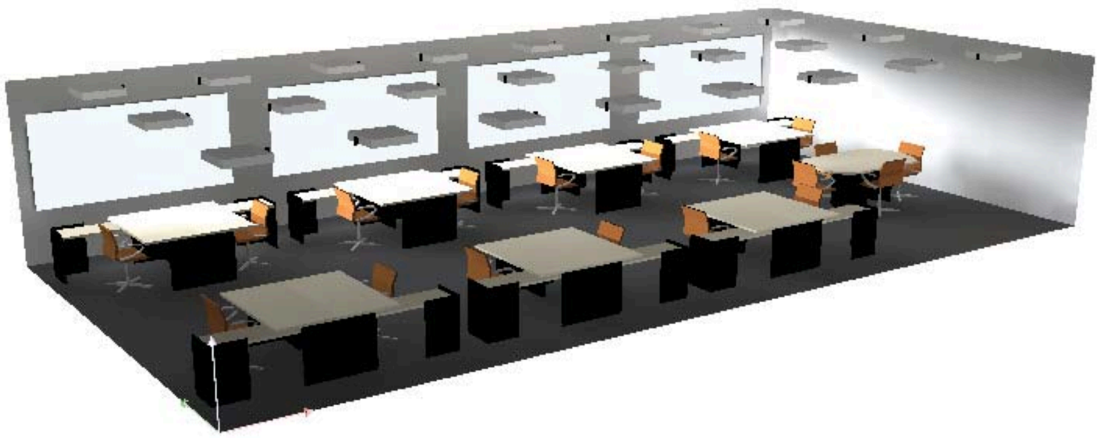
Master Degree in Electrical and Computer Engineering
2015/2016 – Winter Semester

Distributed Real-Time Control Systems
(Sistemas de Controlo Distribuído em Tempo-Real)

PROJECT

Distributed Lighting Control

Details for the 2nd Stage



Prepared by

Alexandre Bernardino

José Gaspar

Instituto Superior Técnico

Department of Electrical and Computer Engineering

Scientific Area of Systems, Decision and Control

1. Introduction

This document provides a few more details regarding the implementation of the 2nd stage of the project. It is organized as follows. First it presents the description of the 2nd stage contained in the original document (Section 2). Then it provides some more detailed information regarding the control network (Section 3), the evaluation metrics (Section 4), the global control (Section 5), the calibration procedure (Section 6), the C++ server (Section 7) and the final demonstration (Section 8).

2. Second Stage (original description)

During the second stage the students will create a control network (I2C or SPI) to allow communication between the luminaires, define the evaluation metrics (energy consumption, user comfort), and develop the global coordinated control. When a luminaire has to change its reference value due to change in its occupation status, it should broadcast this information to the neighbour luminaires so they can compensate quickly for the change. To realize the compensation, it is necessary to model the effects of a change in one luminaire in the measured illuminance of the other. Because this interference between luminaires depends on many factors, a calibration procedure should be made at system startup. For example, turn on one luminaire at each time and measure the illuminance in all other luminaires. The global controller can use either a centralized method (basic objective) or a distributed method (advanced objective). In centralized control, a master luminaire has access to the information of all luminaires (luminance and dimming levels), computes the global optimal compensation values, and forwards to the other luminaires. In decentralized/distributed control, each luminaire can communicate only with its neighbours but no central master exists. A PC will be connected via serial port to one of the luminaires. This luminaire will operate as a router of information between the PC and the other luminaires, coordinate the calibration procedure, and serve as master controller in the centralized control mode. A C++ server with serial and socket communications will run on this PC to allow remote access by a client PC to the state of the system. The client PC should be able to read the state of any luminaire (occupation, led dimming level, illuminance level), set values to the system (occupancy, led dimming level, redefine the reference levels HIGH and LOW), and access system statistics (energy consumption, comfort metrics). The information can be requested in three different modes: actual values, last minute history or real-time stream.

This part will be evaluated through a demonstration of the final distributed control system in the end of the semester (18th December).

A written report containing both parts, and associated software, will be delivered two weeks after the final demonstration until (3rd January, 2016).

3. The Control Network

In principle, the most suited control network for the given setup is the I2C bus, where all nodes can connect and exchange information. This is the alternative that provides the greatest flexibility while minimizing the wiring. Despite the default use of I2C is one-master-multiple-slaves, it can be also used as multi master, where any node can take the initiative of communicating to its neighbours whenever necessary. However, the Wire library for I2C provided with the Arduino has some caveats, in particular it blocks in the writing stage, which stops all other ongoing processing. This is a problem of the Wire library and not of the Arduino itself since the ATMega328 I2C hardware implementation is non-blocking and based on interrupts. There are several alternatives to address this problem:

1- Reduce the PID control loop sampling frequency so that I2C communications have enough time to send the data.

2- Stop the PID control during I2C communications. This is feasible in our system since it is a stable system, so holding the control value is not prejudicial in terms of stability.

3- Implement the I2C communications without the Wire library.

4- Implement communications with SPI (fast but does not allow multiple master operation) or Software Serial (also slow).

Any of these alternatives, or other proposed by the students, are accepted provided that they are properly justified, identifying its advantages and limitations.

It is recommended to exploit all functionalities provided by the communication functions through proper configuration, in order to minimize the communication times, e.g. by selecting appropriate baud rates for the existing cable lengths and define messages with short lengths. The final report should include an analysis of the times taken in the communications.

4. Evaluation Metrics

To properly validate an engineering solution, it is fundamental to define appropriate evaluation metrics, expressing in a quantitative way the requirements addressed in its formulation. For our system we target at minimizing the energy spent in illumination while providing comfort to the users. This is the integral of the instantaneous power along time. Let us assume a nominal (fake) value for the maximum power of the luminaire as 1W. Then, a formula to compute the energy consumed at each desk is:

$$E = \sum_{i=2}^{N} d_{i-1}\left(t_{i-1} - t_i\right)$$

where *i* is the index of the control samples, t_i are the sample times and d_i is the led duty cycle value (between 0 and 1) at sample time t_i. The units of this metric are Joule [J].

While energy minimization is simple to formulate, the comfort criteria is more subjective. We can consider the following rules:

- The system should prevent periods of illumination below the minimum settings defined by an occupation state. A metric to assess this criterion can be defined as the average error between the reference illuminance (l_{ref}) and the measured illuminance (l_{meas}) for the periods when the measured illuminance is below the reference. Let us call this quantity the **Comfort Error**:

  $$C_{error} = \frac{1}{N} \sum_{i=1}^{N} \max(l_{ref}(t_i) - l_{meas}(t_i), 0)$$

  where N is the total number of samples used to compute the metric and t_i are the sampling times.

  The previous expression refers to a single desk. The total average error should be computed as the sum of the average errors at each desk. The units of this metric are [lux].

- The system should prevent sudden variations in the illuminance (flickering) while the reference is at a constant value. Variations of illuminance at a desk should only happen during alterations of the occupation state of that desk. Therefore, a metric to assess this criterion can be defined as the average variation of the illuminance during periods of constant occupation computed by and approximation to the second derivative of the illuminance. Let us call this quantity the **Comfort Variance**:

$$V_{flicker} = \frac{1}{N}\frac{\sum_{i=3}^{N} \mid l_{meas}(t_i) - 2l_{meas}(t_{i-1}) + l_{meas}(t_{i-2}) \mid}{T_s^2}$$

where T_s is the sampling period and $\mid A \mid$ represents the absolute value of A. The transients due to explicit variation of the reference, should be excluded from the formula. Again, the previous expression refers to a single desk. The total average variation should be computed as the sum of the average variations at each desk. The units of this metric are [lux/s^2].

In the report, indicate factors that can influence this metric.

5. The Global Control

A global optimization of the luminaires actuation should solve the linear program:

$$\begin{aligned}
&\min_{d_i} \sum_{i=1}^{N} d_i \\
&s.t. \\
&\begin{cases} \sum_{j=1}^{N} e_{ij} d_j \geq L_i - O_i, & \forall i = 1, \cdots, N \\ 0 \leq d_i \leq 1, & \forall i = 1, \cdots, N \end{cases}
\end{aligned}$$

where N is the number of luminaires (3 in the case of our setup), L_i is the lower bound on the luminance for luminaire i, O_i is the illuminance at desk i due to external light, and e_{ij} is the coupling gain expressing the influence of luminaire i in desk j. The values of O_i and e_{ij} can be estimated by a initial calibration procedure (see Section 6). Once the calibration gains e_{ij} are estimated, the values of O_i can be recomputed at any time t by:

$$O_i(t) = l_i(t) - \sum_{j=1}^{N} e_{ij} d_j(t)$$

The linear program thus depends on the luminance lower bounds, that change when the occupation state of a desk changes, and the external illuminance. To prevent exaggerated communication efforts in the global control, it is suggested that the recomputation of the linear program solution be executed only when (i) the desk occupation states change; and when (ii) there are large external illuminance changes. An alternative to (ii) is to perform this recomputation periodically at a slow rate.

The “ideal” solution for global control would be the implementation of the distributed simplex algorithm of [4] in the microcontrollers, thus yielding a completely distributed control. However, due to the complexity of this modality: student may opt by the following alternatives:

- Implementation of the distributed simplex of [4] in C++ running at the PC.
- Implementation of the centralized simplex algorithm in C++ at the PC.
- Implementation of the centralized simplex algorithm in Matlab in the PC.

The objectives of the project can only be considered achieved when the “ideal” distributed control or any of the aforementioned alternatives are implemented.

In the report the students should clearly justify the implementation choices, analysing their advantages and limitations.

6. The Calibration

To allow for the global optimization, the coupling gains e_{ij} and initial background illuminance of O_i and must be estimated at a calibration phase. A possible procedure is to (i) turn off all luminaires and compute the background illuminance as $O_i = l_i$, (ii) turn on luminaire *j* while keeping the other off and compute the gains e_{ij}, by $e_{ij} = l_i - O_i$. Repeat (ii) for all *j*.

7. The C++ Server

To interface the distributed system with the outer world, groups should develop a C++ server using TCP-IP protocol and asynchronous I/O classes from Boost library ASIO. These classes also provide functionality to read/write data in the serial port, for data exchange with the Arduino. The TCP-IP server should listen for connections at port 17000 and be able to serve multiple clients. A sample client is made available in the course web page and will be used to test the project during the final demonstration. The server should be able to send and receive data from the client in string format. A specification of the communication protocol between client and server is provided in the following table.

Command	**Client Request**	**Server Response**	**Observation**
Get current measured illuminance at desk <i>.	“g l <i>”	“l <i> <val>”	<val> is floating point number expressing measured illuminance in lux.
Get current duty cycle at luminaire i	“g d <i>”	“d <i> <val>”	<val> is floating point number expressing duty cycle in percentage.
Get current occupancy state at desk <i>	“g o <i>”	“o <i> <val>”	<val> is a Boolean flag: 0 – non-occupied, 1 – occupied.

Get current illuminance lower bound at desk <i>	"g L <i>"	"L <i> <val>"	<val> is floating point number expressing illuminance lower bound in lux.
Get current external illuminance at desk <i>	"g O <i>"	"O <i> <val>"	<val> is floating point number expressing background illuminance in lux.
Get current illuminance control reference at desk <i>	"g r <i>"	"L <i> <val>"	<val> is floating point number expressing illuminance control reference in lux.
Get instantaneous power consumtion at desk <i>	"g p <i>"	"p <i> <val>"	<val> is floating point number expressing instantaneous power at desk <i> in Watt. Assume each led nominal power = 1W.
Get instantaneous total power consumption in the system.	"g p T"	"p T <val>"	<val> is floating point number expressing total instantaneous power in Watt. Assume each led nominal power = 1W.
Get accumulated energy consumption at desk <i> since the last system restart.	"g e <i>"	"e <i> <val>"	<val> is floating point number expressing accumulated energy consumption at desk <i> in Joule. Assume each led nominal power = 1W.
Get total accumulated energy consumption since last system restart.	"g e T"	"e T <val>"	<val> is floating point number expressing total accumulated energy consumption in Joule. Assume each led nominal power = 1W.
Get accumulated comfort error at desk <i> since last system restart.	"g e <i>"	"e <i> <val>"	<val> is floating point number expressing the accumulated Comfort Error in lux. See section 4 – Evaluation Metrics
Get total comfort error since last system restart.	"g e T"	"e T <val>"	<val> is floating point number expressing the total Comfort Error in lux. See section
Get accumulated comfort variance at desk <i> since last system restart.	"g v <i>"	"v <i> <val>"	<val> is floating point number expressing the accumulated Comfort Variance in lux/s^2. See section 4
Get total comfort variance since last system restart.	"g v T"	"v T <val>"	<val> is floating point number expressing the total Comfort Variance in lux/s^2. See section

Set occupancy state at desk <i>	“s <i> <val>”	“ack”	<val> is a Boolean flag: 0 – non-occupied, 1 – occupied.
Restart system	“r”	“ack”	Reset all values and recalibrate.

8. Final Demonstration

In the final demonstration the instructors will ask the group to start the system and C++ server, and will connect to the server using a C++ client similar to the one made available in the course page. A subset of the commands listed in the table of the previous section will be requested at the client and the results analysed. General questions about the execution of the project may be made to individual members of the group, including aspects related to the first phase. In case the groups are unable to demonstrate the functionalities via the C++ server, they should prepare alternative ways to operate the system and visualize the data.

As noted in the project guide there is required a global report about the first and second stages of the project. Some information was gathered already at the first stage. In the following paragraphs are a number of documentation guidelines indicating necessary steps to complete the project. Note that despite not rewriting here some questions of the main guide, it is still expected to receive answers to those questions.

D1. (i2c Communications) Indicate the average time required to send a request message and receiving an answer.

D2. (C++ Classes, Sockets, Parallelism) Indicate the C++ classes most relevant for coding the server running in the PC interfacing to the Arduino(s).

D3. (Sockets) List the number of sockets used to implement the server. Describe the functions of the sockets.

D4. (Parallelism) Indicate the methodologies used in the server to implement parallelism in order to answer the various clients.

D5. (Simulator) Considering the formal description of the centralized control problem, list the names of the main variables appearing in the simulator. Indicate experiments needed to calibrate some of the constants present in the simulator and that need to be estimated also in the real environment.

D6. (Centralized or Distributed Control) Describe the implementation of the centralized and/or distributed controller. In particular document concisely the inputs and the outputs. Describe the relationship of the global centralized or distributed controller with the local controllers. Document concisely the implementation of the global controller.

D7. Test the controlled system and comment how it runs. In particular comment the results obtained with the evaluation metrics.

Report

Write a report describing the problem, the designed solutions and the obtained results, mentioning the positive and negative points of the applied techniques. The report must be complete but succinct, with less than 20 pages including graphics and references. Avoid redundancies and exaggerated technical content. Try to summarize as much as possible but do not miss to write the important things. The graphics must be self-contained, i.e. fully labelled and with complete captions.

9. References

[1] Occupancy-based illumination control of LED lighting sensors, D. Caicedo, A. Pandharipande and G. Leus, Lighting Research and Technology 43: 217, 2011.

[2] Daylight integrated illumination control of LED systems based on enhanced presence sensing, A. Pandharipande, D. Caicedo, Energy and Buildings 43, 944–950, 2011.

[3] Distributed Illumination Control With Local Sensing and Actuation in Networked Lighting Systems, A. Pandharipande, D. Caicedo, IEEE Sensors Journal, Vol. 13, No. 3, March 2013.

[4] A distributed simplex algorithm for degenerate linear programs and multi-agent assignments. Mathias Burger, Giuseppe Notarstefano, Francesco Bullo, Frank Allgower. Automatica 48, 2012.

All papers available in the course web page.

– ***Enjoy the project*** –