Light Insight — Emulation of Radiation Traces for Analysis and Evaluation of Solar-Harvesting Algorithms

Lars Hanschke and Christian Renner
{lars.hanschke,christian.renner}@tuhh.de
Hamburg University of Technology
Research Group smartPORT

Jannick Brockmann, Tobias Hamann, Jannes Peschel, Alexander Schell, Alexander Sowarka
Hamburg University of Technology

ABSTRACT

The restricted energy budget of energy-aware and -predictive sensor nodes demands algorithms for adaptive duty-cycling and task scheduling. However, their comparison and development is hindered by the lack of reproducibility of environmental conditions, such as solar radiation. To debug their behavior efficiently, it is also key to replay conditions in which the algorithms react unexpectedly in the laboratory. To investigate real-world issues such as dust or pollen, we enable replaying recorded lighting conditions by building an affordable light box. Our self-developed control circuit and high-power LEDs allow us to repeatedly replay environmental illumination data through previously collected and also artificial current and voltage traces. By using the on-board ADC of the harvester, we are able to replay the conditions exactly as seen by the sensor node. We achieve replaying accuracy below the limits of the sensor node's sensing circuitry, which enables us to directly compare the behavior of nodes running different energy-aware and -predictive algorithms.

CCS CONCEPTS

• **Computer systems organization** → *Embedded and cyber-physical systems*; • **Hardware** → *Power and energy*;

KEYWORDS

energy harvesting, cyber-physical systems, test equipment

ACM Reference Format:

Lars Hanschke and Christian Renner and Jannick Brockmann, Tobias Hamann, Jannes Peschel, Alexander Schell, Alexander Sowarka. 2017. Light Insight — Emulation of Radiation Traces for Analysis and Evaluation of Solar-Harvesting Algorithms. In *Proceedings of ENSsys'17*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3142992.3142994

1 INTRODUCTION

The deployment of self-sustained wireless sensing devices for cyber-physical systems (CPS) powered by solar energy aims at reducing maintenance effort and achieving placement flexibility. However, energy-harvesting devices rely on environmental conditions that experience frequent changes. Forecasting the multiple factors of solar radiation such as clouds, local shadowing and seasonal changes draw attention in research, e.g. in [11], [3] and [9].

ACM ISBN 978-1-4503-5477-6/17/11…$15.00
https://doi.org/10.1145/3142992.3142994

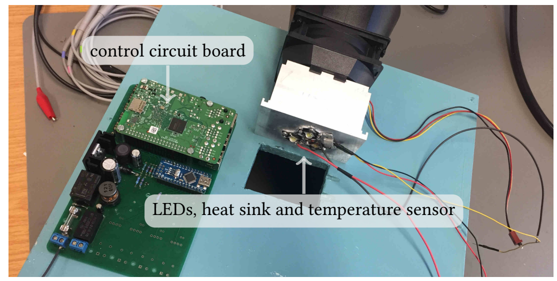

Figure 1: Light box with high-power LEDs; full brightness yields 75 mA **at the solar panel with the used harvester; self-developed control circuit and temperature monitoring.**

Since CPS devices have limited dimensions and an intermittent supply with average intake typically below consumption of fully enabled hardware, their energy resources are restricted. Algorithms adjusting a device's duty cycle, such as [8] or [10], aim at using these resources efficiently.

Comparing these algorithms under equal conditions is hardly feasible in practice: days with equal solar radiation profile are rare and impossible to predict. Especially situations of further interest, e.g. an unexpectedly decreasing energy intake, are hardly reproducible. However, for debugging and optimization, understanding the potentially wrong behavior of algorithms is key for improving. Consequently, the sensor node's harvester has to be fed by an entity offering: stable conditions in repeated experiments, accuracy within the limits of the sensor node's sensing circuitry and easy-to-use exchange of replayed scenarios.

The authors of [6] and [2] substitute solar cells or RF harvesters by controllable current and voltage sources, which allows them to replay previously recorded current-voltage traces. Despite being accurate, their approach is not to replay already happened corner cases; e.g. a failed long-term test. If this situation can be exactly replayed afterwards in the laboratory as observed by the sensor node, additional options of debugging can be applied and consequently modifications in the algorithms can be tested under controlled circumstances.

While reproducing current-voltage traces is sufficient in most cases, practical issues of solar cells and reaction of the sensor nodes cannot be modeled: e.g, dust or pollen may reduce energy intake in reality and also smaller damages can only be investigated in laboratory with the solar cell. Therefore, a manual emulation of these real-world issues should be possible. Additionally, solar cells experience slow transient processes in rapidly changing lighting conditions; investigating these issues without the need for complex changes in hardware structure leads to a deeper insight of the behavior of solar-energy harvesters. Thus, a solution is needed for replaying recorded real-world lighting conditions providing the same solar energy-harvesting conditions, which should be as closely to the scenario seen by the energy-aware sensor node.

Furthermore, while energy-harvesting sensor networks become wide-spread, their joint energy consumption draws more attention. A key question is how to coordinate network tasks considering the joint energy budget of multiple nodes. E.g, spatial monitoring resolution might be reduced in times of low energy intake. Consequently, investigating the cooperation between energy-harvesting nodes experiencing the same or even different environmental conditions is of high interest. Thus, a replaying solution has to be accurate but also affordable to enable researchers to build more of them to form an energy-harvesting replaying network. Achieving this is possible using a simple circuitry with affordable regular hardware parts.

This leads to the development of our light box, depicted in Fig. 1, which uses high-power LEDs, is remotely configurable and uses affordable off-the-shelf hardware components. The light box is able to replay already recorded lighting conditions from the environment but also artificial traces through current and voltage traces by a sensor node to enable subsequent testing of energy-aware and -predictive algorithms in the laboratory.

We used our prototype harvester presented in [5] to calibrate the platform to ensure the replayed scenario is as close to the originally recorded real-world situation as possible. In consequence, no special sensing equipment is needed for the presented accuracy: only a communication channel between lighting control and harvester inside the box is needed to calibrate and start emulations.

The rest of the paper is structured as follows: First, we introduce the goals as well as hard- and software structure of the light box. Second, we show techniques to achieve the accuracy in the range of the harvesting platform and third, we evaluate the performance of the light box concerning reaction of the energy-aware sensor node.

2 THE BOX

The varying nature of environmental energy resources demands new challenges on duty-cycling and task scheduling algorithms. However, if an oddity was observed during a long-term test, it is hard to rebuild the exact environmental conditions to investigate changes in the algorithm. If the environmental conditions were monitored, e.g. a wireless sensor node reports its observations on solar radiation and residual energy level regularly, they can be rebuilt in the laboratory.

Our goal with the light box is to take these real-world data as input and reproduce them so accurately that the sensor node inside the light box experiences the same situation as outside. Furthermore, we want to replay these situations multiple times to alleviate uncontrollable effects, such as varying wireless channel quality.

In this section, we highlight the system structure of our light box and present the utilized hard- and software.

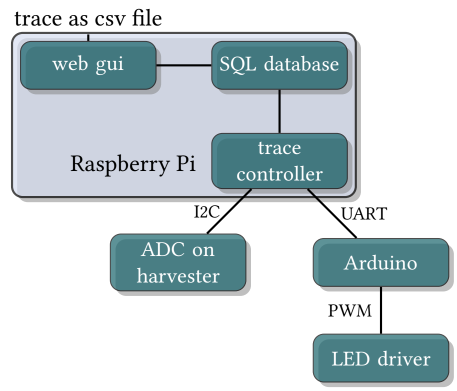

Figure 2: System structure of the light box; input of traces via web page, storage and trace control are hosted by a Raspberry Pi; hardware PWM for LED control is provided by an Arduino Nano.

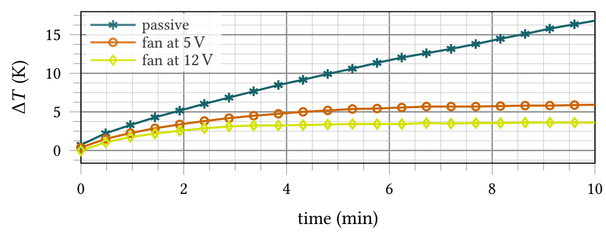

Figure 3: Temperature difference to ambient temperature at heat sink; junction temperature at LED is expected to be 60 K **higher; passive cooling insufficient.**

2.1 System Structure

We used our harvesting platform presented in [5] to record real-world lighting conditions through solar current and capacitor voltage traces during a long-term test. It is also possible to feed custom traces to evaluate corner cases or performance in situations of special interest. These traces contain triples of delivery time, solar current and capacitor voltage.

As shown in Fig. 2, these traces serve as input for the trace controller, which runs on a Raspberry Pi. It is powerful enough to host the easy-to-use web GUI and the SQL database and, for future versions, is capable of handling multiple traces in different boxes at the same time. This gives us the flexibility to create multiple boxes with just one single trace controller. Additionally, it is capable of a variety of communication protocols such as I2C or UART.

However, the PWM signal for controlling the LEDs might be unstable if it is generated in software by an operating system handling other processes. Thus, we integrate an Arduino Nano, which only receives commands via UART and adjusts its output PWM duty cycle upon reception of the configuration command.

To maintain a still compact design, we developed a control circuit board, which hosts all hardware parts needed for operation of the light box. This control circuit and its components are explained in the following.

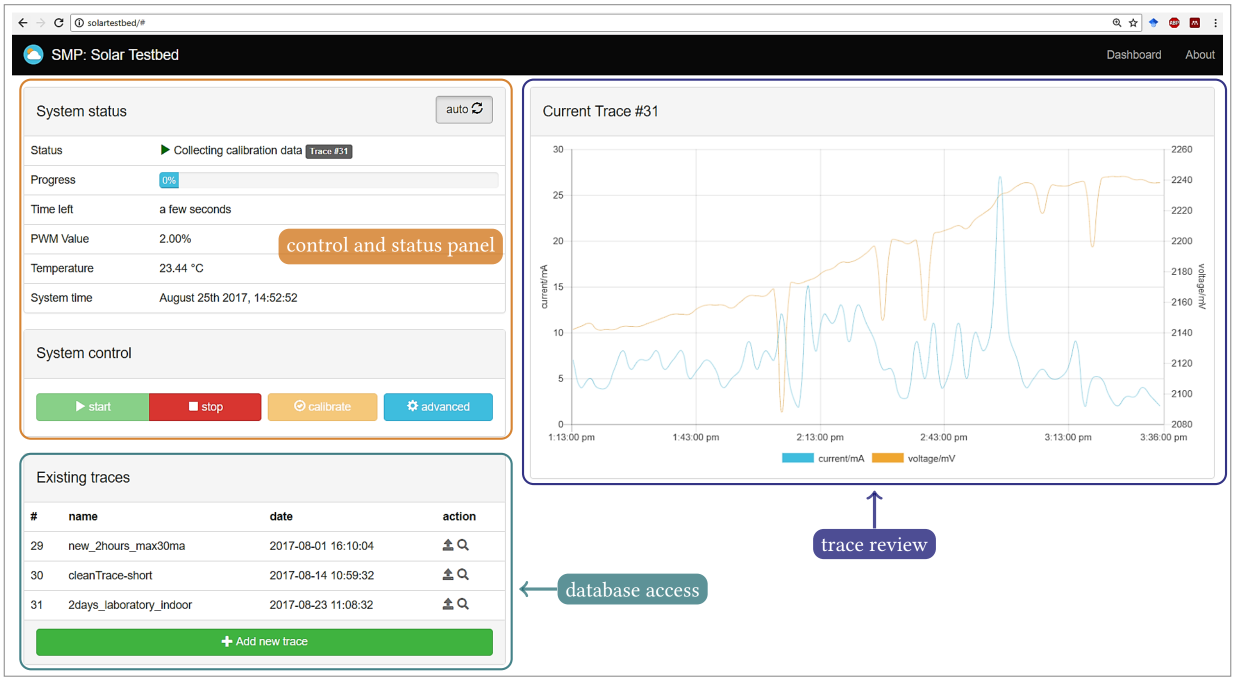

Figure 4: GUI allows comfortable upload of recorded traces via web browser as csv files; traces can be reviewed after upload; system control allows monitoring of progress and experiment control.

2.2 Hardware

The control circuit board of our light box hosts four major components: a Raspberry Pi, an Arduino Nano, a 40 V DC-Converter and a LED driver. Since a single power supply for the whole control circuit increases the manageability of the light box, we need a down-converter providing 5 V for the Raspberry Pi. The Arduino is directly supplied by the Raspberry Pi. Control commands are sent via UART to the Arduino Nano, which generates the PWM signal with adjustable duty cycle to control the LED driver Recom RCD-24-0.70. This driver allows us to control the current through the LEDs and consequently illumination level accurately.

Two high-power LEDs of the Cree MK-R J2 series [4] are served per LED supply string with forward voltage of 11.7 V at 700 mA. Their light temperature of 6200 K is close to what is observed in direct sunlight (6000 K to 6500 K). We also plan to integrate LEDs with different light temperatures, to allow for emulation of the different light phases during a day. The whole circuit board hosts up to five strings with up to ten LEDs, which allows us to extend the box in future versions, e.g. to support harvesters with larger energy intake.

To increase the lighting yield, a reflector is used to steer the beam of the LEDs towards the solar panel. We found this increases the achievable solar current by up to 25%. With reflector and two high power LEDs, we are able to achieve a harvester output current of up to 75 mA with our hardware.

To keep the internal temperature of the box at a low level, an efficient heat dissipation from the LED junction is needed. The authors of [6] also found that this influences their repetition accuracy. Our setup uses a conventional CPU heat sink with active cooling. To attach the LEDs to the heat sink, heat conductive tape was not sufficient. Since the tape only offers a heat conductivity of 1.4 W/mK, the LEDs quickly started overheating. Replacing this thermal transition by regular thermal paste solved this issue. As displayed in Fig. 3, the difference between heat sink and ambient temperature quickly rises beyond 15 K without active cooling. By calculating an equivalent thermal circuit, we deduced that the temperature difference between heat sink and LED junction is approximately 60 K. With an ambient temperature of 25 °C, a heat sink temperature difference of 15 K is equivalent to a LED junction temperature of approximately 100 °C. Since this already influences the lifetime of the LED and the brightness level, we opt for an active cooling solution. We tested two different voltage settings: the fan runs at its maximum speed at 12 V but 5 V are directly available on our control circuit board. Both voltages provide a sufficient cooling, so we decided to use 5 V to save additional voltage converting circuits.

With control modules and two LEDs at maximal brightness, the box consumes 21 W. The total bill of material is around € 110.

2.3 Software

A key requirement of the light box is its remote configurability. The Raspberry Pi hosts a web page with GUI for the light box control, which we show in Fig. 4. In the control panel area, the user can start and stop the experiment, perform a new calibration or shutdown the whole system. Status information such as progress, remaining time as well as heat sink temperature can be monitored during calibration or experiment. Furthermore, the system is automatically shut down if the LED temperature exceeds a threshold. Solar current traces from previous experiments can be uploaded comfortably via

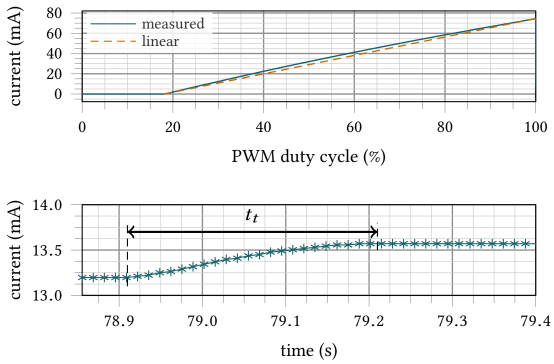

Figure 5: Calibration trace links PWM duty cycle and solar current; LEDs start working above 18% PWM duty cycle; between PWM steps, harvesting circuit experiences transient behavior manifesting in t_t; we observed $t_{t,\max} = 2.93$ s.

web browser in csv format and are stored in a SQL database for later access. They can be checked before starting the experiment in the trace review area to ensure the correct data is loaded.

Reliable communication between web GUI and lighting control is ensured via TCP sockets. Before starting a newly loaded experimental trace, the box is calibrated, c.f. Section 3.1, to match the PWM values and the current. To replay recorded and artificial traces accurately, further steps during calibration are necessary which we explain in the following.

3 TECHNIQUES FOR ACCURATE REPLAY

To replay recorded situations, we use the same hardware platform for recording, calibrating and replaying. We achieve comparability by applying two techniques before starting any experiment which we explain in the following sections.

3.1 Accurate Calibration

We control the lighting conditions in the box by applying a distinct PWM duty cycle to the LED driver, which regulates the current through the LEDs. The resulting light generates current at the solar panel, which charges the supercapacitor and can be monitored by the harvester of the sensor node inside the box.

Before starting the experiment, a calibration is needed to map PWM values to the desired lighting conditions. The calibration curve differs upon used solar cell, distance between LEDs and solar panel and positioning of the solar cell inside the box or the illuminated area on the solar panel, respectively. Since exact placement of the harvester cannot be ensured between experiments, e.g. after flashing a new software on the sensor node, we decided to calibrate the lighting conditions before each experiment.

The calibration procedure works as follows: each PWM value is applied for a calibration time t_c. At the end of this timespan, the Raspberry Pi reads the solar current and the capacitor voltage from the ADC of the harvester and stores them in the so-called PWM-current map. We display the resulting calibration trace in Fig. 5. The difference between measured and ideal curve is mainly caused by non-linear behavior of the Cree LEDs, which is also treated in

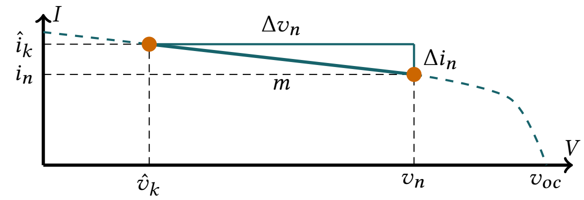

Figure 6: Cap voltage $\hat{v}_k$ during calibration might differ from voltage v_n in replayed trace; instead of wanted current i_n, $\hat{i}_k$ is generated; before replay, the trace is compensated by applying an offset factor Δi_n to each trace point.

the data sheet [4]. Due to the minimum current of the LEDs, a PWM duty cycle of approximately 18% is needed for the lowest illumination level. As our box uses an Arduino Nano to achieve real-time PWM control, values between 0 and 255 are available for adjusting the duty cycle and the lighting level in our box, respectively. Even with wasting a fifth of the PWM values, at full brightness we achieve a step size of 0.36 mA , which is equal to 3 steps of the solar harvester 12 bit-ADC.

Due to the widely studied transient effects of solar cells during calibration, e.g. in [1], a careful adjustment of the calibration time is needed. We define the transient time t_t as the time difference between applying a new PWM signal and stabilizing of the solar current. A stable current is assumed when five subsequent samples at 100 ms sampling interval give the same ADC value. During our calibration tests, we observed a median and maximum transient time of $t_{t,\text{median}} = 1.86$ s and $t_{t,\max} = 2.93$ s, respectively. Applying a calibration time of $t_c = 3$ s to each of the 256 PWM values yields an overall calibration time of 13 min.

3.2 Compensating Power Point Shifting

Our prototype harvester presented in [5] uses a direct charging circuit for the supercapacitor; thus, the voltage of the capacitor influences the power point of the solar panel and consequently the amount of harvested power. In future, we also plan to ensure compatibility for solar harvesting platforms using MPPT devices and algorithms.

During calibration, the load at the solar cell is given by the voltage of the supercapacitor. The load voltage influences the power point of the solar cell and thus the current which is stored in the PWM-current map during calibration. During replay, the capacitor voltage might differ from the voltage during calibration; thus, the generated current is different. During replay, a PWM value with an expected current is applied, but since the capacitor voltage changed, the observed current differs from the expected current due to the calibrated PWM-current map. During compensation, we are looking for the current difference between calibrated value and recorded value: in consequence, a higher or lower PWM value has to be chosen than originally expected.

We depict this issue in Fig. 6. Our recorded solar traces consist of n triples containing time, solar current i_n and cap voltage v_n. The PWM-current map contains k triples, $k \in [0, 255]$, containing PWM value k, the cap voltage $\hat{v}_k$ and the generated solar current $\hat{i}_k$. Since it is expected that $v_n \neq \hat{v}_k$, we observe $i_n - \hat{i}_k = \Delta i_n$. Consequently,

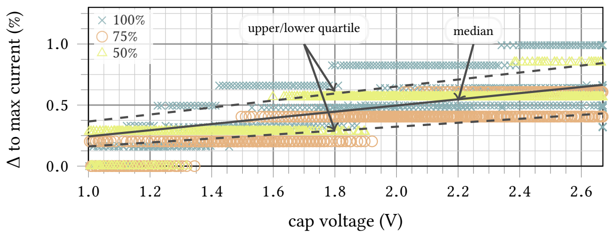

Figure 7: Relative difference between maximal current and observed current at shown cap voltage; less current can be drawn from the solar cell with increasing voltage; linear curve of the median gives gradient for compensation.

the observed solar current during replay may be smaller or larger than expected. If v_n is not closely located to the open circuit voltage v_{oc}, an approximately linear dependency can be found: we can use the known voltage difference Δv_n to calculate

$$\Delta i_n = m \cdot \Delta v_n = m \cdot (v_n - \hat{v}_k). \tag{1}$$

After calculating Δi_n, we choose a compensated PWM value for the current $i_n + \Delta i_n$.

To obtain the gradient m, we recorded current-voltage traces at different illumination levels. We display the results in Fig. 7. Note that we normalized the values to the maximum solar current of the corresponding illumination level to compare them side-by-side. The switching regulator of our harvesting platform works with capacitor voltages down to 1 V, which determines our lower limit. The solar cell has an open-circuit voltage of 4 V whilst the supercapacitor voltage is rated at a maximum voltage of 2.7 V. This ensures the condition that v_n is far off v_{oc}.

We obtain a constant solar current by applying a constant illumination level and measure the current from the solar cell into the capacitor using a 1 Ω measuring shunt. The voltage at the measuring shunt is amplified and sampled with the 12 bit-ADC on-board the harvesting platform.

To compare the different illumination levels, we calculate the difference towards the current at 1 V cap voltage and normalize the difference by dividing by the maximum current. Based on all data points depicted in Figure 7, the linear curve fitting yields a median relative gradient m' of 0.26 %/V. Note that m' has to be multiplied by the maximum current to obtain m which is mandatory for our compensation technique. We also observe a higher relative gradient for higher illumination levels, e.g. for 100% we observe up to 0.45 %/V. Using $\Delta v_n = 1.7$ V and a current of 75 mA yields $\Delta i_n = 0.57$ mA. Since this is already in the range of the step size and thus emulation capability, we expect the influence to be minor compared to the right choice of the calibration time.

4 RESULTS AND PRACTICAL MERIT

In this section, we show the impact of our different calibration and compensation methods and evaluate the repetition accuracy of subsequent experiments.

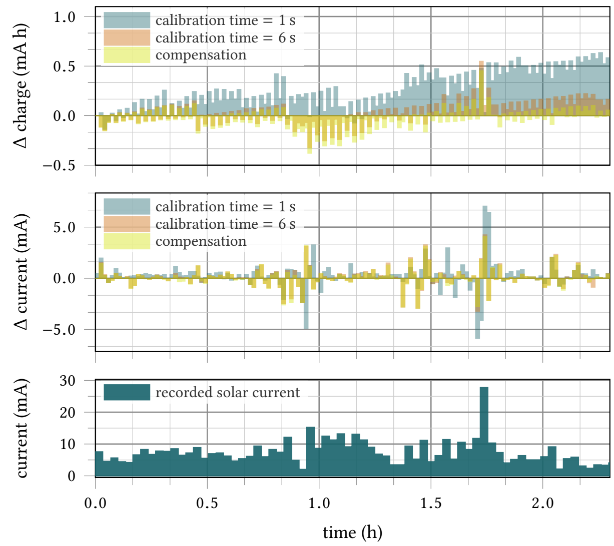

Figure 8: Difference between replayed current trace and observed input at harvester; charge is summed up error during replay; a higher calibration time and compensation of cap voltage difference is needed to increase accuracy.

4.1 Calibration Methods

We display the benefits of the compensation method in Fig. 8. We compare the overall charge fed into the charging circuit during the replay against the charge of the recorded trace. Since most energy-aware duty-cycling algorithms, e.g. [3] or [8], evaluate the battery level of a node, we are interested in the amount of charge fed into the storage unit at the end of the experiment. The same sensor node and harvester is used throughout all experiments.

With short calibration time of 1 s, the charge difference at the end of the simulation is 0.52 mAh, which results in a voltage difference of 37.44 mV with a 50 F capacitor.

Using a calibration time per PWM value of 6 s yields a charge error of 0.01 mAh. With the total amount of charge fed during the emulation of 17.23 mAh, this results in a relative error of 0.58%. The voltage difference with the used 50 F capacitor is 7.2 mV.

The compensation method presented in Section 3.2 yields a charge error of 0.001 mAh, leading to a cap voltage error of only 0.72 mV. Since the ADC compares the capacitor against a reference voltage of ±4.096 V, this result is already below the accuracy which is used for the energy-aware algorithms.

The practical merit is also shown in the difference to the original current traces: the observed trace with the developed compensation in mean only differs by 0.62 mA, which is in the range of the measurement accuracy. The on-board 12 bit-ADC with ±2.048 V reference voltage has a resolution of 0.12 mA. Note that the ADC uses half the reference voltage for solar current compared to cap voltage measurements to obtain more precise results.

This underlines that, within the accuracy range of the used harvester, our box is able to reproduce environmental lighting conditions accurately.

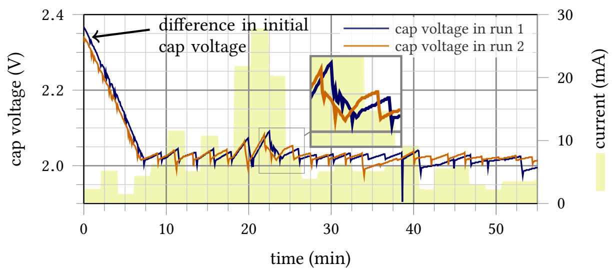

Figure 9: Repeated experiment; different initial cap voltage hinders exactly reproducible voltage traces; controlled discharging of capacitor planned in future versions; reaction of harvester is repeatable, i.e. gradient of cap voltage is equal.

4.2 Repetition Accuracy

To alleviate uncontrollable conditions such as varying radio channel conditions, subsequent repetitions of experiments are needed. We compare two experiment runs using an energy-aware but simple program on our WiFi sensor node. The node operates as long as it has energy and enters a sleep mode if detected otherwise. The sensor node queries the ADC of the harvesting platform four times a second. If the capacitor voltage is larger than 2 V it stays active, otherwise it goes to sleep for 2 min to save energy waiting for the capacitor to charge.

We attach another platform with the same hardware to the inputs of the ADC inside the box and sample values every 100 ms to obtain a timely fine-grained image without influence of the wireless channel. We display the resulting cap voltage trace of both runs and the replayed solar trace in Fig. 9. Since the initial cap voltages differ, the curves do not exactly overlap. A small change of just 20 mV results in unequal decisions of the sensor node. If the nodes are in the same state, their power consumption is equal as well as the solar current which charges the capacitor. This translates to an earlier point in time, at which the threshold voltage of 2 V is reached and thus untimely decision of the node. Consequently, more effort has to be spent to adjust the initial capacitor voltage, which is hard to do manually. Since the used harvesting platform allows controlled discharging of the capacitor, we plan to implement a voltage adjustment at the beginning of each experiment.

The key observation can be seen in the magnification figure. When two nodes in the different runs are in the same state and consequently have the same power consumption, the gradient of the two cap voltage curves is the same; especially when the illumination level changes rapidly. E.g. between approximately 24 and 25 min, the cap voltage increases in both runs by 1 mV each six seconds. The RMSE of the solar current between both runs is 0.07 mA, which is below the measurement accuracy of the solar harvester.

5 RELATED WORK

The authors of [6] present a small and accurate platform for emulating current-voltage traces of solar energy but also RF energy. As RF traces vary much faster, they spend additional effort to increase the responsiveness of their platform. Their superior speed and accuracy comes at a higher price and a more complex electrical circuit. Additionally, practical issues such as partly covering of the solar panel cannot be directly investigated with their platform.

The analog front end of [2] allows for an accurate emulation of a variety of solar cells. Their system supports a very large range, capable of generating up to 9.8 V and a current of up to 1.89 A. However, their focus is not on providing the interface for replaying real-world data or corner cases to evaluate energy-aware algorithms.

A high-accuracy testing platform for solar energy-harvesting devices is presented in [7]. The platform is able to variate lighting conditions in several ways: by distance alteration, dimming, filtering or variation of incident angle. However, with a volume of $0.83\,\mathrm{m}^3$ compared to our box at $0.02\,\mathrm{m}^3$, it is over-sized for small energy-harvesting devices.

6 CONCLUSION & OUTLOOK

We presented an affordable light box, which is able to reproduce environmental conditions repeatedly in the laboratory. We showed that the box achieves measurement accuracy within the range of the harvesting platform which underlines that it is sufficient for investigating algorithms relying on typical low-power platforms.

The box can be easily extended with additional LEDs for larger solar cells or higher light intensity. With costs below € 110, we plan to produce more boxes to build a testbed for nodes experiencing different conditions. Additionally, we plan to integrate a mechanism for charging and discharging of the capacitor to enable fully autonomous operation. Follow-up experiments also embrace interacting nodes in multiple boxes and their joint energy budgeting.

REFERENCES

[1] Giorgio Bardizza, Diego Pavanello, Roberto Galleano, Tony Sample, and Harald Müllejans. 2017. Calibration Procedure for Solar Cells Exhibiting Slow Response and Application to a Dye-Sensitized Photovoltaic Device. *Solar Energy Materials and Solar Cells* 160 (2017).

[2] Stanislav Bobovych, Nilanjan Banerjee, Ryan Robucci, James P. Parkerson, Jackson Schmandt, and Chintan Patel. 2015. SunaPlayer: High-Accuracy Emulation of Solar Cells. In *Proc. of the 14th International Conference on Information Processing in Sensor Networks (IPSN)*.

[3] Alessandro Cammarano, Chiara Petrioli, and Dora Spenza. 2012. Pro-Energy: A Novel Energy Prediction Model for Solar and Wind Energy-Harvesting Wireless Sensor Networks. In *Proc. of the IEEE 9th International Conf. on Mobile Ad-Hoc and Sensor Systems (MASS)*.

[4] Cree 2017. *XLamp MK-R LEDs*. Cree. http://www.cree.com/led-components/media/documents/XLampMKR.pdf Rev. 8C.

[5] Lars Hanschke, Jan Heitmann, and Christian Renner. 2016. Challenges of WiFi-Enabled and Solar-Powered Sensors for Smart Ports . In *Proc. of the 4th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems (ENSsys)*.

[6] Josiah Hester, Timothy Scott, and Jacob Sorber. 2014. Ekho: Realistic and Repeatable Experimentation for Tiny Energy-Harvesting Sensors. In *Proc. of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys)*.

[7] Mojtaba Masoudinejad, Jan Emmerich, Dominik Kossmann, Andreas Riesner, Moritz Roidl, and Michael ten Hompel. 2016. A Measurement Platform for Photovoltaic Performance Analysis in Environments with Ultra-Low Energy Harvesting Potential. *Sustainable Cities and Society* 25 (2016).

[8] Clemens Moser, Davide Brunelli, Lothar Thiele, and Luca Benini. 2007. Real-Time Scheduling for Energy Harvesting Sensor Nodes. *Real-Time Systems* 37 (2007).

[9] Christian Renner. 2013. Solar Harvest Prediction Supported by Cloud Cover Forecasts. In *Proc. of the 1st International Workshop on Energy Neutral Sensing Systems (ENSsys)*.

[10] Christian Renner, Stefan Unterschütz, Volker Turau, and Kay Römer. 2014. Perpetual Data Collection with Energy-Harvesting Sensor Networks. *Transactions on Sensor Networks* 11 (2014).

[11] Navin Sharma, Jeremy Gummeson, David Irwin, and Prashant Shenoy. 2010. Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems. In *Proc. of the 7th Annual IEEE Communications Society Conf. on Sensor Mesh and Ad Hoc Communications and Networks (SECON)*.