

Received September 7, 2021, accepted October 13, 2021, date of publication October 21, 2021, date of current version October 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3121669

# Use of Receiver Operating Characteristic Curve to Evaluate a Street Lighting Control System

JOSE-LUIS POZA-LUJAN<sup>1</sup>, (Member, IEEE),

JUAN-JOSÉ SÁENZ-PENAFIEL<sup>2</sup>, JUAN-LUIS POSADAS-YAGÜE<sup>1</sup>, J. ALBERTO CONEJERO<sup>3</sup>,  
AND JUAN-CARLOS CANO<sup>4</sup>, (Senior Member, IEEE)

<sup>1</sup>Institute of Control Systems and Industrial Computing (ai2), Universitat Politècnica de València (UPV), 46022 Valencia, Spain

<sup>2</sup>Dirección de Investigación, Universidad de Cuenca, Cuenca 010107, Ecuador

<sup>3</sup>Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València (UPV), 46022 València, Spain

<sup>4</sup>Grupo de Redes de Computadores, Universitat Politècnica de València (UPV), 46022 València, Spain

Corresponding author: Jose-Luis Poza-Lujan (jopolu@upv.es)

This work was supported in part by the Spanish Science and Innovation Ministry (MICINN): Interministerial Commission on Science and Technology (CICYT) project Predictable and dependable computer systems for Industry 4.0 (PRECON-I4) with code TIN2017-86520-C3-1-R.

**ABSTRACT** Intelligent control of public lighting is nowadays one of the most challenging issues in smart city deployment. Lighting optimization entails a compromise between comfort, safety, and power consumption, affecting both vehicles and pedestrians. Smart solutions must estimate their characteristics to trade-off users' needs and energy requirements. This paper proposes an intelligent street lighting control system and the Receiver Operating Characteristic (ROC) curve method to evaluate the best number of street lamps to achieve a balance between public road user comfort and system power consumption. The control system is based on the detection of users, mainly pedestrians, using presence sensors. From the detection of a pedestrian by two or more consecutive street lamps it is possible to determine their speed. Knowing the pedestrian speed, allows the system to anticipate and adjust the light intensity of the remaining street lamps, and provide a comfortable view of the street. Using the ROC curve, we evaluate the control algorithm in terms of the number of previous street lamps used. We have tested the system and the method in a model of pedestrians walking down a street. The obtained results show that ROC analysis used to control street lighting allows measuring the whole control system's efficiency by providing a concrete number of previous street lamps.

**INDEX TERMS** Distributed computing, distributed control, networked control systems, sensor systems and applications, intelligent systems, lighting control, ROC curve, smart cities.

## I. INTRODUCTION

The Sustainable Development Goals (SDGs) are considered part of the basis of the research in Smart Cities [1]. The challenge of smart cities is in continuous review and transformation [2] since it has to consider different actors, initiatives, goals, and applications [3]. However, most authors agree that technology must be ubiquitously integrated into the city environment to improve citizens' quality of life and simultaneously progressing towards achieving the SDGs. Its use covers broad technology fields: from the use of information related to Big Data systems [4], to the use of technology embedded in the everyday objects [5]. The extension of the

technology gives rise to different challenges from the management of routine aspects, such as intelligent traffic [6], energy management [7], and roads management [8]; to others that may involve risks, such as cybersecurity [9]. Smart cities include a large number of potential smart sensors such as temperature detectors, ambient sound levels, air quality, and traffic density. With this information, lots of elements interact with the environment to improve the livability of cities. Both city elements and citizens can be considered from an atomic point of view as elements of a system because they share the same environment and interact with them.

Public space lighting, mainly in the streets, is one of the top power consumption sources in any city. Energy management based on worst-case scenarios does not control energy consumption. For instance, in a public road shared by pedestrians

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

and bicycles uses a control policy based on the worst-case (bikes), more energy will be spent without improving user comfort. Smart cities can be endowed with intelligent control systems of light management that would permit to reduce power consumption without reducing users' comfort [10]. These control services are based on the analysis of data provided by sensors and its subsequent processing to consider users' needs and power consumption [11]. As a solution, we can conceive a smart street lighting system as a distributed system with several devices to be controlled, such as the lighting of every lamp in a street or a set of traffic lights in a roundabout.

There are many ways to control these distributed systems, from centralized control, through a cluster-based control, to each system device's autonomous control. The control action evaluation of a distributed system is a complex task. Control actions that are effective for a single element might not be efficient from an overall perspective or vice versa. For example, a traffic light can prioritize vehicles' crossing to clear a street, but this action will cause a traffic jam in the following streets. Another point to be addressed is if sharing valuable information between system components could increase overall control action performance. So, one of the aspects to decide is the scope of the information for control management. For example, if the data provided by a brightness sensor from a street lamp could be practiced to control a neighboring street lamp's lamp lighting or on the other side of the city.

Measure the control action's efficiency on a single component is simple. We only need to consider sensors and actuators involved in the same control loop. In this case, the scope of the information is limited to its component. Depending on the system, the scope of the information may be different. For example, in an urban environment, the information corresponding to vehicles' detection by a streetlight can be used in the next streetlight to optimize lighting. This information can also be used at a traffic light at the end of the street to optimize traffic. Based on these premises, measure the whole system efficiency is one of the main points to be addressed in a distributed intelligent control system.

The starting point to measure is the extraction of system optimization indicators to evaluate how optimal the control policy is applied in the complete system. This step involves sensors that can gather physical parameters and the additional difficulty of determining citizens' needs in each particular context. Intelligent control systems use sensory information to determine the optimal action to do. In a street lighting system, optimal action provides a more comfortable environment with less power consumption. We have to detect the service offered to street or road users and the power consumption to know if the control action is optimal. Power consumption is easy to measure in each device because only it is necessary to add the corresponding electronics. However, measuring the service offered is complicated because users

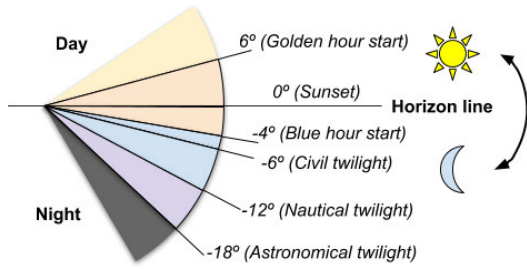
have different requirements due to aspects because of the speed of movement or their lighting, as bikes have.

There are many ways to measure the optimal action in a control system. Simple control loops based on periodic sampling, with one input and one output, are usually evaluated through the error between the action provided, output value, and the action expected or reference value [12]. In smart cities, due to the uncertainty of the users' behavior, and the multiple inputs and outputs to manage, the event-based control is a convenient approach instead of the periodic sampling control approach. When the control is event-based and distributed, measure the optimal action is difficult due to the result depends on the multiple outputs [13]. For example, to diagnose lighting needs on the street, it is necessary to know the light intensity at different points, the street users' needs at these points, and how they expect to evolve as time goes by.

In this paper, we propose the use of the *Receiver Operating Characteristics* (ROC) curves [14] to evaluate the performance of control methods applied to develop a distributed smart street lighting system. A ROC curve is a statistical method used to illustrate the diagnostic ability of a binary classifier. Based on this premise, we are interested in how ROC curves can be used to evaluate how optimal the control is applied in a smart street lighting system. ROC curves will allow us to evaluate different protocols for controlling the light intensity of luminaries.

The main contribution of the article is a technique of evaluating the efficiency of a distributed intelligent dynamic control system. The technique uses the parameters of the ROC curves, considering the control action calculated as a diagnostic of the user's needs. In this way, it is possible to minimize the amount of information or data required, in the proposed case defined by the number of streetlights needed to provide enough information to save energy without decrease the user's comfort. The techniques to control dynamically street lighting are nowadays a hot topic whose impact is especially relevant in the economic and ecological realm [15].

The rest of this paper is organized as follows: Section II presents further details of smart cities' environments and how they can be smartly controlled. Citizens' needs and movements are analyzed, emphasizing how they influence street users' comfort. We also discuss streets and light source (LS) characteristics and considerations for the lighting control proposed in this paper. In the III section, we briefly discuss ROC curves and evaluate the control of the components presented in the previous section using the ROC curves. In the IV section, we describe the evaluation of different control strategies. We also analyze the impact on the efficiency of the system when different source lights communicate between them. Finally, in the Conclusions and Future Work section, we report the main findings and future challenges to be addressed.



**FIGURE 1.** Representation of the twilight and daily times. Depending on the angle of the sun, there is more or less illuminated. A good control system should be able to adapt lighting these characteristics.

**II. STREET LIGHTING CONTROL IN A SMART CITY CONTEXT**

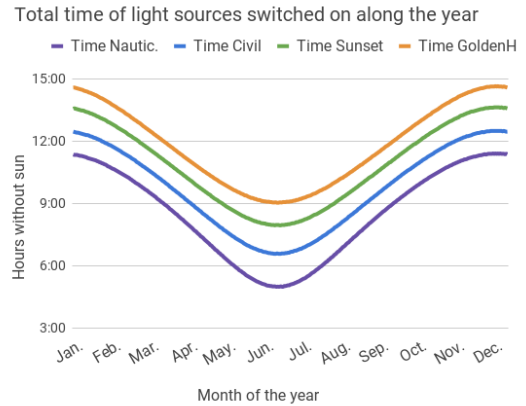
In this section, we review road users, vehicles, and pedestrians’ needs and how these needs condition the implementation of the system.

**A. VEHICLES AND PEDESTRIANS NEEDS**

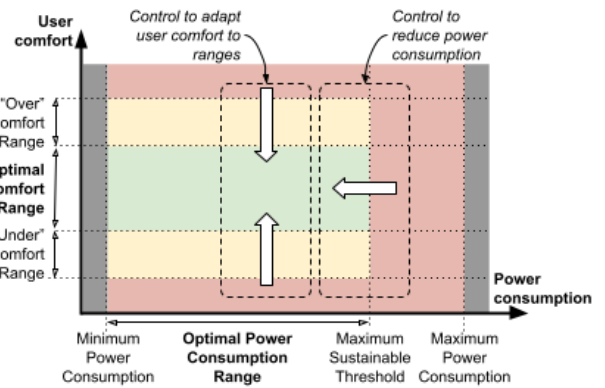
We can consider diverse conditions for determining the lighting of a street. The first factor to be taken into account is the street topology. Streets deviations, joins, and crossings usually require more illumination since the risk of accidents is higher than at straight roads. Besides, not all the parts of a straight street could require the same lighting due to urban considerations. For instance, narrow streets would require more artificial light than wider ones, and streets surrounded by tall buildings or large leafy trees will get dark earlier than streets in open areas. Environmental conditions constitute the second factor to be taken into account since daylight directly depends on the sun’s relative position and atmospheric conditions. In most cases, specific day-times are considered to start street illumination. Traditionally, different twilight are used to determine daylight times (see Figure 1).

On/off switching times can then be programmed as a function of the twilight, without considering atmospheric conditions. Sunset is typically defined as the time after which we require illumination. A simple astronomic computation (Figure 2) reveals that delaying the switching time by a single minute result in approximately 0.15% of energy savings. Moreover, natural lighting depends on the sun’s relative position and, therefore, on daytime and atmospheric conditions. If cloudy conditions are present near dusk, we require illumination to ensure road safety and provide more comfort.

Finally, a third factor consists of street users’ behavior [16]. Significant savings can be hypothetically obtained by a system that only provides illumination when a street is in use, either by vehicles or pedestrians. This information can be obtained through motion sensors and it can be complemented with historical data from traffic in specific locations. Several systems have been developed under this strategy and they will eventually replace current daytime-based controls. In all cases, an intelligent distributed lighting control will allows the system to adapt illumination street conditions in different



**FIGURE 2.** Total hours without sun of each twilight (Y-axis) with respect to the month of the year (X-axis). Times of the figure are contextualized in the 40th parallel north. Depending on the twilight used, there are three hours of difference to switch on street lights.



**FIGURE 3.** Constraints on power consumption and user comfort to control an urban system. The objective is to place consumption and comfort values in the central area. It is necessary to have parameters that probe that the control algorithm places consumption and comfort in the central area.

ways [17], ranging from direct interaction to traffic behavior clustering [18], providing energetic savings once we ensure optimal conditions [5].

**B. POWER SAVING VS. USER COMFORT**

The main parameters to be considered in an urban control system are power consumption and user comfort. The figure 3 shows the environment in which such a control system acts.

Users’ comfort is strongly linked with high street lighting. As a consequence, comfort is linked with higher power consumption. Consequently, the control system’s objective is to consider reducing consumption while maintaining, or even increasing, user comfort. Regarding consumption, there is an upper limit defined by the maximum possible use by the actuator, that is, the Maximum Power Consumption boundary in Figure 3. In street lighting, the maximum consumption occurs when all the electrical and electronic elements (sensors, processors, and actuators) are working. Different changes can reduce the maximum consumption: using lower consumption LED lights [19] or decreasing the number of street lights.

When the maximum consumption cannot be reduced with hardware modifications, it is still possible to offer light only when users need it and modulate the light intensity within user comfort limits. The energy sustainability of a system implies that the consumption of the services offered must not exceed the energy resources available [20]. So that, the *Maximum Sustainability Threshold* is determined by the available energy. Authorities define this threshold, and it depends on available renewable energy sources. Power generation is the main cause of climate change, so the concept of sustainability threshold goes beyond available energy and is defined as available energy that does not generate climate change [21]. Despite the recommended reduction in energy consumption, user comfort needs must be achieved.

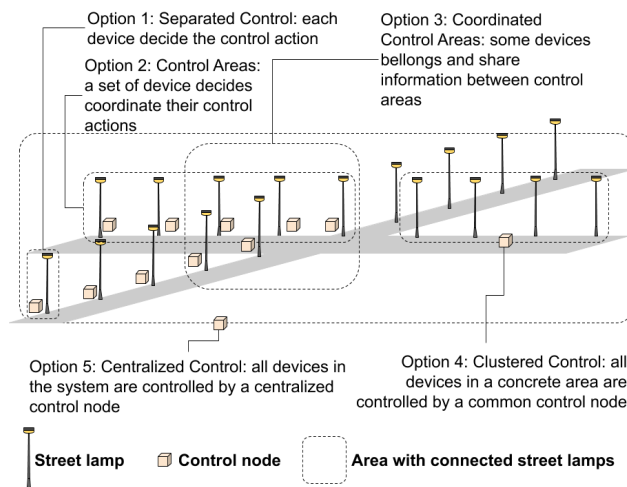
The user comfort measurement is problematic because it depends on subjective perception [22]. As shown in Figure 3, the upper and lower thresholds do not have a specific value and therefore are in a range. In this case, average comfort is the goal, but it seems more appropriate to focus on an area within which most users have an acceptable comfort level. We represent it in Figure 3 and we have called it *Optimal Comfort Range*. Based on previous conditions, the control system’s objective is twofold since it must minimize energy consumption, but focus on maintaining the Optimal Comfort Range. This subjective aspect implies that personal perception follows a normal distribution [23]. Measuring the error between the provided light and the subjective needs of a user is difficult to do. Consequently, it is necessary to have a method to verify that a control strategy is adequate according to the distribution above.

The problem increases in complexity when several control elements must provide comfort to a user, as is the case of various streetlights in the street. The overall system’s efficiency must be measured in terms of the set of components to be controlled. Consequently, for both individual and collective elements, the control strategy is a crucial aspect to achieve optimization objectives. In the next subsection, we present different control strategies used in urban lighting systems.

**C. CONTROL STRATEGIES IN LIGHTING SYSTEMS**

There are many smart street lighting systems managed with different control methods [24]. Usually, the architecture is centralized since the current electrical-based infrastructure is centralized, too. However, there is a current trend to distribute control actions to adapt the system to specific areas. Consequently, in most cases, the control method is based on the *Distributed Control System (DCS)* paradigm and its implications on system performance. A DCS can cover from a small number of nodes up to a significant amount. We show different distributed system control paradigms applied to a street lighting environment in Figure 4.

In Option 1, each streetlight controls its operation area. In other words, the light turns on, and its intensity is regulated based on the conditions detected only by its sensors. In Option 2, street lights are considered within a common control shared space for the users, like a street or a junction.



**FIGURE 4. Distributed system control paradigms applied to a street lighting environment. The point is where the control nodes are located and the scope of these nodes’ connections.**

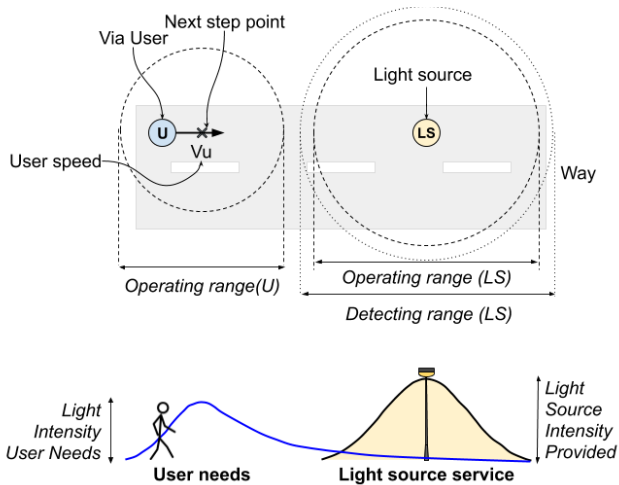
This option allows streetlights to share information to increase control optimization. Option 3 is an extension of Option 2, where two control areas share information to increase, if possible, the optimization of their actions. A dedicated control node can control this last option, or one of the control nodes in the system, assuming an additional load of the communication channel and the control processing cost. In Option 4, a set of streetlights is controlled from a single control node, generally in the same neighborhood, while in Option 5, the entire system, the city, is controlled from a single node. Options 4 and 5 are centralized and, currently, the most commonly implemented in urban environments. This last option is no longer in use due to the advantages of a distributed control for different clusters.

In this work, we configure the control system based on Option 2 since it is the current trend for control systems. Nevertheless, regardless of the option selected, it is necessary to characterize both the user’s lighting needs and the lighting provided. We review this aspect in the next section.

**D. CHARACTERIZATION OF ONE STREET LIGHT POINT VS USER NEEDS**

An intelligent control system must consider that each light source is an entity with its behavior but conditioned to the environment. Because the system focuses on control devices, we have considered naming the elements Light Source (LS) to distinguish it from the urban element Street Light. Figure 5 shows the minimum working environment, which will consist of a light source and a single user.

When carrying out the user’s detection and the corresponding street lighting control, there are more constraints than the detection and control actions. The light must remain in on state during a suitable time period to transit within the light source operating range. Besides, since a light source’s intensity can vary, it is possible to optimize the users’ service.



**FIGURE 5.** Characterization of the user operating range and a single light source detection and operating range. Ranges can be modeled as a normal distribution, centered on the user's next steps and the light source, respectively.

In the case of a single light source, the next characteristics will be used in the control strategy: a range of user detection, a range of environmental condition detection, and an operating range that does not need to coincide. When referring to the detection range, we implicitly refer to the type of detection. A binary detection provides the simplest one: user present and user not present. A step forward will be to consider a user's speed to achieve a more customized light service. This service can also be more accurate if environmental conditions are also taken into account. With this information, we can also define each user's operating range and how it should evolve with time. This is relevant because the user, usually, needs more light intensity ahead of the point where he/she is located. Nevertheless, how much ahead is needed depends on the factors above and conditions.

The level of Lumens required by the user should be determined at every point. Therefore, the control system has to detect, at least, when the user enters into the zone that is illuminated by the luminary under consideration (see the pedestrian in Figure 5). To detect a user, a presence sensor or a motion detector is enough. Of course, these sensors can detect when a user enters the detection zone, and so adapting the light, but using only one of these types of sensors is not possible to estimate the user speed. To overcome this issue, we can estimate the average user speed, considering how the sensors of other luminaries located already detected the user. These calculations are accurate when there is only one user on the street, but if more than one user is present, these sensors cannot correctly distinguish which user it is detected or the direction in which she moves. To partially overcome this limitation, we can place barrier sensors that know when a user passes by a certain point. These sensors have the disadvantage of being installed outside the light source, which implies additional costs. We will consider sensors in the same location as the light source in the rest of our work.

To know the user speed and, thus, to predict when the light should increase at some point, we need at least one distance sensor. It must have enough range to take two measures that permit the calculation of the user speed. Besides, it should be placed at a location that would permit the system to react by increasing the luminaries' light intensity under consideration.

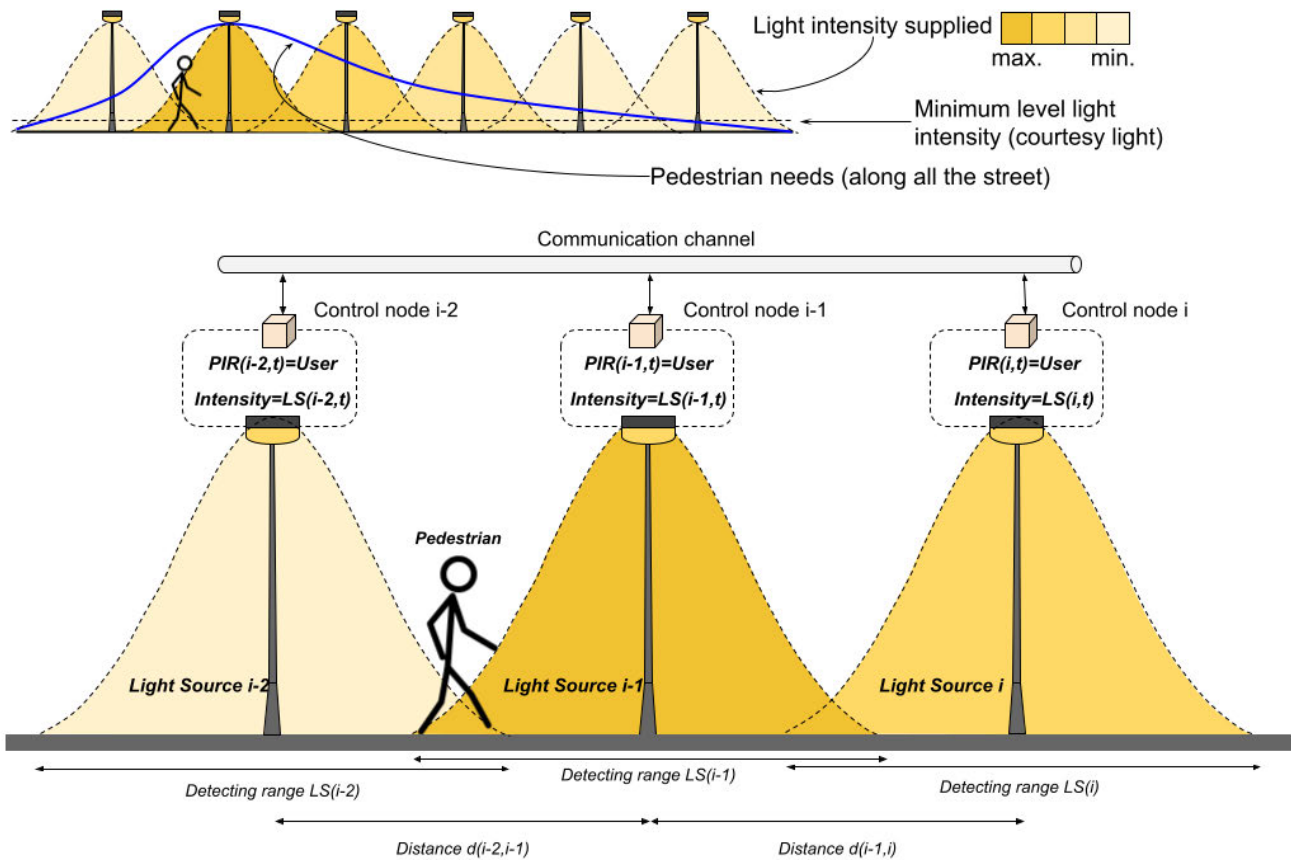
From the state-of-art technology, the current low-cost ultrasound technology does not allow detecting objects at more than 5 meters distance, which is an insufficient detection range in the context of this problem since only people walking at moderate speeds of 3 or less m/s would be detected. Some ultrasound sensors permit to deal with long distances, but the price is high to provide one sensor per street light. The infrared laser technology itself allows objects to be detected at distances higher than 100 meters. However, this technology is costly and can even be harmful in the case of near-infrared. Finally, in some systems, video cameras allow detecting movements and speeds through image analysis. These devices have the advantage of being able to detect several users simultaneously, calculating speeds and sense of movement, see for instance [25]–[27]. In addition to those concerning personal protection laws, the cameras' problem would require the support of infrared bulbs. So, we have discarded such technology for the design of the system.

### E. EXTENDING STREET LIGHT AND USER NEEDS TO ONE STREET

When we consider several streetlights the amount of information, that has to be considered to design an intelligent distributed system, increases according to the system's complexity. Currently, city's streetlights are usually clustered into zones, and it can be understood as a centralized or clustered distributed control system. In these types of systems, any single streetlight cannot make its own decisions. However, we suppose each light source could regulate its light intensity to the environmental and usage conditions. In that case, the system could be optimized for power consumption and user safety (providing suitable light brightness). When a large number of light sources are available, each one of the luminary's sensors can be used to get extra information or improve the existing one [28].

In the case of a street, we expect that a user can traverse it with a suitable amount of light intensity along the way. Here, the system can consider just information from streetlights under consideration and other light sources. Let us list the light sources by  $LS(i)$ , with  $1 \leq i \leq M$ . If  $LS(i)$  has information about forthcoming users before their sensors detect such presence, it can better adapt the light intensity provided. We show a basic example in Figure 6, where we illustrate the basic context of a street where each light source has a *passive infrared sensor* (PIR). Each PIR has an electronic sensor that measures infrared (IR) light radiating from objects. Such sensors are frequently used to detect motions or object movement within its detection range.

The difference in size between the sensing range and the operating range affects only the instant the control will



**FIGURE 6.** Light sources are distributed in the street with PIR sensors used to detect pedestrians. To simplify the system, detection and operation ranges have the same value. The upper part of the figure shows how the pedestrian’s lighting needs vary depending on the distance and how the street lighting must respond to these requirements. The lower part of the figure shows a fragment of the street with the corresponding control nodes and the parameters involved.

actuate the light. A sensing range greater than the operating range would delay the control action to turn on the light. A detection range smaller than the operating range would accelerate the light level since the user would be within the area in which he needs the light when it is detected. To simplify calculations, we assume that the detection and operation ranges of each light source are coincident. That is, the pedestrian detection range is the same as the streetlight illumination range.

We consider the setting shown in Figure 6. Here, the light source  $LS(i - 2)$  detects a user within its detection range at an instant  $t$ . Now, the control algorithm releases the instruction to  $LS(i)$  of increasing its light intensity since a user will need more lighting in a short period of time. If at time  $t + 1$  a user is still detected within the detection range of  $LS(i - 2)$ , the control algorithm should tune the instruction of increasing the light intensity of  $LS(i)$  according to the obtained user speed estimation.

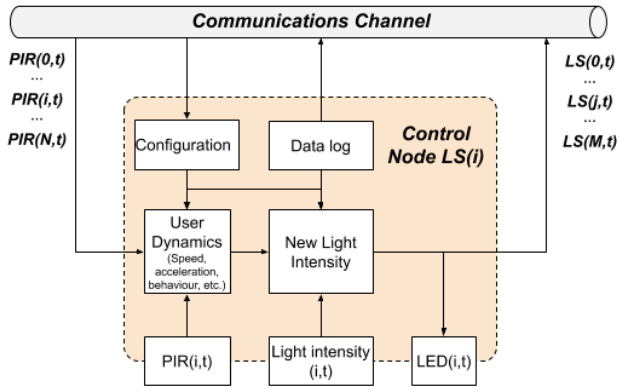
As time instants pass, the light source  $LS(i - 2)$  can continue to detect the user. At a specific moment, the light source  $LS(i - 1)$  can start to detect that same user. If the detection ranges overlap, the user will be detected at the same time by the light sources  $LS(i - 2)$  and  $LS(i - 1)$ .

If the detection ranges are not overlapped, the user will not be detected simultaneously by two luminaries. User may not be detected during some moments, but after a while, the next luminary will be detected on the way  $LS(i - 1)$ . In any case, as distances between two consecutive light sources are known, it is possible to estimate the speed at which the user moves. This permits the adaptation of the light intensity to satisfy the user’s needs and save energy, if possible.

As shown in Figure 6, the number of streetlights that must react to users’ presence can vary depending on the control policy used. With the proposed methodology, it is possible to measure each streetlight’s control algorithm’s performance and check the number of previous lamps necessary to offer adequate comfort without increasing energy consumption.

### III. METHODOLOGY

The section begins with a description of the control system used by each streetlight. This control system allows for a given streetlight to obtain information from a concrete number of previous streetlights. The number of previous streetlights can change for each streetlight, depending on the circumstances. The ROC curves adapted to the experimental environment of the article are described below. Finally, we



**FIGURE 7.** Control node block diagram. We obtain inputs from external sensors (PIR signal from neighboring LS), internal sensors (PIR sensors), or configuration parameters (i.e., LS distance). Learning control algorithms can also use data history. The outputs will be the control action (LS Intensity) and the signals to other LS interested in the current LS data.

explain the confusion matrix’s use to check the control algorithm’s prediction’s correctness.

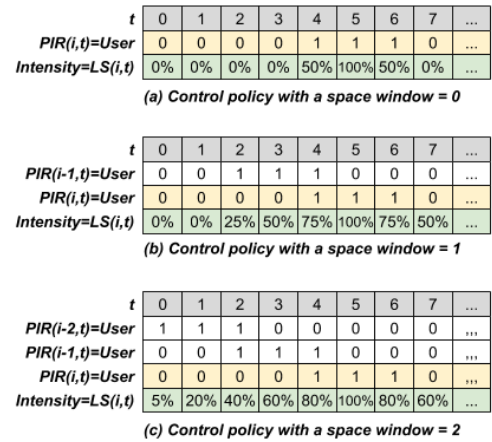
**A. CHARACTERIZATION OF THE STREETLIGHT CONTROL**

To design the control system, we start from the basic case when the pedestrian traffic happens on one-way. It is possible to extend the control model to several pedestrians traveling in one direction or to several pedestrians traveling in two directions. In these cases, each control element must consider several sources and decide to act according to the closest lamppost. We selected one pedestrian and one-way mode because our interest is to check how the ROC curves allow us to select the optimal number of information sources.

We can consider a spatial window of  $N$  preceding light sources or a time window of  $J$  velocity measurements for the control strategy. In any case, to implement the control strategies, it is necessary to have a control node at each light source in our proposal. Each of them must have internal inputs, outputs, and internal data. An example is illustrated in Figure 7.

Basically, the control node of the light source  $LS(i)$  has internal inputs at time  $t$  and the outcome of the motion sensor  $PIR(i)$ . If allowed, it can also have  $N$  additional external inputs of the PIR sensors of the  $N$  preceding light sources at the corresponding instant  $t$  when the PIR detection event is produced. As outputs, we have the light intensity and the PIR sensors outcome of other light sources. As internal data, we can consider the distances of the near light sources from which we obtain the information provided by their sensors  $PIR(i - l)$ , with  $l = 1, \dots, i - 1$ .

The control node has to develop a policy to determine the control action on the light intensity regulation based on the user speed estimation. The use of information from other light sources can improve the results; see Figure 8, where the control action (modulation of the light intensity in the maximum intensity percentage) also considers information from previous light sources whose data is already known.



**FIGURE 8.** Control policies (actions to regulate light intensity) based on the information of previous LS. (a) Example of intensity dynamics of just the single LS. Here, the space window is of length 0 since we do not consider any information from the previous LS. (b) Intensity dynamics of an LS, with a space window of length 1 and (c) of length 2.

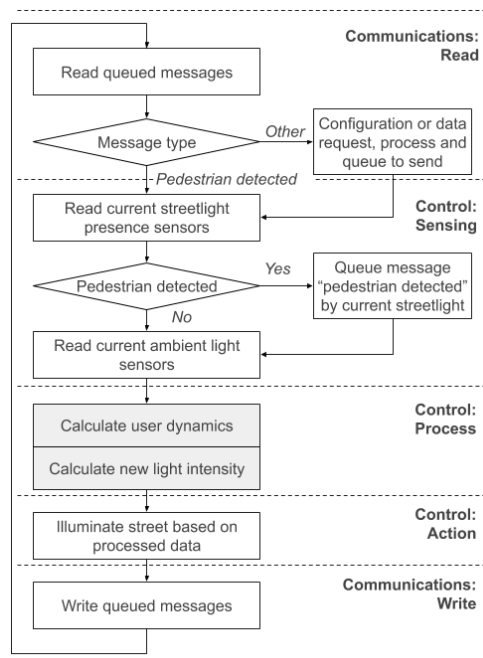
Information from previous light sources provides advantages to optimize control. With the distance between  $LS(i)$  and  $LS(i - 1)$ , it is possible to know the user’s speed since the user’s instant is detected. With two previous light sources, it is possible to know the acceleration and make a better prediction since a user may change their speed going faster or slower. A consideration is that for each light source that sends a message to another, one implies one extra message to be processed in each control network node.

Therefore, we will evaluate the number of previous light sources we need to offer adequate lighting without unnecessarily increasing the communications load and the control load through the sensitivity and specificity parameters. Consequently, we will determine the optimal number of previous light sources that allow a more efficient light adaptation.

Figure 8(a) shows the control action on a single light source’s light intensity using only its own information and using average human speed. We refer to [29] for further information on human dynamics and crowd science. In this case, light intensity suddenly changes when it notices the user presence. In this case, user comfort is very limited since it only considers a few meters ahead.

Figure 8(b) corresponds to the control in which the previous node can provide information regarding if a user is present or not. In this case, the light intensity can increase progressively, and the user will have a better feeling of street lighting than in case (a).

Finally, in Figure 8(c) we refer to a control based on the users’ detection information from previous nodes. With this information, we can implement a smoother light intensity control that will increase users’ comfort. We consider a performance analysis to determine which policy provides the best trade between energy consumption and comfort. To measure the performance of a control policy, we can simulate the control algorithm from the sensory level [30] to the level of simulated behavior of intelligent agents [31].



**FIGURE 9.** Flowchart of the control loop algorithm for a single streetlight. Phases of the loop are organized in the communications-control sequence. The end of the control loop matches with the writing messages phase of communications, and the start of the control loop matches with the read messages phase.

In our case, as we must initially evaluate the use of ROC curves, we have used mathematical simulations. Our goal is to choose a control system that meets specific savings or comfort appropriate to a specific objective. In this sense, the current challenge is to know if the distributed control system correctly predicts users’ needs while maintaining optimum savings. For this analysis, we will vary the number of previous streetlights from which each light source receives the information.

**B. CONTROL LOOP**

Each control node implements the same algorithm, summarized in the flow diagram shown in the figure 9. The algorithm is organized with separate phases concerning communication and control tasks. This separation is justified because the control node is based on embedded micro-controllers [32] because the communications tasks have to use a buffer for the incoming and outgoing message queues. So that, it is necessary to isolate the control loop from the asynchronous nature of the communications. Consequently, the flow chart shows separately the reading of the incoming messages and the outgoing messages’ writing

The first action of the control loop is the reading phase of the incoming messages from the buffer. These messages can arrive from the previous streetlights as well as from other elements of the system. The messages that come from other elements may be data request messages such as monitors that require updating a data history or configuration

messages requested by a higher hierarchical element to, for example, change algorithm parameters. Both message types, data request or configuration, are processed and queued, if necessary, to be sent at the end of the control loop and not delay the lighting action. If the message type is pedestrian detected, it triggers control phases.

The control phase starts with the data acquisition from the sensors. The first sensor read is pedestrian detection. If a person is detected in the detection area, a message is generated to be queued and sent later. In this phase, the message is generated to provide other street elements with the timestamp of the detection as accurately as possible. If no people are detected in the detection area, the rest of the lamppost sensors are read, in our case, the detection of ambient light sensors. With all available data, the processing phase is ready to make a decision and starts the control action.

The processing phase is the phase that implements the control algorithms. In this case, there are two main tasks: the calculation of the prediction of the user’s speed and the calculation of the light intensity. The calculation of light intensity and adaptation to environmental conditions is based on the criteria presented in the figure 8.

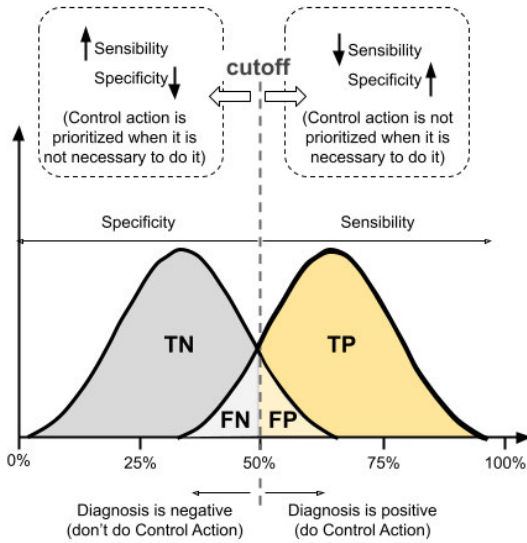
Once the lighting values have been calculated, these values are sent in the action control phase, in which the lamp updates its lighting level. Finally, the communication phase consists of sending messages. In this phase, the messages queued in previous phases are sent to the required destinations.

**C. ROC CURVES ADAPTED TO CONTROL SYSTEMS**

All control systems work reading data from sensors, processing this data, and, as a result, submitting control orders to the actuators. In this process, the control algorithm acts as a diagnostic procedure providing a prognosis and the subsequent treatment in terms of control actions to be carried out. There are several different control algorithms, and it is necessary to have a method to know which is the appropriate algorithm for a specific context. The analysis of Receiver Operating Characteristics (ROC) curves allows evaluating the quality of a diagnostic procedure [14]. A diagnostic method predicts if an event should happen or not. To evaluate its performance, firstly, we have to know which are the real Positive case (P) of an event, and which are the real Negative cases (N). Then, we have to compare the result of the diagnosis method’s prediction with what happens. The diagnostic method provides several predicted events that happened, also called True Positive (TP) or hits. It also detects negative cases, and they do not happen. These are called True Negative (TN) or correct rejections. Both cases lead to a correct control action: do an action in TP cases and not do in TN cases.

The diagnostic method may detect as positive a case that is not happening. These are known False Positive (FP) cases, also called false alarms or type I errors. Finally, a False Negative (FN) case happens when the diagnosis method detects a negative case. These are known type II errors. These last two cases lead to a control error. In FP’s case, we decide to carry out a control action when it is not required. In the NF





**FIGURE 10.** Relationships of control actions with specificity and sensitivity. The figure shows the statistical basis of the model. The diagnosis in the ROC curve corresponds to the interpretation of the information that the control algorithm performs. Figure 11 shows how the TP, FP, FN, and TN indicators are applied to the street light control environment.

case, we decide not to carry out a control action when it was necessary to do it. From the TP, TN, FP, and FN values, it is possible to evaluate the performance of the control system in terms of two parameters, named *sensibility* and *specificity* (see Figure 10).

*Specificity* is the *True Positive Rate* (TPR). We calculate it as the ratio of true positives cases detected (TP) to the total number of real positive cases (P). Therefore, we can design a particular strategy in order to minimize the number of false positives. Applying this concept to a control strategy, a control algorithm will only do the control action when it is sure that the condition to do the control action was really done. *Sensibility* is the *True Negative Rate* (TNR). We calculate the TNR as the ratio of detecting true negatives (TN) to the total number of real negatives (N). We can think of a very sensible control algorithm that will carry out the control action in uncertain cases because we prioritize the control action execution, even if it was unnecessary to do it. Depending on the control algorithm used and the environment in which this control algorithm is applied, we will obtain specific specificity and sensibility values. It is possible to define a value to decide if it is convenient for the control algorithm to be more sensitive than specific, or less sensitive and more specific. This value is named *cutoff value*. A cutoff value close to 0% implies that we execute the control action in most of the cases. Analogously, a cutoff value close to 100% implies that the control action is only carried out if we are entirely sure that we must execute it. This policy is directly related to the consequences of control actions. In this way, the analysis of ROC curves allows us to evaluate and compare different control policies and determine the most appropriate one according to our admissible specificity and sensibility.

**TABLE 1.** Confusion matrix applied to a simple control node. The “true condition” columns show what is really happening in the street, while the rows show the result of the ‘diagnosis’ made by the control algorithm based on the available data.

Predicted	True Condition	
	Positive	Negative
Positive	True Positive: there is a user with light needs, and the light intensity level is increased	False Positive: there is not any user, but the light intensity is increased.
Negative	False Negative: there is a user, but the light intensity is decreased	True Negative: There is no user, and the light intensity is decreased (until the low level in each control iteration).

**D. CONTROL NODE EVALUATION**

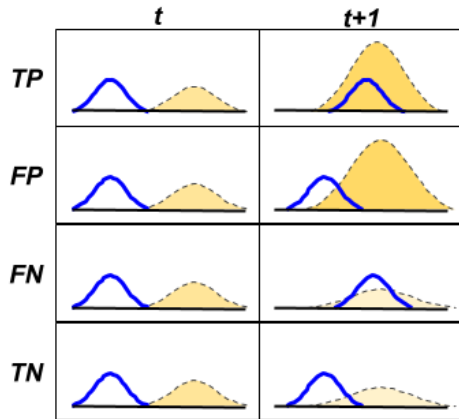
In this work, the confusion matrix represents the relations between the outcome predicted and the system’s true condition. Table 1 describes how every cell of the confusion matrix is applied in a single control node. The rows represent the prediction that the control node did after reading sensors or receiving messages from previous control nodes. The columns represent the true condition, consequently, every cell of table 1 provides us with the result of the prediction in terms of the ROC curve. More precisely, we represent user needs (more lighting needs stand for a positive outcome) versus the diagnosis’s possible result; see Figure 11.

The confusion matrix allows us to determine if a light source is offering an optimal service. We have to identify positive, negative, false positive, and false negative cases and the control actions performed in each case. In Figure 11, we can show these cases for a single light source. Here, we assume that at a time  $t$ , we have a user approaching from the left to a light source that is to the right. We predict where the user will be on time  $t + 1$  and, according to this, we conduct different control actions.

A positive case is that the system predicts a user’s presence at  $t + 1$ , and the user was present at that time. A false positive is when the control action increases the light intensity since it predicts that a user will be there at  $t + 1$ , but the user was not present at that time. This case may be due to several factors, either because the user has stopped or its transit speed is slower than the one considered by the control algorithm. A negative case occurs when the user does need so much light intensity, and the control action decreases the light intensity. Finally, a false negative occurs when the control action decreases the light intensity, and there is not enough lighting to fulfill user needs.

As previously mentioned, ROC curves represent a diagnostic method and control policy’s performance in the present case, in terms of the TPR (or sensibility) and the TNR (specificity). On the one hand, a sensitive system will react when it has little evidence that a user will need more light intensity. On the other hand, a specific system will only increase the light intensity if it is sure that the user will need it. A system

- t** Moment in time used to calculate the control action
- t+1** Moment in time used to calculate the control action
- TP** True Positive control action: light level is increased according to the user requirements.
- FP** False Positive control action: light level is increased but the user does not need this new level.
- FN** False Negative control action: light level does not increase but the user needs more light level.
- TN** True Negative control action: light level does not increase and the user does not need more light level.



**FIGURE 11.** Interpretation of the control action in terms of the confusion matrix. The control action is carried out at time  $t + 1$  from the data available at time  $t$ .

with low sensitivity and high specificity will save pretty much energy. However, it may not satisfy users' light needs. A high sensitive, and low specific, system may increase energy consumption more than what was needed to fulfill user light needs. In essence, an optimal control system would be the one with 100% sensitivity (no false positives) and 100% specificity (no false negatives).

#### IV. EXPERIMENTAL RESULTS

##### A. SIMULATION ALGORITHM

The simulation carried out is based on the algorithm 1, programmed in Python [33]. The algorithm showed a single pedestrian's case, although the algorithm can be called the number of times necessary depending on the number of pedestrians to simulate. Subsequently, it is necessary to create the necessary arrays to perform the calculations of consumption, comfort, and the ROC analysis parameters (TP, FP, FN, and TN). For each pedestrian, a specific speed is calculated within the normal distribution margins based on a standard pedestrian's average speed. Once the data has been initialized, the simulation loop is executed. The simulation loop runs as long as the pedestrian's position is within the street length.

For each simulation step, the `updatePedNeeds()` function calculates the pedestrian needs, along the street conditioned on their position on the street. The values of the pedestrian's lighting needs are calculated for each of the perpendicular points of the streetlights and are stored in the vector `pedNeeds`. The function to obtain the needs is shown in the

##### Algorithm 1 Simulation Algorithm to Calculate the Efficiency of the Different Control Systems

```

/* Simulation variables */
1 const integer numLS = 10;
2 array real pedNeeds[numLS];
3 real pedComf = 0;
4 array real lightSupplied[numLS];
5 real powCons=0;
6 array real currentDif[numLS];
7 real deltaROC = 1.0;
8 array real rocValues[T][4];
9 integer t=0;
10 real pedSpeed = calcPSpeed(1.14,0.305);
/* Simulation loop */
11 while pedPos < streetLenght do
12     pedNeeds = updatePedNeeds(pedPos,pedSpeed);
13     for LS = 0 to numLS do
14         if pedestrianDetected(LS) then
15             lightSupplied = controlStreetLights(LS);
16             break;
17         end
18     end
/* Results update */
19 powCons = updatePowerCons(lightSupplied);
20 currentDif = pedNeeds - lightSupplied;
21 pedComf = UpdatePedComf(currentDif);
22 rocValues = updateROC(currentDif,deltaROC);
/* Next simulation step */
23 t=t+1;
24 pedPos = updatePedestrianPos(t,pedPos,pedSpeed);
25 end
    
```

equation 1. This equation is modeled as a descent linear function from a maximum need at the point of the pedestrian position and with a horizon of 0 Lux in the point  $dMax$ .

$$pedNeed(d) = MaxNeed(\frac{d}{dMax} + 1) \quad (1)$$

The variable  $MaxNeed$  is obtained from the random function that provides a value within the range  $PedNeed_{AVG} \pm PedNeeds_{STD}$  in order to simulate different pedestrian needs. The variable  $dMax$  changes depending on the concrete speed of each pedestrian. This speed is obtained from the random function that provides a value within the range defined by  $PedSpeed_{AVG} \pm PedSpeeds_{STD}$  in order to simulate different pedestrian needs depending on the pedestrian speed. Subsequently, all the streetlights are evaluated to check which of them detects the pedestrian. The streetlight that detects the pedestrian launches the control algorithm and, depending on the algorithm implemented, will cause the change of the light supplied of every streetlight. The function `controlStreetLights()` implements the control algorithm for each streetlight. This means that calculates the different light intensity that every streetlight must have in the corresponding

simulation iteration. Additionally, the function stores the corresponding intensity value of each streetlight in the array of streetlights (*lightSuplied*). It is possible to calculate the values of power consumption equations, pedestrian comfort, and ROC parameters at this stage. Power consumption is calculated with the time interval and the sum of the array's values *lightSuplied*. Regarding power consumption, the formula to calculate is directly obtained by the different light intensities provided each time interval (equation 2). For each of the time intervals, the street consumption will consume the intensity with which each street light is lighting the street (equation 3). The constant *maxLight* is the value of the consumption of the maximum Lux provided by a street light. The result is accumulated in the variable *powCom*.

$$powerCons(T) = \int_{t=0}^T lightSup(t) dt \quad (2)$$

$$lightSup(t) = \sum_{i=1}^{numLS} maxLight * lightSup(i, t) \quad (3)$$

Comfort is calculated as the accumulated value of the difference between the needs of the pedestrian and the light provided by each of the streetlights; see equation (4).

$$pedConf(t) = \sum_{i=1}^{numLS} lightSup(i, t) - pedNeed(i, t) \quad (4)$$

These values are stored in the array *currentDif* in order to calculate the weighted comfort average of the corresponding iteration. The value of *pedCom* is calculated by updating the average of the following array values for each simulation step.

Next, the algorithm calculates the ROC parameter, taking into account the difference between the pedestrian's lighting needs and the provided light level, and considering a percentage margin of goodness  $\delta$  configured at the beginning of the simulation. Basically, in this step, we calculate the number of TP, FP, FN, and TN for each street lamp in the way described in figure 11. Values are stored in the array *rocValues*, a bi-dimensional array with several rows depending on the time simulation steps, and one column to TP, FP, FN, and TN values. This bi-dimensional array offers the values to a post calculation of the different cutoff values' sensitivity and specificity.

As a final step, the simulation time is increased in the defined time interval, and the pedestrian's position is updated.

## B. SIMULATION PARAMETERS

We present the mathematical analysis using the simulation's data and analyzed by Apps Script with Google Sheets [34]. Concerning the street configuration, we have considered that all street lights have the same equipment. We consider that a PIR sensor carries out the detection with a detection range of 20 meters. We have considered that the light sources can emit a maximum of 400 Lux in the lamp's perpendicular point, which offers an active range of 20 meters. The distance between two consecutive light sources has been fixed

at 20 meters. Consequently, the complete path to simulate represents a street about 220 meters.

We have just considered the case of a single pedestrian walking down the street. A single pedestrian allows checking if the set of streetlights reacts appropriately, without taking into account the action due to serving several pedestrians simultaneously. It is not easy to determine a fixed walking pedestrian speed [35]. To define the pedestrian behavior, we have considered data speeds from a set of random 30 pedestrians observed at the university campus. They had a speed average of 1.14 m/s and a standard deviation of 0.305 m/s; these values are considered to simulate the pedestrians. In the analysis presented, we generated 100 pedestrians with a random speed between the minimum value of 0.835m/s (average - standard deviation) and the maximum value of 1.445 (average + standard deviation), considering a normal distribution of pedestrians. In order to ensure that only one pedestrian is walking down the street, the time between two consecutive pedestrians has been set up in 250 seconds.

The pedestrians' needs depend mainly on the corresponding pedestrian speed. For each simulated pedestrian, the comfort condition is calculated based on a maximum value of 400 Lux in the perpendicular of the next street light. This value will make it possible to compare whether the streetlights to which the pedestrian approaches have been turned on with the minimum value on time when the pedestrian arrives to the street light operating range. From this point, we calculate the lighting needs for the pedestrian in the following streetlights. The values of these needs, produce a Pedestrian need's array that we compare with the array of light intensities produced by the different streetlights. The difference between pedestrians' lighting needs and light provided by street lights establish if the control algorithm result is classified as a TP, TN, FP, or FN.

The first control algorithm to evaluate the performance of the service offered by a single streetlight. Since we do not consider the previous streetlight information, the control algorithm assumes a constant user's speed. In this case, single streetlight, we tested three different pedestrians' speeds, and we use the ROC curves to check which speed estimation is more appropriate in the control algorithm. Next, we evaluate how each streetlight's control algorithm improves, using information from one, two, until nine previous streetlights.

## C. TWILIGHT EVALUATION

As described in Figure 6, there is a minimum of light to provide. This minimum is called courtesy light and has the main function to show the pedestrian's complete path. Previous to a detailed analysis, we assessed the relevance of the twilight to the pedestrian's needs. The question to answer in this previous analysis is to know this minimum threshold of the light required to cover each twilight's needs. This will lead us to be able to determine under what conditions the control algorithms can be evaluated. Pedestrian needs have been fixed in 400 Lux on the floor in the space of 1 meter in

**TABLE 2.** Results obtained from a single LS model with three different control policies. Ctrl\_MAX expects that pedestrian is coming at a speed with the standard deviation above average, Ctrl\_AVG expects that pedestrian is coming with the average speed, and Ctrl\_MIN expects that pedestrian is walking the standard deviation lower than the average. The figure 12 shows the corresponding ROC curve.

Sensitivity	Ctrl MAX	Ctrl AVG	Ctrl MIN
100,0	0,0	0,0	0,0
96,1	35,4	34,7	33,9
92,9	50,4	46,4	42,3
80,4	77,4	69,8	62,2
79,2	79,9	72,4	64,9
75,9	83,6	78,2	72,7
74,3	84,7	81,4	78,0
61,5	89,1	87,2	82,4
41,1	96,4	91,7	88,7
38,2	96,7	93,7	92,4
21,9	98,6	96,8	95,6
0,0	100,0	99,1	98,8
0,0	100,0	100,0	100,0

front [36]. The average of the Lux measured by the authors in the university campus was 320 Lux when the golden hour starts, 210 Lux in the sunset, and 92 Lux in the civil twilight. Consequently, to simplify, we determine the courtesy light as 100 Lux in each street light’s measured on the ground from every perpendicular point.

**D. SINGLE LIGHT SOURCE CONTROL POLICY EVALUATION**

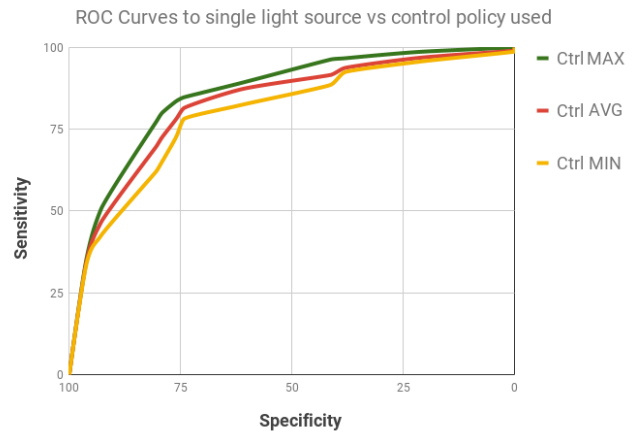
Based on the twilight results, we first consider a single light source control during night conditions. The control of the action is related to the average speed of the pedestrians. There is no method to know exactly the real user speed until the user leaves the operating range. We can only estimate it by looking at the time in which the PIR is active. For the control design, we have considered three predetermined speeds.

- Control with minimum speed (Ctrl MIN): we assume that the user is going slower than the user’s average speed. The predicted speed is the historical average calculated minus the standard deviation.
- Control with average speed (Ctrl AVG): we assume that the user walks at the average speed calculated with the historical data.
- Control with a maximum speed (Ctrl MAX): we assume that the user is going faster than the user’s average speed. The predicted speed is the historical average calculated plus the standard deviation.

Table 2 shows the results obtained from a single LS node with the control policies based on the three previous criteria presented.

We show the results through ROC curves in Figure 12. We can find the main differences in the center of the graph. The faster that the user goes, the better user service is provided by the control algorithm.

We also notice that with a single light source, we have sharp changes in the specificity values of 40% and 75% (X-axis). These changes indicate that groups of users with similar behavior coincide with human behavior models. Some of them go much faster than others.



**FIGURE 12.** ROC curves of a single source light with three different control policies. CTRL\_MIN and CTRL\_MAX policies assume that the pedestrian speed is lower or higher, respectively, by the standard deviation from the average speed (CTRL\_AVG). Sensitivity is represented on the Y-axis and Specificity on the X-axis.

**TABLE 3.** Results were obtained from multiple light sources. We only represent the two precedent light sources to each one since considering more precedent light sources do not significantly increase the system’s performance. All previous light sources can be found in the corresponding figure 13. In the column headers the abbreviation ‘Prev.’ corresponds to the term ‘previous’.

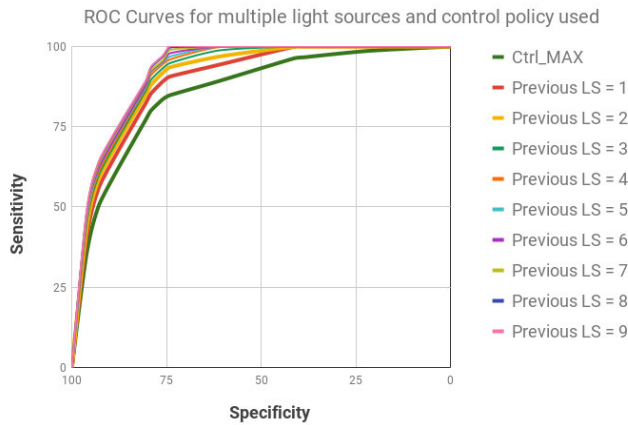
Sensitivity	Ctrl_MAX	Prev. LS = 1	Prev. LS = 2
100,0	0,0	0,0	0,00
96,1	35,5	41,4	43,9
92,9	50,4	55,8	58,7
80,4	77,4	82,8	85,1
79,2	79,9	85,5	87,8
75,9	83,6	89,4	92,1
74,3	84,7	90,6	92,7
61,5	89,1	94,9	97,2
41,1	96,4	100,0	100,0
38,2	96,7	100,0	100,0
21,9	98,6	100,0	100,0
0,0	100,0	100,0	100,0
0,0	100,0	100,0	100,0

**E. MULTIPLE LIGHT SOURCE CONTROL POLICY EVALUATION**

Once we have verified that ROC curves illustrate the control’s performance on a single light source correctly, we consider a street with 10 light sources. The subsequent analysis will permit us to set controls that gather information from several light sources to control actions. Table 3 shows the results obtained from different number of previous LS used.

We show the ROC curves in Figure 13. We have considered different controls, each one using information on a different number of light sources. We have also added the best control algorithms to compare a single street light control algorithm with multiple street lights control algorithms.

Figure 13 shows an optimization (improvement of the TPR) due to the increment of information that a light source control algorithm use from previous light source sensors. As  $LS(i)$  knows the distances of the previous  $LS(i - k)$ , with  $k = 1, \dots, i - 1$ , when a user enters in the operating range of the  $LS(i)$  the user speed is estimated with high accuracy and



**FIGURE 13.** ROC curves of a control policy considering previous light sources to estimate users’ speed. The most optimal curve from the previous experiment (CTRL\_MAX) is used to compare the optimization produced by the inclusion of information from previous streetlights. Sensitivity is represented on the Y-axis and Specificity on the X-Axis.

not only estimated. We also see that adding information from more than one preceding light source increases the TPR, but the LS control action improvement is minimal.

**F. OPTIMIZATION EVALUATION**

To find out the optimization obtained from using the information provided by the different number of streetlights, we use the parameter Area Under the Curve (AUC). The closer to one is the AUC value, the more optimal is the prediction made by the control algorithm [37]. Table 4 shows all the data involved to know if AUC can be used as a suitable parameter that joins power consumption and user comfort. The first column shows the different control strategies based on the previous light sources’ information (LS). The second column (Pow. Con.), provides the proportional value of the power consumption. This value is calculated as the percentage of consumption related to the maximum consumption of the street. The maximum possible consumption is calculated as all the street lamps emit the maximum light when the pedestrian is crossing the street. The third column is obtained as an average of the difference between the Luxes needed and provided. If *Conf.diff.* contains negative values, this means that the light provided is less than necessary. As the control performed is oriented to reduce the light provided to save energy, all values are negative. The fourth column shows the values obtained from the AUC for each case in which a different number of previous streetlights are considered in the control algorithm (first column). The last column shows the calculation of the optimization obtained when comparing the AUC of a specific number of streetlights with the previous one.

We notice that the more sources of information (street lamps) we have, the more comfort is provided. However, as expected, both energy savings and pedestrian comfort from several street lights hardly have not appreciable variations. As the results show, the improvement of AUC obtained by including only the previous lamppost to adapt the control

**TABLE 4.** Overall data obtained. Power consumption (Pow. con.) is calculated as the rate between the maximum consumption and the consumption when the control is applied. Pedestrian comfort (Ped. comf.) is calculated as the average of the difference between the pedestrian needs and the light intensity provided. The Area Under the Curve (AUC) shows the values for each number of previous light sources and the improvement percentage. The improvement is calculated as the quotient of the AUC of the next streetlight by the AUC of the current streetlight and expressing the result as a percentage. In the column header the abbreviation ‘Prev.’ corresponds to the term ‘previous’.

Prev. LS.	Pow. Con.	Ped. Comf.	AUC	Improv.
0	38.64%	-69.86	0.8560	–
1	50.09%	-55.31	0.8919	4.19%
2	59.45%	-42.33	0.9037	1.32%
3	67.18%	-30.29	0.9096	0.65%
4	73.18%	-19.33	0.9146	0.55%
5	76.27%	-11.83	0.9186	0.44%
6	77.82%	-10.81	0.9201	0.16%
7	80.55%	-5.83	0.9221	0.22%
8	80.91%	-5.17	0.9244	0.25%
9	80.98%	-5.09	0.9260	0.17%

action to pedestrian speed (LS = 1) is 4.19%. This previous information allows the streetlight to know the pedestrian’s speed in an approximate way, which makes it possible to dispense with the speculation carried out in the algorithm without prior information. The inclusion of the previous information of a second lamppost, LS = 2, implies an increase in the optimization of 1.32% with respect to the use of the information of a single previous lamppost. When information from the third previous lamppost is added to the control algorithm, the increase is less than 1%. Although it depends on the design requirements, considering a value of LS = 2 or LS = 3 seems the most appropriate to not overload the message communications network nor to increase so much the extra processing control time.

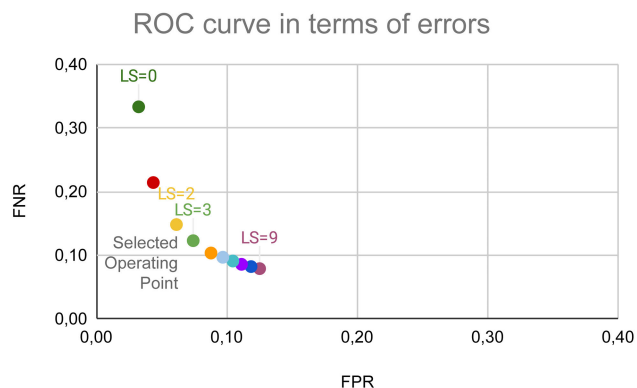
Table 4 shows that there is no option to improve one variable, power consumption, without decrease the other, user comfort, and vice-versa. Consequently, we have a multi-objective problem: minimize consumption and maximize comfort. One method to optimize a multi-objective system is to use the Pareto Front Approximations [38] usually used to optimize a system [39]. With this method, it is possible to determine a region when power consumption and user comfort values can be accepted. In [40], the ROC variables are used to obtain a Pareto Front using the TP, FN, FP, and TN values. Specifically, the variables False Negative Rate (FNR) and False Positive Rate (FPR) are used, shown respectively in the equations 5 and 6.

$$FNR = \frac{FN}{(TP + FN)} \tag{5}$$

$$FPR = \frac{FP}{(TN + FP)} \tag{6}$$

Representing FNR in the y-axis, and FPR in the x-axis, the ROC curve is represented in terms of errors.

Figure 14, shows the curve obtained from the simulations done. As shown in this figure, it is possible to select a point, or a range, when the FNR and FPR rates are minimal.



**FIGURE 14.** ROC curve in terms of errors using the False Negative Rate (FNR) and False Positive Rate (FPR). With this curve, it is possible to select a range or operating point when power consumption and pedestrian comfort values can be considered optimal. The closer an FNR or FPR value is to zero, the more efficient there is in the control algorithm. In this case, the value  $LS = 3$  minimizes both distances.

This point coincides with the LS value of 3; that is, the ROC curve allows adjusting the distributed control system configuration’s value. The ROC curve, in terms of errors, provide useful information. A high FPR implies a high number of street lights switched on, and consequently a high power consumption. As the number of streetlights to be considered decreases, the false-negative rate increases due to the fact that there are streetlights, which are not considered, this is largely due to the road is not illuminated, and only the courtesy light is provided.

**V. CONCLUSION AND FUTURE WORK**

In this article, we have presented how it is possible to use ROC analysis to evaluate which control policy is the most appropriate to a specific context and requirements. We have modeled a street with several luminaries as light sources and with a pedestrian walking down it. We have studied the confusion matrix based on different types of control actions simulated in this environment.

The confusion matrix is suitable as a method to characterize the control action’s efficiency, especially if this control action is based on predictions, as in the case of pedestrian speed. Instead of absolute and relative error, in this case, we use the confusion matrix to determine whether the control action has planned the pedestrian’s position well. We consider that a control action is good when the number of TP and TN is significantly higher than the number of FP and FN. The concept of ‘significant’ in this case refers to a change in the AUC parameter. Thus, when the sensitivity and specificity values approach 1, we know that the control action is close to optimal. In addition, it is possible to tune the algorithm of a control system by deciding whether to be conservative in the action, for example, to illuminate in case of doubt; or specific in the control action, for example, to illuminate only when it is certain that the pedestrian is going to be in the streetlamp action range.

We have tested several control strategies: one, in which light sources respond independently, and another one in which light sources take actions according to the information provided by previous street lamps. We have seen how the communication between several light sources improves their performance. Moreover, other methods can be included that determine if the light bulb is working correctly. When some failure will be expected soon [41], the control system can incorporate a control part for adjusting the behavior of closed light sources.

Initially, we used the ROC curves with a single lamppost to check which control criterion was more suitable to offer a better service. We have tested as a conservative criterion, assuming that the pedestrian’s speed is high, it is more efficient by providing more sensitivity. This result is as expected since the conservative criteria illuminate earlier, which does not affect lighting comfort. The second control strategy tested implies that the control action considers the speed calculated from the pedestrian’s detection in an increasing number of previous streetlights. Since the pedestrian is detected earlier, it is possible to calculate the speed with better precision and, consequently, the specificity increases and the optimization of the service. The results show that the more previous streetlights algorithm considers, the higher is the resulting AUC.

The use of ROC curves, where the success of the control depends on both the successes and the errors, is especially useful in the design of distributed control systems to determine the connections between nodes that are not in motion. In the case presented in the article, the technique is useful for Infrastructure to Infrastructure (I2I) networks. However, it remains to be studied whether if when control nodes are moving, i.e.: in the case of the vehicles, the use of the parameters of both success (TP, TN) and failure (FP, FN) can be more efficient than current adaptive control methods. This aspect is relevant, especially for the Vehicle to Vehicle (V2V) networks or the Vehicle to Infrastructure (V2I) networks.

In this work, we have located light sources along the same line. In the future, we want to study the influence of other physical topology, such as crossroads and roundabouts. The aim will be to study how the environment topology influences the number of necessary connections between light sources. Finally, we also want to study how the ROC curves can determine a control algorithm’s optimal parameters. In this way, given the characteristics of an environment, as the distance between light sources, type, traffic of pedestrians, and so on, the ROC curves can be used to decide the best control algorithm to be used in each case.

**REFERENCES**

- [1] J. M. Klopp and D. L. Petretta, “The urban sustainable development goal: Indicators, complexity and the politics of measuring cities,” *Cities*, vol. 63, pp. 92–97, Mar. 2017.
- [2] V. Albino, U. Berardi, and R. M. Dangelico, “Smart cities: Definitions, dimensions, performance, and initiatives,” *J. Urban Technol.*, vol. 22, no. 1, pp. 3–21, 2015.
- [3] R. P. Dameri, “Searching for smart city definition: A comprehensive proposal,” *Int. J. Comput. Technol.*, vol. 11, no. 5, pp. 2544–2551, 2013.

- [4] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi, "Applications of big data to smart cities," *J. Internet Services Appl.*, vol. 6, no. 1, p. 25, Dec. 2015.
- [5] O. Z. Andrzej adowicz and J. Grella, "Energy saving in the street lighting control system—A new approach based on the EN-15232 standard," *Energy Efficiency*, vol. 10, no. 3, pp. 563–576, 2017.
- [6] A. Das, P. Dash, and B. K. Mishra, "An innovation model for smart traffic management system using Internet of Things (IoT)," in *Cognitive Computing for Big Data Systems Over IoT*. Cham, Switzerland: Springer, 2018, pp. 355–370.
- [7] H. Su, E. Zio, J. Zhang, L. Chi, X. Li, and Z. Zhang, "A systematic data-driven demand side management method for smart natural gas supply systems," *Energy Convers. Manage.*, vol. 185, pp. 368–383, Apr. 2019.
- [8] A. H. Alavi, H. Hasni, N. Lajnef, and K. Chatti, "Continuous health monitoring of pavement systems using smart sensing technology," *Construct. Building Mater.*, vol. 114, pp. 719–736, Jul. 2016.
- [9] Z. A. Baig, P. Szweczyk, C. Valli, P. Rabadia, P. Hannay, M. Chernyshev, M. Johnstone, P. Kerai, A. Ibrahim, N. Syed, N. Peacock, and K. Sansurooah, "Future challenges for smart cities: Cyber-security and digital forensics," *Digit. Invest.*, vol. 22, pp. 3–13, Sep. 2017.
- [10] M. Ryu, J. Yun, T. Miao, I.-Y. Ahn, S.-C. Choi, and J. Kim, "Design and implementation of a connected farm for smart farming system," in *Proc. IEEE SENSORS*, Nov. 2015, pp. 1–4.
- [11] A. Haans and Y. A. W. de Kort, "Light distribution in dynamic street lighting: Two experimental studies on its effects on perceived safety, prospect, concealment, and escape," *J. Environ. Psychol.*, vol. 32, no. 4, pp. 342–352, Dec. 2012.
- [12] B. Chachuat, B. Srinivasan, and D. Bonvin, "Adaptation strategies for real-time optimization," *Comput. Chem. Eng.*, vol. 33, no. 10, pp. 1557–1567, Oct. 2009.
- [13] F.-L. Lian, J. Moyne, and D. Tilbury, "Network design consideration for distributed control systems," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 2, pp. 297–307, Mar. 2002.
- [14] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
- [15] P. Chiradeja, S. Yoomak, and A. Ngaopitakkul, "Economic analysis of improving the energy efficiency of nanogrid solar road lighting using adaptive lighting control," *IEEE Access*, vol. 8, pp. 202623–202638, 2020.
- [16] V. Mehta, "Walkable streets: Pedestrian behavior, perceptions and attitudes," *J. Urbanism*, vol. 1, no. 3, pp. 217–245, Nov. 2008.
- [17] M. Mahoor, F. R. Salmasi, and T. A. Najafabadi, "A hierarchical smart street lighting system with brute-force energy optimization," *IEEE Sensors J.*, vol. 17, no. 9, pp. 2871–2879, May 2017.
- [18] J. F. De Paz, J. Bajo, S. Rodríguez, G. Villarrubia, and J. M. Corchado, "Intelligent system for lighting control in smart cities," *Inf. Sci.*, vol. 372, pp. 241–255, Dec. 2016.
- [19] D. Alarcón, E. Balvís, R. Bendaña, A. Conejero, D. Fernández, J. Higón, J. Isidro, and H. Michinel, "Mathematical modeling and numerical simulation of heat dissipation in led bulbs," *Thermal Sci.*, vol. 24, no. 3, pp. 1877–1884, 2020.
- [20] D. Griggs, M. Stafford-Smith, O. N. Gaffney, J. N. Rockström, M. C. Öhman, P. Shyamsundar, W. Steffen, G. Glaser, N. Kanie, and I. Noble, "Sustainable development goals for people and planet," *Nature*, vol. 495, no. 7441, pp. 305–307, 2013.
- [21] S. Chu, Y. Cui, and N. Liu, "The path towards sustainable energy," *Nature Mater.*, vol. 16, no. 1, pp. 16–22, Jan. 2017.
- [22] M. Frontczak and P. Wargocki, "Literature survey on how different factors influence human comfort in indoor environments," *Building Environ.*, vol. 46, no. 4, pp. 922–937, 2011.
- [23] C. Villa and R. Labayrade, "Energy efficiency vs subjective comfort: A multiobjective optimisation method under uncertainty," in *Proc. Building Simulation Conf.*, Sydney, NSW, Australia, 2011, pp. 1905–1912.
- [24] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, and A. Al-Fuqaha, "Smart cities: A survey on data management, security, and enabling technologies," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2456–2501, 4th Quart., 2017.
- [25] X.-B. Cao, H. Qiao, and J. Keane, "A low-cost pedestrian-detection system with a single optical camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 58–67, Mar. 2008.
- [26] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1037–1045.
- [27] S. Gomez, B. Ivorra, and A. M. Ramos, "Designing optimal trajectories for a skimmer ship to clean, recover and prevent the oil spilled on the sea from reaching the coast," *Appl. Math. Nonlinear Sci.*, vol. 3, no. 2, pp. 553–570, Dec. 2018.
- [28] N. Yoshiura, Y. Fujii, and N. Ohta, "Smart street light system looking like usual street lights based on sensor networks," in *Proc. 13th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Sep. 2013, pp. 633–637.
- [29] G. K. Still, *Introduction to Crowd Science*. Boca Raton, FL, USA: CRC Press, 2014.
- [30] R. Fujdiak, P. Mlynek, J. Misurec, and J. Slacik, "Simulation of intelligent public light system in smart city," in *Proc. Prog. Electromagn. Res. Symp.-Spring (PIERS)*, 2017, pp. 2515–2519.
- [31] J. A. Rincon, J.-L. Poza-Lujan, V. Julian, J.-L. Posadas-Yagüe, and C. Carrascosa, "Extending MAM5 meta-model and JaCallVE framework to integrate smart devices from real environments," *PLoS ONE*, vol. 11, no. 2, Feb. 2016, Art. no. e0149665.
- [32] D. Hristu-Varsakelis, W. S. Levine, R. Alur, K. E. Arzen, J. Baillieul, and T. A. Henzinger, *Handbook of Networked and Embedded Control Systems (Control Engineering)*. Basel, Switzerland: Birkhäuser, 2005.
- [33] T. E. Oliphant, "Python for scientific computing," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 10–20, May 2007.
- [34] J. Ferreira, *Google Apps Script: Web Application Development Essentials*. Sebastopol, CA, USA: O'Reilly Media, 2014.
- [35] E. Papadimitriou, G. Yannis, and J. Golias, "A critical assessment of pedestrian behaviour models," *Transp. Res. F: Traffic Psychol. Behav.*, vol. 12, no. 3, pp. 242–255, 2009.
- [36] S. Fotios, J. Unwin, and S. Farrall, "Road lighting and pedestrian reassurance after dark: A review," *Lighting Res. Technol.*, vol. 47, no. 4, pp. 449–469, Jun. 2015.
- [37] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [38] T. Tušar and B. Filipič, "Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the projection method," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 225–245, Apr. 2015.
- [39] L. Torralba-Morales, G. Reynoso-Meza, and J. Carrillo-Ahumada, "Tuning and comparison of design concepts applying Pareto optimality. a case study of cholette bioreactor," *Revista Iberoamericana de Automática e Inform. Ind.*, vol. 17, no. 2, pp. 190–201, 2020.
- [40] R. Aler, J. Handl, and J. D. Knowles, "Comparing multi-objective and threshold-moving ROC curve generation for a prototype-based classifier," in *Proc. 15th Annu. Conf. Genetic Evol. Comput. Conf. (GECCO)*, 2013, pp. 1029–1036.
- [41] W. Zhao, T. Shi, and L. Wang, "Fault diagnosis and prognosis of bearing based on hidden Markov model with multi-features," *Appl. Math. Nonlinear Sci.*, vol. 5, no. 1, pp. 71–84, Jan. 2020.



**JOSE-LUIS POZA-LUJAN** (Member, IEEE) was born in València, Spain, in 1969. He received the B.S. degree in science and mathematics education from the Universitat de València, Valencia, in 1990, and the B.S. degree in systems and computing and the Ph.D. degree in computer science from the Universitat Politècnica de València, Valencia, in 1999 and 2012, respectively.

Since 2002, he has been a Lecturer and a Researcher at the UPV. He is currently the Vice-Dean of Communication and Promotion of the School of Informatics, UPV. He is responsible of the master's subjects soft skills and agile project management. He has led more than 100 final degree projects. He has participated in different stays in HVA (The Netherlands), TUAS (Finland), and Telecom (France). He has published 13 articles in indexed journals, two books, two book chapters, and 107 articles in congresses. He has participated in 22 technological projects and in 11 projects of educational innovation. He has been the Project Co-ordinator of a university project of education innovation focused on the cross-curricular competencies in the ICT degree. He has been the Main Researcher and Project Co-ordinator of the CoSki21 Project, and a Technical Co-ordinator of a European Project of the H2020 Program. His research interests include distributed systems architectures, intelligent devices, project management, and soft skills. He received the prize for teaching excellence of the Schools of Informatics in 2007. He received the Prize for Teaching Excellence, recognized by the National Spanish Agency in 2013.



**JUAN-JOSÉ SÁENZ-PENAFIEL** was born in Cuenca, Ecuador, in 1989. He received the degree in systems engineering from the Universidad de Cuenca, Ecuador, in 2014, and the M.Sc. degree in computer engineering and networks from the Universitat Politècnica de València, Spain, in 2018, where he is currently pursuing the Ph.D. degree. He is also a Data Analyst and a Journals Administrator at Research Direction of the Universidad de Cuenca, Ecuador. He has participated in research projects, since 2015, as a Research Assistant at the University of Cuenca. He has published one article in an indexed journal articles and four papers in conferences. He is also responsible for research data at the University of Cuenca. His research interests include distributed systems and artificial intelligence.



**JUAN-LUIS POSADAS-YAGÜE** received the M.Sc. and Ph.D. degrees in computer engineering from the Universitat Politècnica de València, Spain, in 1996 and 2003, respectively. He has taught computer architecture at the School of Engineering in Computer Science, since 1997. He is currently an Associate Professor with the Department of Computer Engineering, Universitat Politècnica de València. He is also the Director responsible of the teaching management with the

Department of Computer Engineering. His research interests include distributed systems, real-time mobile robot control and coordination, code delegation, multi-agent systems, and embedded systems.



**J. ALBERTO CONEJERO** received the degree in mathematics from the Universitat de València (UV), and the Ph.D. degree from the Universitat Politècnica de València (UPV), Spain, in 2004, receiving the Outstanding Dissertation Award. Since 2009, he has been an Associate Professor at UPV. He has been an Invited Researcher of Bowling Green St. University (USA), Kent St. University (USA), Università del Salento (Italy), Universität Tübingen (Germany), and Czech Academy of Sciences (Czech Republic). He has been awarded with the Teaching Excellence Prize of UPV in 2014. He is currently the Director of the Department of Applied Mathematics, UPV, and responsible of the M.Sc. Program on Mathematics Research at UPV. He is the author of more than 60 articles in international journals. His research interests include dynamical systems, partial differential equations, graph theory, and the multidisciplinary applications of mathematics to computer science, engineering, and biotechnology.



**JUAN-CARLOS CANO** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the Polytechnic University of Valencia (UPV), Spain, in 1994 and 2002, respectively. From 1995 to 1997, he worked as a Programming Analyst at the IBM's Manufacturing Division, Valencia. He is currently a Full Professor with the Department of Computer Engineering, UPV. His current research interests include vehicular networks, mobile ad hoc networks, and pervasive computing.

...