy system in real time from any location with an internet connection. The purpose of this laboratory is to allow students to learn about renewable energy systems from equipment using real world inputs. It allows students to see live data being generated in real time, and to see the effects of changing light conditions on the outputs in real time. It also allows students to see data live, as it would actually be generated, instead of analysing data such as the daily average. The fact that a remote laboratory such as this allows students to access this information at any time, even from their own home, allows the way that students approach education to change to allow a greater flexibility in their undergraduate engineering study.

References

- Cooper, M. and J. M. Ferreira (2009). "Remote laboratories extending access to science and engineering curricular." Learning Technologies, IEEE Transactions On **2**(4): 342-353.
- Corter, J. E., et al. (2007). "Constructing reality: A study of remote, hands-on, and simulated laboratories." ACM Transactions on Computer-Human Interaction (TOCHI) **14**(2): 7.
- Feisel, L. D. and A. J. Rosa (2005). "The role of the laboratory in undergraduate engineering education." Journal of Engineering Education **94**(1): 121-130.
- Lindsay, E., et al. (2007). Remote laboratories in engineering education: Trends in students' perceptions. Proceedings of the 2007 AAEE Conference.
- Lowe, D., et al. (2011). "Labshare: Towards Cross-Institutional." Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines: Scalable E-Learning Tools for Engineering and Science Disciplines: 453.
- Lowe, D., et al. (2012). "Uts remote labs, labshare, and the sahara architecture." Using Remote Labs in Education: Two Little Ducks in Remote Experimentation **8**: 403.
- Lowe, D., et al. (2009). "Evolving Remote Laboratory Architectures to Leverage Emerging Internet Technologies." Learning Technologies, IEEE Transactions On **2**(4): 289-294.
- Lowe, D., et al. (2008). "Reflecting professional reality in remote laboratory experiences." Remote Engineering and Virtual Instrumentation, ME Auer and R. Langmann,(Eds), Proc. of REV 2008, Düsseldorf, Germany.
- Lowe, D. and N. Orou (2012). Interdependence of booking and queuing in remote laboratory scheduling. Remote Engineering and Virtual Instrumentation (REV), 2012 9th International Conference on, IEEE.
- Sauter, M., et al. (2013). "Getting real: the authenticity of remote labs and simulations for science learning." Distance Education **34**(1): 37-47.
- Sousa, N., et al. (2010). "An Integrated Reusable Remote Laboratory to Complement Electronics Teaching." Learning Technologies, IEEE Transactions On **3**(3): 265-271.
- Yazidi, A., et al. (2011). "A web-based remote laboratory for monitoring and diagnosis of ac electrical machines." Industrial Electronics, IEEE Transactions on **58**(10): 4950-4959.

Copyright

Copyright © 2015 Liam D. Lyons, Matthew A. Joordens: The authors assign to AAEE and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to AAEE to publish this document in full on the World Wide Web (prime sites and mirrors), on Memory Sticks, and in printed form within the AAEE 2015 conference proceedings. Any other usage is prohibited without the express permission of the authors.