

# Monitoring LED Lights with Current Signatures

Johnny Verhoeff    Akshay Uttama Nambi S.N.    Marco Zuniga Zamalloa    Bontor Humala  
TU Delft    Microsoft Research    TU Delft    TU Delft  
The Netherlands    India    The Netherlands    The Netherlands  
j.verhoeff@me.com    akshay.uttama@gmail.com    m.a.zunigazamalloa@tudelft.nl    bontorhumala@student.tudelft.nl

**Abstract**—Artificial lighting is a pervasive element in our daily lives. Researchers from different communities are investigating challenges and opportunities related to artificial lighting but from different angles: *energy disaggregation*, to monitor the status of light bulbs in buildings; and *communication*, to transmit information wirelessly. We argue that there is an unexplored synergy between these two communities. When a light bulb modulates its intensity for *communication*, it also affects the current it draws. This current *signature* is unique and could be used by *energy disaggregation* methods to identify the lights’ status. These signatures however will be exposed to interference (collisions of signatures) and distortions due to power line effects. To overcome these problems, we build upon coding schemes to assign interference-resilient signatures, and we develop custom hardware to ameliorate distortions introduced by power lines. We validate our framework in a proof-of-concept testbed, perform simulations to test scalability, and use energy traces from real homes to evaluate the impact of other electric loads.

## I. INTRODUCTION

Lighting accounts for 17-38% of total electricity consumption in buildings [1]. A significant part of this could be reduced by providing real-time information on individual light energy consumption to occupants and building managers. Our work is motivated by the potential synergy between two areas that investigate challenges related to artificial lighting, but that hitherto has not been explored much.

*Area 1: Visible Light Communication (VLC).* A new lighting infrastructure providing services beyond illumination. Due to their high energy efficiency, LED lights are rapidly replacing traditional light bulbs. By 2030, 86% of all lighting installations are expected to be LEDs [2]. And thanks to advances in VLC, LED lights can now modulate their intensities to transmit data. These advances are creating a new range of exciting applications. For example, Philips has transformed the lighting infrastructure of a French supermarket to provide localization services to its customers [3]. Thus, in the future we may be replacing the LED lights in our buildings with VLC lights that act as beacons to provide localization. If this occurs, we could leverage VLC to monitor in real time the status of individual lights (i.e. to check if they are *on* or *off*).

*Area 2: Energy Disaggregation.* The difficulty of monitoring light bulbs. Energy disaggregation aims at taking the entire energy signal of a building, and separate it into appliance specific data [4], [5]. Many appliances, such as fridges and HVAC systems, have unique energy profiles (signatures) that can be identified from the aggregated energy measured at the smart meter. But lights are difficult to monitor. Buildings can

have tens or thousands of lights with the same energy profile. Some recent studies explore various types of infrastructures to achieve that monitoring goal: light sensors [6], power line communication [7] and lighting control systems [8]. We propose a new alternative: to combine energy disaggregation methods with the up-and-coming VLC infrastructure.

*Our approach.* We assume that a building has VLC lights sending periodic beacons (IDs) for applications such as indoor localization. These VLC lights do not have any type of backchannel (e.g. no radio module or power line communication). When these lights modulate their light intensity, they also ‘modulate’ the current drawn to power themselves. These unique current patterns could be decoded at the smart meter. But the correct identification of these signatures would need to overcome two problems: (i) *interference*, many lights are *on* at the same time, and thus, their signatures will collide; (ii) *distortions*, the signatures will get distorted due to non-linear elements present in power lines. In light of these challenges, we present **iLED**, a framework to monitor the status of individual lights in real time using a single energy meter. iLED aims to provide valuable insights to occupants and building managers on electricity cost at individual light level, energy optimization, occupancy estimation and other applications. Our specific contributions are three-fold:

- 1) *Analytical framework (Secs. IV and V).* To tackle the problem of interference, we build upon coding theory to provide lights with a special ID that is resilient to collisions.
- 2) *Platforms (Sec. VI).* To tackle the problem of distortion, we develop custom VLC lights to overcome the non-linear effects caused by power lines (for energy disaggregation).
- 3) *Evaluation (Sec. VII).* We validate our approach with a small scale testbed to showcase the ability of iLED to overcome interference and distortion problems. To test scalability, we perform simulations. To analyze the effect of other electric loads, besides lighting, we use traces from real homes.

## II. RELATED WORK

The general area of energy disaggregation is vast [4]. We focus only on the most relevant work aimed at disaggregating the energy consumed by lighting.

**Leveraging electromagnetic signatures to identify the use of lighting.** Recent studies propose using high-frequency energy meters ( $> 1MHz$  sampling) to monitor electromagnetic interference (EMI) as a mean to identify unique signatures in appliances [9], [11]. Gulati et. al [11] can identify the presence

of LEDs powered by switch mode power supplies (SMPS), and ElectricSense [9] can detect the presence of different types of CFL lamps based on their consumption power. These studies can identify if lights are on. But they can not identify which *specific* light is on because it is difficult to distinguish EMI signatures when multiple instances of the same appliance are active (e.g., lights).

**Adding or exploiting existing infrastructure to monitor individual lights.** *There are other types of infrastructures that are suitable to provide individual light monitoring.* If a building already has these types of infrastructure, our method would not be required. Below we describe these approaches.

*Radio Channels & Building Control Networks (BCN).* Some lights have BLE or WiFi radios to control them [10]. By default, such lights can use their radios to report their status in real time. Some buildings have actuators to record lighting events. When a light switch is turned on, it pings an actuator that forwards the event to a central controller. An actuator can be connected to a single light or many lights in a room. In [8], the authors correlate these switching events with information from a smart meter to provide fine-grained decomposition of lighting usage. The decomposition can be per light or room, depending on how many lights are connected to each actuator.

*Power over Ethernet (PoE).* Nowadays Ethernet cables can provide sufficient energy to power *some* LEDs [12]. Thus, PoE can enable by default per-light monitoring because every LED light has a network address. PoE however has some limitations. First, it is an inefficient way of transferring power. The voltages are very low, between 30 and 60 V. These low voltages imply big currents, which lead to energy wastage on the transfer line (Ethernet cable). Second, PoE can only support low-power loads, 13 W for CAT3 and 25 W for CAT5, which limit the types of LEDs that can be used.

*Power line communication (PLC).* With PLC, appliances use adaptors to transfer data over the power line. A few research efforts have explored the use of PLC to collect data from appliances [7], [13], [14]. In [13], PLC is studied as the communication backbone for smarthomes. The study found that the performance of PLC can degrade rapidly due to high power motors and external interference (power lines are big antennas). Murthy et. al [14] looked deeper into the reasons behind PLC's performance degradation, and found that it is also affected by the distance between appliances and the number of simultaneous transmissions. Batra et. al [7] extended the above pieces of work by exploiting performance degradation itself as a signature. The same appliance (e.g. lights) located at different distances cause distinct reductions in bandwidth. Based on this pattern, the authors identify which appliance is *on*, but only if one of them is active.

Our work has some resemblance with PLC, but there is an important difference. PLC is a brittle technology, the power line carries two types of 'signals': changes in *current* due to energy consumption, and low *voltage* pulses to transfer data. PLC's low voltage pulses are easily distorted by interference. iLED, on the other hand, carries only one type of signal: changes in current. Since power meters are designed by default

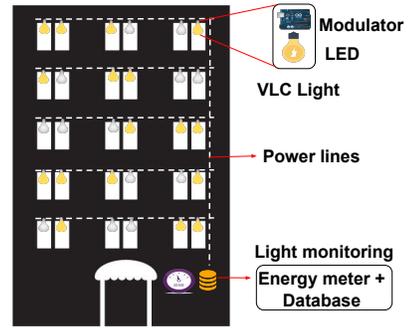


Fig. 1: iLED framework.

to measure accurately current flows, iLED is less affected by interference in the power line.

**Modulating light intensity and current simultaneously.** VLC is only concerned with modulating light intensity [15], [16], not with modulating the current drawn. We create custom hardware to achieve both modulation goals simultaneously.

### III. SYSTEM DESIGN

The aim of this work is to develop a framework to monitor the use of lights in buildings with a single energy meter. We focus on legacy buildings, that is, buildings that do not have any extra infrastructure beyond the basic power lines (AC or DC). Due to standard requirements, many buildings have separate power lines for lighting. For scenarios with a single power line for all loads, we present some analysis in Sec. VII-D. We assume that a smart meter is connected to the lighting power line and that the building has VLC-lights broadcasting periodically their unique IDs (the VLC lights are standalone devices with no backchannel). Our system is depicted in Fig. 1 and has three main components:

**VLC lights.** We use commercial LED lights with additional custom build parts to modulate their light intensity and current consumption. A simple On-Off Keying (OOK) scheme is used for modulation, where 0/1 symbols are sent by turning the LED off/on at high frequencies. Our modulation does not create any flickering effects.

**Energy meter.** The energy meter measures the current consumed by *all* light fixtures in a building. Due to the high modulation frequency of our *VLC lights*, we need a high-frequency, high-sensitivity energy meter. Some commercial meters do not provide this capability, others do but are expensive [17]. We develop our own energy meter (albeit intrusive) by embedding a resistor in the power line.

**Lighting monitoring system.** This component identifies the status of individual lights from the aggregated energy consumption. Our framework requires no training phase. This component only requires a database with the IDs of lights in the building. With this information, active IDs are decoded continuously from the aggregated signal.

### IV. CODES FOR iLED

To obtain IDs that are resilient to interference we build upon the field of coding theory. Our first task is to identify a code that abides by four key requirements in our application:

*Scalability.* Given  $L$  bits, we can obtain  $2^L$  unique IDs. But only a very small subset of those  $2^L$  IDs have resilience to interference. Each light will need an ID from that small subset. Thus, buildings with many lights may need increasingly longer IDs. But if the IDs get too long, we may not be able to transmit them in a timely manner.

*Balance.* In radio communication, it is not so important the pattern of 0's and 1's present in a code. But for VLC it is. If there are too many 0's in a row, flickering effects will occur. We require codes with a balanced sequence of symbols.

*Synchronicity.* Lighting switching events in buildings can occur in a synchronous manner, for example a switch turning *on* many lights in a room, or in an asynchronous manner, lights being turned on/off independently in different rooms. We need codes that perform well under both circumstances.

*Resilience.* Considering  $X_1$  as the ID of LED<sub>1</sub> (available from a database) and  $Y$  as the aggregated signal measured by the smart meter (which may or may not contain  $X_1$ ). The *lighting monitoring* component will correlate these two signals to identify if the ID is present (i.e. to check if the light is on). The correlation of two signals is defined as,

$$R(\tau)_{xy} = \sum_{i=0}^{L-1} x_1(i) \times y(i + \tau) \text{ with } \tau = 0, 1, 2, \dots, L \quad (1)$$

where,  $\tau$  is the lag of one signal relative to the other, and  $L$  is the length of the code. Assuming that a total of  $k$  IDs are present in  $Y$ , including  $X_1$ ; the above sum can be decomposed into two sums, one evaluating the *auto-correlation* of  $X_1 = \sum x_1(i) \times x_1(i + \tau)$ , and the other evaluating the *cross-correlation* between  $X_1$  and the other IDs present in  $Y$ :  $\sum x_1(i) \times x_j(i + \tau) + \dots + \sum x_1(i) \times x_k(i + \tau)$ . Ideally, we would like codes that give an *auto-correlation* of  $L$  for  $\tau = 0$  and zero for  $\tau \neq 0$ , and a *cross-correlation* of zero for all values of  $\tau$ . Such correlations would generate a single spike when a code is present and zero the rest of the time.

Next we describe succinctly different types of codes and highlight their pros and cons w.r.t. our scenario.

#### A. Orthogonal code sequences [21]

**Pros: scalability and resilience.** Orthogonal codes scale well. For a sequence of length  $L$ , we get  $L - 1$  codes. Orthogonal codes also have excellent resilience. When a code is present in the aggregated signal, the auto-correlation is  $L$ . If the code is not present the cross-correlations are zero.

**Cons: synchronicity and balance.** Orthogonal codes only work in synchronous scenarios. In asynchronous scenarios, the auto-correlation can be low and the cross-correlations are no longer zero [18]. Furthermore, in orthogonal codes, the number of consecutive zeros and ones increase as the code length increase, which would lead to *flickering effects*.

#### B. Pseudorandom Noise code sequences

**Pros: synchronicity and balance.** PN codes *can work in synchronous and asynchronous scenarios*. Furthermore, they have *great balancing properties* [19], which guarantees no flickering effects. Any PN sequence of length  $L$  contains an

TABLE I: Comparison of PN and Gold codes.

$L$	PN codes			Gold codes		
	# of codes	Cross-corr.	$m$	# of codes	Cross-corr.	$m$
31	6	11	1	33	9	1
63	6	23	1	65	17	1
127	18	41	1	129	17	3
511	48	113	2	513	33	7
1023	60	383	1	1025	65	7
2047	176	287	3	2049	65	15

almost perfect balance of ones and zeros:  $(L + 1)/2$  ones and  $(L - 1)/2$  zeros, and these symbols alternate in a rather homogeneous manner.

**Cons: resilience and scalability.** PN codes *have an average resilience*. The auto-correlation is good ( $L$  when  $\tau = 0$  and  $-1$  when  $\tau \neq 0$ ), but the cross correlations are greater than zero. If multiple lights are *on*, their cross-correlations could add up to values similar to the auto-correlation, which would lead to false positives. PN codes also *scale poorly*. The maximum number of codes in sequence of length  $L$  grows slowly, as it depends on the prime factors present in  $L$ .

#### C. Gold code sequences [20]

**Pros: synchronicity, balance and scalability.** Gold codes maintain the good properties of PN codes (balancing and (a)synchronous operation), and add scalability. The scalability of Gold codes is even better than that of orthogonal codes.

**Cons: resilience.** The resilience of Gold codes is lower than orthogonal codes, but higher than PN codes. If the pair of PN sequences are 'well chosen', the auto-correlation is given by,

$$R(\tau) = \begin{cases} L & \text{if } \tau = 0 \\ \{-t(n), -1, t(n) - 2\} & \text{if } \tau \neq 0 \end{cases} \quad (2)$$

$$t(n) = \begin{cases} 1 + 2^{\frac{n+1}{2}} & \text{for odd } n \\ 1 + 2^{\frac{n+2}{2}} & \text{for even } n \end{cases} \quad (3)$$

where,  $n$  is the number of LFSR registers, and the cross correlation can only take one of three possible values [20],

$$R_{xy}(\tau) = \{-t(n), -1, t(n) - 2\} \quad (4)$$

Table. I shows the code length ( $L$ ), number of codes ( $C$ ), peak cross-correlations and maximum concurrent transmitters ( $m$ ) for PN and Gold codes. We can see that for the same code length, the number of Gold codes generated is much higher and the peak cross-correlations are lower. **Thus we use Gold codes as the mechanism to generate the IDs for iLED.**

## V. INDOOR LIGHTING MONITORING

We now describe the two main challenges we need to overcome to use Gold codes for iLED.

#### A. Minimizing number of concurrent LED transmitters

In a building, numerous LEDs will be *on* at any given time, and thus, they will be transmitting their codes simultaneously. Since the cross-correlations of Gold codes are not zero, these correlation values could add up (or subtract) leading to false positives and false negatives (as explained later). To ensure that the LEDs' IDs can be decoded successfully at the smart meter, we first derive the maximum number of codes  $m$  that can be

transmitted simultaneously. Then we propose a probabilistic approach to guarantee with probability  $p$  that no more than  $m$  LEDs transmit their code at any given time.

1) *Maximum number of concurrent transmitters:* When a Gold code is present in the aggregated energy signal, the auto-correlation value is equal to the length of the code  $L$  and for all other codes the cross-correlation value is bounded by  $t(n)$ , which can be positive or negative (c.f. Eq. 4). Let  $T$  be the threshold used to determine if the correlation value is high enough to deem that a light is *on*. Due to the non-zero cross-correlations we can have false positives or false negatives. If a code is not present, but the cross-correlations with other codes are positive, the correlation values may add up to values greater than  $T$  (false positive). If, on the other hand, a code is present but the cross-correlations are negative, the final correlation may fall below  $T$  (false negative). To avoid these problems, we need to guarantee that:

$$R = L - m \times t(n) > T \quad (\text{avoid false negative})$$

$$R = m \times t(n) < T \quad (\text{avoid false positive})$$

By equalizing the above equations, we can compute (i) the maximum number of concurrent transmitters ( $m$ ) for a reliable decoding process, and (ii) the threshold value ( $T$ ),

$$m = \frac{L}{2 \times t(n)} \quad \text{and} \quad T = \frac{L}{2} \quad (5)$$

Now let us give an example to determine the parameters  $m$  and  $T$  for a building with less than 1026 lights. Considering a 10-register LFSR, the code length is 1023 ( $2^n - 1$ ), the maximum number of codes (lights) is 1025 ( $2^n + 1$ ), the maximum cross-correlation  $t(n) = 65$ , and thus, the maximum number of concurrent transmitters is 7.

2) *Probabilistic approach:* The maximum number of LEDs  $m$  that can transmit their IDs concurrently (c.f. Eq. 5) can be low compared to the number of LEDs active in a building. In our prior example, a building with 1025 lights could have at most 7 active lights at any given time. To ease this constraint, we propose a probabilistic approach. Every LED divides its local time into slots of length  $L/f$ , where  $L$  is the code length and  $f$  is the modulating frequency. For a code length of 1025 and a modulation frequency of 10 kHz, these time slots will be approximately of 0.1 seconds. At each time slot, the LED sends its ID with probability  $p$ .

Considering that we have  $C$  valid codes, but only  $m$  can be active simultaneously, we can model the probability of having  $m$  or less active transmissions as a binomial distribution,

$$\text{CDF: PR}(X \leq m) = \sum_{i=0}^m \binom{C}{i} \times p^i \times (1-p)^{C-i} \quad (6)$$

where  $X$  is the random variable representing the number of LEDs transmitting their IDs at any given time. The probability  $\text{PR}(X \leq m)$  needs to be high to avoid interference. To determine the probability  $p$ , a user could set  $\text{PR}(X \leq m) = 1 - \epsilon$ , where  $\epsilon$  denotes the likelihood of having more than  $m$  concurrent transmissions. The trade-off for selecting  $\epsilon$  is between time and accuracy. A small  $\epsilon$  increases the likelihood of correctly decoding (monitoring) IDs, but it also increases

the time required to hear from all LED lights. A large  $\epsilon$  has the opposite trade off. Once a user defines  $\epsilon$ , and obtains  $p$ , we can calculate the expected polling time. That is, how frequently we can monitor the LEDs. This expected polling time is given by  $E[t] = \frac{L}{f} \times \frac{1}{p}$ . In Sec. VII-C, we will see that iLED is capable of accurately monitoring hundreds of lights every few minutes.

## B. Correlation with (0,1) symbols

Coding schemes have been developed mainly to suit the needs of radio communication, where bits can be mapped to  $\pm 1$  symbols. But in our system the aggregated signal only takes 0/1 values. We thus have a mismatch: the codes stored in the *lighting monitoring component* (database) contain  $\pm 1$  symbols, but the aggregated signal measured at the smart meter is generated by 0/1 codes. To overcome this problem, we perform a transformation. First, we will obtain correlation values for codes using  $\pm 1$  symbols (as in the case of radio communication). Then, we obtain correlation values for our system. Finally, we show that the former correlation (which is what we want), is a function of the latter.

*Correlation between a  $\pm 1$  signal and  $\pm 1$  symbols.* Let  $x = s(t)$  be the received aggregated signal composed by  $m$  distinct codes with symbols (+1,-1),  $c_i(t)$  be the code sequences and  $y = c_0(t)$  be the code we want to identify in the received signal. The correlation in (1) can be rewritten at  $\tau = 0$  as,

$$R(0)_{s c_0} = \sum_{t=0}^{L-1} s(t) \times c_0(t) \quad , \quad \text{where} \quad s(t) = \sum_{i=1}^m c_i(t)$$

$$R(0)_{s c_0} = \sum_{t=0}^{L-1} \left\{ c_0(t) \times \sum_{i=1}^m c_i(t) \right\} \quad (7)$$

The final correlation is the sum of all cross-correlations plus the auto-correlation of  $c_0(t)$ , if  $c_0(t)$  is in  $s(t)$ .

*Mapping function.* To derive correlation values for  $\pm 1$  symbols as a function of correlation values with 0/1 symbols, a mapping function needs to be defined [19]. Let  $r$  denote +1 or -1 and  $b$  denote 0 or 1,  $b$  can be written as  $b = \frac{1-r}{2}$ .

*Correlation between a 0/1 signal and  $\pm 1$  symbols.* Let  $x^1 = s_1(t)$  be the received signal at the smart meter, which is composed of  $m$  distinct codes formed with 0/1 symbols. Let us denote each one of this binary codes as  $c_i^b(t)$ . Let us also denote  $y = c_0(t)$  as the code we are looking for, but this code is stored at the database with  $\pm 1$  symbols. The correlation of these two signals can be written as,

$$R(\hat{0})_{s_1 c_0} = \sum_{t=0}^{L-1} s_1(t) \times c_0(t) \quad \text{and} \quad s_1(t) = \sum_{i=1}^m c_i^b(t)$$

$$R(\hat{0})_{s_1 c_0} = \sum_{t=0}^{L-1} \left\{ c_0(t) \times \sum_{i=1}^m c_i^b(t) \right\}$$

Substituting  $c_i^b(t)$  with the mapping  $b = \frac{1-r}{2}$ , we can rewrite  $R(\hat{0})_{s_1 c_0}$  as,

$$R(\hat{0})_{s_1 c_0} = \sum_{t=0}^{L-1} \left\{ c_0(t) \times \sum_{i=1}^m \frac{1 - c_i(t)}{2} \right\} \quad (8)$$

$$R(\hat{0})_{s_1 c_0} = \frac{m}{2} \times \sum_{t=0}^{L-1} c_0(t) - \frac{1}{2} \times \sum_{t=0}^{L-1} \left\{ c_0(t) \times \sum_{i=1}^m c_i(t) \right\} \quad (9)$$

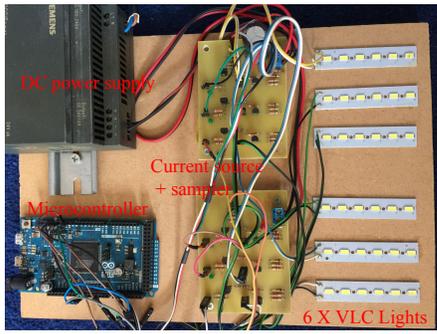


Fig. 2: iLED DC testbed.

*Transformation.* The second term in the above equation represents the correlation coefficient described in (7) and can be rewritten as,

$$\hat{R}(0) = \frac{m}{2} \times \sum_{t=0}^{L-1} c_o(t) - \frac{1}{2} \times R(0) \quad (10)$$

$$R = m \times \sum_{t=0}^{L-1} c_o(t) - 2 \times \hat{R} \quad (11)$$

where,  $m$  is the number of distinct transmitters and the term  $\sum_{t=0}^{L-1} c_o(t)$  represents the sum of the code sequence. For balanced Gold codes, the number of ones equals the number of zeros plus one as described in (IV-B), hence  $\sum_{t=0}^{L-1} c_o(t) = -1$ . Finally we obtain:

$$R = -m - 2 \times \hat{R} \quad (12)$$

Upon calculating the correlation  $\hat{R}$  at the smart meter, we transform it to  $R$ , and only then, we compare the correlation with the threshold  $T$  to determine if an LED is active or not.

## VI. ILED TESTBED

For iLED to work in practice, we need square waves to allow proper superposition of signals when multiple LEDs are modulating. The codes, however, can get distorted due to non-linear elements in DC and AC power lines. In this section we describe our custom hardware to tackle the distortion problem.

### A. DC testbed

Our DC setup includes VLC Lights and an energy meter.

**VLC Light** consists of standard LED lights with modulators. The modulator switches on and off the LED based on the code that is assigned to the light fixture. LEDs are usually powered by connecting a current limiting resistor in series with the voltage source. However, depending on the load, the supplied voltage may fluctuate, which would also cause fluctuations in the current drawn. This causes *distortions* in the generated signal. To eliminate this distortion, we build a current source which delivers a constant current independent of the load. Thus, the resultant current drawn has two distinct values (0 for the LED *off* state and some constant current for the LED *on* state). A signal from a microcontroller is used to toggle the current source for modulating the LED.

**Energy meter:** We developed a low cost energy meter by connecting a resistor in series between the DC power supply and the LEDs. The voltage drop over the resistor is linearly proportional to the current that flows through it and is measured directly using a microcontroller's ADC channel.

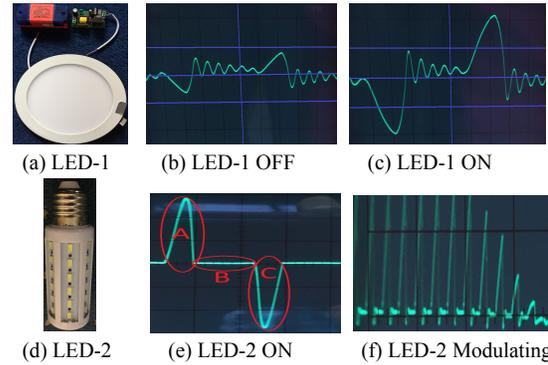


Fig. 3: Commercial LEDs, their current signatures along with the custom built VLC light.

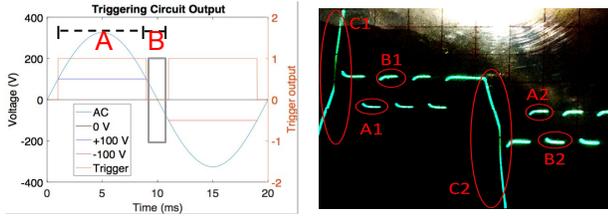
**DC testbed:** Fig. 2 shows the DC testbed. The setup includes six LEDs, the current sources to power them and a microcontroller. The energy meter measures the aggregated current consumption of the LEDs and decodes the received signal to determine the presence of an LED signature.

### B. AC testbed infrastructure

Unlike the DC setting, in AC settings the supplied voltage is a sinusoidal wave. Hence the design of the VLC light and energy meter is not trivial. We first discuss the problems with modulating light and current simultaneously in commercial light bulbs. Then, we describe our custom hardware design to overcome these problems.

**Problems modulating standard LEDs.** Commercial LEDs are powered in several ways for efficiency, (i) by using an external switching mode power supply (SMPS) to convert AC to DC, or (ii) by directly connecting to AC with a built-in rectifier circuit. These two approaches introduce distortions. The SMPS approach (Fig. 3(a-c)) leads to highly distorted waveforms, which cannot be decoded. Fig. 3(b-c) show the current signatures when the LED is *off* and *on*. The second approach has a smoother waveform but has its own set of limitations, c.f. Fig. 3(d-f). First, the time available for modulation is limited, c.f. Fig. 3(e). These type of LEDs work in a duty cycle manner. Regions A and C are the only times the LED is actually operational, which means that the time available for modulation is only 8 ms in a 20 ms period (40% of modulation time). Hence the time taken for an LED to transmit its code would be longer as compared to a DC setting. Second, and more important, the amplitude changes. We cannot obtain the necessary square waves to create superimposing codes. To showcase this point, we modified the internal circuitry of a commercial LED to toggle it by adding two transistors. Fig. 3(f) shows the result of the modulation. We can see the modulation leads to triangular waves with different amplitudes.

**Custom LED modulator.** Due to the distortions caused by the two commercial LEDs analyzed above, we decided to design a custom hardware that includes a *triggering circuit* to indicate when to modulate the LED, a *current source* to provide constant current when the LED state is on, and a *voltage source* to power the triggering circuit without distorting the signals during the modulation periods.



(a) Trigger circuit

(b) Current signature



(c) AC testbed

Fig. 4: iLED AC testbed.

The *triggering circuit* is used to determine when the AC voltage crosses a certain voltage threshold to power an LED and start/stop modulating. In our setup, the trigger setpoint voltage is set to 100 V as shown in Fig. 4(a). It can be seen that the LED's *on* time is roughly 16 ms i.e.,  $\frac{16}{20} = 0.8$ , resulting in 80% of the time for modulation.

A *current source* is used to power the LED such that the current drawn is constant. Fig. 4(b) shows the current signature when an LED is modulating at 500 Hz. It can be seen that, in regions A1 and A2 the current drawn is zero, indicating the LED off state with a '0' symbol encoding. In regions B1 and B2 the LED is on and we can observe a '1' symbol encoding. The current source guarantees that the current drawn by the LED (when it is *on*) is a constant value. A *voltage source* is used to power the triggering circuit and the current source.

**Energy meter:** A high-frequency, high-sensitivity energy meter is needed to detect small current changes. But in this case we need to cope with two additional problems: the voltage can be positive or negative, and we need to detect exactly the periods where modulation occurs without the help of global clocks or external signals. We use a series resistor as current sampler. To overcome the polarity of AC voltage over the resistor (positive and negative values), the output voltage is lifted with an offset before reading it. A triggering circuit is used to identify the modulation regions. The energy meter and lights are all in synch because the sinusoidal voltage in power lines is the same for the entire electrical network and acts as a global clock. Note that high-end commercial energy meters could be used too [17], but they could cost between 300-1000\$.

**AC testbed:** Our AC setup includes 3 VLC lights with commercial LEDs rated at 15 watts. Each LED has its own modulator and microcontroller. The microcontrollers are programmed to generate gold codes of certain length. The energy meter is connected in series to measure the current and decode

the information from the aggregated signal. Fig. 4(c) shows our testbed setup with VLC lights and energy meter.

## VII. EVALUATION

In this section, we describe our experimental evaluation and show the efficacy of iLED w.r.t. accuracy, real-time monitoring, scalability, and the presence of other electric loads.

**Working of iLED.** Consider a legacy building as described in Sec. III, with VLC lights programmed with unique Gold codes of length  $L$ . We do not know the physical locations of the lights, but all the Gold codes are stored in a database, which can be accessed by the energy meter. When a VLC light is turned on it starts transmitting the unique code and the energy meter measures the aggregated signal that contains signatures from multiple lights. The energy meter then computes the correlation values between the measured aggregated signal and each code in the database using the framework presented in Sec. V-B to determine which lights are on/off.

We utilize the following metrics to quantify the performance:

**F-measure** as a comprehensive metric of accuracy combining the precision and recall, which includes the effects of false positives ( $f_p$ ) and false negatives ( $f_n$ ),

$$F = 2 \frac{P \cdot R}{P + R}, \text{ where } P = \frac{t_p}{t_p + f_p} \text{ and } R = \frac{t_p}{t_p + f_n}.$$

**Latency** as the time taken to identify the status of lights.

### A. Numerical results

In Sec. V-A1 we derived bounds for the maximum number of concurrent transmitters, and proposed a probabilistic framework to maintain the number of active ID transmissions below that bound (Sec. V-A2). We will now shed some light about what this theoretical framework means in terms of monitoring LED lights. Table II captures the theoretical performance of our system. A constant modulating frequency of  $10kHz$  is used by all VLC lights.

Consider a building that has 500 LED fixtures and the facility manager would like them to send their codes continuously and simultaneously. To avoid interference problems at the smart meter, we will need to assign each LED a code length of more than two million bits (fifth row on the left side of Table II: *Without Prob. Approach*). This code length would take almost 3.5 minutes to transmit at  $10kHz$ . This is a good polling time, every 3.5 minutes the facility manager can monitor the status of all 500 lights, but an unreasonable time to get just a beacon for many VLC applications. To overcome this problem, we use our probabilistic approach. Now consider the right side of Table II. If a facility manager is willing to tolerate a very small error  $\epsilon = 0.001$ , the code length is reduced significantly (from more than two million bits to 511 bits). The manager would need to wait only 13.24 seconds (on average) to poll lights.

### B. Testbed evaluation

1) *DC testbed:* Our DC testbed includes six LEDs as shown in Fig. 2. Each LED generates a unique code when its *on*. A seventh code is also stored in the database to represent an LED in an *off* state.

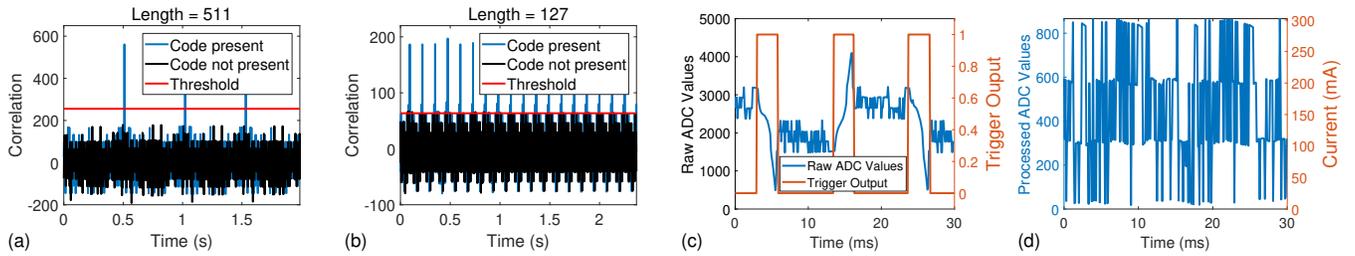


Fig. 5: (a) Correlation values for  $L=511$  in DC testbed (b) Correlation values for  $L=127$  in DC testbed (c) Aggregated current consumption in AC testbed (d) Processed current in AC testbed.

TABLE II: Concurrent transmitters ( $m$ ), code length ( $L$ ) and latency without and with the probabilistic approach.

Without Prob. Approach			With Prob. approach			
$m$	$L$	Latency (s)	$m$	$L$	Latency ( $\epsilon=0.001$ ) ( $\epsilon=0.1$ )	
7	511	0.0511	9	7	6.31	0.06
31	8191	0.8191	33	31	2.22	0.19
63	32767	3.2767	65	63	8.91	0.77
127	131071	13.107	129	127	3.79	0.93
511	2097151	209.72	513	511	13.24	5.62
1023	8388607	838.86	1025	1023	53.10	22.50
2047	33554431	3355.4	2049	2047	65.36	37.64

**Case without interference:** Code length 511, modulation frequency  $1kHz$ . When there are six concurrent VLC lights, a code length of 511 or higher is required to ensure no interference occurs (see Table. I). To determine if an LED is *on*, the energy meter computes the correlation between the LED's ID and the aggregated signal measured at the power meter. If a code is present, the correlation will have peaks of length  $L$ . If the code is not present, the correlation will always be lower than the threshold  $T = L/2$ . Fig. 5(a) shows the correlation values when a code is present and not present (seventh code) in the received aggregated signal. We can see that when the code is present, correlation values have peaks around 511 (blue peaks). When the code is not present the correlation values are always below the threshold ( $T=511/2$ ).

**Case with interference:** Code length 127, modulation frequency  $1kHz$ . To demonstrate the importance of selecting the right code length given  $m$  maximum concurrent transmitters, let's take a code length that is less than 511, say 127, for each of the six LEDs. From our analysis, we can see that only *three* concurrent transmitters should be present so that no interference occurs for this code length (see Table I). However, if all six VLC lights transmit their code simultaneously there will be interference and the accuracy of identifying lights will be reduced. Fig. 5(b) shows the correlation values when a code is present (blue line) and not present (black line) in the received signal. The correlation values for the code that is present have clear peaks crossing the threshold. However, the correlation values of a code that is not present also have instances which cross the threshold (peaks crossing the red line), leading to false positives. This is due to the high number of concurrent transmitters for the chosen code length.

2) *AC testbed:* Our AC testbed includes three LEDs each rated at 15 watts, c.f. Fig. 4(c). We use four gold codes, one

for each of the three LEDs and one to represent an LED in the *off* state. From our analysis, a code of length 127 or higher would avoid interference. A constant modulating frequency of  $10kHz$  is used by all VLC lights. Fig. 5(c) shows the aggregated signal and the corresponding trigger output. To extract the modulated signal, we first filter the aggregated signal when the trigger output is low, and then, normalize the raw ADC values. Fig. 5(d) shows the normalized ADC values and the corresponding current drawn by the three LEDs.

Fig. 6(a) shows the correlation values when a code is present and not present in the aggregated signal. When a code is present, the correlation values have peaks around the code length (127). Since the codes present in the received signal were accurately identified all the time, the F-measure is 1 with no false positives. In both testbeds, AC and DC, the experiments were conducted more than ten times and each run was around five to ten minutes long.

### C. Scalability analysis

To validate the applicability of iLED in buildings with thousands of lights we perform a scalability analysis using simulations. We test the scalability of our approach by employing the probabilistic method described in Sec. V-A2, which dictates that all VLC lights only transmit their codes with a probability  $p$ . Let us consider a building with 129 LEDs, which would require Gold codes of length  $L = 2^n - 1 = 127$ .

*Analyzing the trade-off between  $\epsilon$ , accuracy and polling time.* Fig. 6b shows the accuracy of identifying an LED, when 129 LEDs are transmitting their code sequences of length 127 with  $\epsilon = 0.1$ . The  $x$ -axis indicates the time in seconds, the left  $y$ -axis indicates the number of concurrent transmitters at each time slot, and the right  $y$ -axis indicates the percentage of unique LED IDs that have been successfully transmitted. The dotted line indicates the maximum number of concurrent transmitters such that no interference takes place (for  $L = 127$ ,  $m$  should be 3 c.f. Table I). Fig. 6b also shows the corresponding F-measure. When the number of concurrent transmitters are below or equal to the line, there exists minimal interference and it does not affect the accuracy (F-measure is equal to one, no false positives/negatives). However when the number of concurrent transmitters is higher than  $m$ , depending upon the cross-correlation with other codes, the interference may be high enough to affect the accuracy. In some instances there are no false positives as the cross-correlation between the codes is not high enough to interfere (see time period 0.1 s with

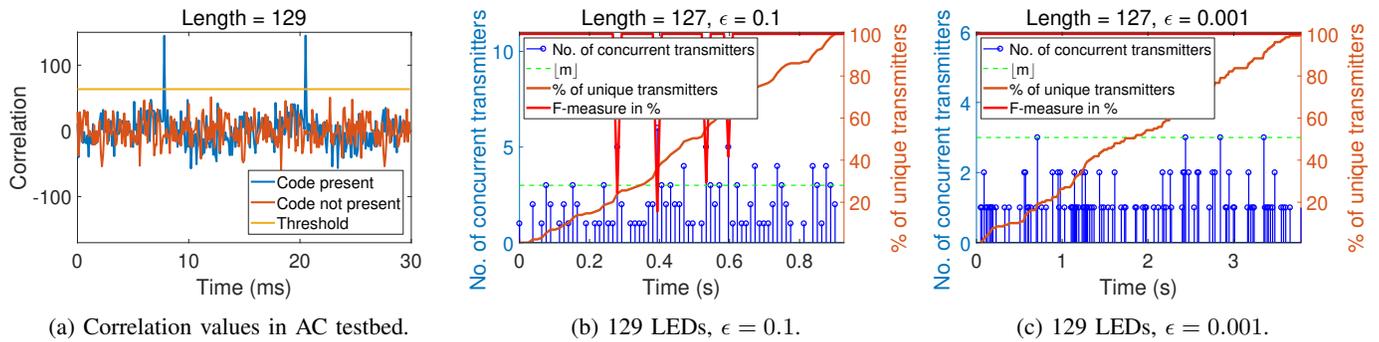


Fig. 6: Scalability analysis.

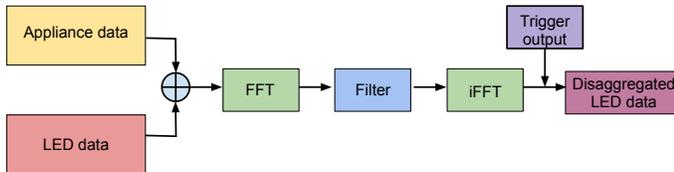


Fig. 7: Disaggregating LED data from combined data.

4 transmitters). In other instances however the interference is high and the F-measure drops (see time period 0.3-0.4 s). Note that the percentage of transmitters that successfully transmit its code at least once increases linearly over time, since in each slot all transmitters have an equal probability of being chosen. In our evaluation, the total time required to identify all 129 LEDs is around 1 second.

Fig. 6c shows the evaluation result when  $\epsilon = 0.001$ . It can be seen that over time, the number of concurrent transmitters is always below the maximum number of concurrent transmitters ( $m=3$ ). Hence the F-measure in this setup is always equal to 1. However, the time required to successfully identify all the LEDs is around 4 seconds. Extensive scalability analysis of iLED has been performed for different building settings with thousands of lights. For example, a building with 2049 LEDs transmit their IDs under 38 seconds when  $\epsilon = 0.1$ .

#### D. Impact of other loads

Until now, we have discussed iLED's performance only in scenarios where lights are present. While some buildings have separate power lines for lighting, households and legacy buildings usually have a single power line. In this section, we use a data-driven approach to describe the impact of other electric loads and show that iLED is still resilient enough to monitor the status of lights. iLED requires high frequency smart meters, which are expensive to deploy. Taking advantage of the fact that current traces can be combined, because power (P) is a linear function of current (I); we use high frequency measurements of several households from open datasets and merge them with the LED traces obtained from our testbed. This combined signal includes both appliances and LED data.

At the core of our method lies the concept of Fast Fourier Transforms (FFT). Fig. 7 shows the overview of our approach. For our explanation we define the following type of traces: (1) LED data, which is the current trace obtained in our testbed

for three lights, (2) Appliance data, which is the aggregated current trace of appliances obtained from the public databases; and (3) Combined data, which merges LED and Appliance data into a single trace.

1) *Combining Appliance and LED data:* There are very few energy datasets that have high frequency smart meter data. We use two datasets *viz.*, REDD(<http://redd.csail.mit.edu>) and UK-DALE(<https://www.doc.ic.ac.uk/~dk3810/data/>), which provide high frequency samples for the total energy consumption of a household, 15kHz and 16kHz respectively. To combine the appliance data from open datasets with our LED data, we ensure two things (i) phase alignment: align the x-axis of both signals with respect to the zero crossing points of voltage; and (ii) polarity alignment: align the y-axis of signals to follow the positive and negative parts of voltage oscillation. We use the data from House 3 of the REDD dataset, which includes energy consumption from diverse set of appliances such as Lighting (6 fixtures), Fridge, Electric furnace, Microwave, Sockets and CE appliances. Fig. 8(a) shows the aligned appliance and LED data.

2) *Filtering Appliance and LED data:* In order to discern the LED data from combined data, we use Fourier transforms. Our method has two main components: determine active frequency and filtering.

*Determine active frequencies.* The goal of this step is to identify the most active frequencies in the appliance data, so we can later filter them out. First, we perform a FFT over the *appliance* data, as shown in Fig 8(b). Any frequency component with an intensity higher than  $\delta$  is recorded in an array. The selection and impact of  $\delta$  in iLED's performance is clarified later.

*Filtering.* The goal of this step is to obtain the LED data from the combined data. First, we perform a FFT over the *combined* data. Then we filter out all the frequencies determined previously by the  $\delta$  parameter. This step will remove some information from the LED data, because some frequencies of the LED and appliance data overlap, but not much information is lost. Finally we apply an inverse FFT (IFFT) to reconstruct the LED data in the time domain, as shown in Fig. 8(c).

3) *Monitoring and decoding accuracy:* From, this step onwards we follow the general iLED framework. The LED data in the time domain is processed using the triggering signals, as shown in Figs.8(c-d). This step has two important

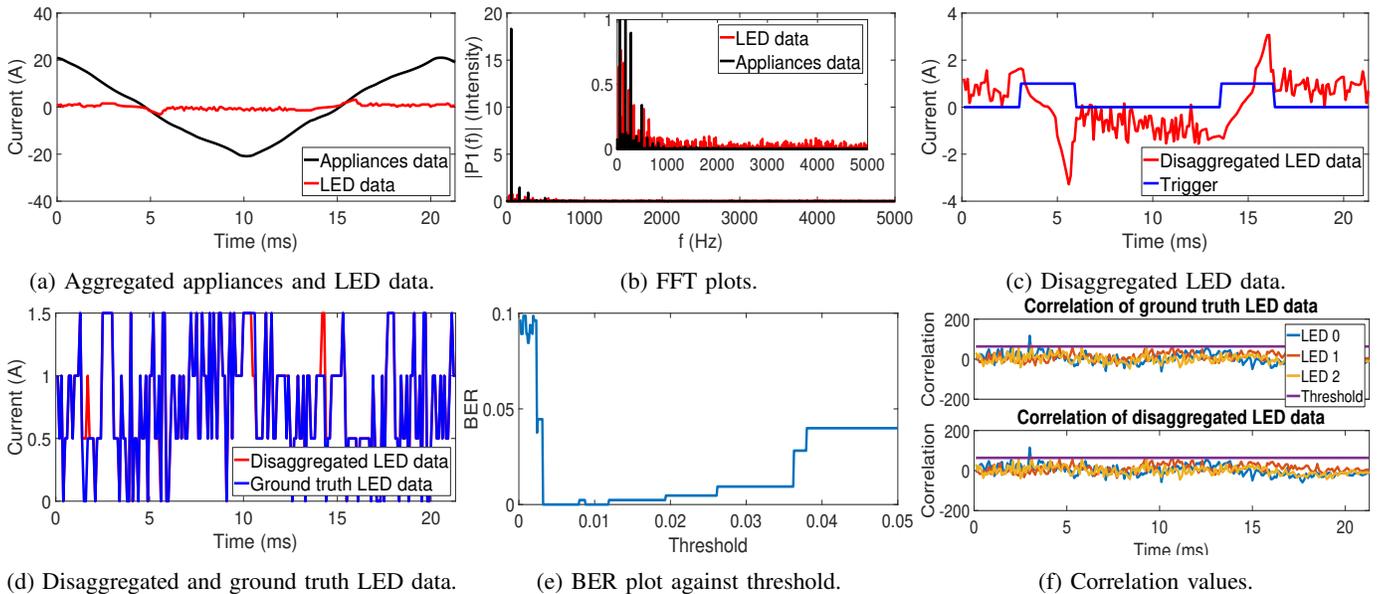


Fig. 8: Disaggregating LED data from appliances data.

results: the decoding process and the correlation process.

*Decoding:* We compare the filtered LED signal with the ground truth LED signal in Fig.8(d). We can see that there are some bit errors in the code. To evaluate the impact of  $\delta$ , we compute bit error rates (BER) across various intensity thresholds, as shown in Fig. 8(e). We can observe that in general BER values are low. Thus, there is a wide range over which the values of  $\delta$  would lead to a good performance.

*Correlation:* While the BER is low, the key question to answer is: *can we accurately identify the status of the LED in spite of these errors?* To this end, we compute the correlation values of the filtered LED data, and for comparison, we also compute the correlations with the ground truth LED data. Fig. 8(f) shows both correlation results. In general, when a code is present in the signal, we can still obtain peaks showing (correctly) that the LED was *on* (even with a few bit errors). The reason for this resilience is that the correlation outcomes are not affected greatly with a few bit errors because correlations are sums over *all bits* in a code. Thus the proposed iLED framework can accurately identify LEDs when other loads are present. We evaluated our approach across other households data in the REDD dataset (i.e., House 5) and other datasets such as UK-DALE. The results show similar trends.

## VIII. CONCLUSIONS

The key contribution of our work is the proposition of a *new tool* to monitor the status of lights via energy disaggregation. At the core of that contribution is the idea of enabling lights to modulate interference-resilient IDs in the *current domain*. Our proposal provides a unique perspective compared to what is available in the state-of-art, because we do not rely on the ‘default’ signatures that light bulbs create, but rather shape them without affecting the main purpose of illumination. iLED is a novel concept, but there is plenty of room for improvement. The next step is to test iLED in a real setting at a larger scale, including the effects of various electric loads.

## REFERENCES

- [1] Trends in lighting in commercial buildings [online] <https://www.eia.gov/consumption/commercial/reports/2012/lighting/>
- [2] Forecast of Energy Savings from LED Lighting [online] <http://energy.gov/eere/ssl/ssl-forecast-report>
- [3] Lighting in supermarkets [online] <http://www.lighting.philips.com/main/cases/cases/food-and-large-retailers/carrefour-lille.html>
- [4] K. C. Armel, A. Gupta, G. Shrimali, A. Albert. Is disaggregation the holy grail of energy efficiency? The case of electricity. *In Energy Policy*.
- [5] S.N.A.U.Nambi, Personalized Energy Services: A Data-Driven Methodology towards Sustainable, Smart Energy Systems, 2017
- [6] Y. Kim, T. Schmid, Z. M. Charbiwala, and M. B. Srivastava. ViridiScope: design and implementation of a fine grained power monitoring system for homes. *In UbiComp*, 2009.
- [7] N. Batra, M. Gulati, P. Jain, K. Whitehouse, and A. Singh. Bits and watts: improving energy disaggregation performance using power line communication modems: poster abstract. *In BuildSys*, 2009.
- [8] F. Englert et al. LightMon: Apportioning the Effect of Light Switching Events on the Electricity Consumption of Buildings. *In EWSN*, 2009.
- [9] S. Gupta, M. S. Reynolds, and S. N. Patel. Electrisense: single-point sensing using emi for electrical event detection and classification in the home. *In UbiComp*, 2010.
- [10] Philips Hue [online] <http://www2.meethue.com/en-in/>
- [11] M. Gulati, S. Sundar Ram, and A. Singh. An in depth study into using emi signatures for appliance identification. *In BuildSys*, 2014.
- [12] Power over Ethernet [online] <http://luxreview.com/article/2016/08/what-nobody-tells-you-about-power-over-ethernet>
- [13] N. Roy et al. Performance of the latest generation powerline networking for green building applications. *In BuildSys*, 2013.
- [14] R. Murty, J. Padhye, R. Chandra, A. R. Chowdhury, and M. Welsh. Characterizing the end-to-end performance of indoor powerline networks. *Microsoft Research*, 2008.
- [15] S. Schmid et al. EnLighting: An Indoor Visible Light Communication System based on Networked Light Bulbs *In SECON*, 2016.
- [16] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: a visible light based positioning system. *In NSDI*, 2014.
- [17] Accuenergy Energy Meter [Online] <https://www.accuenergy.com/product/acudc-dc-power-energy-meter>
- [18] E. H. Dinan and B. Jabbari. Spreading codes for direct sequence cdma and wideband cdma cellular networks. *IEEE Comm. Magazine*, 1998.
- [19] A. Mitra. On pseudo-random and orthogonal binary spreading sequences. *In Journal of Information Techn*, 2008.
- [20] R. Gold. Maximal recursive sequences with 3-valued recursive cross-correlation functions. *In IEEE Trans. on Information Theory*, 1968.
- [21] S. S. Aghaian. Hadamard matrices and their applications, *In Lecture Notes in Mathematics*, 1985.