This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Simulation of Fuzzy Queueing Systems with a Variable Number of Servers, Arrival and Service Rates

Enrique Muñoz
Enrique H. Ruspini, *Life Fellow, IEEE*

***Abstract*—Fuzzy Queueing Theory (FQT) is a powerful tool to model queueing systems taking into account their natural imprecision. The traditional FQT model assumes a fixed number of servers with constant fuzzy arrival and service rates. It is quite common, however, to encounter situations where the number of servers, the arrival rate, and the service rate change with time. In those cases the variability of these parameters can significantly impact the behavior of the queueing system and complicate its analysis. In this paper we propose a simulation method to deal with these problems that leads to realistic estimations of the behavior of the queueing system.**

To demonstrate the applicability of the technique we have employed it to solve a real world problem: the prediction of the impact of electric vehicles on power networks. Electric vehicles have recently become a popular topic for research and development, mainly for their potential ability to reduce emissions and dependence on fossil fuels. Their demanding charging requirements can, nevertheless, significantly impact the performance of the power grid. In this paper we propose to model the recharging process as a fuzzy queueing problem. This approach is more realistic than previous treatments while leading to accurate descriptions of the performance of the system.

***Index Terms*—Fuzzy Queueing Theory, Fuzzy Simulation, Electric Vehicles, Power Grid, Electric Demand.**

I. Introduction

QUEUEING theory (QT) [1] is the branch of mathematics that studies waiting lines or queues. A queue is formed when "clients" arrive to a "place" asking for a service to a "server" with limited resources. If the server is not immediately available the client needs to wait and join a waiting line. The use of QT allows the study of different processes associated to queues including arriving at the queue, waiting in it, and being served. The wide range of applications of QT, including traffic flow, telecommunications and facility design, gives a clear idea of its ability to answer these questions.

One drawback of QT is that it relies on models with parameters that are assumed to be known exactly. In the majority of real-world applications, however, there is substantial uncertainty about their values [2]. For this reason QT has been extended using fuzzy-set theory resulting in the field known as Fuzzy Queueing Theory (FQT). Unfortunately, while FQT provides powerful tools to deal with uncertainty, many issues that have been addressed in traditional QT have been scarcely studied by FQT. This is the case of queueing systems with a varying number of servers, arrival and service rates, which are the focus of this paper. While there have been previous efforts to apply fuzzy-system approaches to discrete-time, time-delay, estimation and control problems [3], [4], previous studies have not employed FQT to study queueing systems combining these three characteristics.

Problems characterized by such a combination of time-dependent parameters are, in general, not amenable to mathematical analysis, whether applying traditional QT or FQT. For this reason, a simulation method that randomly generates arrivals and user characteristics was developed to gain insights into the behavior of these complex systems and the estimation of measures of system performance. Arrival and service rates are modeled as time-varying Poisson processes with fuzzy parameters while the number of servers can change during execution time depending on certain problem-specific restrictions.

In order to explore the possibilities offered by this simulation methodology it was applied to a complex real world problem: the analysis of the potential impact of Electric Vehicles (EVs) on power consumption and generation. EVs rely upon batteries with high storage capacity to provide energy to their engines. The need to provide adequate driving range between recharges results in significant charging demands, which, as the number of EVs increases, are expected to have considerable impact on electric power system design and operation [5]. While this problem has been frequently studied in the recent past, the treatments have not been realistic and do not consider the complexities that motivated the development of the simulation techniques presented in this paper. In particular, the methods presented in this work have focused on the design of an intelligent battery charger capable of recharging the battery in a reasonable period of time while avoiding overload of the electric network. The methodology was applied to two possible future scenarios, corresponding to the years 2014 and 2020, for EV-related demand in Spain.

This paper is organized as follows. Section II briefly discusses time-varying queueing systems with a variable number of servers, arrival and service rates. The simulation method is introduced in Section III. Section IV presents a case study of the impact of EVs on the power grid. A real-world scenario based on models of the Spanish power grid is analyzed in Section V. Simulation results are shown in Section VI, while Section VII presents the research conclusions.

E. Muñoz and E. H. Ruspini are with the Laboratory of Collaborative Intelligent Systems, European Centre for Soft Computing, Mieres, Asturias, Spain. e-mail: {enrique.munoz, enrique.ruspini}@softcomputing.es.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

II. Queuing systems with time-varying parameters

The queueing models typically considered by QT are based on a number of assumptions about the behavior of users and the nature of the distributions modeling their arrival rates that do not match the complexity of many common problems. In many important applications, even when the arrivals are continuous and their inter-arrival times are independent, the system model parameters change with time and may be dependent on the characteristics of particular users and servers. A 24-hour calling service, for example, usually receives more calls during the day than late at night. The system may also need to support a mix of users that have changing characteristics and needs that are attended by multiple servers of dissimilar nature and number. A calling center may employ more responders, with a different mix of skills, from 8:00 to 14:00, than, say, from 0:00 to 6:00.

The complexity of these systems leads to models that are more difficult to analyze using mathematical tools as evidenced by the lack of literature on queues with variable number of servers and rates as compared with that dealing with the behavior of queues with constant parameters. Massey [6], Wang et al. [7], and Green et al. [8], among others, have addressed issues related to queues having rates that change with time. Queueing systems with variable number of servers have been considered by Brodi et al. [9] and Ivaneshkin [10]. Going one step further and combining all three problems considerably complicates matters [11], [12].

Beyond these issues, the use of crisp parameter values, such as those of the exponential distributions modeling arrival rates, is not realistic in a majority of applications since it is difficult to collect the data required to make accurate objective estimates, which are often provided subjectively by an expert. These considerations have led to models where arrival and service patterns are described using linguistic terms or fuzzy numbers. Although this type of approach leads to more realistic representations and has a larger number of possible applications than conventional treatments, few studies have been published on the topic [13]. The first effort in this direction was that of Prade [14], who pointed out that, in practical situations, most of the parameters are not known exactly. In the same sense the work of Li et al. [15] also stresses the necessity of using fuzzy queueing systems to obtain more realistic results. Further following this idea, Buckley [16], who later extended his work to fuzzy queueing systems[17], introduced the concept of uncertainty in QT discussing elementary queueing systems where arrivals and departures follow a possibility pattern. Additional approaches include those proposed by Kao et al. [18] and Chen et al. [19], which adopt parametric programming methods to solve different queueing problems, and that of Pardo et al. [13], which dealt with queueing systems where the length of the queue is limited and where both arrival and service rates are fuzzy numbers. Negi and Lee [20] noted, however, that analytical approaches can be very complicated and are generally unsuitable for predictive purposes, leading them to propose a simulation method in order to analyze fuzzy queues. On the same note, Wu [21] introduced a simulation technique able to deal with fuzzy queueing problems with time-dependent arrivals but that cannot cope with a variable number of servers nor with variations in service rates.

III. Simulation of a queueing system with a variable number of servers, arrival and service rates

The simulation method proposed in this work extends that introduced by Wu [21]. The main contribution of this new method is the consideration of time-varying service rates, and the dynamic modification of the number of servers. It will be assumed that arrivals follow a time-changing Poisson process with fuzzy arrival rate $\tilde{\lambda}(t)$. The service rate for a customer that starts service at time t is also modeled by a varying Poisson process with fuzzy rate $\tilde{\mu}(t)$. The number of servers c(t) is also function of time.

A. Notation and definitions

A fuzzy subset $\tilde{a}$ of $\mathbb{R}$ is defined by its membership function $\tilde{a}\colon \mathbb{R} \to [0,1]$. A fuzzy number [22] is a fuzzy subset of $\mathbb{R}$ that satisfies the following conditions:

(i) $\tilde{a}$ is normal, i.e., $\exists x \in \mathbb{R}$ such that $\tilde{a}(x) = 1$,
(ii) the alpha-cuts of $\tilde{a}$ are closed intervals of $\mathbb{R}$ for all $\alpha \in [0,1]$,
(iii) the support of $\tilde{a}$ is bounded.

We denote by $\{\oplus, \ominus, \otimes, \oslash\}$ the binary operations between two fuzzy numbers $\tilde{a}$ and $\tilde{b}$, representing fuzzy addition, subtraction, multiplication and division, respectively. In what follows, we will use these binary operators to combine also fuzzy and crisp numbers assuming that the latter correspond to the classical characteristic function of a crisp set. We denote by $\mathcal{F}(\mathbb{R})$ the set of all fuzzy numbers. Finally, a function $\eta : \mathcal{F}(\mathbb{R}) \to \mathbb{R}$, mapping fuzzy subsets into real numbers is called a *defuzzification function* if $\eta(\chi_m) = m$, where χ_m is the (crisp) characteristic function of the singleton $\{m\}$ and m is a real number.

B. Poisson processes with variable fuzzy arrival rate

In order to understand the simulation method presented in this paper it is necessary to clarify the concepts of counting and Poisson processes. A counting process is a stochastic process $\{N_t\}_{t\geq 0}$ where N_t represents the number of events that have occurred up to time t. A Poisson process is a counting process such that the times between two consecutive events (inter-arrival times) are independent random variables that follow an exponential probability distribution with parameter λ.

If X_i is the i-th inter-arrival time in a Poisson process, its cumulative distribution function is $F(x) = 1 - \mathrm{e}^{\lambda x}$. We resort, as customary, to the application of the inverse function $F^{-1}(u) = -\frac{1}{\lambda}\ln(1-u)$ to generate random inter-arrival times having an exponential distribution, where u is an uniformly-distributed random number.[1]

This generation procedure must be modified to deal with dynamic variations in the arrival rate. Let $\lambda(t)$ be the arrival

[1]In what follows the term "random number" is employed to indicate a random number uniformly distributed in $[0,1]$.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

rate and let λ be an upper bound of $\lambda(t)$, that is $\lambda(t) \leq \lambda$, for all $t \geq 0$. Following the work of Wu [21], we consider a Poisson process P with constant parameter λ, and define a counting process Q to be a time-varying Poisson process with parameter $\lambda(t)$ if whenever an event occurs in process P, it occurs in Q with probability $\lambda(t)/\lambda$. In order to generate inter-arrival times for this case we resort to an iterative scheme such as that given in Algorithm 1, which iteratively increments the inter-arrival time until the probability $\lambda(t)/\lambda$ is greater or equal than a random number.

Algorithm 1: Pseudo-code used to generate inter-arrival times in crisp QT.

```
Input: t: current time, λ(t): arrival rate, λ:
       maximum arrival rate.
Output: next inter-arrival time.
begin
    s = t;
    repeat
        u = generate random number;
        s = s - (1/λ) ln u;
        v = generate random number;
    until v ≤ λ(s)/λ;
    return s-t;
end
```

To generalize this algorithm to fuzzy arrival rates, we employ Zadeh's extension principle [23] and the procedure proposed by Wu et al. [21], thus leading to the formula $g(x) = \eta(-\ln u \oslash \tilde{\lambda})$, to generate inter-arrival times following a Poisson process with constant arrival rate, where $\tilde{\lambda}$ is the fuzzy arrival rate and where u is a random number.

To extend this procedure to time-varying Poisson processes, let $\tilde{\lambda}(\tilde{t})$ be the arrival rate, and let $\tilde{\lambda}$ be its upper bound, with $\eta(\tilde{\lambda}(\tilde{t})) \leq \eta(\tilde{\lambda})$ for all $\tilde{t}$ with $\eta(\tilde{t}) \geq 0$. Note that we assume here, for the sake of generality, that the time $\tilde{t}$ is fuzzy so as to be able to model events that occur *around* some time. The procedure is, however, still applicable to crisp times. We consider then a Poisson process P with constant fuzzy parameter $\tilde{\lambda}$, and define a counting process Q to be a time-dependent Poisson process with fuzzy parameter $\tilde{\lambda}(\tilde{t})$ if whenever an event occurs in process P, it occurs in Q with probability $\eta(\tilde{\lambda}(\eta(\tilde{t})))/\eta(\tilde{\lambda})$. Following a process analogous to that used for the crisp case we can use the simulation procedure shown in Algorithm 2 to generate inter-arrival times. This algorithm iteratively increments the inter-arrival time until the probability $\eta(\tilde{\lambda}(\eta(\tilde{t})))/\eta(\tilde{\lambda})$ is greater or equal than a random number.

C. Variable number of servers

The modeling and simulation of a variable number of servers requires the consideration of additional issues, such as the specification of the end-of-shift policy employed to handle the customers that they are serving [12]. Two different disciplines can be considered: *pre-emptive* or *exhaustive*. When employing the *pre-emptive* discipline a customer in service is sent back to the queue if the server is scheduled to leave, while in the *exhaustive* discipline it is assumed that servers always complete the current job before leaving. If an estimate of the service time, or the actual value, is available, it is possible to use a third discipline, that we will call *refuse* discipline. In this case a server only accepts a customer if the server will not leave before the service process is completed. This new discipline can be useful when service times are known in advance or when servers are not allowed to stay longer than their scheduled time, as is the case in the problem that will be studied in Section IV. The simulation techniques discussed in this paper are able to deal with these three disciplines.

Algorithm 2: Pseudo-code used to generate inter-arrival times in FQT.

```
Input: t̃: current time, λ̃(t): arrival rate, λ̃:
       maximum arrival rate.
Output: next inter-arrival time.
begin
    s̃ = t̃;
    repeat
        u = generate random number;
        s̃ = s̃ ⊕ (− ln u ⊘ λ̃);
        v = generate random number;
    until v ≤ η(λ(η(s̃)))/η(λ̃);
    return s̃ ⊖ t̃;
end
```

Turning now our attention to the modeling of a variable number n of servers let $\tilde{s}_i$, $\tilde{f}_i$, $i = 1, \ldots, n$, denote the times when a server starts and ends being available, respectively. Handling of users by servers is dependent on the particular end-shift discipline:

(i) when a *pre-emptive* discipline is used, the user is directly assigned to the server. When the server is scheduled to leave, the user is reinserted in the queue and its service time is reduced by the amount of time spent on the server
(ii) if an *exhaustive* discipline is employed, the user is assigned as in previous case but the server remains operational until the service is provided
(iii) if a *refuse* discipline is utilized, it is necessary to check, before assigning a user to a server, if the server will be able to provide in time the total service required. If that is the case, then the user is assigned to the server, otherwise the user waits in the queue.

In some problems, such as the one discussed in Section IV, additional constraints further restrict the assignment of users to servers. In applications involving power generation, for example, constraints on maximum power consumption or on the rate of consumption may prevent the addition of new servers.

D. Simulation techniques

The simulation method is based on the random generation of five classes of random events: arrivals, departures, allocations, start-of-server-shifts and end-of-server-shifts. Arrivals are sequentially generated by means of Algorithm 2, employing a time-varying exponential distribution with fuzzy parameter $\tilde{\lambda}(t)$. In a similar fashion service times are generated, whenever specific users are assigned to a server, following a time-dependent exponential distribution with fuzzy parameter $\tilde{\mu}(t)$. Allocation events are created whenever users can be assigned

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

to a server taking into account applicable service restrictions. Allocation and service times, together with service policies and restrictions, determine when a user leaves the system, thus resulting in a departure event. Start-of-server-shift and end-of-server-shift events are generated at times $\tilde{s}_i$, $\tilde{f}_i$.

The skeleton of the simulation method, described by Algorithm 3, features an initialization procedure and a loop that handles the next event according on its type. The initialization procedure entails setting initial values for the variables $\tilde{t}$, $\tilde{T}$, $\tilde{t}_A$, $\tilde{t}_D$, $\tilde{t}_L$, $\tilde{t}_S$ and $\tilde{t}_F$, that represent current time, end time, next arrival time, next departure time, next allocation time, next time when a server starts working, and next time when a server stops working, respectively. Other important variables that do not appear explicitly in Algorithm 3 but that are processed by its ancillary techniques are $Queue$, which represents the list of users that are waiting to receive service independently of the chosen queue policy, $Service$, which is the list of users that are using a server, $\tilde{\lambda}(t)$, which corresponds to the parameter of the time-varying exponential distribution that controls inter-arrival times; and $\tilde{\mu}(t)$, which represents the fuzzy rate of the exponential distribution that generates service times.

Algorithm 3: Pseudocode for the simulation method.

```
begin
    initialize();
    while η(t̃) < η(T̃) do
        if η(t̃_A) < η(t̃_D) AND η(t̃_A) < η(t̃_L) AND
        η(t̃_A) < η(t̃_S) AND η(t̃_A) < η(t̃_F) then
            handleNewArrival();
        end
        else
            if η(t̃_D) < η(t̃_L) AND η(t̃_D) < η(t̃_S) AND
            η(t̃_D) < η(t̃_F) then
                handleNewDeparture();
            end
            else
                if η(t̃_L) < η(t̃_S) AND η(t̃_L) < η(t̃_F) then
                    handleNewAllocation();
                end
                else
                    if η(t̃_S) < η(t̃_F) then
                        handleNewServerStart();
                    end
                    else
                        handleNewServerFinish();
                    end
                end
            end
        end
    end
end
```

The procedure to handle a new arrival is described by Algorithm 4. Users are first added to the queue but are not assigned to servers until it is determined that free servers are available and that assignation to a free server will not violate problem-specific restrictions. If there is a free server the algorithm generates an allocation time that meets such constraints. Section IV presents an example of such calculation. The next arrival is then generated by the time-varying Poisson process.

Allocation events are managed by Algorithm 5. This process simply involves computing the service time for the first user in the queue according to the queueing discipline and the time-varying nature of service time. If the end-of-shift policy is *pre-emptive* or *exhaustive*, the user can then be assigned to the server without any further considerations. If, on the other hand, the policy is *refuse*, each free server is checked to determine if it is able to provide service for the required service time. In that case the user is allocated to the server, otherwise an allocation event is not generated. The allocation process is continued if there are still waiting users and free servers capable of satisfying the problem-specific constraints.

Algorithm 4: Pseudocode for the handling of arrivals.

```
begin
    Add task to Queue;
    if There is any free server then
        t̃_L=GenerateAllocationTime();
    end
    t̃_A=Generate next arrival time using
    exponential distribution with parameter λ̃(t);
end
```

Algorithm 5: Pseudocode for the handling of allocations.

```
begin
    Obtain first task from Queue;
    if task has not been previously allocated then
        Generate service time using exponential
        distribution with parameter μ̃(t);
    end
    if Policy is pre-emptive or exhaustive then
        Remove task from Queue;
        Add task to Service;
    end
    else
        if If any free server can guarantee full service then
            Remove task from Queue;
            Add task to Service;
        end
    end
    if Queue is not empty then
        if There is any free server then
            t̃_L=GenerateAllocationTime();
        end
    end
end
```

Algorithm 6 handles departure events. Users that have been served are eliminated from the list of users of the corresponding server after recording important data about their service, including arrival, waiting, and service times. Servers scheduled to leave are also removed from the list of available servers. If the server remains available, then it is assigned to the first user in the queue taking into account end-of-shift policy constraints. The procedure ends by computing the next departure time.

Algorithm 7 shows the procedure employed to manage the start of a server's shift. New servers are added to the service queue and the value of $\tilde{t}_S$ is updated to record the time when the sever becomes available, thus creating a new start-of-server-shift event. Finally, if there are users in the queue, a new allocation event is generated since a new server has been created and, hence, it is free.

Algorithm 8 manages the end of a server's shift. If a server is free, it is removed from the list of available servers. If the

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

Algorithm 6: Pseudocode for the handling of departures.

```
begin
    Store task data;
    Remove task from Service;
    if server was scheduled to finish its shift after this task then
        Remove server from the list of available
        servers;
    end
    else
        if Queue is not empty then
            Obtain first task from Queue;
            if task has not been previously allocated then
                Generate service time using
                exponential distribution with
                parameter μ̃(t);
            end
            if Policy is pre-emptive or exhaustive then
                Remove task from Queue;
                Add task to Service;
            end
            else
                if If any free server can guarantee full service then
                    Remove task from Queue;
                    Add task to Service;
                end
            end
        end
    end
    t̃_D=finish time of the next finishing task;
end
```

Algorithm 7: Pseudocode for handling the start of a server's shift.

```
begin
    Add the server to the list of available
    servers;
    t̃_S = moment at which next unavailable server
    starts its shift;
    if Queue is not empty then
        t_L=GenerateAllocationTime();
    end
end
```

server is not free, the treatment of the event is dependent on the end-of-shift discipline. In the case of an *exhaustive* discipline, the service continues until its scheduled termination time, when the server is removed. If, on the other hand, the chosen discipline is *pre-emptive*, the server is removed and the user is returned to the queue although its service time is reduced by the service time already provided.[2] Finally, the value of $\tilde{t}_F$ is updated to determine the time of a new end-of-server-shift event.

Information collected during the simulation permits to compute service-quality measures such as the size of the queue and the number of servers at specified points in time.

E. Computational complexity

Estimation of the computational complexity of the simulation procedure presented in this paper is an important issue which, unfortunately, is difficult to treat in a formal way for the same reasons that prevent the derivation of mathematical expressions describing the behavior of such a complex system.

[2]When the discipline is *refuse* it is impossible for a server to be occupied when its removal time arrives.

Algorithm 8: Pseudocode for handling the end of a server's shift.

```
begin
    if server is free then
        Remove server from the list of available
        servers;
    end
    else
        if Policy is exhaustive then
            Schedule server to finish its shift
            after task completion;
        end
        else //The policy is pre-emptive
            Reduce task's service time by the
            amount of time spent in the server;
            Add the task to Queue;
            Remove server from the list of
            available servers;
        end
    end
    t̃_F = moment at which next available server
    finishes its shift;
end
```

Furthermore, the algorithms described in previous sections are stochastic in nature as they rely on Montecarlo approaches and, as such, their own behavior can only be adequately described through statistical measures. In consideration of these difficulties an empirical complexity estimation has been carried out in the context of the problem of prediction of the impact of EV recharging on power networks. The results of this study are discussed in Section VI-E.

IV. A CASE STUDY: IMPACT OF ELECTRIC VEHICLES ON POWER GENERATION AND TRANSPORTATION

This Section is devoted to the application of the modeling and simulation approach discussed in this paper to the prediction of the impact of EV recharging on the Spanish electric network. The particular problem motivating this study was the design of a system of intelligent battery chargers capable of recharging the battery in a reasonable period of time while avoiding overload of the power network. To meet the latter requirement the chargers need to know the non-EV and EV demand at every instant in time. Because of design considerations the chargers cannot communicate with any agent capable of delivering this information thus necessitating the a-priori production of detailed, reliable, and timely information about network load and user behavior.

A. Previous studies of the impact of EVs

Growing concerns about air pollution and climate change have motivated the search for transportation solutions that reduce emissions and dependence on fossil fuels. EVs have been developed and produced as a solution to this problem [24], [25], [26]. The main disadvantage of EVs is, however, their dependence on batteries with high energy storage capacity and significant recharging requirements. These requirements may potentially lead to major demands for power as the number of operational EVs increases.

The problem of analyzing the potential impact of EVs on power consumption and generation has been studied in the

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

recent past although the underlying approaches have ignored significant aspects of the recharging process. An interesting review of this topic may be found in the work of Wang et al. [27]. Early works on this problem assumed that users would prefer to recharge their vehicles during off-peak hours, that is, when the global power demand is lower, as exemplified by the studies of Denholm and Short [28], Scott et al. [29], and Parks et al. [30], who assumed that the recharging process would be controlled by energy producers. It is reasonable, however, to assume that drivers will plug their EVs when it is more convenient for them. In this regard the works of Hadley and Tsvetkova [24] and of Hadley [24] are more realistic. These approaches, however, rely on simple assumptions to approximate EV energy requirements such as considering that all vehicles will recharge at the same time starting from completely depleted batteries. More recently, authors have obtained more accurate estimates [31], [32] using either analytic or stochastic methods, which take into account user habits represented by distributions of arrival times and average trip distance. In spite of these advances methodological limitations of these approaches strongly suggest that the problem might be better approached by sophisticated approaches such as FQT.

Fuzzy Queueing Theory provides better tools to handle the multiple sources of uncertainty that characterize the EV recharging problem such as the nature of the client-arrival process and the recharging time needed by each user. These processes depend on time, since, for example, it is reasonable to expect that most users will start charging their EVs when they arrive home at night. In this work we estimate parameters associated with such uncertainties because of the current lack of empirical evidence on the behavior of EV users.

In addition, the number of available servers also varies with time, being limited by two variables:

(i) the maximum power demand that the grid can support, which is a function of the total amount of power capable of being safely allocated to all energy users,
(ii) the maximum power slope, which, again, reflects limitations of the power system in rapidly adapting to surges in user demand.

These constraints demand a complex, realistic, simulation approach based on the application of techniques such as those provided by FQT.

B. Analysis of the impact of EVs

In EVs the battery is the only device capable of storing energy being, at the same time, the component with the highest cost, weight and volume [26]. A battery converts chemical energy into electrical energy required to power the vehicle systems. Battery recharges reverses this process thus consuming electrical energy.

The need to recharge EVs adds to existing requirements imposed on the power grid by other consumers of electrical energy. Existing electrical power systems have been designed to respond to instantaneous consumer demand. This non-EV demand depends on factors such as temperature, time of day, or season. Figure 1 shows a curve depicting typical variations in the hourly demand to the Spanish power network. Two valleys may be observed, at night and afternoon, when demand falls due to lesser consumer activity, while demand peaks when users are more active in the morning and evening. In addition, this curve also shows an estimate of the total demand (EV plus non-EV) on the network.

This Section is concerned with the application of FQT to estimate the global impact, at a country or state level, of EV demand placing emphasis on undesirable consequences, such as inability of the power system to meet EV recharging requirements.

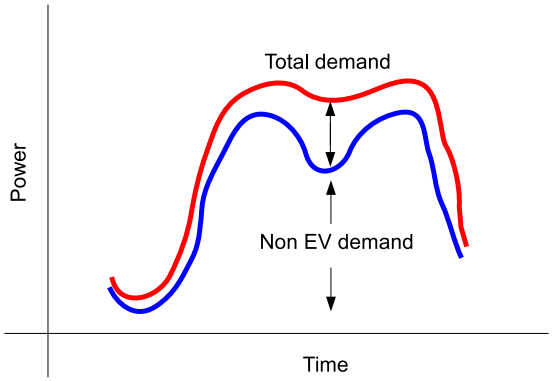

Fig. 1. Typical non-EV Demand and typical Total Demand for Spain.

C. Modeling the EV recharging process as a queueing problem

To model the process of recharging EVs it is necessary to analyze its characteristics, which include the nature of the arrival process, the service time distribution, the number of servers, the number of places in the system, the system population, and the queue discipline. It is assumed that there exists a central controller capable of delaying recharges. This controller is assumed to know the energy required by each user (and, therefore, its expected service time) and the non-EV demand at any moment in time, including future accurate predictions. This process is illustrated in Figure 2.

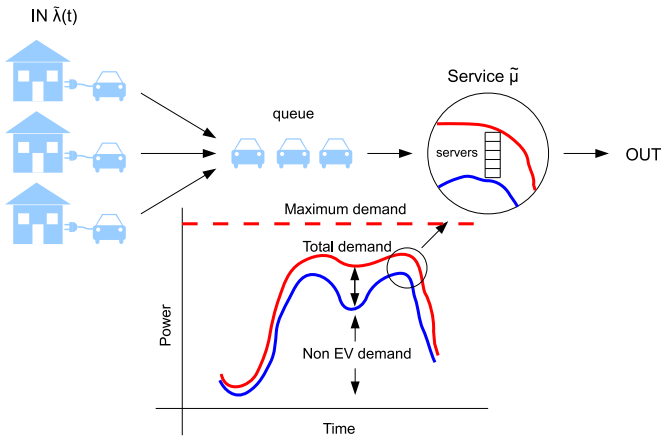

Fig. 2. EV recharging process as a queueing problem.

It is assumed that a new user arrives to the system when he plugs his EV to recharge its battery. Inter-arrival times are

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

assumed to be independent random variables that follow an exponential distribution with a time-dependent fuzzy parameter $\tilde{\lambda}(t)$. It is also assumed that the service required is that needed to completely recharge the battery from its state at arrival time. Service times are assumed to be random variables following a fuzzy exponential distribution $\tilde{\mu}$.

In our treatment each recharger is a virtual entity corresponding to the amount of power available to recharge some EV for the required time (i.e., the service time). All servers are assumed to be capable of delivering the same amount of power. At any moment in time only some servers may be actually available as the electrical network imposes restrictions related to its maximum allowable power demand and maximum allowable instantaneous changes to power demand (i.e., maximum slope).

The times when servers start and end of their shift depend on the non-EV demand n, and maximum allowable demand, d_{max}. The number of servers available at each moment is the difference between non-EV demand and maximum demand divided by the server power s, i.e., $\frac{d_{max}-n}{s}$. At the beginning of the simulation the number of available servers depends on the initial difference between maximum demand and starting non-EV demand. If, later, this difference increases as much as that of the power capable of being delivered by a server, then that server becomes available. Correspondingly, if the difference decreases enough a server is scheduled to finish its shift. The server being removed is chosen using a last-in first-out discipline, that is, the last server made available is the first that leaves. This procedure simplifies the allocation of tasks with a long service time. The total number of servers is the maximum possible value of available servers.

Allocation of users to servers depends on the total demand for power (including both non-EV demand and EV) and maximum slope allowable for that demand. When a user enters the system it is allocated an available server as soon as the increment in power demand does not exceed the maximum allowable power slope, m_{max}. Algorithm 9 shows the procedure that assures that this constraint is not violated. The instantaneous slope of the demand curve is given by $m = \frac{\Delta d(t)}{\Delta t}$, where $d(t)$ and t represent total instantaneous demand and time, respectively. The method searches for the smallest Δt that makes m less or equal than m_{max}, or $\Delta t \geq \frac{\Delta d(t)}{m_{max}}$. Straightforward solution of this inequation is, however, not simple since $\Delta d(t)$ depends on t. To solve it the Algorithm iteratively recalculates Δt until it converges to a stable value. This iterative search starts from the last server departure or assignation of a user to a server since that was the last time at which demand changed. Note that it is not possible to allocate a user to a server before the current time, so Δt will be, at least, the difference between the current time and the last time at which demand changed. The *refuse* end-of-shift policy was chosen in our simulation due to the need to avoid stability problems if strict maximum demand constraints are violated.

In this problem there are no restrictions on the maximum number of customers allowed in the system including those being serviced. Similarly there are no constraints on the size of the user population although the number of EV owners may be bounded. This assumption is valid since it is reasonable to expect that, during the whole simulation, the actual number of users in the queue will be much lower than the total number of EVs. This conjecture was, in fact, found to be true in our simulations as the number of users in the queue never went above 30% of the total number of EVs.

To close this Section we must note that our simulations employed a slightly modified version of the first-come first-served queueing discipline, which selects the first user in the queue who does not violate any restriction. This policy guarantees that the queue will not be blocked by users with very demanding requirements while being also fairer to those users who arrived first, since, as soon as they do not violate any condition, they can start charging.

Algorithm 9: Pseudocode for the calculation of the allocation time of a user to a free server.

```
begin
    lastAllocationDemand = Obtain Non-EV Demand at
    the time when last task was allocated;
    lastAllocationDemand += Number of servers *
    Power used by a server;
    demandAfterAdding = currentDemand + Power used
    by a server;
    while possible time is not stable do
        possible time =
        t̃ ⊕ ((demandAfterAdding − lastAllocationDemand) / (maximum demand increment));
        demandAfterAdding = prediction of demand
        at time possible time + (Number of server
        + 1) * Power used by a server;
    end
    return possible time;
end
```

V. Study of the Impact of EVs on Spanish Electric Network

The estimation of the potential impact of EVs on the Spanish electric network requires the estimation of several model parameters, including non-EV demand, arrival rates, and service times.

A. Prediction of non-EV demand

The first parameter studied was the electric demand in environments where EVs are not currently present. REE is the Spanish organization in charge of controlling power networks. This organization periodically publishes information about hourly demand to the power grid. From this historical database, we extracted daily demand data for every day between 2005 and 2010. Each data point was a 24-dimensional vector representing hourly demand during the day. Application of fuzzy clustering techniques [33], specifically fuzzy c-means [34], produced a fuzzy partition with six clusters having prototypes shown on Figure 3.

Cluster 6, corresponding the highest demand scenario, was employed in the simulation. A set of 24 Mamdani rules, each corresponding to one hour of the day, was generated to represent the prototype of this cluster. The antecedent of each

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

TABLE I
PERCENTAGES OF DISPLACEMENT BY TIME FRAME.

	From 4:01 to 7:00	From 7:01 to 8:00	From 8:01 to 9:00	From 9:01 to 12:00	From 12:01 to 15:00	From 15:01 to 18:00	From 18:01 to 21:00	From 21:01 to 0:00	From 0:01 to 4:00
Total	3.6%	6.5%	8.7%	14.4%	20.9%	21.4%	19.2%	4.7%	0.8%
Home	4.6%	2.0%	6.6%	31.7%	67.7%	38.8%	67.1%	74.9%	83.6%
Workplace	85.0%	65.6%	30.4%	9.4%	10.1%	11.6%	1.9%	3.4%	4.7%

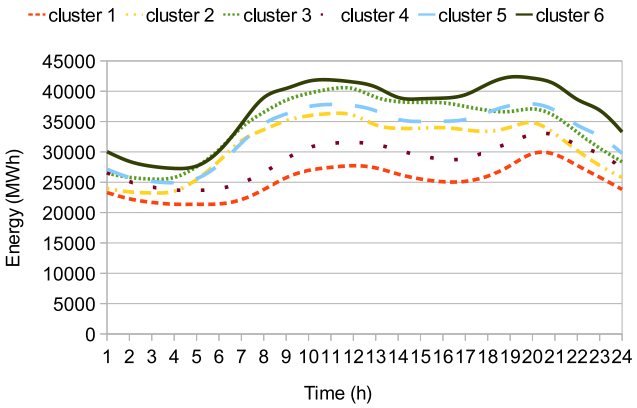

Fig. 3. Prototypes of demand clusters.

rule was a triangular fuzzy set with support $h-1, h, h+1$, where h is the hour, while the consequent represented the power demand corresponding to the cluster prototype.

B. Arrival rate $\tilde{\lambda}(t)$

We assume that recharging occurs either at home or at the workplace where it is easier to install a recharger. It is also assumed that recharge requests are made immediately after arriving since it is then convenient to plug the vehicle. Estimates of arrivals were derived from Movilia [35], a study sponsored by the Spanish Ministry of Public Works, which analyzes the mobility patterns of Spanish citizens. We have employed, in particular, statistics about displacements grouped by time frame and, among them, those having arrival destinations at either home or workplace. These data are shown in Table I.

Estimation of the arrival rate $\tilde{\lambda}(t)$ requires also knowledge about the number of daily displacements of EV users. We assume that EV users behave as the average Spanish citizen, who travels 3.3 times a day. From this value and an assumed number of total EVs it is possible to approximate the number of vehicles recharging on each time frame as $\lambda(t) = n * r * d(t) * (h(t) + w(t))$, where $\lambda(t)$ is the estimated proportion of displacements at each time frame, n is the total number of EVs, d is the proportion of displacements per time frame, r is the number of trips per day, and h and w are, respectively, the proportion of displacements heading home and workplace. As these crisp values are approximate estimates of arrival rates it is more realistic to model them by means of triangular fuzzy sets defined by the points $(0.85\lambda(t), \lambda(t), 1.15\lambda(t))$.

C. Service rate $\tilde{\mu}$

The time $\tilde{c}$ necessary to fully charge an EV may be expressed as $\tilde{c} = \tilde{b} \oslash \tilde{p}$, where $\tilde{b}$ is its total energy recharge requirement and $\tilde{p}$ is the power level of the recharger. The total recharge need $\tilde{b}$ was estimated from Movilia data about the average distance traveled per day, while the consumption per unit of distance was specified by EV manufacturers.[3] We assume the charge level $\tilde{p}$ was 3.5 kW. The service rate $\tilde{\mu}$ is then calculated as $1 \oslash \tilde{c}$. The value employed in the simulation is shown in Figure 4.

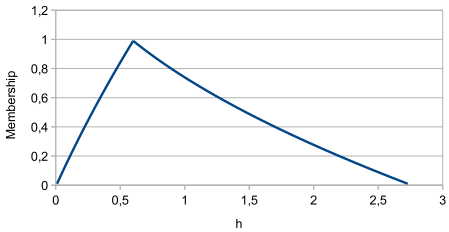

Fig. 4. Fuzzy set modeling average service time.

D. Other parameters

Finally, maximum power demand was estimated as 43.58 GW, corresponding to an increment of 2.5% above the maximum value of non-EV demand, while the maximum demand slope utilized in the simulation was 7.1 GW/h, equal to the maximum slope observed between 2005 and 2010.

VI. RESULTS

This Section is devoted to the discussion of simulation results involving two scenarios: the first involving expected conditions in 2014, when the Spanish Ministry of Industry, Tourism and Commerce [36] projects a number of about 250,000 EVs in Spain, while the second employs the corresponding prediction for the year 2020 when the number of EVs is projected to be 2,500,000 [37].

We compare our simulation approach with two related methods capable of dealing with complex queueing problems, those of Negi and Lee [20] and Wu [21]. For the sake of convenience, we will refer to these approaches as Negi and Wu, respectively, while that presented in this paper will be called VANSAS.[4] Negi, one of the initial efforts to employ FQT, simulates fuzzy queues with constant fuzzy arrival and

[3] In our simulation we considered only one type of EV with the worst consumption among all those specified.

[4] Short for VAriable Number of Servers, Arrival, and Service rates

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

TABLE II
RESULTS FOR SCENARIO 1

	L_q	L	T	T_q	Constraint violations
Negi	0.0 $_{(0.0)}$	12000.5 $_{(8660.3)}$	28.41 min $_{(18.5)}$	0.0 s $_{(0.0)}$	-
Wu	0.0 $_{(0.0)}$	18990.9 $_{(11651.7)}$	54.3 min $_{(54.3)}$	0.0 s $_{(0.0)}$	16388.1 $_{(133.5)}$
VANSAS	2.7E-03 $_{(0.2)}$	18973.1 $_{(11627.9)}$	54.3 min $_{(54.3)}$	0.2 s $_{(1.4)}$	-

TABLE III
RESULTS FOR SCENARIO 1

	L_q	L	T	T_q	Constraint violations
Negi	14.5 $_{(293.6)}$	113575.8 $_{(73711.3)}$	27.6 min $_{(16.3)}$	0.2 s $_{(3.6)}$	-
Wu	15070.2 $_{(24306.4)}$	204708.7 $_{(107189.6)}$	58.6 min $_{(71.8)}$	258.3 s $_{(1696.1)}$	246885.0
VANSAS	0.4 $_{(3.1)}$	189843.7 $_{(116312.7)}$	54.3 min $_{(54.3)}$	0.2 s $_{(1.3)}$	-

service rates and a fixed number of servers. Wu, which allows a higher degree of flexibility, allows also variable fuzzy arrival rates. VANSAS further generalizes these methods, being capable of handling changing fuzzy arrivals and service rates, and a variable number of servers. Furthermore VANSAS allows treatment of complex restrictions on the assignment of users to servers.

A. Parameters for Negi and Wu

The service rate employed in our Negi comparison simulation is the same as that used in VANSAS. Negi's method does not permit, however, to specify variable number of servers, arrival and service rates, thus making it necessary to define constant values for these parameters. To cover the range of possible arrival rates we have represented it as a triangular fuzzy set defined by the minimum arrival rate, median arrival rate, and maximum arrival rate, which were obtained from the values given in Section V-B. The number of servers was computed as the difference between the maximum non-EV power demand and the maximum total power demand divided by the power provided by a charger, i.e., $\frac{maxTotalDemand - maxNonEVDemand}{PowerCharger}$, or 354,048 servers. This estimate, which corresponds to the maximum number of users that can simultaneously recharge while guaranteeing system stability, takes into account the restriction on maximum power demand. The maximum-slope constraint cannot be modeled in Negi.

In our comparison simulation employing Wu we have employed the same arrival and service rates as for VANSAS. The number of servers was the same as that employed with Negi.

B. Scenario 1: 250,000 EVs

This Section presents simulation results for the first scenario focusing on four important parameters characterizing performance of the queuing system: the expected number of users waiting in the queue (L_q), the expected number of users in the system (L), the expected waiting time (T_q) and the expected total time spent in the system (T). We indicate also the number of times that the maximum-slope constraint was violated. Results are shown in Table II, showing average values and standard deviations (as subscripts) after 1000 simulations with Negi and 100 days of simulation employing Wu and VANSAS. Constraint violations are given in number of times per day.

The results obtained, for waiting users and waiting times, employing Wu and VANSAS are similar, while those of Negi are significantly different. These large differences in estimates—even when the same fuzzy service rate was employed in all approaches—are due to the inability of Negi to deal with Variable arrival rates, thus underestimating the number of users in the system, and, in addition, differences in the methods employed to represent fuzzy numbers. Negi favors generation of values around the point with maximum membership while Wu and VANSAS tend to produce values around the point obtained by defuzzification.

The estimates for expected number of waiting users and waiting time are, on the other hand, very low for all approaches as the system can easily cope with 250,000 EVs. The results for VANSAS, however, show values that are slightly higher than zero since it is the only approach that takes into account the maximum-slope demand. Wu violated, on the average, this constraint more than 16,000 times on a simulated day. The corresponding number cannot be computed for Negi, due to its reliance on theoretical analysis of queues with variable parameters.

C. Scenario 2: 2,500,000 EVs

The large increase in the number of EVs potentially imposes serious demands on the system. Results obtained for this scenario are shown in Table III.

Negi estimates, at every point in time, a very low number of users waiting in the queue, which is higher than VANSAS's but lower than Wu's. Further examination of methodological behavior shows, however, that is due to the restriction that limits maximum system demand. This value is also likely be an overestimation, as compared with VANSAS, since Negi (as is also true of Wu) is unable to deal with a variable number of servers. The total number of users L is again underestimated for the same reasons discussed in the previous Section. The expected total time in the system is similar to that of Scenario 1 since the expected waiting time is very small. It is important to

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

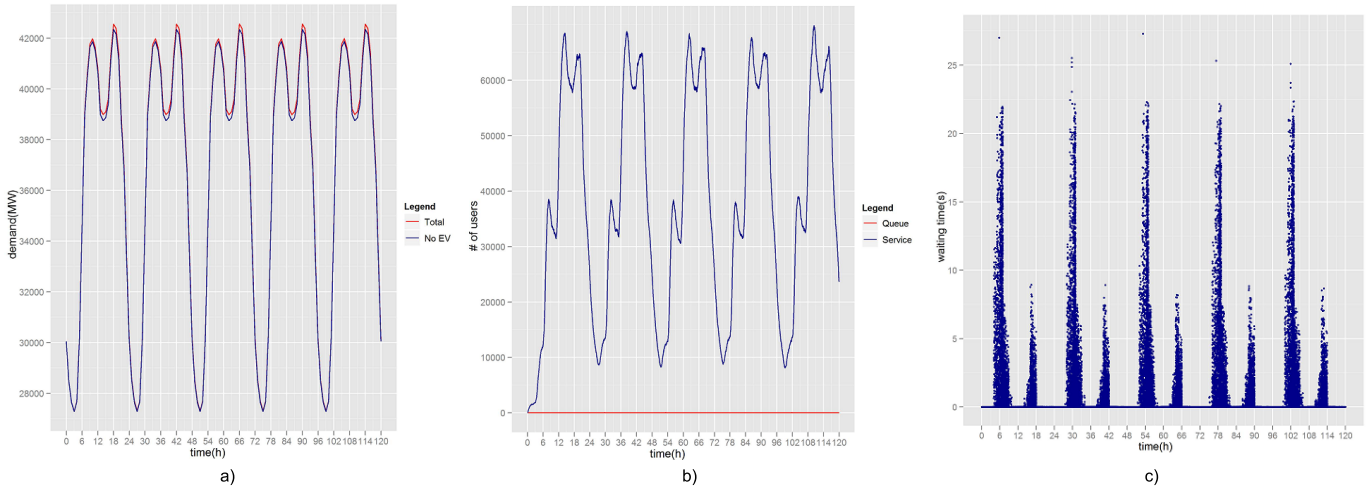

Fig. 5. Results for Scenario 1.

note that some of the simulations by Negi were non-stable (20 out of 1000) since demand and queue grew without bounds.

The expected number of queued users in Wu is very high, unlike that obtained using VANSAS, leading to question Wu's assumption that the number of servers is constant. This number was set, to prevent instability, as that of the servers available when the non-EV demand was at its highest. At other moments, particularly when the demand is much lower, the number of available servers is higher—a situation that cannot be modeled in Wu—leading to questionable results about unnecessary waiting times when there are more arrivals. The same considerations indicate that the expected waiting time computed by Wu, around 4 minutes, is too high, as also reflected in the total number of users and waiting times, which are higher than those produced by VANSAS. Furthermore, the large number of daily violations of system constraints clearly shows the inapplicability of Wu's method to this type of problem.

D. Analysis of the results from the perspective of the electric network

We turn now our attention to the analysis of simulation results from the perspective of total demand on the electric network. Special attention has been paid to waiting times as they are an indicator of network saturation problems while suggesting potential issues with charger design and behavior.

1) Scenario 1: 250,000 EVs: The results of the simulation for the first scenario are shown in Figure 5. Figure 5 a) illustrates power demand, comparing demand non-EVs and total (non-EV + EV) demand. Figure 5 b) shows the number of customers that are being served and that of those waiting in the queue. The number of available servers changes during the day having its highest value at 15:00 when a large number of clients return home. The network impact of 250,000 EVs is, however, quite negligible from a global perspective. As shown in Figure 5 c), showing user waiting times (one point representing one user), the number of waiting users is nearly zero. The maximum waiting time is very small around 25 seconds. From a design viewpoint these results imply that, at least under the assumptions of this scenario, it is not necessary to provide the charger with any form of intelligent behavior.

2) Scenario 2: 2,500,000 EVs: The results for projected values for the year 2020 are shown in Figure 6. Figure 6 a) illustrates the results for EV demand. A tenfold increase in the projected number of EVs clearly leads to a much larger demand. Results about number of servers and queue size, shown in Figure 6 b), are very similar to those obtained for Scenario 1. The time pattern followed by the number of servers is similar to that of the previous scenario although the number of servers is now ten times larger. The increment in the number of EVs does not, however, dramatically affect queue size, which remains very low as seen in Figure 6 c). Although the number of waiting users is larger, as seen by the increase in the density of points, the maximum waiting time remains around 25 seconds.

From a global perspective, results from both simulations show that EV demand will not dramatically affect the Spanish electric network. Demand peak values will be obviously higher but this growth in demand will be gradual, thus permitting energy producers to adaptively minimize impact on the network. These results also lead us to conclude that intelligent chargers will not be required to prevent network crashes.

E. Empirical estimation of computational complexity

To estimate empirically the computational complexity of the simulation we studied the empirical behavior of execution time with respect to simulated time and number of EVs.

Figure 7 a) shows the dependence of execution time on simulated time. Ten simulation runs, each involving 100,000 EVs, were conducted for simulated times ranging from one to ten days. Each data point corresponds to the average run execution time in milliseconds for each simulated time. In addition, a linear trendline (dashed red line in the figure) was found to approximate the results with sufficient accuracy ($R^2 = 0.9998$), indicating that the execution time increases linearly with respect to simulated time.

Figure 7 b) displays empirical dependence of execution time on number of EVs. Ten simulation runs, each involving one

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

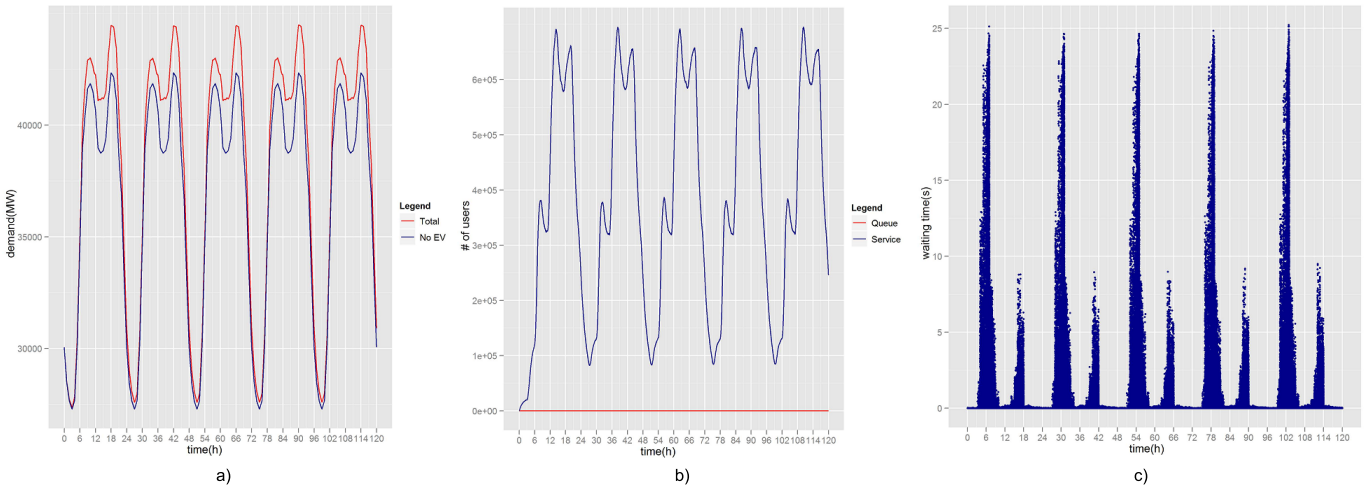

Fig. 6. Results for Scenario 2.

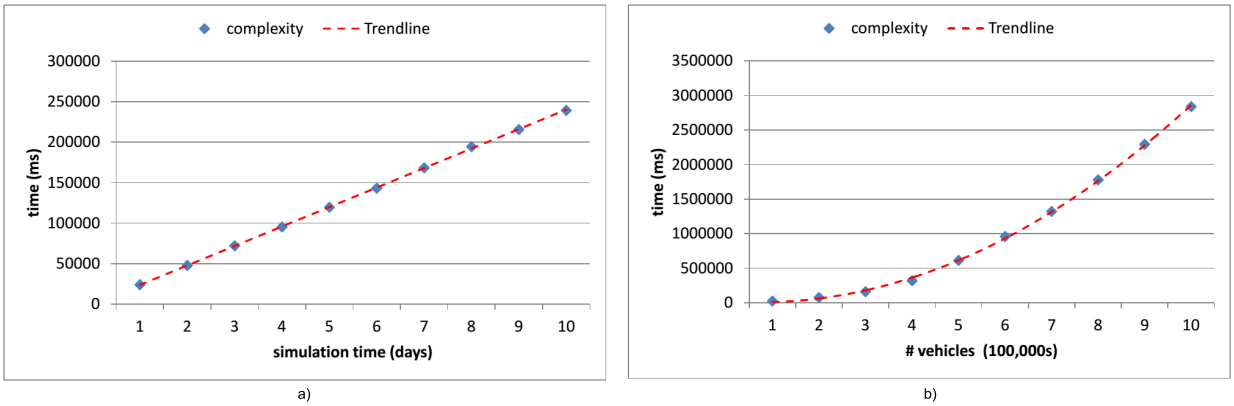

Fig. 7. Analysis of complexity with regard to simulated time and vehicles.

simulated day, were conducted for vehicle numbers ranging from 100,000 to 1,000,000. Each data point corresponds to the average run execution time in milliseconds for each number of EVs. These results can be approximated with sufficient accuracy by a quadratic function ($R^2 = 0.9995$), showing a $O(n^2)$ dependence of simulated times on the number of EVs.

VII. Conclusions

An important class of queuing problems is characterized by features that prevent their analysis employing analytical approaches. These peculiarities include several time-varying parameters including arrival rates, service rates, and number of servers. Values of these parameters are typically only known in an imprecise and uncertain manner. Furthermore, server behavior is subject to complex restrictions on system capacity and stability. In this work we introduce a new FQT simulation approach that considers these difficulties, which were only partially addressed by prior efforts.

This approach was applied to an important practical problem, the estimation of future demand, on the Spanish power grid, for electric-vehicle recharging. In this problem, arrival and service rates fluctuate during the day depending on user preferences and power availability. The latter parameter also determines the number of available servers, which is, therefore, also time-dependent. Lack of empirical data, due to lack of prior experience with massive EV recharging, is the source of uncertainties about system parameters. The nature of power generation systems also imposes constraints on service behavior.

We have compared our results to those obtained by two other FQT methods. Although straightforward comparison between methodologies with different scopes and limitations is not simple, the results obtained show, nonetheless, significant consistencies—empirical differences being easy to explain as prior efforts lacked the required generality or violated behavior constraints. We have also studied computational complexity by examining the dependence of execution time on key system parameters, which show that the approach is capable of coping with foreseeable increases on system size. We further conclude that it extends, in substantive ways, the applicability of FQT while producing realistic estimates of system behavior. This study is an important step in the modeling of queuing systems characterized by complex behavior under complex constraints. Future improvements can be expected as methods for the analysis of fuzzy systems improve and availability of empirical data suggests enhancements to modeling techniques.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

ACKNOWLEDGMENT

This research work has been partially supported by the Spanish MEC (ENERGOS Project TIN2009-00000).

REFERENCES

[1] L. Kleinrock, Queueing Systems, Volume 1, Wiley-Interscience, 1975.
[2] J. Buckley, "Fuzzy Queueing Theory", Fuzzy probabilities, studies in fuzzinnes and soft computing. Springer, pp. 61-60, 2005.
[3] X. Su, P. Shi, L. Wu, and Y.-D. Song, "A novel control design on discrete-time Takagi-Sugeno fuzzy systems with time-varying delays," IEEE Transactions on Fuzzy Systems, 2012.
[4] X. Su, P. Shi, L. Wu, and Y.-D. Song, "A Novel Approach to Filter Design for T-S Fuzzy Discrete-Time Systems With Time-Varying Delay," IEEE Transactions on Fuzzy Systems, vol. 20, no. 6, pp. 11141129, 2012.
[5] J.A.P. Lopes, F.J. Soares, P.M.R. Almeida, "Integration of electric vehicles in the electric power system". *Proc. of the IEEE*, vol. 99, no. 1, pp. 168-183, 2011.
[6] W. A. Massey, "The Analysis of Queues with Time-Varying Rates for Telecommunication Models", Telecommunication Systems, vol. 21, pp. 173204, 2002.
[7] W.P. Wang, D. Tipper, S. Banerjee, A simple approximation for modeling nonstationary queues, Proceedings of the INFOCOM96, pp. 255-262, 1996.
[8] L. Green, P. Kolesar, The pointwise stationary approximation for queues with nonstationary arrivals. Management Science, Vol. 37, no. 1, pp. 84-97, 1991.
[9] S. M. Brodi and N. A. Migalko, "A queueing model with variable number of servers", Journal of Mathematical Sciences, Vol. 40, no. 4, p. 458, 1988.
[10] A. I. Ivaneshkin, "Optimizing a Multiserver Queuing System with a Variable Number of Servers", Cybernetics and Systems Analysis, Vol. 43, No. 4, pp. 542-548, 2007.
[11] Ingolfsson, A., E. Akhmetshina, S. Budge, Y. Li, X. Wu. "A survey and experimental comparison of service level approximation methods for non-stationary M/M/s queueing systems". INFORMS Journal of Computing vol. 19, pp. 201-214, 2007.
[12] R. Stolletz, "Approximation of the non-stationary M(t)/M(t)/c(t)-queue using stationary queueing models: The stationary backlog-carryover approach". European Journal of Operational Research, vol. 190, n. 2, pp. 478493, 2008.
[13] M.J. Pardo, D. de la Fuente, Optimal selection of the service rate for a finite input source fuzzy queuing system, *Fuzzy Sets and Systems*, vol. 159, pp. 325-342, 2008.
[14] H. M. Prade, "An outline of fuzzy or possibilistic models for queuing systems," in Proc. Symp. Policy Anal. Inform. Syst., in Durham, NC,1980, pp. 147153.
[15] R.J. Li and E.S. Lee, "Analysis of fuzzy queues", Comput. Math.Applicat., vol. 17, no. 7, pp. 11431147, 1989.
[16] J.J. Buckley, Elementary queuing theory based on possibility theory, *Fuzzy Sets and Systems*, vol. 37, pp. 43-52, 1990.
[17] J.J. Buckley, T. Feuring, Y. Hayashi, Fuzzy queuing theory revisited, *Internat. J. Uncertainty Fuzziness Knowledge-based Systems*, vol. 9, no. 5, pp.527-537, 2001.
[18] C. Kao, C. Li, S. Chen, Parametric programming to the analysis of fuzzy queues, Fuzzy Sets and Systems 107, 93100, 1999.
[19] S.P. Chen, Solving fuzzy queuing decision problems via a parametric mixed integer nonlinear programming method, *Eur. J. Oper. Res.*, vol. 177, no. 1, pp. 445-457, 2007.
[20] D.S. Negi, E. S. Lee, Analysis and simulation of fuzzy queues, Fuzzy Sets and Systems, 46, 321-330, 1992.
[21] H.C. Wu, Simulation for queueing systems under fuzziness, *Intern. J. Syst. Sci.*, vol. 40, no. 6, pp. 587-600, 2009.
[22] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, 1995.
[23] L.A. Zadeh, The Concept of Linguistic Variable and Its Application to Approximate Reasoning I, II and III, Information Sciences, pp. 43-80, vol. 8, pp. 199-249, pp. 301-357, 1975.
[24] S.W. Hadley, A.A. Tsvetkova, "Potential Impacts of Plug-in Hybrid Electric Vehicles on Regional Power Generation", The Electricity Journal, Vol. 22, no. 10, pp 56-68, 2009.
[25] R.E. Uhrig, "Greenhouse Gas Emissions from Gasoline, Hybrid-Electric, and Hydrogen Fueled Vehicles", EIC Climate Change Technology Conference, pp. 1-6, 2006.
[26] J. Larminie, J. Lowry. Electric Vehicle Technology Explained. West Sussex, England: J. Wiley, 2003.
[27] R. C. Green II, L. Wang, M. Alam, "The impact of plug-in hybrid electric vehicles on distribution networks: A review and outlook", Renewable and Sustainable Energy Reviews, vol. 15, pp. 544553, 2011.
[28] P. Denholm, W. Short, "An evaluation of utility system impacts and benefits of optimally dispatched plug-in hybrid electric vehicles", Nat. Renewable Energy Lab. (NREL), Golde, CO, Tech. Rep., 2006.
[29] M. J. Scott, M. Kintner-Meyer, D. Elliott, W. Warwick, "Impacts assessment of plug-in hybrid vehicles on electric utilitites and regional U.S. power grids. Part I", Pacific North West National Lab., Richland, WA, PNNL-SA-61687, 2007.
[30] K. Parks, P. Denholm, T. Markel, Costs and Emissions Associated with Plug-In Hybrid Vehicle Charging in the Xcel Energy Colorado Service Territory, NREL/TP-640-41410, National Renewable Energy Laboratory, 2007.
[31] J. Taylor, A. Maitra, M. Alexander, D. Brooks, M. Duvall, "Evaluation of the impact of plug-in electric vehicle loading on distribution system operations". Power and Energy Society General Meeting 2009, pp. 16, 2009.
[32] J.C. Kelly, J.S. MacDonald, G.A. Keoleian, "Time-dependent plug-in hybrid electric vehicle charging based on national driving patterns and demographics", Applied Energy, vol. 94, pp. 395405, 2012.
[33] E.H. Ruspini, "A new approach to clustering", Information and Control, vol. 15, n. 1, pp. 22-32, 1969.
[34] J.C. Bezdek and S. K. Pal, Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data, IEEE Press, 1982,
[35] Spanish Ministry of Public Works, Mobility of the People Resident in Spain Survey [Encuesta de movilidad de las personas residentes en España (Movilia 2006/2007)]. (2006).
[36] Spanish Ministry of Industry, Tourism and Commerce, Integral Strategy for the Introduction of the Electric Vehicle in Spain [Estrategia Integral Para El Impulso Del Vehículo Eléctrico En España]. (2010).
[37] Spanish Ministry of Industry, Tourism and Commerce, National Action Plan of Renewable Energies in Spain [Plan De Acción Nacional De Energías Renovables De España (PANER) 2011 - 2020]. (2010).

Enrique Muñoz Enrique Muñoz obtained his degree in Computer Science Engineering at the University of Murcia in 2006. In 2007 he received a masters degree on Advanced Information Tecnologies and Telematics. In 2010 he finished his PhD (Artificial Intelligence) at the University of Murcia, obtaining the "Best Doctoral Dissertation Award" in Computer Science. He has written several publications (papers in journals, conferences, book chapters, etc.), he is member of the editorial board of the Soft Computing journal, and has been TPC member in several journals and international conferences. He has participated into several research projects. His research interests include multi-agent systems, machine learning, and optimization strategies.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TFUZZ.2013.2278407

Enrique H. Ruspini Enrique H. Ruspini received his degree in Mathematics from the University of Buenos Aires, Argentina, in 1965 and his doctoral degree in System Science from the University of California at Los Angeles in 1977. He is currently the Director of the Collaborative Intelligent Systems Laboratory at the European Centre for Soft Computing in Mieres (Asturias), Spain. He is also a Distinguished Lecturer of the IEEE Computational Intelligence Society. Previously he was a Principal Scientist at the Artificial Intelligence Center, SRI International, in Menlo Park, California. Dr. Ruspini has also held positions at the University of Buenos Aires, the University of Southern California, the UCLA Brain Research Institute, and Hewlett-Packard Laboratories.

Dr. Ruspini is one the earliest contributors to the development of fuzzy-set theory and its applications, having introduced its use to the treatment of numerical classification and clustering problems. He has also made significant contributions to the understanding of the foundations of fuzzy logic and approximate-reasoning methods.

Dr. Ruspini, who is the recipient of the 2009 Fuzzy Systems Pioneer Award of the IEEE Computational Intelligence Society, is a Life Fellow of the IEEE, a First Fellow of the International Fuzzy Systems Association, a Fulbright Scholar, an European Union Marie Curie Fellow, and a SRI International Fellow. In 2004, he received the Meritorious Service Award of the IEEE Neural Networks Society for leading the transition of the Neural Networks Council into Society status. He is a former member of the IEEE Board of Directors and President (2001) of the IEEE Neural Networks Council.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.