

# Exploiting Timed Automata Based Fuzzy Controllers for Voltage regulation in Smart Grids

Giovanni Acampora, Vincenzo Loia, Autilia Vitiello

Department of Computer Science, University of Salerno, Fisciano, Salerno 84084

Email: {gacampora, loia, avitiello}@unisa.it

**Abstract**—The large-scale deployment of the Smart Grid paradigm will support the evolution of conventional electrical power systems toward active, flexible and self-healing web energy networks composed of distributed and cooperative energy resources. In a Smart Grid platform, the optimal coordination of distributed voltage controllers is one of the main issues to address. In this field, the application of traditional control paradigms has some disadvantages that could hinder their application in Smart Grids where the constant growth of grid complexity and the need for massive pervasion of Distribution Generation Systems (DGSs) require more scalable, more flexible control and regulation paradigms. To try and overcome these challenges, this paper proposes the concept of a decentralized non-hierarchical voltage regulation architecture based on intelligent and cooperative smart entities. The distributed voltage controllers employ traditional sensors to acquire local bus variables and mutually coupled oscillators to assess the main variables that characterize the operation of the global Smart Grid. These variables are then amalgamated by a novel fuzzy inference engine, named Timed Automata based Fuzzy Controllers, in order to identify proper control actions aimed at improving the grid voltage profile and reducing power losses.

**Index Terms**—Distributed Generation Systems, Voltage regulation, Timed Automata based Fuzzy Controllers.

## I. INTRODUCTION

The Smart Grid[1] is the convergence of information and operational technologies applied to the electric grid, providing sustainable options to customers and improved security [2][3]. It could play a strategic role in supporting the evolution of conventional electrical grids toward active, flexible and self-healing web energy networks composed of distributed and cooperative energy resources [4][5]. Advances in research on Smart Grids could increase the efficiency of modern electrical power systems by: (i) supporting large-scale penetration of small-scale Distributed Generation Systems (DGSs); (ii) facilitating the integration of Renewable Energy Sources; (iii) reducing system losses and Green House Gas emissions; (iv) increasing the reliability of electricity supply to customers. From this perspective, a crucial issue is how to increase the penetration of DGS into Smart Grids by properly coordinating their operation and by restricting their negative impact on grid operations and control. In this context, effective network voltage regulation is one of the main issues to address [6][7]. Voltage regulation in electrical grids is normally achieved by the optimal coordination of a large set of regulating devices, all used with the aim of supporting the load buses voltages and improving the power quality at the distribution level [8][9]. The presence of DGSs in electrical grids increases the complexity

of this regulation process since their power injections, intrinsically random for generators based on renewable sources, could alter the voltage profiles on network buses. This requires the design of a proper methodology aimed at assuring effective grid operation by the optimal coordination of DGS operations with traditional voltage regulating devices. This methodology should require low computational effort in order to assure an effective on-line solution to the voltage regulation problem. In order to address these needs, the application of fuzzy logic theory has been proposed in the literature [10][11]. Fuzzy logic can handle problems with imprecise and incomplete data, and it can model nonlinear functions of arbitrary complexity [12][13][14].

However, the solution to the voltage regulation problem should be prompt, reliable and accurate, but, above all, it should be highly adaptive in order to effectively manage the intrinsic time varying phenomena affecting the smart grid operation. For this reason, in this work, a novel fuzzy inference engine, named Timed Automata based Fuzzy Controllers (TAFCs)[15], is exploited to regulate the reactive power flows injected by the DGSs into the electrical grid. Thanks to its high dynamic features, a TAFC is capable of modelling a system characterized by frequent changes over the time, and, consequently, it results highly suitable for the application in the electrical power systems context.

As shown in the experimental results section, where our proposal has been compared with a conventional fuzzy controller, the proposed strategy results in an effective and suitable method to solve voltage regulation problem.

## II. PROBLEM FORMULATION

The ultimate goal of the voltage regulation process is to identify the optimal asset of voltage regulating devices required to deliver higher service quality, in terms of voltage profile flattening and power losses minimization. Roughly speaking, the solution of the voltage regulation problem requires the identification of the voltage regulation devices asset that minimizes, for each grid state, a set of objectives subject to a number of inequality constraints.

The voltage regulation devices asset is described by the following vector:

$$y = [Q_{dg,1}, \dots, Q_{dg,N_g}, \dots, Q_{cap,1}, \dots, Q_{cap,N_c}, V_{FTS,N_f}, m] \quad (1)$$

where  $Q_{dg,i}$  is the reactive power injected by the  $i^{th}$  ( $i = 1, \dots, N_g$ ) DGS connected to the Smart Grid and available for

regulation;  $Q_{cap,j}$  is the vector of the reactive power injected by the  $j^{th}$  ( $j = 1, \dots, N_c$ ) capacitor bank;  $V_{FTS,k}$  is the set point of the  $kk^{th}$  ( $k = 1, \dots, N_f$ ) Flexible AC Transmission System (FACTS) and  $m$  is the tap position of the HV/MV line tap changing transformer. Note that the vector  $y$  takes values in the solution space  $\Omega$ :

$$y \in \Omega \Leftrightarrow \begin{cases} tap_{min} \leq m \leq tap_{max} & i = 1, \dots, N_g \\ Q_{dg,min,i} \leq Q_{dg,i} \leq Q_{dg,max,i} & i = 1, \dots, N_g \\ Q_{cap,min,j} \leq Q_{cap,j} \leq Q_{cap,max,j} & j = 1, \dots, N_c \\ Q_{FTS,min,k} \leq Q_{FTS,k} \leq Q_{FTS,max,k} & k = 1, \dots, N_f \end{cases} \quad (2)$$

When solving the voltage regulation problem, several constraints must be satisfied. They include the technical constraints for correct system operation in terms of allowed voltage ranges for the network buses (i.e.  $V_{min,i} \leq V_i \leq V_{max,i}$   $i = 1 : n$ ) and maximum allowable currents for the  $n_l$  power lines (i.e.  $I_l \leq I_{max,l}$   $l = 1 : n_l$ ).

The objectives to be minimized in identifying the optimal voltage regulating devices asset takes into account both technical and economic aspects (i.e. power losses, reactive power costs, voltage magnitude). Since the design objectives are in competition, the voltage regulation problem has no unique solution and a suitable trade-off between objectives has to be identified.

### III. THE PROPOSED APPROACH

To address the voltage regulation problem in a Smart Grid, an innovative solution approach is proposed here.

The reactive power flows injected by the DGSs into the electrical grid are regulated according to the following control strategy: The proposed algorithm first acquires the global variables characterizing the actual operation of the Smart Grid. These variables are then processed in order to estimate the current grid state. This could be classified on the basis of the mean grid voltage magnitude according to the following set of minimal states:

- 1) *Grid Voltage Low*: This state is characterized by high load demand. As a result, high power line loading and, consequently, high power losses and a very low value of the mean grid voltage magnitude are expected on the grid.
- 2) *Grid Voltage High*: This state is characterized by low load demand. As a result, low power line loading and, consequently, low power losses and a very high value of the mean grid voltage magnitude are expected on the grid.
- 3) *Normal Operation*: This state represents the nominal operating point of the electrical grid. Both the power losses and the value of the mean grid voltage magnitude lie within allowable intervals.

Once the actual Smart Grid state has been assessed, the algorithm identifies proper voltage control strategies for each node. This is implemented according to the following control rules:

- If the Smart Grid state is classified as *Grid Voltage Low* (respectively *Grid Voltage High*), then the control objective is to raise (respectively reduce) the mean grid voltage magnitude  $V_m$  by properly increasing (respectively decreasing) the reactive power injected by the DGSs into the grid. In order to obtain a uniform rise in voltage magnitude, the  $i^{th}$  controller compares the local bus voltage magnitude ( $V_i$ ) with the mean grid voltage magnitude ( $V_M$ ) and:

- 1) If  $V_i < V_M$ , it increases (respectively decreases) the injected reactive power  $Q_{dg,i}$  by a quantity directly (respectively inversely) proportional to  $|V_M - V_i|$ , namely:

$$\Delta Q_{dg,i} = \alpha_i(V_i) \cdot (V_M - V_i)$$

respectively

$$\Delta Q_{dg,i} = \frac{\alpha_i(V_i)}{V_i - V_M}$$

- 2) Otherwise, it increases (respectively decreases) the injected reactive power  $Q_{dg,i}$  by a quantity inversely (respectively directly) proportional to  $|V_M - V_i|$ , namely:

$$\Delta Q_{dg,i} = \frac{\alpha_i(V_i)}{(V_i - V_M)}$$

respectively

$$\Delta Q_{dg,i} = \alpha_i(V_i) \cdot (V_M - V_i)$$

- If the Smart Grid state is classified as *Normal operation*, then the control objective is to improve the voltage profile (namely to reduce the voltage magnitude deviation). To address this issue, the  $i^{th}$  controller compares the local bus voltage magnitude ( $V_i$ ) with the mean grid voltage magnitude ( $V_M$ ) and:

- 1) If  $V_i < V_M$ , it increases the injected reactive power  $Q_{dg,i}$  by a quantity directly proportional to  $|V_M - V_i|$ , namely:

$$\Delta Q_{dg,i} = \alpha_i(V_i) \cdot (V_M - V_i)$$

- 2) Otherwise, it decreases the injected reactive power  $Q_{dg,i}$  by a quantity directly proportional to  $|V_M - V_i|$ , namely:

$$\Delta Q_{dg,i} = \alpha_i(V_i) \cdot (V_M - V_i)$$

where  $\alpha_i(V_i)$  is a monotonically increasing function of the actual value of the voltage magnitude of the  $i^{th}$  bus.

These user-supplied human language rules could be effectively implemented by a Fuzzy Inference System [16]. This system converts these linguistic rules into their mathematical equivalents, thereby simplifying the job of the system designer and resulting in much more accurate representations of the way systems behave in the real world. However, the solution to voltage regulation problem should be characterized by also other characteristics such as reliably and accuracy, but above

all it should be highly adaptive in order to effectively manage the intrinsic time varying phenomena affecting the smart grid operation. In the light of these needs, our proposal is to use a TAFC in order to face the dynamic changes which highly characterize power systems. The exploited fuzzy inference engine is described in detail in the next section.

#### IV. TIMED AUTOMATA-BASED FUZZY CONTROLLERS

This section is devoted to describe the Timed Automata-based Fuzzy Control. This novel fuzzy inference engine improves classic inference methods in terms of dynamism and ability to model variable structure systems that are characterized from a discontinuous nonlinear behaviour. As mentioned, this kind of behaviour strongly characterizes power systems. Consequently, a standard control system could not be sufficient to manage a so complex framework and an alternative scenario is necessary. This scenario is provided by Timed Automata based Fuzzy Controllers.

Before formally defining TAFCs, it is necessary to introduce *timed fuzzy controllers*, i.e., a novel fuzzy inference engine that extends fuzzy control idea with three additional concepts: *control configuration*, *control era* and *control time*. In detail, a timed fuzzy controller uses the control time as a clock moving the controlled systems through several time intervals, known as *control eras*. Each control era is characterized by a specific *control configuration* which is formed by: 1) the number and typology of fuzzy variables and 2) the number and structure of relationships among variables.

**Definition 1 (Fuzzy Logic Controller):** A fuzzy logic controller  $c$ , with  $n$  inputs and  $m$  outputs, can be considered as a function mapping a vector  $\underline{x} \in \mathbb{R}^n$  with an output vector  $\underline{y} \in \mathbb{R}^m$ :

$$c : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

where  $\underline{y} = c(\underline{x})$  is computed by applying fuzzy inference operators. ■

**Definition 2 (Timed Fuzzy Logic Controller):** Let  $C = \bigcup_{n \in \mathbb{N}} \bigcup_{m \in \mathbb{N}} \{c : \mathbb{R}^n \rightarrow \mathbb{R}^m\}$  be the collection of all possible fuzzy controllers and let

$$E = \{e_i = [t'_i, t''_i] \mid i = 1 \dots k, k \in \mathbb{N}, \\ t'_i, t''_i \in \mathbb{R}^+ \text{ with } t'_1 = 0, t''_i = t'_{i+1} \text{ and } t'_i < t'_{i+1}\}$$

be the set of  $k$  disjoint control eras. The  $i^{th}$  control era, known as  $e_i$ , starts at  $t'_i$  and ends at  $t''_i$ . Once defined  $C$  and  $E$ , let  $f : E \rightarrow C$  be the function that maps each control era to the corresponding control configuration and let  $g : \mathbb{R}^+ \rightarrow E$  a function which associates a time instant  $t_j$  with the corresponding control era. Then, a *timed fuzzy control* is a function composition  $\varphi = f \circ g : \mathbb{R}^+ \rightarrow C$  that maps a time instant  $t_j \in \mathbb{R}^+$  belonging to the  $i^{th}$  control era to a fuzzy controller  $f_i \in C$  representing the associated control configuration. ■

**Observation 1:** If two instants  $t_j$  and  $t_{j+1}$ , with  $t_j < t_{j+1}$ , belong to the same  $i^{th}$  control era, the function  $\varphi$  returns the same function  $f_i$ :

$$\varphi(t_j) = f(g(t_j)) = f_i = f(g(t_{j+1})) = \varphi(t_{j+1})$$

$$\text{if } t'_i \leq t_j < t_{j+1} < t''_i \quad \blacksquare$$

**Observation 2:** Each fuzzy controller  $c$  is a timed fuzzy controller  $\varphi_c$  on 1 control era with following parameters:

$$\begin{aligned} E &= \{[0, t''_1], t''_1 > 0\} \\ C &= \{c\} \\ f([0, t''_1]) &= c \\ g(t) &= [0, t''_1] \quad \forall t \in \mathbb{R}^+ \\ \varphi_c &= f(g(t)) = c \quad \forall t \in \mathbb{R}^+ \quad \blacksquare \end{aligned}$$

A TAFC is a particular implementation of the function  $\varphi$  based on a theory deriving from formal language: Timed Automata [17].

##### A. Timed Automata

A timed automaton is a standard finite-state automaton extended with a finite collection of real-valued *clocks* providing a straightforward way to represent time related events, whereas automata-based approaches cannot offer this feature. The clocks can be reset to 0 (independently of each other) with the transitions of the automaton, and keep track of the time elapsed since the last reset. The transitions of a timed automaton are labeled with a *guard* (a condition on clocks), an *action* or *symbol* on alphabet  $\Sigma$ , and a *clock reset* (a subset of clocks to be reset). Intuitively, a timed automaton starts execution with all clocks set to zero. Clocks increase uniformly with time while the automaton is within a node. A transition may be taken only if the current values of the clocks satisfy the associated constraints. The set of behaviours expressed by a system modeled by means of a timed automaton is defined by a *timed language*, i.e., a collection of *timed words*. Both timed concepts are defined in the follows.

**Definition 3:** A *time sequence*  $\tau = \tau_1 \tau_2 \dots$  is an infinite sequence of time values  $\tau_i \in \mathbb{R}$  with  $\tau_i > 0$ , satisfying the following constraints:

- 1) *Monotonicity:*  $\tau$  increases strictly monotonically; that is,  $\tau_i < \tau_{i+1}$  for all  $i \geq 1$ .
- 2) *Progress:* For every  $t \in \mathbb{R}$ , there is some  $i \geq 1$  such that  $\tau_i > t$ .

A *timed word* over an alphabet  $\Sigma$  is a pair  $(\sigma, \tau)$  where  $\sigma = \sigma_1 \sigma_2 \dots$  is an infinite word over  $\Sigma$  and  $\tau$  is a time sequence. A *timed language* over  $\Sigma$  is a set of timed words on  $\Sigma$ . ■

**Definition 4:** For a set  $X$  of clock variables, the set  $\Phi(X)$  of *clock constraints*  $\delta$  is defined inductively by

$$\delta := x \leq c \mid c \leq x \mid \neg \delta \mid \delta_1 \wedge \delta_2$$

where  $x$  is a clock in  $X$  and  $c$  is a constant in  $\mathbb{Q}$ , the set of nonnegative rationals.

A *clock interpretation*  $\nu$  for a set  $X$  of clocks assigns a real value to each clock; that is, it is a mapping from  $X$  to  $\mathbb{R}$ . We say that a clock interpretation  $\nu$  for  $X$  satisfies a clock constraint  $\delta$  over  $X$  iff  $\delta$  evaluates to true using the values given by  $\nu$ . For  $t \in \mathbb{R}$ ,  $\nu + t$  denotes the clock interpretation which maps every clock  $x$  to the value  $\nu(x) + t$ , and the clock interpretation  $t \cdot \nu$  assigns to each clock  $x$  the value  $t \cdot \nu(x)$ .

For  $Y \subseteq X$ ,  $[Y \mapsto t]\nu$  denotes the clock interpretation for  $X$  which assigns  $t$  to each  $x \in Y$ , and agrees with  $\nu$  over the rest of the clocks. ■

Now, a precise definition of timed transition table, which determines the timed automaton behaviour, is given:

*Definition 5:* A *timed transition table*  $\mathcal{A}$  is a tuple  $\langle \Sigma, S, S_0, C, E \rangle$ , where

- $\Sigma$  is a finite alphabet,
- $S$  is a finite set of states,
- $S_0 \subseteq S$  is a set of start states,
- $C$  is finite set of clocks, and
- $E \subseteq S \times S \times \Sigma \times 2^C \times \Phi(C)$  gives the set of transitions. An edge  $\langle s, s', a, \lambda, \delta \rangle$  represents a transition from state  $s$  to state  $s'$  on input symbol  $a$ . The set  $\lambda \subseteq C$  gives the clocks to be reset with this transition, and  $\delta$  is a clock constraint over  $C$ . ■

If  $(\sigma, \tau)$  is a timed word viewed as an input to an automaton, it presents the symbol  $\sigma_i$  at time  $\tau_i$ . If each symbol  $\sigma_i$  is interpreted to denote an event occurrence then the corresponding component  $\tau_i$  is interpreted as the time of occurrence of  $\sigma_i$ . Given a timed word  $(\sigma, \tau)$ , the timed transition table  $\mathcal{A}$  starts in one of its start states at time 0 with all clocks initialized to 0. As time advances, the values of all clocks change, reflecting the elapsed time. At time  $\tau_i$ ,  $\mathcal{A}$  state from  $s$  to  $s'$  using some transition of the form  $\langle s, s', \sigma_i, \lambda, \delta \rangle$  reading the input  $\sigma_i$ , if the current values of clocks satisfy  $\delta$ . With this transition the clocks in  $\lambda$  are reset to 0, and thus start continuing time with respect to the time of occurrence of this transition. Formally, this timed behaviour is captured by introducing *runs* of timed transition tables.

*Definition 6:* A run  $r$ , denoted by  $(\bar{s}, \bar{\nu})$ , of a timed transition table  $\langle \Sigma, S, S_0, C, E \rangle$  over a timed word  $(\sigma, \tau)$  is an infinite sequence of the form

$$r : \langle s_0, \nu_0 \rangle \xrightarrow[\tau_1]{\sigma_1} \langle s_1, \nu_1 \rangle \xrightarrow[\tau_2]{\sigma_2} \langle s_2, \nu_2 \rangle \xrightarrow[\tau_3]{\sigma_3} \dots$$

with  $s_i \in S$  and  $\nu_i \in [C \rightarrow \mathbb{R}]$ , for all  $i \geq 0$ , satisfying the following requirements:

- *Initiation:*  $s_0 \in S_0$  and  $\nu_0(x) = 0$  for all  $x \in C$ .
- *Consecution:* for all  $i \geq 1$ , there is an edge in  $E$  of the form  $\langle s_{i-1}, s_i, \sigma_i, \lambda_i, \delta_i \rangle$  such that  $(\nu_{i-1} + \tau_i - \tau_{i-1})$  satisfies  $\delta_i$  and  $\nu_i$  equals  $[\lambda_i \mapsto 0](\nu_{i-1} + \tau_i - \tau_{i-1})$ .

■

The timed transition table together with the run concept are the main notions used in our approach to embed dynamism in the standard FLC definition.

### B. Merging Timed Automata and Fuzzy Controllers

In an attempt to provide an instrument to model the function  $\varphi$  and consequently to obtain an implementation of a timed fuzzy controller, Timed Automata based Fuzzy Controllers (TAFCs) are proposed. TAFCs represent an integration between two theories: fuzzy control and timed automata.

A TAFC implements a timed fuzzy controller through a timed automaton whose behaviours define all the potential sequences of control eras (and the related control configurations)

which the system may cross during its life-cycle. More in detail, TAFCs are able to manage the control configurations by associating each of them with a state in the timed automaton. Timed Automata support control eras progression by using the *run* concept (definition 6) where the  $i^{th}$  transition moves the system from the  $i^{th}$  control era to the  $(i+1)^{th}$  one.

However, in order to change system's configuration among different automaton states, it is necessary to extend timed automata definition by exploiting a collection of so-called *transformation operators*:

$$C_{op} = \{ \oplus, \ominus, \boxplus^k, \boxminus^k, \boxdot^k, \sim, \ltimes, \rtimes, \bowtie, \otimes, \oslash, \odot, *, \alpha \}$$

where  $\oplus$  and  $\ominus$ , respectively, adds and deletes a fuzzy variable from the current configuration;  $\boxplus^k$ ,  $\boxminus^k$  and  $\boxdot^k$ , respectively, adds, removes and modifies  $k$  fuzzy rules related to the current configuration;  $\sim$  changes the fuzzy implication method of the rule base related to the current configuration;  $\otimes$ ,  $\oslash$  and  $\odot$ , respectively, adds, deletes and modifies a term to a fuzzy variable in the current configuration;  $\alpha$  modifies the defuzzify method of configuration;  $\bowtie$  modifies the fuzzy aggregation method;  $*$  changes default value of an output variable in the current configuration;  $\ltimes$  and  $\rtimes$  change, respectively, the lower bound and upper bound of the universe of discourse of a fuzzy variable in the current configuration.

Once that the automaton computation starts over a given timed word, the state transitions will opportunely modify the current control configuration in the successive one. In this vision, a timed word determines how and when to execute the switching among successive control eras. So, a timed word coincides with the control time concept.

The collection  $C_{op}$  will be used to extend timed automata and formally define TAFC as a couple of two components: an extended timed automaton that describes the dynamic evolution of a system and a fuzzy logic controller modeling the control behaviour of system during first phase of its existence.

### C. Defining an extended timed automaton

Once defined the set of transformation operators  $C_{op}$  it is necessary to redefine the Timed Automaton concept in order to introduce a novel kind of transition edges capable of changing the control configuration of the modeled system. In particular, the standard transitions set of timed automata  $E$  is replaced with the following:

$$E_C \subseteq S \times S \times \Sigma \times 2^C \times \Phi(C) \times C_{op}^*$$

where  $C_{op}^*$  represents the set of all possible sequences of transformation operators, i.e.,  $C_{op}^* = \cup_{n \geq 1} C_{op}^n$  where  $C_{op}^n$  is the set of all possible sequences of  $n$  operators with  $n \geq 1$ . Now, it is possible to provide an extended definition of a timed automaton:

*Definition 7:* A *timed control transition table*  $\mathcal{A}_C$  is a tuple  $\langle \Sigma \cup \{\epsilon\}, S, S_0, C, E_C \rangle$ , where

- $\Sigma$  is a finite alphabet;
- $\epsilon$  represents the empty event, that is, when it is on a transition, the crossing of this transition depends only by temporal constraints;

- $S$  is a finite set of states;
- $S_0 \subseteq S$  is a set of start states;
- $C$  is finite set of clocks;
- $E_C \subseteq S \times S \times \Sigma \times 2^C \times \Phi(C) \times C_{op}^*$  gives the set of transitions. An edge  $\langle s, s', a, \lambda, \delta, o^n \rangle$  represents a transition from state  $s$  to state  $s'$  on input symbol  $a$  which can be also the empty event. The set  $\lambda \subseteq C$  gives the clocks to be reset with this transition,  $\delta$  is a clock constraint over  $C$  and  $o^n \in C_{op}^*$  is a sequence of  $n$  transformation operators, with  $n \geq 1$ , defined in order to change the current control configuration of modeled system. ■

In each sequence of transformation operators an operator can be repeated in order to execute the same task on different arguments. Besides, it is important to establish that the operators are executed in the same order of their definition in the sequence (definition 9).

**Definition 8:** Let  $F$  be a fuzzy controller and let  $o \in C_{op}$  be a transformation operator then  $G = o(F)$  is the fuzzy controller obtained to apply the operator  $o$  on fuzzy controller  $F$ . ■

**Definition 9:** Let  $F$  be a fuzzy controller and let  $o^n \in C_{op}^*$  be a sequence  $o^n = (o_1, o_2, \dots, o_n)$  where  $o_i \in C_{op} \forall i \in \{1, 2, \dots, n\}$  then  $G = o^n(F) = (o_n(o_{n-1}(\dots(o_2(o_1(F))\dots)))$  is the fuzzy controller obtained to apply the operators  $o_1, o_2, \dots, o_{n-1}, o_n$  on fuzzy controller  $F$  in the same order of listing in the sequence  $o^n$ . ■

#### D. TAFC formal definition

At this point, it is possible to give a formal definition of a TAFC and the properties characterizing its dynamic behaviour.

**Definition 10:** A TAFC  $T$  is an ordered pair composed by an initial control configuration, represented by a fuzzy controller named  $F^0$ , together with a timed control transition table  $T_C$ . Formally:

$$T = (F^0, T_C). \quad \blacksquare$$

The TAFC properties which define the dynamic behaviour of a system are: *control evolution* and *control run*.

The control evolution is a mapping among the states  $S$  contained in  $T_C$  and the collection of possible control configurations obtained starting from  $F^0$ . More in detail, the control evolution is a mathematical succession, generated in an inductive way, which maps each state in  $S$  with a one or more control configurations obtained by sequentially applying over  $F^0$  the transformation operators in  $S \times S \times \Sigma \cup \{\epsilon\} \times 2^C \times \Phi(C) \times C_{op}^*$ . Then:

**Definition 11 (Control evolution):** Let  $T = (F^0, T_C)$  be a TAFC defined over a timed control transition table  $\langle \Sigma \cup \{\epsilon\}, S, S_0, C, E_C \rangle$  with  $S = \{s_0, s_1, \dots, s_{|S|-1}\}$  the finite set of automaton states; let  $F^*$  be the collection of all possible fuzzy controllers; let  $\Omega = \{n_1, n_2, \dots, n_{|\Omega|}\}$  be a subset of ordered sequences in  $C_{op}^*$ , that is,  $n_i = (o_1, o_2, \dots, o_{|n_i|}) \forall i \in \{1, 2, \dots, |\Omega|\}$ , employed to define

the edges in  $E_C$ . Then, the control evolution  $\Psi$  over a state  $s^0 \in S_0$  is:

$$\Psi : \mathbb{N} \rightarrow S \times F^*$$

defined inductively, as follows:

The base case ( $i = 0$ ). Let  $s^0 \in S_0$  be an initial state of  $\langle \Sigma \cup \{\epsilon\}, S, S_0, C, E_C \rangle$ , then:

$$\Psi(0) = (s^0, F^0)$$

The inductive step ( $i > 0$ ). Let  $\Psi(i-1)$ , with  $i > 1$ , be defined as:

$$\Psi(i-1) = (s^{i-1}, F^{i-1})$$

where  $s^{i-1} \in S$  and  $F^{i-1} \in F^*$ , then:

$$\Psi(i) = (s^i, F^i)$$

with  $s^i \in S$ ,  $F^i = n_i(F^{i-1})$ ,  $n_i \in \Omega$  and  $\langle s^{i-1}, s^i, a, \lambda, \delta, n_i \rangle \in E_C$ . ■

More intuitively, the expression (3) shows the sequence of pairs composing a control evolution over  $s^0 \in S_0$  together with the fuzzy transformations obtained by exploiting the  $n_i$  sequences of operators.

$$\begin{aligned} \Psi(0) : s^0 \in S_0 &\rightarrow F^0 \\ &\downarrow n_0 \\ \Psi(1) : s^1 \in S &\rightarrow F^1 \\ &\downarrow n_1 \\ \Psi(2) : s^2 \in S &\rightarrow F^2 \\ &\downarrow n_2 \\ &\vdots \\ &\downarrow n_{j-1} \\ \Psi(j) : s^j \in S &\rightarrow F^j \\ &\downarrow n_j \\ &\vdots \end{aligned} \quad (3)$$

The image of function  $\Psi$ ,  $I_\Psi$ , can be finite or infinite. This depends upon the topology of graph modeling the component  $T_C$  of the TAFC. Indeed, if the topology of  $T_C$  contains cycles then various FLCs can be associated with a same state  $s_h \in S$ . In order to explain it, let  $\langle s^k, s^{i-1}, a, \lambda, \delta, n_k \rangle \in E_C$  be an edge entering in the state  $s^{i-1}$  already crossed, that is, an edge forming a cycle. If it is crossed the current fuzzy controller stored in the state  $s^{i-1}$  is replaced by a new one obtained executing the sequence of operators  $n_k$  on fuzzy controller stored in the state  $s^k$ . Then, crossing again the edge  $\langle s^{i-1}, s^i, a, \lambda, \delta, n_i \rangle \in E_C$  also the controller fuzzy stored in  $s^i$  is replaced. Indeed, even if the sequence of operators  $n_i$  is unchanged from the previous crossing of the edge, a new controller fuzzy is stored in  $s^i$  because  $n_i$  is executed on the different fuzzy controller which has been stored in  $s^{i-1}$  in the previous step. So, states  $s^{i-1}$  and  $s^i$  are both associated with a new fuzzy controller because of the cycle formed by the edge  $\langle s^k, s^{i-1}, a, \lambda, \delta, n_k \rangle$ .

Obviously, the control evolution only represents a mapping between the states of timed automaton  $T_C$  and the collection of control configurations computable starting from  $F^0$  by applying different sequences of operators in  $\Omega$ ; no dynamic

aspects are considered in the control evolution definition and, therefore, it is necessary to introduce the idea of control run extending the initial idea of the run of standard timed transition table (definition 6).

**Definition 12:** Let  $\Psi$  be a control evolution, then a control run  $r_c$ , denoted by  $(\bar{s}, \bar{\nu})$ , of a timed transition table  $\langle \Sigma \cup \{\epsilon\}, S, S_0, C, E_C \rangle$  over a timed word  $(\sigma, \tau)$  and a collection of sequences of transformation operators  $\Omega = \{n_1, n_2, \dots, n_{|\Omega|}\} \subseteq C_{op}^*$ , is an infinite sequence of the form

$$r_c : \langle s^0, \nu_0 \rangle \xrightarrow[\tau_1]{\sigma_1, n_1} \langle s^1, \nu_1 \rangle \xrightarrow[\tau_2]{\sigma_2, n_2} \langle s^2, \nu_2 \rangle \xrightarrow[\tau_3]{\sigma_3, n_3} \dots$$

with  $s^i \in S$  and  $\nu_i \in [C \rightarrow \mathbb{R}]$ , for all  $i \geq 0$ , and  $n_i \in C_{op}^*$ , for all  $i \geq 1$ , satisfying the following requirements:

- **Initiation:**  $s^0 \in S_0$  and  $\nu_0(x) = 0$  for all  $x \in C$ ;
- **Consecution:** for all  $i \geq 1$ , there is an edge in  $E_C$  of the form  $\langle s^{i-1}, s^i, \sigma_i, \lambda_i, \delta_i, n_i \rangle$  such that  $(\nu_{i-1} + \tau_i - \tau_{i-1})$  satisfies  $\delta_i$  and  $\nu_i$  equals  $[\lambda_i \mapsto 0](\nu_{i-1} + \tau_i - \tau_{i-1})$ ;
- **Atomicity:** The operators of sequence  $n_i \in C_{op}^*$  are atomic operations and their computation time is equals to 0, i.e., they do not modify the duration of permanence in the automaton state  $s^i, (\tau_i - \tau_{i-1})$ ;
- **Evolution:** each state  $s^i$  of a pair  $\langle s^i, \nu_i \rangle$  in  $r_c$  is mapped on a FLC  $F^i$  as described by the control evolution  $\Psi$ .

If  $T = (F^0, T_C)$  is a TAFC which models a given system then the set of control run  $r_c$  defined over the timed language  $L$ , generated by  $T_C$ , completely describes the collection dynamic behaviours of the system, whereas, the control run  $r_c$  defined over a single word  $w_i \in L$  defines a precise dynamic behaviour of the system, so  $w_i$  defines the control time.

**Definition 13 (Control Time):** If  $T = (F^0, T_C)$  is a TAFC and  $T_C$  is a timed automaton recognizing the timed language  $L = w_1, w_2, w_3, \dots, w_i, \dots$  and  $w_i$  is a timed word and  $r_c$  is a control run defined over  $w_i$  then  $w_i$  is a Control Time of the system. ■

Finally, it is possible to give a formal description of control era and control configuration concepts.

**Definition 14 (Control era and control configuration):** If  $r_c$  is a control run defined over the Control Time  $w_i = (\sigma, \tau) \in L$ :

$$r_c : \langle s^0, \nu_0 \rangle \xrightarrow[\tau_1]{\sigma_1, n_1} \langle s^1, \nu_1 \rangle \xrightarrow[\tau_2]{\sigma_2, n_2} \langle s^2, \nu_2 \rangle \xrightarrow[\tau_3]{\sigma_3, n_3} \dots$$

then time interval between the instant  $\tau_i$  and  $\tau_{i+1}$  is the  $i^{th}$  control era of system and the FLC  $F^i$  which depicts the system during the same interval is defined as the  $i^{th}$  control configuration. ■

Therefore, it is important to notice how a TAFC is an implementation of timed fuzzy controller thanks to the concept of the control evolution and control run which perform the task accomplished by previously defined function  $\varphi$ .

Both the control evolution and the control run are potentially based on the infinite concept. In fact, the control evolution can exploit an infinite application of control operators in  $\Omega$  to compute the mappings between the state  $s^i$  and the FLC

$F^i$ , whereas, the control run uses a timed word, defined as an infinite sequence of ordered pairs, to describe the dynamic behaviour of the system. Consequently, in order to simulate the behaviour of a TAFC during the first  $n$  control eras, the  $n^{th}$ -order control evolution and control run are introduced.

**Definition 15 ( $n^{th}$ -order Control Evolution):** If  $\Psi$  is a control evolution then the set

$$\Psi^n = \{\Psi(i) = (s^i, F^i) | i = 1, 2, \dots, n\}$$

which contains the first  $n$  ordered pairs computed by  $\Psi$  through definition 11 is the  $n^{th}$ -order control evolution. ■

**Definition 16 ( $n^{th}$ -order Control Run):** Let

$$r_c : \langle s^0, \nu_0 \rangle \xrightarrow[\tau_1]{\sigma_1, n_1} \langle s^1, \nu_1 \rangle \xrightarrow[\tau_2]{\sigma_2, n_2} \langle s^2, \nu_2 \rangle \xrightarrow[\tau_3]{\sigma_3, n_3} \dots$$

be a control run defined over a control evolution  $\Psi$ , then  $n^{th}$ -order control run  $r_c^n$  is the sequence of the first  $n$  elements of  $r_c$ :

$$r_c^n : \langle s^0, \nu_0 \rangle \xrightarrow[\tau_1]{\sigma_1, n_1} \langle s^1, \nu_1 \rangle \xrightarrow[\tau_2]{\sigma_2, n_2} \dots \xrightarrow[\tau_{n-1}]{\sigma_{n-1}, n_{n-1}} \langle s^n, \nu_n \rangle$$

where the mapping between the automaton states  $s^i$  and the FLC  $F^i$  is computed by the  $n^{th}$ -order control evolution related to  $\Psi$ . ■

## V. EXPERIMENTAL RESULTS

This section discusses the application of the proposed methodology to the task of voltage regulation for the IEEE 30-bus test system depicted in fig. 1.

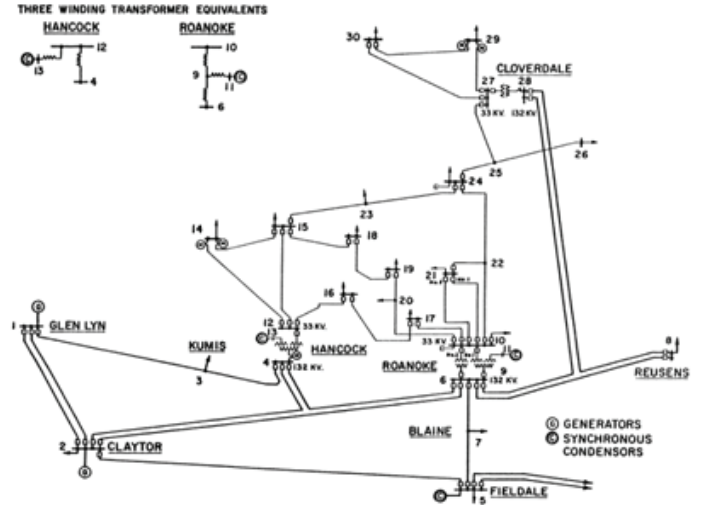


Fig. 1. The IEEE 30-bus test system

Six dispatchable DGs have been considered in our experiments. The regulating devices asset is, therefore, identified by 6 variables:

$$\mathbf{y} = [Q_{dg,1}, Q_{dg,2}, Q_{dg,3}, Q_{dg,4}, Q_{dg,5}, Q_{dg,6}] \quad (4)$$

In our experiment, each variable is regulated by an identical voltage TAFC. As far as the regulation performances are

concerned, detailed simulation analyses that consider highly variable load patterns have been developed.

In particular, the TAFC  $T = (C, T_A)$  is exploited for regulating the reactive power flows injected by the DGSs where  $C$  is a fuzzy logic controller representing the initial configuration of  $T$  and modeling the control behaviour of each variable during first phase of its existence, whereas,  $T_A$  is the extended timed automaton that describes the dynamic evolution. In particular, in the Fig. 2, the timed automata  $T_A$  used to regulate the reactive power flows injected by the DGSs during a simulation run over a time equal to 150 minutes is shown.

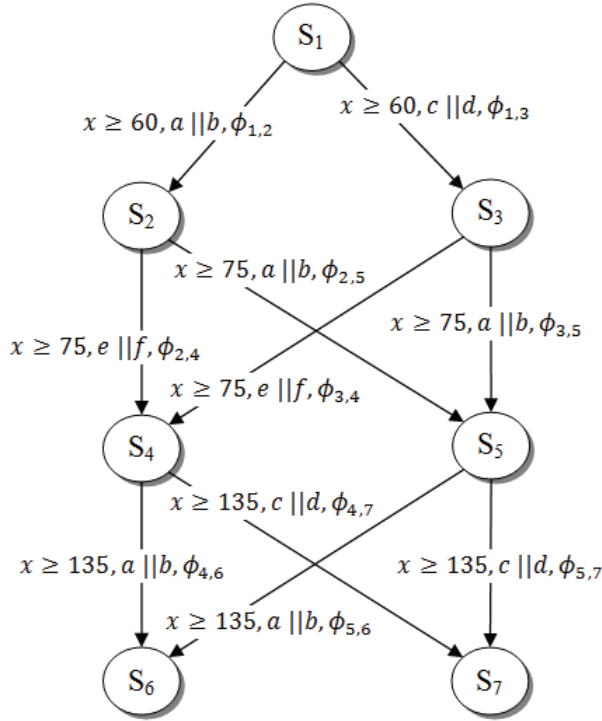


Fig. 2. The timed automata  $T_A$

In details, the timed automaton  $T_A$  is composed by seven states  $S_1, S_2, \dots, S_7$ . Each state stores a control configuration able to regulate the reactive power flows during the corresponding control era. For sake of simplicity, in the Fig. 2, the description of the transformation operators is replaced by a function  $\phi_{i,j}$  capable of calculating the set of operators needed to transform the control configuration related to the state  $S_i$  in the control configuration related to the state  $S_j$ . The events described through the letters  $a, b, c, d, e, f$  on the automaton transitions correspond to the following conditions related to the voltage magnitude  $M$ :

- $a \Leftrightarrow M < 0.99$
- $b \Leftrightarrow M > 1.01$
- $c \Leftrightarrow M > 0.99$
- $d \Leftrightarrow M < 1.01$
- $e \Leftrightarrow M < 0.97$
- $f \Leftrightarrow M > 1.03$

In order to show the better performances provided by our strategy, the proposed TAFC  $T$  has been compared with a standard fuzzy controller  $F$ .

The evaluation of the provided performances has been based on technical and economic aspects, and in particular on the power losses (see fig. 3), reactive power costs (see fig. 4), and voltage magnitude (see fig. 5).

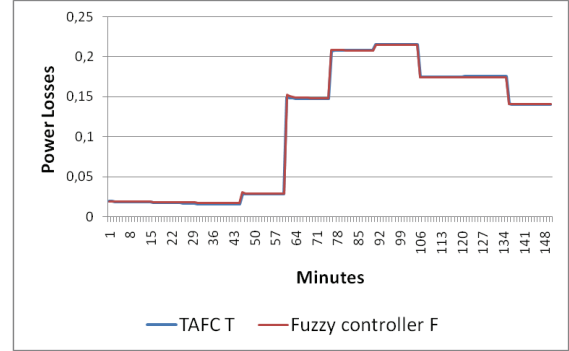


Fig. 3. The power losses values for the TAFC  $T$  and the conventional fuzzy controller  $F$

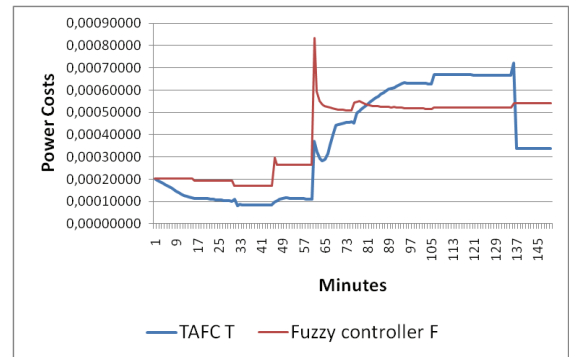


Fig. 4. The reactive power costs values for the TAFC  $T$  and the conventional fuzzy controller  $F$

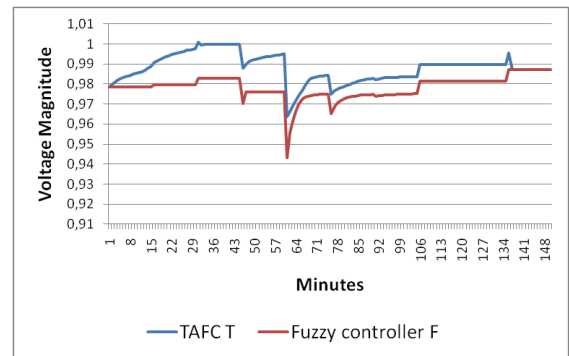


Fig. 5. The voltage magnitude values for the TAFC  $T$  and the conventional fuzzy controller  $F$

In order to effectively measure the superiority of the our strategy, the Mean Absolute Errors (MAEs) regarding to the considered performance aspects have been computed for the

compared controllers,  $T$  and  $F$ . More in detail, the MAE for each aspect has been calculate as:

$$MAE = \frac{1}{N} \cdot \sum_{i=1}^N |G(i) - T(i)| \quad (5)$$

where  $N$  is the number of iterations of the simulation (equal to the number of minutes of simulation) and  $G(i)$  and  $T(i)$  are the guessed and the desired value for the performance aspect (power losses, reactive power costs and magnitude voltage) in the  $i$ -th iteration, respectively. In particular, the value 0 has been considered as the desired value for the power losses and reactive power costs, whereas, 1 has been considered as the desired value for the magnitude voltage.

The table I shows the computed MAEs concerning the performance aspects for our proposal and the compared conventional fuzzy controller. As highlighted by the perceptual improvement shown in the last column of the table I, the exploited TAFC  $T$  provides better performances than a conventional fuzzy controller.

TABLE I  
MAES OF OUR CONTROLLER  $T$  AND THE STANDARD  $F$  FOR ALL PERFORMANCE ASPECTS. IN THE LAST COLUMN ARE REPORTED THE RELATIVE IMPROVEMENTS.

| Performance Aspect   | MAE of $T_A$ | MAE of $F$ | Rel. impr. |
|----------------------|--------------|------------|------------|
| Power losses         | 0,1142       | 0,1146     | 0,35%      |
| Reactive power costs | 3,7245E-4    | 4,0137E-4  | 7,2%       |
| Magnitude Voltage    | 0,0119       | 0,0216     | 44,9%      |

## VI. CONCLUSIONS

Modern trends in Smart Grids are oriented toward the employment of advanced regulation architectures that move away from the older centralized paradigm to a system distributed in the field with an increasing pervasion of smart sensors where central controllers play a smaller role. In supporting this complex task, this paper proposes the concept of a decentralized non-hierarchal voltage regulation architecture based on intelligent and cooperative smart entities. The distributed voltage controllers employ traditional sensors to acquire local bus variables and mutually coupled oscillators to assess the main variables that characterize the operation of the global Smart Grid. These variables are then amalgamated by a TAFC in order to identify proper control actions aimed at improving the grid voltage profile and reducing power losses. As far as the regulation performances are concerned, the simulation studies developed confirm the effectiveness of the proposed strategy in solving the voltage regulation problem also in the presence of highly variable load patterns.

## REFERENCES

- [1] C. Cecati, G. Mokryani, P. Siano, A. Piccolo, "An Overview on the Smart Grid Concept," *36th IEEE Annual Conference of the Industrial Electronics Society (IECON 2010)*, 7-10 November 2010, Phoenix, Arizona, USA, pp. 3316 - 3321, ISBN:978-1-4244-5226-2
- [2] V.Madani, R.L.King, "Strategies to meet grid challenges for safety and reliability," *International Journal of Reliability and Safety*, vol. 2, no.1-2, 2008.

- [3] S. M. Amin, B.F.Wollenberg, "Toward a Smart Grid," *IEEE Power and Energy Magazine*, vol.3, no 5, pp. 34-38, Sept/Oct. 2005.
- [4] A. Schroder, I. Laresgoiti, K.Werlen, B. Schowe-von der brelie, A. Schnettler, "Intelligent self-describing power grids," *Proc. of the 20th International Conference and Exhibition on Electricity Distribution, 2009. CIGRE 2009*, June 2009, Prague, Czech Republic.
- [5] R.L. King, "Information services for smart grids," *Proc. of 2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, July 2008.
- [6] G. Joos, B.T. Ooi, D. ;cGillis, F.D. Galiana, R. Marceau, "The potential of distributed generation to provide ancillary services" in *Proc. of IEEE Power Engineering Society Summer Meeting 2000*, vol. 3, pp. 1762 - 1767, July 2000.
- [7] G. Bontempi, A.Vaccaro, D.Villacci "An adaptive local learning based methodology for voltage regulation in distribution networks with dispersed generation," *IEEE transactions on Power Systems*, vol.21(3), Aug. 2006, pp.1131-1140.
- [8] K. Tomsovic, T. Hiyama, "Intelligent control methods for systems with dispersed generation" in *Proc. of IEEE Power Engineering Society Winter Meeting 2001*, vol.2, pp. 913 - 917, Feb. 2001.
- [9] M.I. Marei, E.F. El-Saadany, M.M.A. Salama, "Flexible distributed generation: (FDG)" in *Proc. of IEEE Power Engineering Society Summer Meeting 2002*, vol.1, pp. 49-53, Jul. 2002.
- [10] Spatti, D.H.; da Silva, I.N.; Usida, W.F.; Flauzino, R.A.; "Fuzzy Control System for Voltage Regulation In Power Transformers," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol.8, no.1, pp.51-57, March 2010.
- [11] Hassan A.R, Sadrul Ula A.H.M, "Design and implementation of a fuzzy controller Based Automatic Voltage Regulator for a Synchronous Generator," *IEEE Trans. on Energy conversion*, vol 9, no 3, September 1994.
- [12] M. Pipattanasomporn, H. Feroze, S. Rahman, "Multi-Agent Systems in a Distributed Smart Grid: Design and Implementation," *IEEE PES 2009 Power Systems Conference and Exposition (PSCE'09)*, Mar 2009, Seattle, Washington, USA.
- [13] N. Higgins, et al., "Concept of Intelligent Decentralised Power Distribution Automation with IEC 61850, IEC 61499 and Holonic Control," *Proc. IEEE Conference on Systems, Machine and Cybernetics*, 2008.
- [14] P. Siano, C. Citro, C. Cecati, A. Piccolo, "Smart Operation of Wind Turbines and Diesel Generators According to Economic Criteria," *IEEE Transactions on Industrial Electronics*, forthcoming, 2011, pp. 1-12.
- [15] G. Acampora, V. Loia, A. Vitiello, "Hybridizing fuzzy control and timed automata for modeling variable structure fuzzy systems," *Fuzzy Systems (FUZZ)*, 2010 *IEEE International Conference on*, pp.1-8, July 2010.
- [16] Z. Mi, F. Wang, "Substation Reactive Power and Voltage Control Using Fuzzy Control Theory," *Proc. of 2009 International Conference on Industrial and Information Systems*, Haikou, China, Apr. 2009.
- [17] R. Alur, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, pp. 183235, 1994.