Mid-term SCDTR project report 2021/2022

Bruno Miguel da Silva Cebola, 93030
MSc in Electrical and Computer Engineering
Instituto Superior Técnico
bruno.m.cebola@tecnico.ulisboa.pt

I. Introduction

With the growing concern for the environment and the surge in electricity prices, during the last decade there has been an increased effort to develop more eco-friendly and energy efficient mechanisms in the most diverse areas. One of these endeavors aims to create more efficient illumination systems, taking advantage of the most recent technologies such as state of the art LED's and new controllers.

This is exactly the focus of this project. The main goal is to develop a system that can ensure the comfort of users at their desks by maintaining a certain amount of light. For testing purposes, this is going to be done in a simulated office space inside a shoe box. During this first part of the project only a standalone LED is going to be controlled with the aim to ensure that all the actions regarding its individual workload are correct before moving into a mesh-like system.

Having in mind the objectives for this project, it also makes sense to have a grasp on the challenges that it poses. Since the goal is to save energy it is mandatory that the control can produce a stable output and is able to respond quickly to external disturbances, being this the presence of external light, for example.

Even though having the ability to handle external factors would be enough to make this problem anything but trivial, it is not the only challenge that needs to be surpassed. Due to the diverse parts that the system is composed of, from the hardware components to the software, there are other problems to have in like: the electronics response time, the way the light is reflected, the micro-controller capabilities, among others.

The methods to overcome the problems listed above can vary depending on the specificity of the imposed constraints. However, there are some key factors that need to be present in virtually any approach. These are: a controller, to handle and generate the control sequences that make the LED's maintain the desired light; a simulator, so that there is a representation of the current state of the system in order to optimize its response; a signal conditioner, to ensure that the control sequence doesn't force the system to try pushing outside its physical constrains.

Through this report, all the methodologies to overcame this challenges and all the details of its implementation are going to be explained, from the basic equations to the actual working system, so that it gets clear how to implement an efficient and energy saving light system.

II. Methodologies

In this section of the report it is going to be explained how the different parts that compose the system work, in a theoretical way. This explanation will cover the underlying mathematical expressions that represent each sub-model and how they are expected to function.

A. Light system

In the ideal office space each desk would have its own light system, which can interact with the surrounding ones, in order to maintain a comfortable amount of light. Such system would be implemented as depicted in Figure 1.

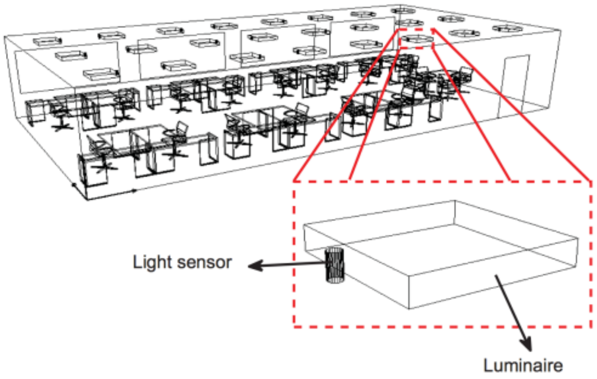


Fig. 1: Light system montage

For a scaled model this can be achieved through the use of a LED for the luminaire and a LDR for the light sensor. The necessary circuit for this mechanism to be implemented is shown in Figure 2.

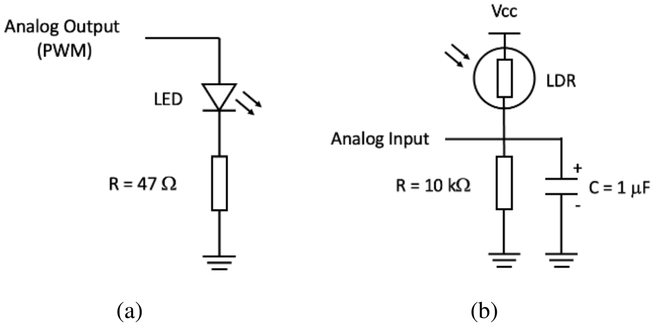


Fig. 2: Luminaire: (a) LED and (b) LDR

Due to the configuration of the montage, where the LED and LDR are placed next to each other on the upper part of the office model, what is perceived by the light sensor is the reflected light and not the emitted one, as shown in Figure 3. However, it can be considered that these values are pretty much the same. This two component montage is in its essence a luxmeter.

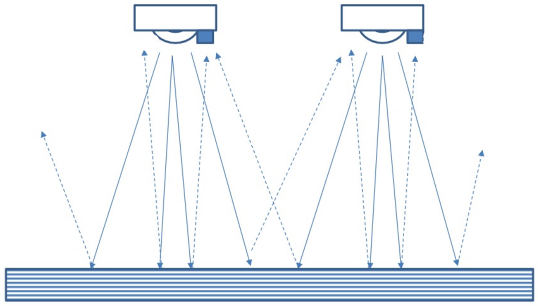


Fig. 3: Light path

Even though the reference measurement is the lux (quantity of light present in the office), the variable to be controlled is the voltage placed on the terminals of the LED. This way, it is important to define a relation between this two measurements, as done in (1), where $l(t)$ is the lux, $u(t)$ is the voltage, G is the gain and $o(t)$ is the component that accounts for external disturbances (which are 0 in a perfect environment).

$$l(t) = Gu(t) + o(t) \tag{1}$$

It is important to note that the gain G of the system depends on the hardware itself and on the model of the office and due to this it has to be calibrated before each iteration, as it will be explained later.

At this stage, it also makes sense o explain that the LDR is a components that changes its resistance based on the amount of light that receives. Due to this, it is necessary to define the equation that maps the relation between the voltage on the terminals of the LDR to its resistance (2), which is based on kirchhoff's laws for voltage applyed to the circuit in Figure 2.

$$R_{LDR} = R * (\frac{V_{cc}}{V_{LDR}} - 1) \tag{2}$$

Having the LDR's resistance it is possible to obtain is relation with the amount of light presented in the office (3).

$$L = 10^{\frac{log_{10}(R_{LDR})-b}{m}} \tag{3}$$

Finally, the constants m and b need to set so that this relationship can be as linear as possible so that it can then be expressed as in (1).

B. General controll architecture

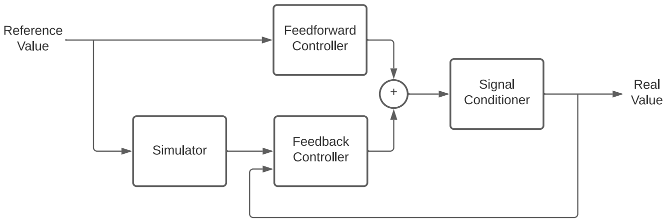


Fig. 4: Controller architecture

In Section II-A it was explained the physical working of the light system, but this would be useless without a proper controller to make sure that there is a steady amount of light in the office.

As shown in the diagram of Figure 4, the architecture of the solution for this project's goal - energy saving light system - is very simple, consisting of four distinct modules: a feedforward controller, a feedback controller, a simulator and a signal conditioner. All of these blocks work together in a closed-loop sequence to ensure the stability of the system and none is more important than the other, as all of them have very specific roles.

Starting with the feedforward controller, this has the role to take the system to meet the reference value by continuously pushing in that direction. It does not look at the current state of the system as it only has the reference value has goal.

The feedback controller, similarly, also tries to move the system to reach the reference value. However, it does this by trying to reduce the error between the current state and the predicted one, generated by the simulator.

This way, the simulator is responsible for generating trustworthy representations of the state of the system at any given time in order for the feedback controller to take them as the reference input.

By combining these three blocks it is possible to generate the control sequence needed to stabilize the system and make it respond to disturbances. Nevertheless, since all the devices have physical restrains it is necessary to introduce a block to condition this control signal so that this limits are not broken. It is due to this limitations that the response is neither immediate or equal to the simulated one.

C. Simulator

As briefly explained, the simulator aims to create a reliable representation of the state of the system. In order to understand its underlings its first necessary to understand exactly how the system works.

The actual response of a real system is not immediate as one would want. Instead, there is an initial response time, τ, due to the capacitance's of the different electronics present in the system. This time constants is not always the same as it depends on the final desired value.

In order to accurately simulate the response of the system at any given time it is necessary to know this time constants, the initial and final values of the variable to be controlled (voltage at the LED's terminals) and the instant to simulate. The way the system evolves is given by (4).

$$v(t) = v_f - (v_f - v_i) \cdot \exp - \frac{t - t_i}{\tau(x_f)} \tag{4}$$

In Figure 5 is possible to see a comparison between the perfect response and the simulated one, for a final tension of 1V with a τ of 0.5s.

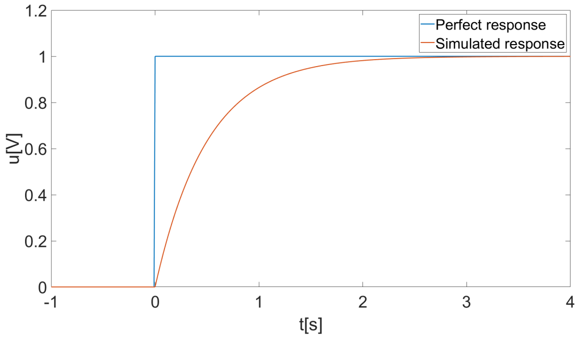


Fig. 5: Simulated output

D. Feedback controller

It makes sense that the next block to be explained is the feedback controller as it takes the output of the simulator as one of its inputs. This is also the most complex block of the system, because by trying to compensate the final value based on an error there is a more intricate structure to ensure a reliable output.

The chosen controller architecture was a PI controller, like the own depicted in Figure 6. Although there are other options, this sort of controller is perfect for our situation since it can counteract the effects of static errors, due to its proportional part, and respond to quick variations, because of the integrator that sums the error across time.

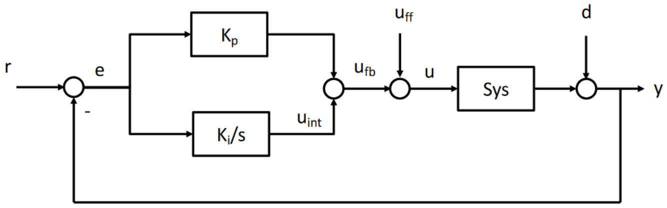


Fig. 6: Proportional Integral Controller

The output of this controller is then the sum of the proportional and integrative parts as expressed in (5), where K_p and K_i are respectively the proportional and integrative gains. These gains could be computed through complex equations so that they could be as perfect as possible, however due to the practical nature of this project (which implies the existence of unpredictable circumstances) it is preferable to determine this gains by trial and error.

$$u_{fb}(t) = K_p e(t) + K_i \int_{t_0}^{t} e(\lambda)\, d\lambda \tag{5}$$

Due to the integrative part of the controller it is possible to track the error through time by adjusting u_{int}. This grants the possibility to continuously having a representation of the error, which in normal situations is beneficial. However, when the system cannot counteract this error there is a new problem that arises: buffer over-saturation or windup.

Lets take a more practical explanation to understand this problem. When in perfect conditions, the only source of light observed by the luxmeter if the LED, which means that if the reading is not correct the only thing that needs to be done is changing the dimming of the LED to make it more or less intense.

Now, if external light can reach the office, the system will try to compensate it by reducing the light emitted by the LED until an extreme point where the LED will eventually turn off. However, since it is not possible to remove light from the system, if the external lighting is superior to the desired value, the integral part of the controller will continue to sum this error. Due to this, if the said external lighting is removed, the system instead of going immediately to the reference lux value it will have to wait a certain amount of time until the integrative buffer goes back to normal values and starts working properly again.

To solve this problem it is necessary to implement a anti-windup mechanism to ensure that if the system is put under such circumstances the buffer instead of continuing to increased it stays on a fixed level.

E. Feedforward controller

Contrary to the functioning of the feedback controller, the feedforward only tries to move the system in the right direction by analysing the reference value, making this controller much simpler.

This controller performs only a single operation (6) to force the exit of the system to be equal to needed voltage for the reference lux. Although this may seem useless, this controller actually helps to speed the process of reaching the desired value.

$$u_{ff} = \frac{lux_{ref}}{G} \tag{6}$$

F. Signal conditioner

As depicted in Figure 4, the outputs of the two controllers (feedforward and feedback) are summed together to get the final control signal.

The problem is that neither of the controllers have in consideration the physical system, as they only focus on mathematical expression. This makes necessary to insert a block that can condition the signal to meet the physical constrains. The control signal is then given by (7).

$$u_{cont} = \begin{cases} 0, \; u_{ff} + u_{fb} < 0 \\ V_{cc}, \; u_{ff} + u_{fb} > V_{cc} \\ u_{ff} + u_{fb}, \; otherwise \end{cases} \tag{7}$$

This sort of output conditioning will cause the system not to respond as fast as theoretically possible, but will ensure that the response it stable and as quick as the hardware allows.

III. Experimental setup

In this section of the report it is going to be analyzed how the project was taken from theory and put to practice. It will be explained how the different modules were implemented, how the office models works as a whole and which were done to improve the stability of the system.

A. *Office model*

As already explained, in order to test this light controlling mechanism, a model of a real office was created. Due to its similar shape, a shoe box was the best alternative to this implementation. To be as similar as possible to a real office, the lighting system (LED and LDR) was attached to the box cover, as shown in Figure 7, and all the electronics were placed outside of the box, as it can be seen in Figure 8.

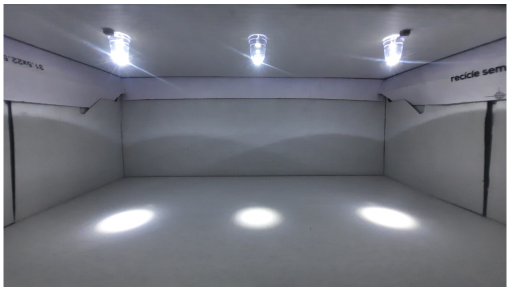


Fig. 7: Office model interior - lighting system

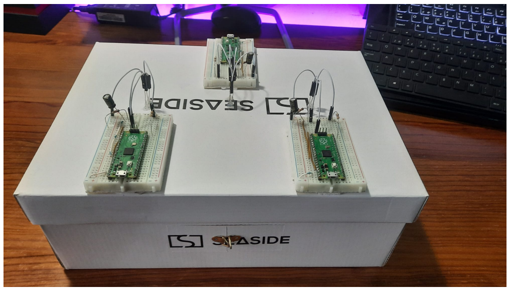


Fig. 8: Office model exterior - Electronics

This model allows to meet all the requirements need to properly tests this mechanism. Te box is white providing the best reflection possible, its fully empty and so almost all the light can be collected, when closed it isolates all exterior light and it has a small windows to allow the entry of outside light. However, there are also some problems, because since the box is not perfect the reflection path is not as linear as it should and some light will be absorbed. Nevertheless these are small problems that do not need to be taken into account for this project.

On the box cover is attached the circuit from Figure 2 controlled using a Raspberry Pico. It is important to mention that due to the characteristics of this micro-controller the signal that are either read or written to its analog pins are integers from 0 to 4096, which means that the LED is powered using a PWM wave, whose duty cycle defines the intensity of the light.

B. *Relation between duty cycle and lux*

As explained in Section II-A, the first constants that must be determined are m and b so that the relations between the different components can be defined.

The graphic in Figure 9 show the desired relation between the LDR's resistance and the lux. With this it is possible to set $b = 6.2$ so that the expression that models the relation can fall in the middle of the possible area (highlighted in grey).

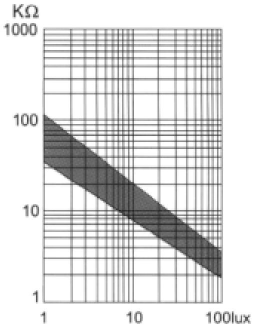


Fig. 9: Relations between lux and LDR's resistance

Next, to determine m is necessary to perform incremental steps on the intensity of the LED and register the value read by the luxmeter. The values in Table I refer to the linear regression squared error, R^2, for different values of m.

Table I: Value read by the luxmeter for incremental steps

m	R^2
-0.7	0.985
-0.8	0.993
-0.9	0.997

For $m = -0.9$ it is possible to obtain the most linear relation between the voltage at the LED's terminal and the lux, as it can be seen in Figure 10.

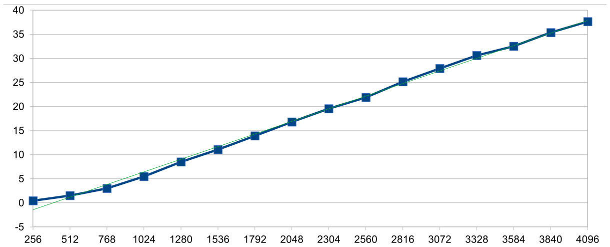


Fig. 10: Relations between LED's voltage and lux

By applying a linear regression to the data points it is possible to determine the equations that best defines this line and that can be mapped into (1). The slope of this line would be the best value for G. However, as explained previously, this value can change between experiments and therefor must be computed at the beginning of each new iteration.

In order to do this, since its a linear relation, it is only needed to take two points (one for low voltage and one for high voltage) and compute the slope of the line that unites them. Whit this value calculated is possible to establish the relation between the duty cycle of the LED and its lux.

C. *Tau in function of duty cycle*

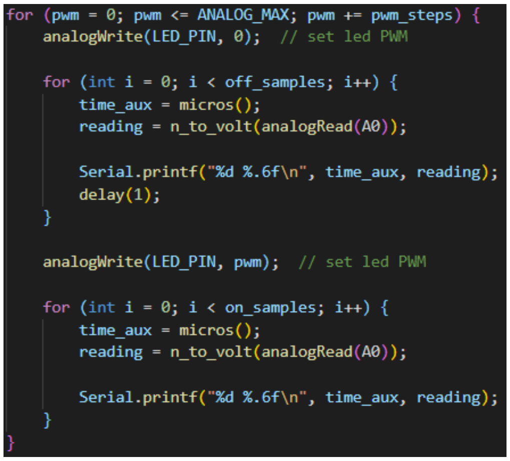
The only other variable that need to be modeled before starting the system is the time constant, τ, which appears due to the condenser connected to the LDR. If this element was not to be present, the response would be immediate, but would not steady because as the LED is powered by a PWM the LDR's resistance would change with the same frequency as the PWM.

Even though this component is connected to the LDR, it is possible to establish a relation between the time constant that it introduces in the system and the voltage at the terminal of the LED. This happens because there is a direct mapping between what happens in the luxmeter and the LED.

Theoretically this value would be given by (8) if only the condenser's capacitance was at stake. However this is not the case, because all of the electronics present in the system contribute to this time constant, even if its only in a small fraction.

$$\tau(x) = R_{eq}(x) \cdot C, R_{eq}(x) = \frac{R \cdot R_{LDR}(x)}{R + R_{LDR}(x)} \tag{8}$$

Due to this, it is needed to perform some computation *a priori* to using the system, in order to find how this constants relates with the voltage supplied to the LED. To do this it is necessary to perform a series of readings *per* LED step that dist only in microseconds from each other, as shown in the code in Figure 11.



```
for (pwm = 0; pwm <= ANALOG_MAX; pwm += pwm_steps) {
    analogWrite(LED_PIN, 0);  // set led PWM

    for (int i = 0; i < off_samples; i++) {
        time_aux = micros();
        reading = n_to_volt(analogRead(A0));

        Serial.printf("%d %.6f\n", time_aux, reading);
        delay(1);
    }

    analogWrite(LED_PIN, pwm);  // set led PWM

    for (int i = 0; i < on_samples; i++) {
        time_aux = micros();
        reading = n_to_volt(analogRead(A0));

        Serial.printf("%d %.6f\n", time_aux, reading);
    }
}
```

Fig. 11: Code to read multiple values *per* LED step

By doing this multiple readings is possible to obtain the graph depicted in Figure 12. Due to the components not being in their best conditions and the office model not being perfect is visible that the readings are very noisy. This way, by doing a moving average, is possible to clear the data, whih can now be processed.

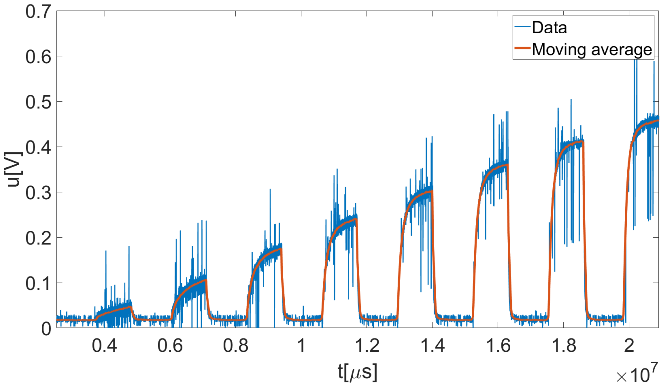


Fig. 12: Evolution of the voltage on the LDR for different LED steps

Through a Matlab script it is possible to analyse this data and find an equation that computes the values for τ across the different levels. This equation is displayed in (9) and its graphical representation along with the experimental $\tau's$ can be seen in Figure 13.

$$\begin{aligned}\tau(n) = 0.39 + 8.1 \cdot 10^{-4}n - 1.86 \cdot 10^{-6}n^2 + \\ +1.39 \cdot 10^{-9}n^3 - 4.95 \cdot 10^{-13}n^4 + \\ +8.58 \cdot 10^{-17}n^5 - 5.82 \cdot 10^{-21}n^6\end{aligned} \tag{9}$$

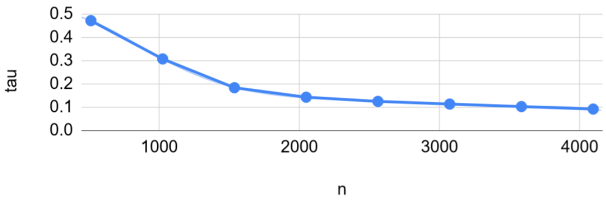


Fig. 13: Relation between τ and LED level

As expected, with the increase of the voltage at terminals of the LED the time constant gets incrementally lower, which means that the more abrupt the change the faster the system will be able to respond.

D. *Feedforward controller only*

Now that all the constants that are needed for the system are defined, let's start analysing each block and putting them together.

In a first instance lets use only the feedforward controller, as it is the simpler control mechanism to be implemented. In Figure 14 it is possible to visualize that the control signal is equal to the reference signal, which was expected due to its expression (6). When in perfect conditions (no external disturbances) this controller is able to lead the system to a correct value in the fastest way, as shown in Figure 15.

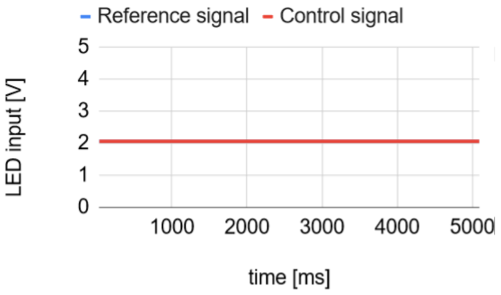


Fig. 14: Feedforward controller only - control signal

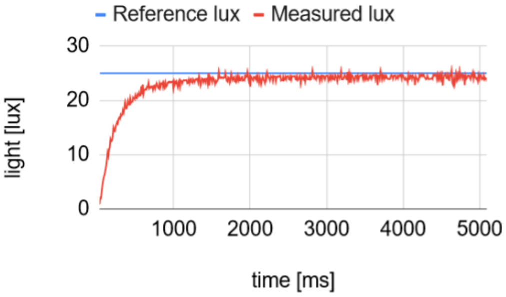


Fig. 15: Feedforward controller only - measured lux

However, when external disturbances are added, as shown in Figure 17, this controller fails to produce any change in the control sequence to react to this modifications, as it is depicted in Figure 16.

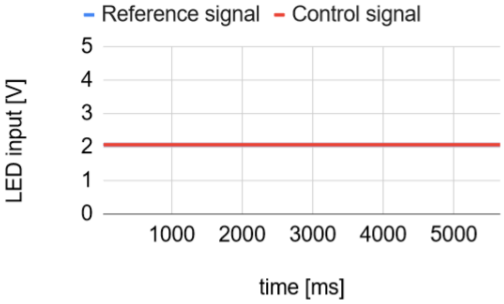


Fig. 16: Feedforward controller only with external disturbances - control signal

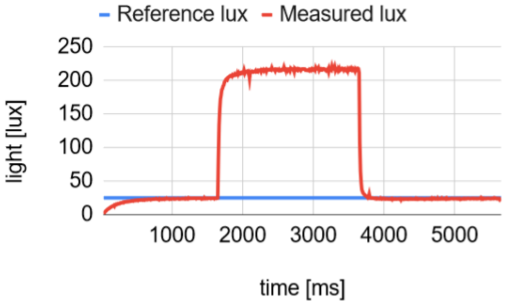


Fig. 17: Feedforward controller only with external disturbances - measured lux

E. *Simulator in action*

In order to correct the errors related to external disturbances to the system it is necessary to use a feedback controller. None the less, to do so, a working simulator is crucial and that is what will be analyzed in this part.

To properly assess the simulator, no other component of the controller (feedback or feedforward) will be put into action. In Figure 18 it is possible to see that even though the real response isn't exactly equal to the simulated one, it is very similar.

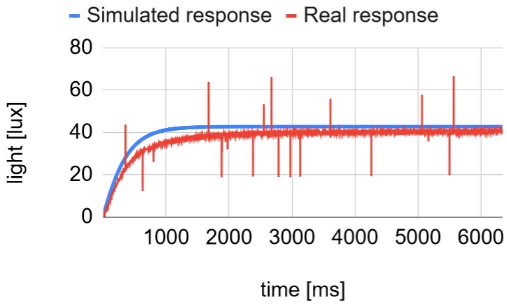


Fig. 18: Simulated response vs real response

This response was simulated using (4), where v_f and v_i are computed at run time and τ is computed using (9). Is this way possible to conclude that although the output isn't a perfect match, this simulation can create a trustworthy representation of the desired state of the system.

F. *Feedback controller + simulator*

Having the simulator working, it is now possible to use it in conjunction with the feedback controller.

As already explained, this controller has a PI structure which means that there are two different parts to be aware off. Nevertheless, since it is not possible in this project to deactivate only one of them it is enough to analyze it as a whole.

Similarly to the feedforward controller, when there are no external disturbances this controller is able to produce an acceptable output, as it can be seen in Figure 20. However, to do so it takes muck longer and the final value isn't as accurate, as shown in Figure 19.

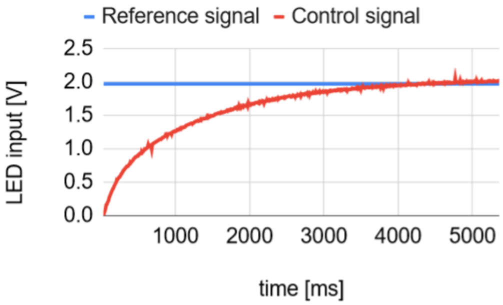


Fig. 19: Feedback controller only - control signal

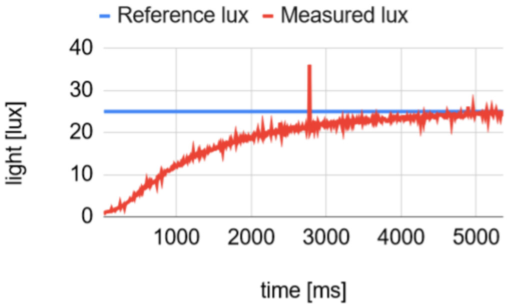


Fig. 20: Feedback controller only - measured lux

This controller gets interesting when we turn to the case were external disturbances are introduced into the system. In this case, as it is displayed in Figure 21, the system tries to correct the extra amount of light by turning the LED off. This is only possible due to the feedback and comparison to the expected value, provided by the simulator.

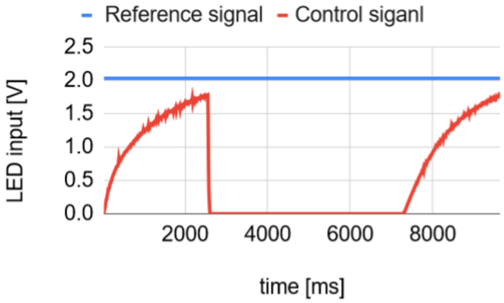


Fig. 21: Feedback controller only with external disturbances - control signal

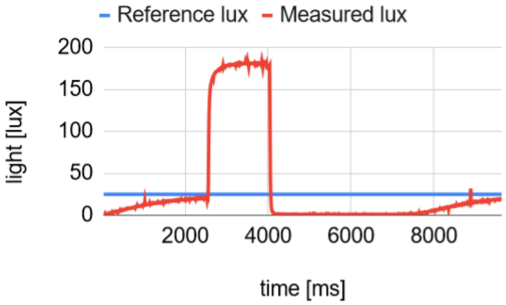


Fig. 22: Feedback controller only with external disturbances - measured lux

Nevertheless, as already explained in Section II-D, without an anti-windup mechanism, when the external disturbances are remove the system cannot response immediately as the integrative buffer is over-saturated. This effect can be analyzed by comparing the time intervals from the measured lux in Figure 22 and the control signal in Figure 21.

G. Feedback and feedforward controller working together

By combining both controller it is possible to take advantages of the characteristics of both: speed from the feedforward controller and stability from the feedback controller.

Again both situations were tested: without and with external disturbances. When in perfect conditions, the system is able to reach the reference lux as it can be seen in Figure 24. It is also possible to see that this did not happen as fast as in the feedforward controller, but happened faster than in the feedback one. By analysing Figure 23 it is also possible to conclude that the control signal is not equal to the reference one, but it is very similar right from the start.

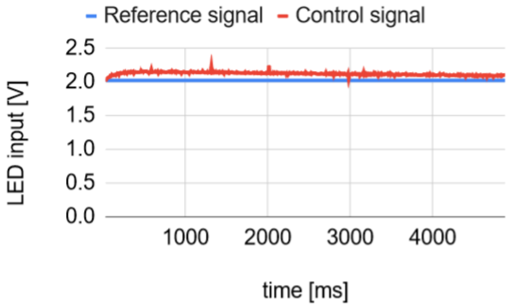


Fig. 23: Feedback controller + Feedforward controller - control signal

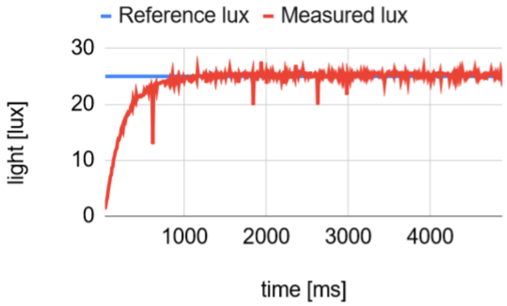


Fig. 24: Feedback controller + Feedforward controller - measured lux

When moving to the case were there are external disturbances, it is possible to see in Figure 25 that the LED gets turned off to counteract the effects of the excess light, as it is displayed in Figure 26. However, since there is still no anti-windup, the response remains not immediate, as it happened with the feedback controller.

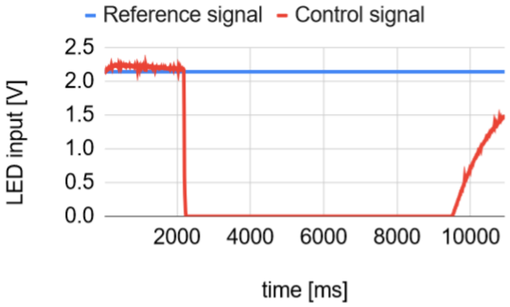


Fig. 25: Feedback controller + Feedforward controller with external disturbances - control signal

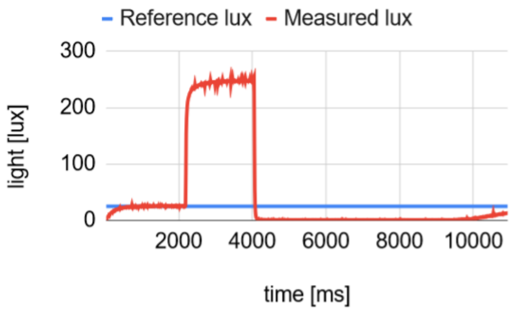
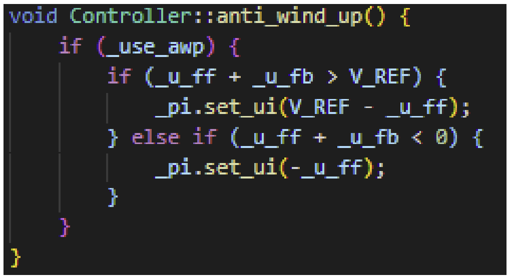


Fig. 26: Feedback controller + Feedforward controller with external disturbances - measured lux

H. *Introduction of an anti-windup mechanism*

The basic idea behind an anti-windup mechanism lies on either keeping the buffer at the steady value when it reaches a peak or minimum or to reset it and start over in a loop.

In Figure 27 it is possible to see the implementation of said mechanism. When the needed control signal surpasses the maximum allowed value (V_{cc}) the buffer is forced to stay in this value or below if the feedforward controller is in action. On the other hand, if the needed control signal is negative then the buffer is forced to also become negative or zero, again depending on either the feedforward controller is in action or not.



```
void Controller::anti_wind_up() {
    if (_use_awp) {
        if (_u_ff + _u_fb > V_REF) {
            _pi.set_ui(V_REF - _u_ff);
        } else if (_u_ff + _u_fb < 0) {
            _pi.set_ui(-_u_ff);
        }
    }
}
```

Fig. 27: Anti-windup code snippet

AS it would be expected, when there are no external disturbances the anti-windup mechanism does not change the functioning of the system. This way it is possible to see that Figures 28 and 29 are almost identical to Figures 23 and 24, respectively.

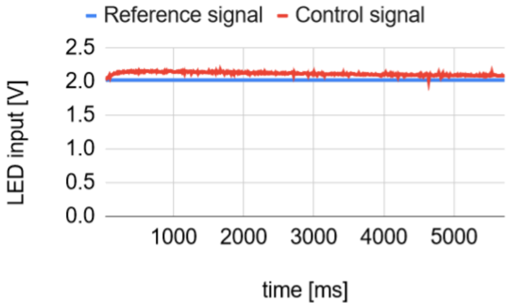


Fig. 28: Full control architecture - control signal

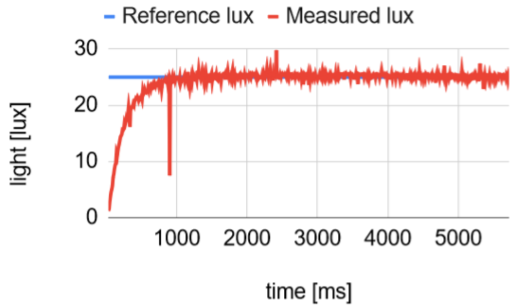


Fig. 29: Full control architecture - measured lux

However, this time when external disturbances are added to the system and the suddenly removed, Figure 31, the system can respond instantly in order to maintain the desired level of light inside the office space. This is possible to see in Figure 28 where the control signal becomes zero when external disturbances are added and start to increase as sudden as this disturbances are removed.

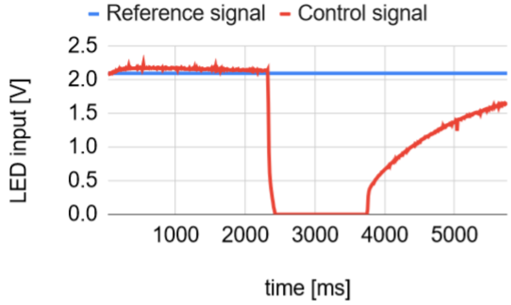


Fig. 30: Full control architecture with external disturbances - control signal

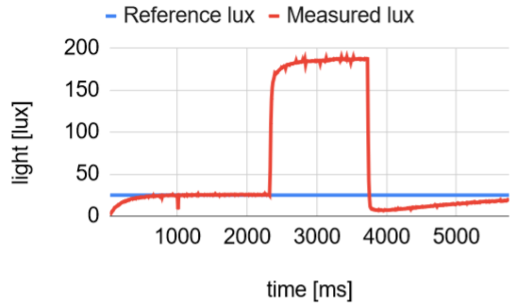


Fig. 31: Full control architecture with external disturbances - measured lux

I. *Interface with the computer*

In previous sections it was explained how each controller affects the system trough the use of images that displayed the system responses. In order to achieve this level of specificity, the blocks to be used were hard-coded. However, during a real execution of the system it is also important to be able to control which mechanisms are put into action and which are not. Furthermore, it is also important to be able to collect some information about the status of the system In order to do this a simple interface was implemented.

In Table II are listed a set of commands available to interact with the system.

Table II: Some of the available interface commands

Command	Response	Description
r i val	”ack” or ”err”	Set illuminance ref at luminaire i
g r i	r i val	Get illuminance ref at luminaire i
w i val	”ack” or ”err”	Set feedforward usage at desk i
b i val	”ack” or ”err”	Set feedback usage at desk i
a i val	”ack” or ”err”	Set anti-windup usage at desk i
b x i	b x i val_1, ..., val_n	Get last minute of data on var x

From the above commands it is important two mention the ’w i val’, ’b i val’ and ’a i val’, since are the one that can control which blocks of the system are used and therefor directly impact its functioning. Its is also relevant to mention that it is possible to obtain the last minute of data either for the measured lux or for the the LED duty cycle through the use of the command ’b x i’, where x can be ’l’ or ’d’ accordingly. It is also possible to live stream these two variables.

The above commands are the ones that allow to control the system and get information about its state, but the main question of this project is yet to be answered: does this solution work? In order to answer, it is necessary to have metrics that allow to understand if this sort of system is worth to implement, these being: the energy consumption, the visibility error and the flicker error. In Table III are listed the three commands available to get the needed measurements.

Table III: Commands to get the base metrics

Command	Response	Description
g e i	e i val	Get accumulated energy consumption at desk i
g v i	v i val	Get accumulated visibility error at desk i
g f i	f i val	Get accumulated flicker error at desk i

To finally it only remains to explained how these values are computed. The accumulated energy consumption corresponds to the sum of the energy spent by the LED across time. This can be expressed as (10), where $t_n - t_{n-1}$ is the sampling interval, d_{n-1} is the LED duty cycle during that interval and P_i is the LED’s nominal power.

$$E_i = P_i \sum_{n=1}^{N} dn - 1(t_n - t_{n-1}) \tag{10}$$

The visibility error can be defined as the average error between the reference illuminance, $L(t_i)$ and the measured illuminance, $l(t_i)$ for the periods when the measured illuminance is below the reference. The mathematical expression for this metric is given by(11).

$$V = \frac{1}{N} \sum_{n=1}^{N} max(0, L(t_i) - l(t_i)) \tag{11}$$

Finally, the last metric missing to explain is the accumulated flicker error. This measurement allows to perceive if the system is able to produce a steady output or if the LED is changing very frequently in intensity. With resort to (12) it is possible to calculate the instantaneous flicker and by summing all the instantaneous values it is possible to obtain the total flicker error (13.

$$f_n = \begin{cases} (|l_n - l_{n-1}| + |l_{n-1} - l_{n-2}|)/(2T_s) \\ \quad if\,(l_n - l_{n-1}) \cdot (l_{n-1} - l_{n-2}) < 0 \\ 0,\, otherwise \end{cases} \tag{12}$$

$$F = \frac{1}{N} \sum_{n=1}^{N} f_n \tag{13}$$

IV. Conclusions

At this middle stage of the project, it is safe to conclude that a solution like the one developed so far can be used to control the light system of an office space. This happen because the controller architecture is capable to react to diverse disturbances and maintain a quality level of light.

However, for as many upper points that this solution has, there are also negative ones. The most notorious is regarding the fact that currently only one LED is being controlled, which means that if other LED’s are to be placed in the system they will be perceived as external disturbances instead of others systems to work and cooperate with.

The next part of the project will aim towards creating this mesh like system so that the energy and power of this control architecture can be taken to its maximum, while taking advantage of the lessons learnt during the development of a standalone LED controller.

References

[1] CDS Light-Dependent Photoresistors
https://datasheetspdf.com/pdf-file/711172/Token/PGM5659D/1

[2] Distributed Real-Time Control Systems 2021/2022 course slides
https://fenix.tecnico.ulisboa.pt/disciplinas/SCDTR/2021-2022/2-semestre/slides