1

Discrete-Event Supervisory Control for Under-Load Tap-changing Transformers (ULTC): from synthesis to PLC implementation

Ali A. Afzalian[1], S. M. Noorbakhsh[2] and W. M. Wonham[3]
[1]*Department of Electrical Engineering, Abbaspour University of Technology, Tehran, Iran*
[2]*Department of Electrical Engineering, Islamic Azad University-Boroujen Branch, Iran*
[3]*Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada*

1. Introduction

Discrete-event systems (DES) can be found as essential integrated subsystems in many complex systems, e.g. electrical power systems. Under-load tap-changing (ULTC) transformers, which obviously have discrete-event behaviour, are widely used in transmission systems to take care of instantaneous variations in the load conditions in substations. In this chapter, the voltage control problem in ULTC is solved in different modes of operation, using DES-based solutions. These solutions include: DES supervisory control, timed DES supervisory control and a hierarchical structure for the control system. It is shown that the specifications are controllable and the closed loop control system is non-blocking. A heuristic method has been used for easier implementation of the supervisor on a Programmable-Logic-Controller (PLC), to overcome the general implementation problems, as well as the implementation problem caused by the Auto/Manual mode of ULTC operation. A step-by-step procedure is developed to generate a ladder diagram code which implements the DES supervisor on a PLC.

A discrete-event system (DES) is a dynamic system that evolves in accordance with the sudden occurrence of physical events at possibly unknown irregular intervals (Ramadge & Wonham, 1989). The supervisory control technique is an effective analytical tool for automation and control of DES (Ramadge & Wonham, 1987). Discrete-event models are generally used to describe systems where coordination and control are required to ensure the orderly flow of events, and/or to prevent the occurrence of undesired chains of events. DES can be employed to describe a wide variety of behaviors in industrial and physical systems. These include control and scheduling of electrical power systems, manufacturing systems, queuing systems, communication protocols, and database management systems. The behavior of electrical power systems can be characterized by interactions between continuous dynamics and discrete-event dynamics.

In the last two decades, discrete-event systems have been studied by researchers from different fields, with respect to modeling, analysis and control. Several models have been

proposed and investigated. These models can be classified as *untimed* DES models and *timed* DES models. In an untimed model, when considering the state evolution, only the sequence of states visited is of concern. That is, only the logical behavior is of interest. In a timed model, both logical behavior and timing information are considered. (Brandin & Wonham, 1994) adjoined to the structure of untimed DES (Ramadge & Wonham, 1989) the timing features of timed transition models. The BW framework, which is used in this chapter, retains the concept of maximally permissive supervision introduced in (Brandin & Wonham, 1994), allows the timed modeling of DES, admits subsystem composition, and admits forcing and disablement as means of control. Different synthesis methods have been developed and implemented as the software TCT (for untimed models) and TTCT (for timed models) (Wonham, 2009) to compute controllers that are optimal in the sense that the controlled system not only satisfies the specifications but is also as permissive as possible. TCT and TTCT are used in this study for synthesizing the supervisory controllers.

There are good reasons for organizing the control of large systems in a distributed hierarchy structure. Among these are: deeper understanding facilitated by the hierarchical structure, reduction in complexity of communication and computation, modularity and adaptability to change, robustness, and generalization. The supervisory control of discrete-event systems can be designed to be hierarchically structured. In the present chapter, implementation of this approach to a control problem in electrical power systems is also discussed.

A power system, in its simplest representation, comprises a set of lines intersecting at nodes (buses). Energy is injected at buses by generators, and loads can be considered as negative injections. The flow of power along lines to and from buses is a phenomenon of primary interest in power system operation and control. Transformers with tap-changing facilities constitute an important means of controlling voltage throughout electrical power systems at all voltage levels. Transformers with off-load tap-changing facilities can help to maintain satisfactory voltage profiles. Under-load tap-changing transformers (ULTC) can be used to take care of daily, hourly, and minute-by-minute variations in system conditions. ULTC may be controlled either automatically or manually (Kundur, 1994). Many dynamic subsystems in a power system exhibit discrete-event behavior. Typically, the continuous dynamics relate to components that obey physical laws. Event-driven discrete behavior results from logical rules that govern the system. The continuous trajectory of the system state can be interrupted by discrete control actions and uncontrolled disturbances, which may be frequent or infrequent. The time scale for these events changes from milliseconds, through seconds and minutes, to hours, days, and weeks or longer (Fink, 1999).

Discrete-event systems theory has been applied to problems in electrical power systems (Prosser, 1995, Selinsky et al., 1995; Lee & Lim, 2004; Lin et al., 2004; Afzalian et al., 2006, Afzalian & Noorbakhsh, 2008; Afzalian & Wonham, 2006 & 2009; Noorbakhsh & Afzalian, 2007a&b & 2009). These applications include: supervisory control, modeling and analysis, and monitoring and diagnosis. The synthesis of a DES-based supervisory control for ULTC was introduced in (Afzalian et al., 2006), where the ULTC along with different specifications (control logics) were modeled as automata. The automatic voltage controller of a tap-changer transformer can be regarded as a discrete–event system. The processes associated with this system may be regarded as asynchronous and discrete in time and/or state space. A DES generating a formal language can be considered as a representation of this tap-changer transformer (plant).

Though supervisory control (SC) theory has received substantial attention for over a decade in academia, industrial applications are scarce. The main reason for this seems to be a discrepancy between the abstract supervisor and its physical implementation. Typically, finite-state automata describe the plant, specification and supervisor. But the step to a physical implementation is not necessarily straightforward. In the special case of industrial systems, where PLC-control is of great importance, the gap between the event-based asynchronous automata world and the synchronous signal based PLC-world has to be bridged. The asynchronous event-driven nature of the supervisor is not straightforwardly implemented in the synchronous signal-based PLC. The first attempt to implement a DES supervisory control on a PLC was made by Leduc (Leduc & Wonham, 1995, Leduc, 1996). PLC implementation of DES supervisory control was discussed in (Leduc & Wonham, 1995; Fabian & Hellgren, 1998; Dietrich et al., 2001; Hellgren et al., 2002; Jiang & Darabi, 2002; Gasper 2002, Max et al. 2002, Vieira et al. 2006, Noorbakhsh & Afzalian 2007a&b, Manesis & Akantziotis, 2005; Noorbakhsh, 2008; Afzalian & Noorbakhsh, 2008; André et al., 2009).
After a brief review of DES supervisory approaches, this chapter deals with the modeling of ULTC as an automaton. Control specifications in each mode of operation are also modeled as finite automata. As a first solution, supervisory controllers are designed for the ULTC in Automatic and Auto/Manual modes of operation. The second solution employs the timed DES approach to design a supervisory control for the ULTC. A hierarchical structure for the supervisory control of the problem is also investigated as the third solution to the ULTC control problem. A two-level hierarchy structure has been used to control the ULTC. A manager has been introduced in the high-level to shut down the system in certain contingencies. The manager deals with an abstract model of the plant in the high-level, and so can apply the control requirements easily. It is shown that a high-level manager can easily supervise the plant using this abstract model of the low-level subsystem, i.e. the low level closed loop control system of ULTC. A step-by-step transformation procedure transferring the automaton of the designed supervisor into a ladder diagram for PLC is presented in Section 7.
The contributions of the chapter are summarized as follows:

1- Discrete-event system modeling of an under-load tap-changing transformer and its control specification.
2- Evaluation of the required properties for the supervisory control system, i.e. controllability, non-blocking, and freedom from conflict.
3- The synthesis of a DES-based supervisory control in the monolithic, modular and hierarchical structures.
4- Systematic approaches for the implementation of supervisory control solutions.

2. Supervisory Control of DES

The supervisory control problem for a discrete-event system is formulated by modeling the plant as well as its control logic (specifications) as finite automata. To solve the supervisory control problem, it is necessary to show that a controller which forces the specification to be met exists and is constructible (Wonham, 2009).

2.1 Discrete-Event Models

A DES model is specified by: the set of states (including an initial state, and marker states which in some applications can be desired or target states), the set of events, and the state transition function of the system. Formally, a DES is represented by an automaton G = (Q, Σ, δ, q_0, Q_m) in which Q is a finite set of states, with $q_0 \in Q$ as the initial state and $Q_m \subseteq Q$ being the desired (marker) states; Σ is a finite set of events (σ) which is referred to as an alphabet; and finally δ is a transition mapping δ: Q×Σ →Q, δ(q, σ)= q' which gives the next state q' after an event σ occurs when G is in the state q. In general δ is only *partially* defined on Q×Σ. G plays the role of the plant and, together with its states, events and transition operator (mapping) models a physical process. G is called a generator, as it generates a set of strings (sequences of events). In other words it generates a language L(G), consisting of strings of events which are physically possible in the plant.

Let Σ^* denote the set of all finite strings of symbols in Σ, including the empty string, denoted ε. A *prefix* of a string *s* is an initial subsequence of *s*, i.e. if *r* and s are strings in Σ*, *u* is a prefix of *s* if *ur*=*s*. A set which contains all the prefixes of each of its elements is said to be *prefix closed*. Clearly, Σ^* is a prefix closed set. As some sets of strings may not contain all of their prefixes, the *prefix closure* of a set *A*, denoted by $\overline{A}$, is defined which contains all the prefixes of each element of *A*. If $A=\overline{A}$, then the set A is prefix-closed. If *A* is not prefix-closed, then $A \subset \overline{A}$. The language *L*(*G*) is the set of all event sequences which are physically possible in the plant. $L(G)=\{s \mid s \in \Sigma^*, \delta(q_0, s) \text{ is defined}\}$. Clearly, *L*(*G*) is a subset of Σ^*, and *L*(*G*) is also prefix-closed, because no event sequence in the plant can occur without its prefix occurring first. Those strings which can be extended to a marker state are of particular importance. The *marked* behavior, denoted by $L_m(G)$, is the sublanguage of *L(G)* consisting of all strings which reach some marker state. L_m(G) is a subset of *L*(*G*) and can be formally given as: $L_m(G)=\{s \in L(G) \mid \delta(q_0, s) \in Q_m\}$.

A discrete-event system is said to be *non-blocking* if $\overline{L_m(G)} = L(G)$. This means that there always exists a sequence of events which takes the plant from any (reachable) state to a desired (marker) state. In some applications of DES models, it is necessary to consider several independent and asynchronous processes simultaneously. There is a procedure called *synchronous product* which combines two DES (G_1 and G_2) into a single, more complex DES, i.e. $G_3 = G_1 \,||\, G_2$. The synchronous product defines new states for G_3 as ordered pairs of states from G_1 and G_2. The event set for G_3 is the union of event sets for G_1 and G_2. The initial and marker states of G_3 are defined similarly.

2.2 Controllable Specifications and Non-blocking Supervisor

A discrete-event plant must be controlled based on certain specifications (required behavior logic). By adjoining controller structure to the plant, it is possible to vary the language generated by the closed loop system within certain limits. The desired performance of such a controlled plant will be specified by stating that its generated language must be contained in some specification language. It is often possible to meet these specifications in a minimally restrictive way, called optimal supervision in the DES literature.

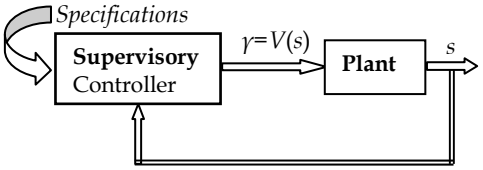


Fig. 1. Block diagram of a supervisory control system.

Suppose $G=(Q, \Sigma, \delta, q_0, Q_m)$, is a nonempty DES representing the plant which must be controlled. $\Sigma = \Sigma_c \cup \Sigma_u$ is the set of controllable and uncontrollable events in the plant. Σ_c is the set of controllable events; these can be enabled or disabled by an external agent (supervisor). A possible set of enabled events which includes some controllable events and all uncontrollable events is called a *control pattern* (γ). Uncontrollable events (Σ_u) are always enabled by their nature. Then it is clearly true that $\Sigma \supseteq \gamma \supseteq \Sigma_u$. The set of all control patterns, which is actually a family of sets, is defined as: $\Gamma = \{\gamma \in Pwr(\Sigma) \mid \gamma \supseteq \Sigma_u\}$. A supervisory control for the plant G is any function V: $L(G) \rightarrow \Gamma$. The pair (G, V) is written V/G, to suggest the concept of "G under the supervision of V".
The plant along with the supervisor forms a closed loop system (Fig. 1). The Plant G, generates strings of events $s \in L(G)$ and sends them to the supervisor as a feedback signal. The supervisory controller, which has been designed based on a required behavior of the plant (specifications), first determines implicitly in which state the system is working and then sends a list of events which are allowed to be enabled in that particular state, as a control signal to the plant. The supervisory controller is actually a DES synthesized using specifications in such a way as to guarantee the required behavior of the plant. The *closed behavior* of the system is defined to be the language $L(V/G) \subseteq L(G)$ described as follows:

- $\varepsilon \in L(V/G)$
- If $s \in L(V/G)$, $\sigma \in V(s)$, *and* $s\sigma \in L(G)$, then $s\sigma \in L(V/G)$
- No other strings belong to $L(V/G)$

In other words, the closed loop system only generates either the "empty" string or a string of the plant which is concatenated immediately by an event declared by the supervisor as allowed. Clearly $L(V/G)$ is nonempty and closed. The marked behavior of V/G is: $L_m(V/G)=L(V/G)\cap L_m(G)$. In other words, the strings reaching marker states in V/G are exactly the strings of $L_m(G)$ that survive under supervision by V. It is always true that $\emptyset \subseteq L_m(V/G) \subseteq L_m(G)$. The supervisor V is said to be *non-blocking (for G)* if $\overline{L}_m(V/G)=L(V/G)$. A language K representing some specification of a plant G is said to be *controllable (with respect to G)* if its prefix-closure $\overline{K}$ doesn't change under the occurrence of uncontrollable events in G. In other words, K is controllable if and only if $\overline{K}\,\Sigma_u \cap L(G) \subseteq \overline{K}$, where $\overline{K}\,\Sigma_u = \{s\sigma \mid s \in \overline{K},\ \sigma \in \Sigma_u\}$. Therefore the controllability condition on specification K only constrains $\overline{K} \cap L(G)$. Based on this definition, to test the controllability of K, one only needs to test its closure $\overline{K}$. The existence of an optimal (marking) non-blocking supervisory controller is proved in (Wonham, 2009). Let $K \subseteq L_m(G)$, $K \neq \emptyset$. Then there exists a supervisory controller V such that $L_m(V/G)=K$ if and only if K is controllable. The supervisory control of a discrete-event system enforces the controllable and non-blocking

behavior of the plant that is admissible under the given specification. The optimal solution to the supervisory control problem is the supremal controllable sublanguage (of the specification language). The DES representing the supremal supervisor typically has a large state size. Its state size is of order the product of state sizes of the plant and specification (plant control logic) DES models. Actually, the supremal supervisor contains redundant information about transition constraints which are already enforced by the plant itself. Therefore, the state size of the supremal supervisor can be reduced without affecting controlled behavior of the closed-loop system (Su & Wonham, 2004). A reduced supervisor has the following advantages:

- Easier implementation.
- The simpler structure may provide the designer with better understanding of the supervisor's control actions.
- The supervisor reduction is useful in the design of modular controls, where optimal local modular supervisors may admit quite small reduced versions that are simple and practical to implement.

It is shown in (Su & Wonham, 2004) that, finding a supervisor of minimal size is an NP-hard problem. Usually, a supervisor is looked for which is smaller than supremal supervisor (S) that does the job. The TCT procedure, *supreduce* (Plant, Supervisor, *condat*($\cdot$)) procedure calculates a small equivalent implementation of the supervisor (S_r) such that the following conditions are satisfied: $L(G) \cap L(S_r) = L(S)$ and $L_m(G) \cap L_m(S_r) = L_m(S)$.

The following steps can be done to design and implement a supervisory controller for a given plant (G) and given specifications:

1) Model the plant (components) as automata.
2) Model the specifications as DES and construct one DES, called *EDES*, representing all the specifications together. This can be done by the "*meet*" operation in TCT.
3) Find the non-blocking supervisory controller using the "*supcon*" operation in TCT i.e. *SUPER*=supcon(*G*, *EDES*).
4) There are some redundant constraints in *SUPER*, as the latter embodies a controller with a larger than necessary number of states and/or number of transitions. To simplify the supervisor the command "*supreduce*" in TCT can be used. In this procedure certain (automated) heuristics are employed to reduce the supervisor. The reduced supervisor has exactly the same control action as the original, but is structurally more economical.

This was a quick review of DES supervisory control. The TDES model is briefly reviewed in next subsection.

2.3 Timed Discrete-Event Systems

This section briefly reviews the *TDES* model proposed by (Brandin & Wonham, 1994). First, a finite automaton $G_{act} = (A, \Sigma_{act}, \delta_{act}, a_0, A_m)$ is introduced, which is called an *activity transition graph* (ATG) to describe the untimed behavior of the system. In G_{act}, A is a finite set of *activities*, Σ_{act} is the finite set of *events*, a partial function $\delta_{act} : A \times \Sigma_{act} \to A$ is the activity *transition* function, $a_0 \in A$ is the *initial* activity, and $A_m \subset A$ is the subset of *marked*

activities. In order to construct a TDES model, timing information is introduced into G_{act}. Let N denote the nonnegative integers. In Σ_{act}, each event σ will be equipped with a *lower time bound* $l_\sigma \in N$ and an *upper time bound* $u_\sigma \in N \cup \{\infty\}$ such that $l_\sigma \le u_\sigma$. Then the set of events is decomposed into two subsets, the *prospective* events $\Sigma_{spe} = \{\sigma \in \Sigma_{act} | u_\sigma \in N\}$ and the *remote* events $\Sigma_{rem} = \{\sigma \in \Sigma_{act} | u_\sigma = \infty\}$. For a detailed discussion and interpretation see (Wonham, 2009). The lower time bound would typically represent a delay, while an upper time bound is a hard deadline.

For each $\sigma \in \Sigma_{act}$, the timer interval T_σ is defined as $T_\sigma = \begin{cases} [0, u_\sigma] & if \sigma \in \Sigma_{spe} \\ [0, l_\sigma] & if \sigma \in \Sigma_{rem} \end{cases}$. The TDES defined by (Brandin & Wonham, 1994) is a finite automaton $G = (Q, \Sigma, \delta, q_0, Q_m)$ which can be displayed by its *timed transition graph* (*TTG*). The state set Q is defined as $Q = A \times \prod \{T_\sigma | \sigma \in \Sigma_{act}\}$. A state $q \in Q$ is of the form $q = (a, \{t_\sigma | \sigma \in \Sigma_{act}\})$, where $a \in A$ and $t_\sigma \in T_\sigma$. The initial state $q_0 \in Q$ is defined as $q_0 = (a_0, \{t_{\sigma,0} | \sigma \in \Sigma_{act}\})$, where $t_{\sigma,0} = \begin{cases} u_\sigma & if \sigma \in \Sigma_{spe} \\ l_\sigma & if \sigma \in \Sigma_{rem} \end{cases}$.

The set $Q_m \subseteq Q$ is given by a subset of $A_m \times \prod \{T_\sigma | \sigma \in \Sigma_{act}\}$. The event set Σ is defined as $\Sigma = \Sigma_{act} \cup \{tick\}$, where the additional event *tick* represents the passage of one time unit. The state transition function $\delta : Q \times \Sigma \to Q$ is defined as follows. For any $\sigma \in \Sigma$ and any $q = (a, \{t_\tau | \tau \in \Sigma_{act}\}) \in Q, \delta(q, \sigma)$ is defined, written $\delta(q, \sigma)!$, if and only if one of the following conditions holds:

- $\sigma = tick$ and $\forall \tau \in \Sigma_{spe}; \delta_{act}(a, \tau)! \Rightarrow t_\tau > 0$
- $\sigma \in \Sigma_{spe}$ and $\delta_{act}(a, \sigma)!$ and $0 \le t_\sigma \le u_\sigma - l_\sigma$
- $\sigma \in \Sigma_{rem}$ and $\delta_{act}(a, \sigma)!$ and $t_\sigma = 0$

When $\delta(q, \sigma)!, q' = \delta(q, \sigma) = (a', \{t'_\tau | \tau \in \Sigma_{act}\})$ is defined as follows:

- if $\sigma = tick$ then $a' = a$ and for all $\tau \in \Sigma_{act}, t'_\tau := \begin{cases} t_\tau - 1, & if \ \delta_{act}(a, \tau)! \wedge t_\tau > 0 \\ t_\tau, & otherwise \end{cases}$
- if $\sigma \in \Sigma_{act}$ then $a' = \delta_{act}(a, \sigma), t'_\sigma = t_{\sigma,0}$, and for $\tau \in \Sigma_{act}$ if $\tau \ne \sigma$ then
  $t'_\tau := \begin{cases} t_\tau & if \ \delta_{act}(a', \tau)! \\ t_{\tau,0} & otherwise \end{cases}$

The function δ is extended to $\delta : Q \times \Sigma^* \to Q$ in the natural way.

The *closed* behavior, the strings that are generated by *G*, and *marked* behavior, the strings that are generated by G and lead to a marker state, of the TDES G are defined by $L(G) = \{s \in \Sigma^* | \delta(q_0, s)!\}$ and $L_m(G) = \{s \in \Sigma^* | \delta(q_0, s) \in Q_m\}$, respectively. *G* is called *non-blocking* if $\overline{L_m(G)} = L(G)$. As in untimed supervisory control, the set Σ_{act} is partitioned into two subsets Σ_c and Σ_u of controllable and uncontrollable events. An event δ that can

preempt the event *tick* is called a *forcible* event. The set of forcible events is denoted by Σ_{for}. A forcible event can be either controllable or uncontrollable. By forcing an enabled event in Σ_{for} to occur, the event *tick* can be disabled. In this framework a supervisor repeatedly decides to disable or enable each event in $\Sigma_c \cup \{tick\}$.

The simplest way to visualize the behavior of a TDES *G* under supervision is first to consider the infinite reachability tree of *G* before any control is operative (Wonham, 2006). Each node of the tree corresponds to a unique string *s* of *L*(*G*). At each node of the tree the subset of *eligible* events can be defined by $Elig_G(s) := \{\sigma \in \Sigma \,|\, s\sigma \in L(G)\}$. In order to define the notion of *controllability* a language $K \subseteq L(G)$ is considered to define: $Elig_K(s) := \{\sigma \in \Sigma \,|\, s\sigma \in \overline{K}\}$. K is *controllable* with respect to *G* if, for all $s \in K$, $Elig_K(s) \supseteq \begin{cases} Elig_G(s) \cap (\Sigma_u \cup \{tick\}), & Elig_K(s) \cap \Sigma_{for} = \phi \\ Elig_G(s) \cap \Sigma_u, & Elig_K(s) \cap \Sigma_{for} \neq \phi \end{cases}$.

The control objective is, for the given plant language $L(G_p)$ and the specification language $L(G_s)$, to find a supervisor such that the closed loop language is, in the sense of set inclusion, the largest sublanguage of $L_m(G_p) \cap L_m(G_s)$ which is controllable w.r.t G_p and also non-blocking, written $\sup C(L_m(G_p), L_m(G_s))$.

2.4 Hierarchical Control Structure

A brief overview of hierarchical supervisory control for DES is given in this section. The reader is referred to (Wonham, 2009) for a detailed discussion. Roughly speaking, a complex system is one made of a large number of parts that interact in a non-simple way (Simon, 1962). In such systems, the whole is more than the sum of the parts. In other words, given the properties of the parts and the laws of their interaction, it is not trivial to infer the properties of the whole. Often, complexity takes the form of hierarchy. Hierarchical structure is a common feature of control solutions of complex dynamic systems. A complex system is composed of subsystems which, in turn have their own subsystems until some lowest level of elementary subsystems is reached. The scope of a control action is defined by the breadth of its temporal horizon and/or by the depth of its logical dependence in a task breakdown. The broader the temporal horizon of control subtasks, or the deeper its logical dependency on other controls, the higher it is said to reside in the hierarchy. Hierarchical systems possess some general features that are independent of their specific application (Zhong & Wonham, 1990).

The DES supervisory control can be designed to be hierarchically structured. Fig. 2 shows a two-level hierarchy consisting of a low level plant and controller, e.g. as field level, and a high-level plant and controller, e.g. as management level [7, 9]. The actual plant, for example a tap-changing transformer, is controlled in the real world by the operator; while the high-level plant is an abstract and simplified model of the actual plant that is employed for decision-making in the ideal world by the manager, e.g., the substation manager in an electrical power system. The high-level plant model is refreshed or updated every so often via the report channel from the actual plant. Alternatively, this report channel can be interpreted as carrying information sent by the operator to the manager, in terms of significant events. The information channel from the plant to the low-level controller

provides the conventional feedback path. The low-level controller applies conventional control to the plant through the "control law" channel.

How is the hierarchical loop closed? The function of the "command" channel is to convey the high-level manager's command to the operator, which in turn must translate (compile) these commands into corresponding low-level control signals which will actuate the plant. State changes in the plant will eventually be conveyed in summary and abstract form to the management level via the report channel. The high level plant is updated accordingly and then provides appropriate feedback to the manager through the "advice" channel. The command centre of a complex system, such as an electric power distribution system or a micro-grid, can be considered as the site of the "high-level plant" where a high-level decision maker (manager) is in command. The external (real) world and those (operators) coping with it are embodied in the low-level plant and controller.

The problem to be addressed concerns the relationship between the required or expected behavior of the high-level model (G_h) by the manager, and the actual behavior implemented in the plant (G_l) by the operator. It will turn out that a relationship of *hierarchical consistency* constrains the report channel from the low to the high level. In other words, it is necessary to refine the information conveyed by this channel, before a consistent hierarchical control structure can be achieved. The information sent up by the operator to the manager must be timely, and sufficiently detailed for various critical low level situations to be distinguished.

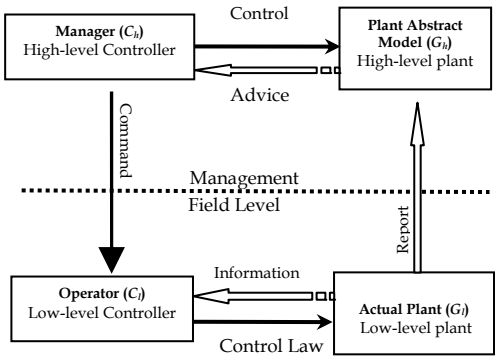


Fig. 2. A two-level hierarchy control system

2.5 Hierarchical Control Action in a Two-Level Controlled DES

Suppose the actual plant is modeled by an automaton $G_l=(Q, \Sigma, \delta, q_0, Q_m)$ that generates a language $L_l := L(G_l) \subseteq \Sigma^*$ as its uncontrolled behavior. Σ^* is the set of finite strings s, for which the extended transition map $\delta : Q \times \Sigma^* \to Q$ is defined.

Recall from DES supervisory control (section 2) that to every specification represented by a closed language E_l, there corresponds a supervisor as the (closed) supremal controllable sublanguage sup $C(E_l \cap L(G_l))$. The following notation is used for this supervisor: $M\uparrow$:= sup $C(M)$. The refined information flow through the "report" channel consists of strings of

significant events, represented by symbols in a high-level alphabet T. Thus the "report" can be modeled as a causal map $\theta : L_l \to T^*$ with the following properties: $\theta(\varepsilon)=\varepsilon$, $\theta(s\sigma)=$ either $\theta(s)$ or $\theta(s)\tau$ for some $\tau \in T$, where $s \in L_l$ and $\sigma \in \Sigma$.

An abstract model for the plant in the high level can be given as an automaton G_h that generates the language $L_h := \theta(L_l) \subseteq T^*$. The high-level controller C_h that observes only the strings of L_h must be able to make meaningful control decision. The following steps and related TCT procedures were proposed to formulate the suitable control structure (Wonham, 2009): Adopt the usual supervisory structure having the same type as in G_l (**Supcon(.,.)**)

1) Refine the state structure of G_l (**Recode(.)**)
2) Extend the high-level event alphabet T (**Vocalize(.,.)**)
3) Find the corresponding structure for G_h (**Higen(G_l)**)
4) Partition this extension into controllable and uncontrollable subsets to provide the manager with the ability to set up specifications in terms of controllable events. This is achieved by converting the G_l to a new DES called "output-control-consistent" in which each output event is unambiguously controllable or uncontrollable. (**Outconsis(G_l)**)
5) Design a high-level supervisory control using a given specification (E_h) for G_h (**Supcon(.,.)**)

The behavior E_h expected by the manager in G_h may be larger than what the operator can actually realize. In other words the manager may be over-optimistic in respect to the effectiveness of the command-control process. But if E_h is not larger than what the operator can realize at the low level, i.e. the equation $\theta((\theta^{-1}(E_h))^{\uparrow})=E_h$ holds for every closed and controllable language $E_h \subseteq L_h$ then, the pair (G_l, G_h) is said to possess hierarchical consistency. Achieving this equality in the hierarchical control system requires a further refinement of the transition structure of the DES model of the low-level plant, in other words, enhancement of the information sent up to the high-level. Such enhancement might or might not be feasible in an application. In TCT, hierarchical consistency can be achieved by running the **Hiconsis(G_l)** procedure.

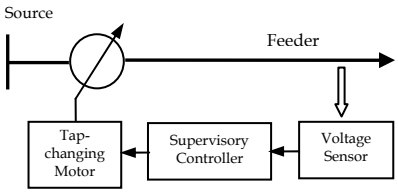


Fig. 3. Block diagram of control system for automatic changing of transformer taps

The two-level hierarchy discussed here can be extended to any number of levels. Once hierarchical consistency has been achieved for the bottom level and first level up, the construction may be repeated on assigning state outputs in the first level and bringing in the next higher level.

3. Tap-Changing Transformer

Transformers with tap-changing facilities constitute an important means of controlling voltage throughout electrical power systems at all voltage levels. Transformers with ULTC are widely used in transmission systems. For example, Ontario Hydro provided ULTC facilities on most 500/230 kV autotransformers and on all "area supply" transformers stepping down from 230 kV or 115 kV to 44 kV, 27.6 kV, or 13.8 kV (Kundur, 1994).

Whereas many articles considered ULTC as a nonlinear element in the power system models for voltage stability studies, a Petri net based model for tap-changer has been used in a framework of differential, switched algebraic and state-reset equations (Hiskens & Sokolowski, 2001). The control logic for tap-changer transformers can be found in the literature (Ohtsuki et al., 1991; Kundur, 1994; Otomega et al., 2003) as well as in manufacturers' catalogues (e.g. (GE Consumer Industrial, 2005)) in varying detail. When the voltage is not "normal" (i.e. is outside a desired limit) then after a time delay the controller changes the tap ratio to restore the voltage, i.e. bring it back into its dead-band. The delay is used to prevent unnecessary tap changes in response to transient voltage variations and to introduce the desired time delay before a tap movement. Fig. 3 shows the block diagram of a ULTC.

The timing behavior of the ULTC suggests a TDES approach to the supervisory control solution. To synthesize a supervisory control for the ULTC, the designer needs to be equipped with DES (TDES) models of the plant and the control specifications which are given in section 4. In sections 4, 5, and 6, DES, hierarchical structure, and TDES approaches are employed respectively to implement the supervisory control for the ULTC.

4. DES Supervisory Control for ULTC

In this section, the DES models of the plant and the control logic governing the ULTC are discussed. The models will be used later to study implementation of the supervisory controller.

4.1 DES Modelling of the Plant

As shown in Fig. 3, a ULTC (plant) consists of three components: Voltmeter, Timer, and Tap-changer. Each component is modeled as a DES. Then DES models of plant components are synchronized to form the plant model.

Voltmeter: The (measured) load voltage (V_l) must be within a dead-band ($V_o \pm$ ID), where: V_o is the set point, $\Delta V := V_o - V_l$, is the Voltage Deviation and ID: Insensitivity Degree, which is defined as the maximum admissible variation of the voltage before originating a command to change the tap. Voltmeter reports the following events associated with the load voltage: (Fig. 4):

- Voltmeter Initialized (ev11)
- Report $|\Delta V| >$ ID and ΔV is Negative (ev10)
- Report $|\Delta V| <$ ID (Voltage Recovered) (ev12)
- Report $|\Delta V| >$ ID and ΔV is Positive (ev14)
- Report Voltage exceeds V_{max} (ev16)

Timer: The timer times out after a certain delay *Operating Time* (OT). The following events are associated with the timer (Fig. 4):

- Timer Starts (ev21)
- Timer Blocks and Resets (ev25)
- Timer Times out (ev27)
- Timer Resets (ev23)

Tap-changer: The transformer tap-changer controls the transformer ratio "manually" or "automatically" in order to keep the power supply voltage practically constant, independently of the load. If the tap increase (decrease) is successful, the system returns to a state and waits for another command. If the tap increase (decrease) operation fails, the controller changes to the Manual mode, and waits for another command.

It is assumed here that the tap-changer has 5 steps. Events associated with the TAP-CHANGER are (Fig. 4):

- Tap down command (ev31)
- Tap down successful (ev32)
- Tap up command (ev33)
- Tap up successful (ev34)
- Tap up/down failed (ev30)

DES models of three plant components will be synchronized in order to get an automaton for the plant.

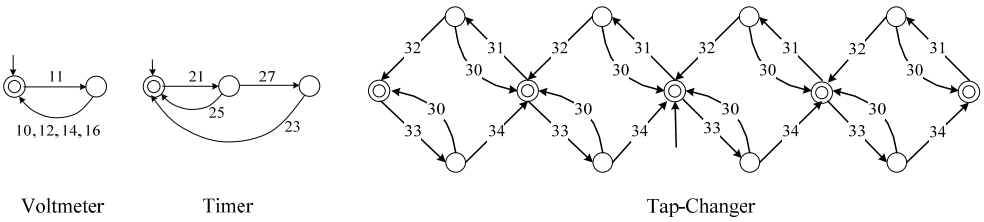


Fig. 4. DES models of different components of the ULTC

4. 2 Control Specifications

The control logic for an under-load tap-changing transformer is normally provided by the manufacturer and/or by the designer. A control logic which is given in (GE Consumer Industrial, 2005) by the GE company is employed in this chapter. The control logic is modeled by suitable automata, which will be described in this section.

The coordination control of the ULTC transformer and other FACTS (Flexible AC Transmission Systems) devices can be achieved by defining appropriate specifications (Thukaram et al., 2004; Kim & Lee, 2005). DES models of these specifications can be used to design modular supervisors. In a hierarchical control structure, the coordination control can be considered as higher level control logic.

There are two modes of operation: "Automatic" and "Manual".

I. Automatic Mode

The tap-changer works in Automatic mode according to the following logic (control specifications):

a. If the voltage deviation $| \Delta V |$ > ID and ΔV is Negative (ev10) then the timer will start and when it "times out", i.e. reaches its maximum (ev27) then a "tap increase command" (ev33) will be made and the timer will be "reset" (ev23).

b. If the voltage deviation $| \Delta V |$ > ID and ΔV is Positive (ev14) then the timer will start and when it "times out" i.e. reaches its maximum (e27) then a "tap decrease command" (ev31) will be made and the timer will be "reset" (ev23).

c. If the voltage returns to the dead-band (ev12), because of smooth system dynamics or a tap-changer or some other system events, then the timer is blocked and reset (ev25).

d. If the voltage exceeds the value set for "Quick Lowering" (ev16), then the timer OT becomes 0 seconds and therefore the lowering tap command (ev31) happens instantaneously.

Fig. 5 shows the DES model of the control specification in the Automatic mode. It actually implements all above logics in a single automaton. The automatic voltage controller of a tap-changer transformer can be regarded as a discrete–event system. The processes associated with this system may be thought of as asynchronous and discrete in time and/or state space. A DES generating a formal language can be considered as a representation of this tap-changer transformer (plant).

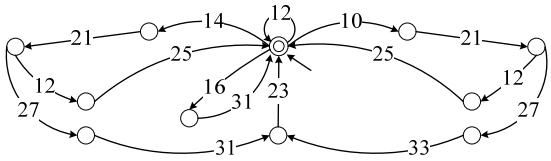


Fig. 5. DES model of the control logic (specification) for ULTC in Automatic mode.

II. Incorporating Operator Override (Auto/Manual mode)

If a fault in tap increase or decrease happens (ev30), or the operator forces the system from Automatic to Manual mode at any time (ev43), the system moves to the Manual state and waits for the operator. In the Manual mode of operation, a model for the operator action is needed to switch the modes and to override in abnormal situations.

OPERATOR: Events associated with the OPERATOR are (Fig. 6-a):

- Enter "Automatic" Mode (ev41)
- Enter "Manual" Mode (ev43)

The operator can force the system from Automatic to Manual mode at any time (ev43). System switches to Manual mode from Automatic mode by the following events:

- A "Manual" command from the operator (ev43).
- An abnormal situation such as, failed tap up/tap down (ev30).

In Manual mode the system is waiting for "Tap-up", "Tap-down", "Automatic", or "Stop" commands. When returning to Automatic mode the controller is reinitialized at "state 0" of

the Automatic mode specification. A specification for the Auto/Manual mode (SPEC2) can be achieved by inserting some transitions after the occurrence of ev31 and ev33 and also by adding a new state as the "Manual-operation" state. "Manual" command (ev43) takes the system from any state (*) to the Manual-operation state. Then ev41 takes this state back to the initial state. Fig. 6-b shows the DES model for control specification in the Auto/Manual mode.

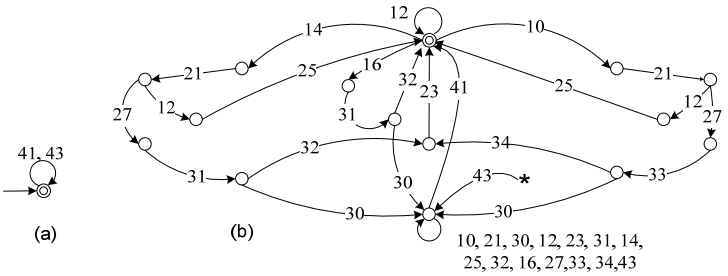


Fig. 6. a) An automaton for the operator, b) DES model for control specification in the Auto/Manual mode. The transition 43 from * represents the similar transition from all states to the "Manual operation"

4. 3 Design of the DES Supervisor

The plant and the specification DES models are implemented in the TCT software. Brief descriptions of the TCT procedures which are used in this chapter are given in the Appendix. The supervisory control and its reduced mode have been designed separately for the Automatic and Auto/Manual modes of operation.

I. Automatic Mode

The supervisor and the control data for the ULTC in the Automatic mode are calculated using TCT.

SUPER1 = Supcon(PLANT1,SPEC1) (78,171)
CONDAT1 = Condat(PLANT1,SUPER1) Controllable.
SIMSUP1 = Supreduce(PLANT1,SUPER1,CONDAT1) (22,92;slb=20)

SIMSUP1 is the reduced order supervisor with 22 states and 92 transitions.

II. Auto/Manual mode

The operator override is incorporated in the model by the control specification shown in Fig. 6-b. Using this specification and the new plant model which is synchronized by the "Operator" automata, the supervisory control is synthesized.

SUPER2 = Supcon(PLANT2,SPEC2) (198,831)
CONDAT2 = Condat(PLANT2,SUPER2) Controllable.
SIMSUP2 = Supreduce(PLANT2,SUPER2,CONDAT2) (12,54;slb=11)
SIMCD2 = Condat(PLANT2,SIMSUP2) Controllable.
MPS = Sync(PLANT2,SIMSUP2) (198,831) Blocked_events = None
true = Isomorph(MPS,SUPER2;identity)

As can be seen, the supervisor state-transition size has been reduced significantly from (198, 831) to (12, 54). The reduced order supervisory control in Auto/Manual mode (SIMSUP2) is shown in Fig. 7.

4.4 Verification of the Results

It is guaranteed by the theorems and procedures of supervisory control (Wonham, 2009) that are employed in this chapter, that the supervisor is non-blocking and meets the control specification in an 'optimal', that is, minimally restrictive fashion. It is shown in section 4.3 that such a controller which forces the specifications of ULTC to be met, exists and is constructible. In this section, the proposed supervisor is verified by inspecting its behavior when an increase in the voltage is reported.

Suppose that ev14 has occurred. Therefore a tap decrease is required until the voltmeter reports that the load voltage is restored, i.e. $|\Delta V| <$ ID (ev12). The event sequence applied by the proposed supervisory control is specified in the DES shown in Fig. 7. The trajectory includes states 0, 3, 7, 10, 11, 6 and finally 0, which is a marker state (the dotted path in Fig. 7). At state (0), events 21, 31, and 33 are disabled. This means that all components of the plant are disabled except for the voltmeter. If the voltmeter reads ev14, the closed loop system will be sent to state (3) by the supervisor. At state (3), the timer starts. When it "times out", i.e. reaches its maximum (ev27) the system goes to state (10). At state (10), the tap-up command (ev33) is disabled by the supervisor, and thus the transformer can only receive a tap-down command, i.e. (ev31). The ev31 takes the system to state (11). At state (11), if the tap-down is successful (ev32), the supervisor sends the plant to state (6), where first the timer will be reset (ev23), and then the system goes to state (0). State (0) is a marker state where the voltmeter is activated to read new voltages. If the voltmeter reads ev12 i.e., the voltage is restored, then the system stays in this state, unless the voltmeter reads ev10 or ev14. If ev10 occurs, the trajectory would go through states 2, 6, 9, 8, 6, and 0.
Similarly, one can follow the supervisor actions on this trajectory.

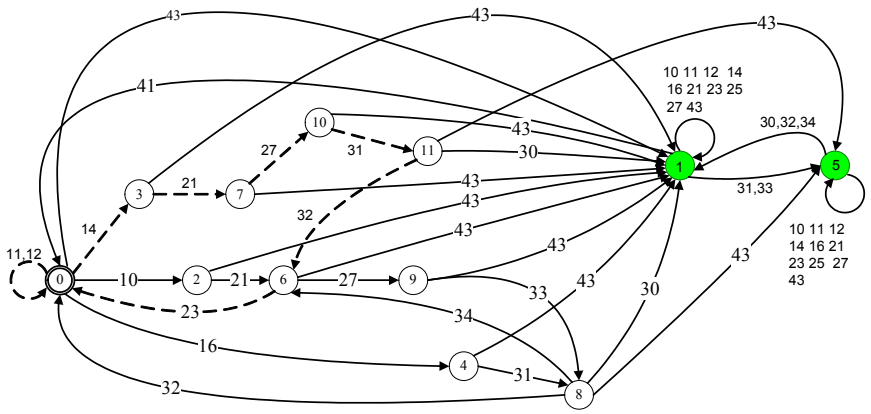


Fig. 7. The reduced order supervisor in Auto/Manual mode.

5. Hierarchical Solution

High level management executes a "Stop" command only after the occurrence of abnormal behavior in the plant, such as a specific number of tap-up/down failures, to shut down the regulation mechanism of the tap-changer. As described in section 2.5, the following steps are taken to synthesize a hierarchical supervisory structure.

1) A supervisor has been synthesized for the Automatic mode of the ULTC (SUPER1) and is considered as the low level plant.

2) Using vocalization, an abstract model for the supervisor in the Automatic mode (SUPER1) is developed, with the objective of letting a high-level manager execute a system Shutdown (ev61 in Fig. 8-a). The shutdown specification (SP_STOP) will require that both tap-up (ev31) and tap-down (ev33) commands along with the Timer (ev21) specification be disabled (Fig. 8-b). A supervisory control is synthesized again after adding the DES models for the manager and the shut-down logic to the plant (SUPER3).

```
SUPER3 = Supcon(PLANT3,SPEC3)  (100,228)
CONDAT3 = Condat(PLANT3,SUPER3)  Controllable.
SIMSUP3 = Supreduce(PLANT3,SUPER3,CONDAT3)  (29,123;slb=28)
```

Significant events corresponding to tap-up/down failure (ev30) and the shutdown (ev61) are vocalized.

```
MINSUP3 = Minstate(SUPER3)  (82,201)
VMSUP3 = Vocalize(MINSUP3,[[*,61,61],[*,30,30])   (118,252)
RVSUP = Recode(VMSUP3)  (118,252)
RVSUP_H = Higen(RVSUP)  (3,3)
```

Reasonably, a small abstraction model (Fig. 9-a) of the low-level controlled behavior is achieved (3 states vs. 29 states).

3) The specification shown in Fig. 9-b, is used to shut the system down after 3 occurrences of tap-up/down failure (ev300). Event labels 300 and 611 are new labels for vocalized events in the high-level.

4) The high level supervisor has been synthesized after finding a hierarchical and output consistent version of the high-level plant. The reduced order version of the high-level supervisor is shown in Fig. 9-c.

```
OC_P = Outconsis(RVSUP)  (119,252)
HC_P = Hiconsis(RVSUP)  (123,268)
false = Isomorph(HC_P,OC_P)
X = Hiconsis(OC_P)  (123,268)
true =Isomorph(HC_P,X;[[101,102],[102,103],[103,104],[104,105],[105,106],[106,107],
[107,108],[108,109],[109,110],[110,112],[111,113],[112,114],[113,115],[114,116],[115,117],
[116,118], [117,101],[118,111]])
SUPER_H = Supcon(PLANT_H,SPEC_H)  (103,330)
CONDAT_H = Condat(PLANT_H,SUPER_H)  Controllable.
SIMSUP_H = Supreduce(PLANT_H,SUPER_H,CONDAT_H)  (4,96;slb=4)
```

As shown in Fig. 9-c, the top manager can easily control the plant using a simple automaton which generates the required performance for the closed loop system.

Devices such as timers, transformers, etc. in the field-level may be provided by different vendors, and hence may have different specifications, i.e. control logic. Obviously, the hierarchical structure for the supervisory control is the appropriate solution in such cases. The DES models of the plant and the control logic can be achieved using the given technical

specifications from the vendors. While technical specifications can differ from one vendor to another, the differences can easily be accounted for in the DES models.

The hierarchical control structure can also be employed to synthesize coordination control of ULTC transformer and some FACTS devices.

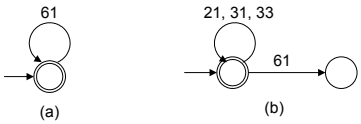


Fig. 8. DES models a) Manager, b) System Shut-down specification

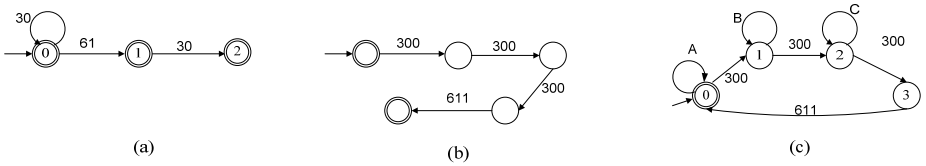


Fig. 9. DES models in the high level a) an abstract model of the low level plant, b) the control logic, c) The reduced order of the high level supervisor control for ULTC, where A, B, and C are lists of some events.

6. TDES supervisory control for ULTC

In this section the timed DES approach is employed to solve the supervisory control problem of the ULTC. First the plant and control logic are modeled as TDES, and then the supervisory control is designed in the different modes of operation.

6.1 TDES representation of the Plant

As discussed in Section 2, the system components are modeled by the corresponding ATGs for their untimed behavior first. When adding timing features, the time bounds (lower and upper) for the events of the system are defined. The plant consists of two main components:

Voltmeter: The voltmeter reports events associated with the load voltage using these events:

- Initialize Voltmeter (ev11 , [0,inf])
- Report $|\Delta V|$ > ID and $\Delta V > 0$ (ev14 , [0,inf])
- Report $|\Delta V|$ < ID and $\Delta V < 0$ (ev10 , [0,inf])
- Report $|\Delta V|$ < ID – i.e. Voltage Recovered (ev12 , [0,inf])
- Report Voltage exceeds V_{max} (ev16 , [0,inf])

Tap-Changer: The transformer tap-changer controls the transformer ratio "manually" or "automatically" in order to keep the power supply voltage practically constant, independently of the load. If the tap increase (decrease) is successful, the system returns to a state and waits for another command. If the tap increase (decrease) operation fails, the controller changes to the Manual mode, and waits for another command. It is assumed here that the tap-changer has 5 steps. Events associated with the Tap-Changer are:

- Tap up command (ev33 [5, inf]),
- Tap up successful (ev34 [0, inf]),
- Tap up/down failed (ev30 [0, inf]),
- Tap down command with 5s delay (ev31 [5, inf]),
- Tap down command without delay (ev35 [0, inf]),
- Tap down successful (ev32 [0, inf]),

The ATGs for the voltmeter and tap-changer are shown in Fig. 10. In order to find the whole system's model, the composition (analogous to synchronous product in untimed DES) of the ATGs of the system is found first, and then the TTG of the plant is worked out by converting the ATG to TTG.

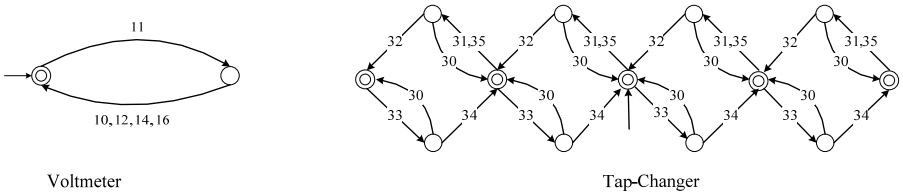


Fig. 10. ATGs for (a) Voltmeter (b) Tap-Changer.

6.2 TDES representation of Control Specifications

There are two modes of operation: "Automatic" and "Manual".

I. Automatic Mode

The tap-changer works in Automatic mode according to the following logic (control specifications):

If the voltage deviation $|\Delta V|$ > ID and ΔV is Negative (ev10) then the timer will start and when it times out, i.e. the time delay in occurrence of ev31 elapses, then a "tap increase " event (ev33) will occur and the timer will reset.

a) If the voltage deviation $|\Delta V|$ > ID and ΔV is Positive (ev14) then the timer will start and when it times out then a "tap decrease" (ev31) will occur and the timer will reset.

b) If the voltage returns to the dead-band (ev12), because of smooth system dynamics or a tap change or some other system events, then no tap change will occur.

c) If the voltage exceeds the value set for "Quick Lowering" (ev16), then the lowering tap command without delay (ev35) happens instantaneously.

Fig. 11 shows the TDES model of the control specification in the Automatic mode. It actually implements all the above logic in a single TDES. Notice that because the events tap-up/down command (31, 33, 35) are needed to preempt *tick* in some states of the above specifications, these events should be defined as "forcible" events (Section 2).

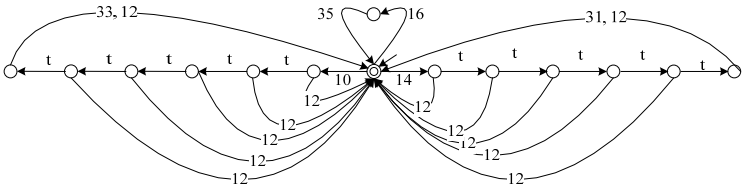


Fig. 11. TTG of the control specifications in Automatic Mode

II. Auto/Manual mode

In this mode of operation, the operator action is modeled to switch the modes and to override in abnormal situations. Events 41 and 43 are defined for operator actions:

- Enter "Automatic" Mode (ev41, [0,inf]),
- Enter "Manual" Mode (ev43, [0,inf])

The operator can force the system from Automatic to Manual mode at any time (ev43). System switches to Manual mode from Automatic mode by a "Manual" command from operator (ev43), or an abnormal situation such as, failed tap-up/tap down. In Manual mode the system is waiting for "Tap-up", "Tap-down" or "Automatic" commands. On returning to Automatic mode the controller is reinitialized at state 0 of the Automatic specification. A specification for the Auto/Manual mode (SPEC2) can be achieved by inserting suitable transitions after the occurrence of ev31 and ev33 and also by adding a new state as the "Manual-operation" state. The "Manual" command (ev43) takes the system from any state (*) to the Manual-operation state. Then ev41 takes this state back to the initial state. Fig. 12 shows the TDES model (TTG) for the control specification in Auto/Manual mode.

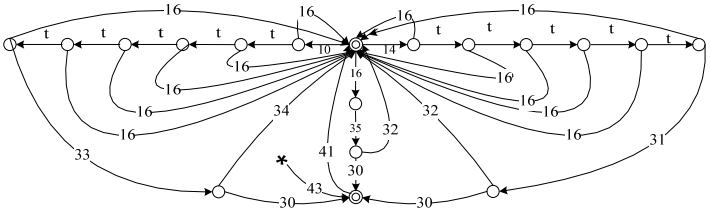


Fig. 12. TTG of the control specifications in Auto/Manual mode. The transition 43 from * represents similar transitions from all states to the "Manual operation" state

6.3 Design of TDES Supervisors

The plant and the specification TDES models are implemented in the TTCT software. The supervisory controller has been designed for the Automatic and Auto/Manual modes of operation separately. The supervisor and the control data for the ULTC in the Automatic mode are calculated using TTCT.

SUPER1 = Supcon(PLANT1,SPEC1) (198,300)
MINSUPER1 = Minstate(SUPER1) (52,79)

So a supervisory controller has been found for the Automatic mode of operation with 52 states and 79 transitions. In the Auto/Manual mode, the operator override is incorporated in the model by the control specification shown in Fig. 12. Using this specification and the

new plant model which is composed by the "Operator" ATG (which has one state and two transitions i.e. 41 and 43), the supervisory control is synthesized:

SUPER2 = Supcon(PLANT2,SPEC2) (231,543)
MINSUPER2 = Minstate(SUPER2) (56,130)
PMINSUP = Project(MINSUPER2, 'tick') (26,53)

As can be seen, the supervisor state-transition size is (56,130) after applying the "Minstate" operation. By projecting out *tick* from the supervisor, its transition structure can be displayed as the *timed activity transition graph* (TATG). While the TATG suppresses *tick*, it does incorporate the constraints on ordering of activities induced by time bounds. The TATG of the supervisor for Auto/Manual mode is shown in Fig. 13.

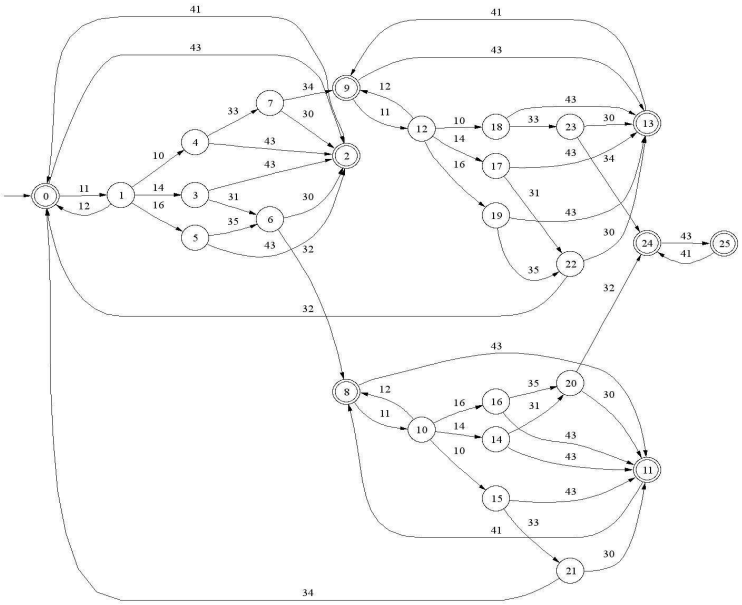


Fig. 13. TATG of the supervisory controller for Auto/Manual mode of operation

7. PLC Implementation of the Auto/Manual ULTC Supervisor

Though supervisory control theory has for over a decade received substantial attention in academia, industrial applications are scarce. Typically, finite-state automata describe the plant, specification and supervisor, and the step to a physical implementation is not necessarily straightforward. In the special case of industrial systems, where PLC-control is of great importance, the gap between the event-based asynchronous automata world and the synchronous signal based PLC-world has to be bridged (Fabian & Hellgren, 1998). The supervisor implementation is a matter of making the PLC behave as an automaton. However, there are a number of problems associated with the implementation in practice, and at the time of writing few guidelines for this can be found. Some generic problems are reported in (Fabian & Hellgren, 1998; Noorbakhsh & Afzalian, 2007a&b; Afzalian & Noorbakhsh 2008; Noorbakhsh, 2008).

One of the most important problems in the PLC-Implementation of a DES supervisory control system concern with the size of the automaton describing the behavior of the closed-loop system. Here, implementation of the untimed ULTC supervisor SUPER2 has been considered. Because of the large size of SUPER2, we have to use its reduced-order version (Fig. 7). There are some limitations in the algorithm applied to reduce the order of a supervisor in TCT software that need attention in developing the PLC ladder diagram. The reduced version of a supervisor generates all possible strings in the original model of the supervisor plus some "superfluous" strings; the latter cannot be eliminated without paying a possibly undesirable price in state size. In any case the new DES model may generate some strings that cannot be generated in the original model. Therefore, it is possible that some of these strings have no particular physical interpretation. For example, consider the following string in the reduced supervisor SIMSUP2 (Fig. 7): *s1: 11, 14, 21, 27, 31, 32, 27, 33, 34, 23*. The string *t:11, 14 ,21, 27, 31, 32* means that after an over voltage (ev14) and after a delay in the timer (ev27), the tap ratio of the transformer is decremented successfully (ev32). The event 27 after the string "t" in the string "s1" hasn't any meaning in the real system. Therefore from a physical point of view, the string "s1" can never occur in the real-world.
By inspecting all possible strings which can be generated in the DES model Fig. 7, we concluded that this problem can be solved by dividing each of the states 6 and 8 into two new states. The modified supervisor is shown in Fig. 14.
The limitation in the tap steps (Fig. 4 .c) is considered in the reduced supervisor (Fig. 14) in the state 9 and state 10 which can be reached by a string such as: *s2:11, 10, 21, 27, 33, 34, 23, 11, 10, 21, 27, 33, 34, 23, 11, 10, 21 ,27*. After this string the supervisor guides the system in Manual operation mode anyway.

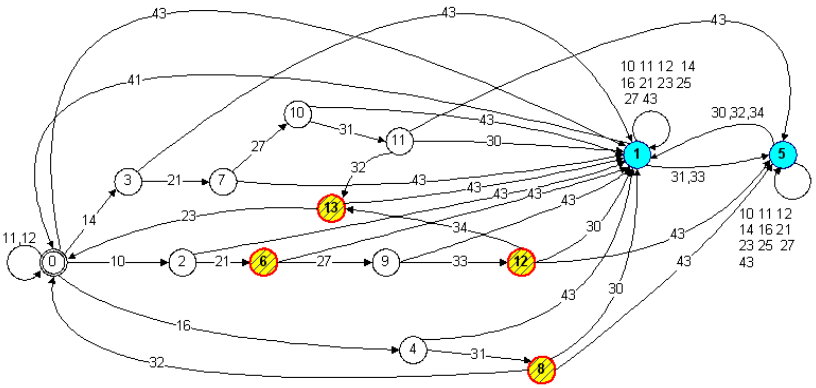


Fig. 14. The modified automaton of the reduced supervisor

The Manual mode should be performed by the operator. State 1 and state 5 in Fig. 14 correspond to Manual operation mode. Indeed when the plant operates in one of the states 1 or 5 the operator is responsible for controlling the required behavior of the plant. Therefore before finding the PLC ladder diagram for the supervisor, we need to extract the automatic part of the supervisor by deleting state 1 and state 5 along with the corresponding transitions in Fig. 14. The extracted automaton is shown in Fig. 15. Finally, the automaton in Fig. 15 is used to implement the ULTC supervisor on a PLC as a ladder diagram.

A straightforward way to implement an automaton on a ladder diagram is to represent each state and each event as an internal Boolean variable, and let the transitions be represented by a Boolean AND between the state variable and the event variable. When a transition occurs the next state is set and the previous state is reset (Fabian & Hellgren, 1998). Following this straightforward approach, a ladder diagram is developed to represent the ULTC supervisor shown in Fig. 16. The ladder code can be downloaded directly into the memory of a PLC. Now the PLC guarantees the required behavior (control specifications) of the plant in Automatic mode of operations. When a fault in tap increase or decrease occurs (ev30), or the operator forces the system from Automatic to Manual mode at any time (ev43), the system moves to the Manual state and waits for the operator commands. Indeed in this situation PLC does nothing. If the system has been switched to Manual mode, then whenever the operator changes the operation mode of ULTC from Manual to Automatic (ev41), the PLC will be reinitialized at "state 0" (Fig. 15).

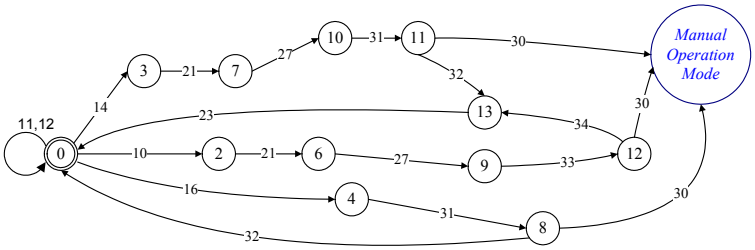


Fig. 15. The final DES model which is converted into a ladder diagram as the controller

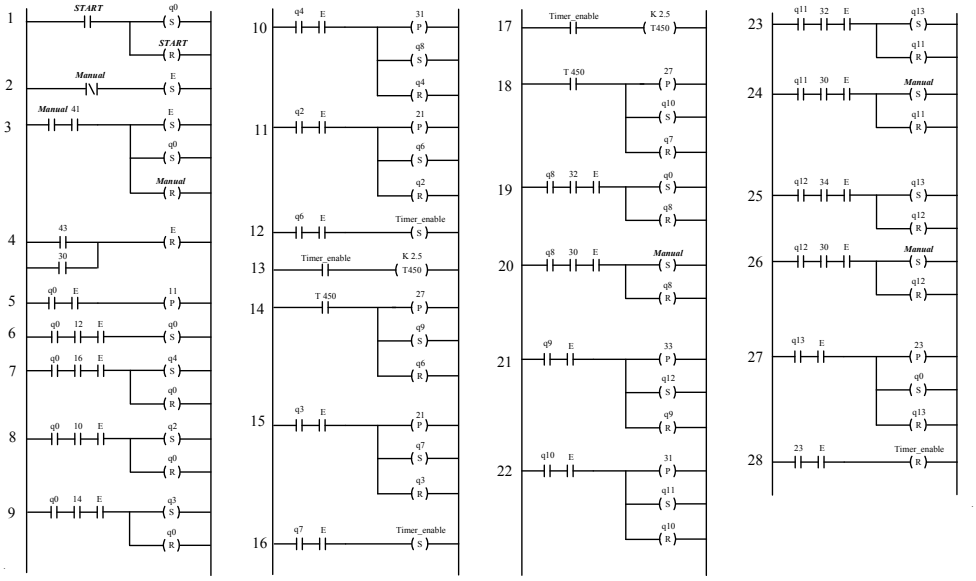


Fig. 16. Converted ladder diagram of the automaton shown in Fig. 15.

8. Conclusions

In this chapter, different solutions based on supervisory control of DES have been proposed and implemented for a control problem in electrical power systems. The problem of voltage regulation by ULTC was first modeled in terms of plant components and control specification. Controllability of the specification was evaluated and, by use of the TCT software, supervisory controllers were designed in different modes of operation including a two-level hierarchical structure. It is guaranteed by the synthesis procedure that the designed supervisors are optimal and non-blocking. The state size of the supervisory controllers was reduced for easier implementation. In the hierarchical supervisory control structure, the abstracted plant model in the high level was controlled by another supervisor, or manager, to handle the ULTC in failure situations.
The synthesis study shows that hierarchical supervisory control structure can be applied as a solution to the control problem in electrical power substations. Designers of protective systems for electrical power systems can use the proposed solutions to design appropriate supervisory control systems and to verify their control logic for ULTC. The hierarchical control structure can also be employed to synthesize the coordination control of ULTC transformers and certain FACTS devices, where DES models are available.
The designed supervisory controllers can be implemented by programmable logic controllers (PLC) to be used in real world. Generalizing this design approach to an electrical grid where many ULTCs and other switches are integrated is considered for future research work. Using a step-by-step procedure, a ladder diagram was developed for implementation of the designed Auto/Manual untimed ULTC supervisor that can be directly downloaded into a PLC. The generated PLC codes can be used in the real-time control of electrical power systems.

Appendix

A quick review of the TCT commands used in this chapter:

DES3= **supcon** (DES1, DES2)
for a controlled generator DES1, forms a trim recognizer for the supremal controllable sublanguage of the marked ("legal") language generated by DES2 to create DES3. This structure provides a proper supervisor for DES1.
DAT3= **condat** (DES1, DES2) returns control data DAT3 for the supervisor DES2 of the controlled system DES1. If DES2 represents a controllable language (with respect to DES1), as when DES2 has been previously computed with supcon, then *condat* will display the events that are to be disabled at each state of DES2. In general condat can be used to test whether a given language DES2 is controllable: just check that the disabled events tabled by *condat* are themselves controllable (have odd-numbered labels).
DES3= **supreduce** (DES1, DES2, DAT2) is a reduced supervisor for plant DES1 which is control-equivalent to DES2, where DES2 and control data DAT2 were previously computed using Supcon and Condat. Also returned is an estimated lower bound slb for the state size of a strictly state-minimal reduced supervisor. DES3 is strictly minimal if its reported state size happens to equal the *slb*.

DES2= **minstate**(DES1) reduces DES1 to a minimal state transition structure DES2 that generates the same closed and marked languages, and the same string mapping induced by vocalization (if any). DES2 is reachable but not necessarily coreachable.
DES2= **project** (DES1, NULL/IMAGE EVENTS) is a generator of the projected closed and marked languages of DES1, under the natural projection specified by the listed Null or Image events.
DES2=**vocalize** (DES1,[STATE-OUTPUT PAIRS]) has the same closed and marked behaviors as DES1, but with state outputs corresponding to selected state/event input pairs.
DES2= **outconsis** (DES1) has the same closed and marked behaviors as DES1, but is output-consistent in the sense that nonzero state outputs are unambiguously controllable or uncontrollable. A vocal state with output V in the range 10...99 may be split into siblings with outputs V1 or V0 in the range 100…991.
DES2= **hiconsis** (DES1) has the same closed and marked behaviors as DES1 but is hierarchically consistent in the sense that high-level controllable events may be disabled without side effects. This may require additional vocalization together with change in the control status of existing state outputs. **hiconsis** incorporates and extends **outconsis**.
True/False= **isomorph** (DES1, DES2) tests whether DES1 and DES2 are identical up to renumbering of states; if so, their state correspondence is displayed.
DES2= **higen** (DES1) is defined over the state-output alphabet of (vocalized) DES1, and represents the closed and marked state-output (or 'high-level') behaviors of DES1.

9. References

Afzalian, A.; Saadatpoor, A. & Wonham, W.M. (2006). Discrete-event system modeling and supervisory control for under-load tap-changing transformers, *Proceedings of 2006 IEEE International conference on Control Applications (CCA'06)*, pp.1867-1872, 2006, Munich, Germany.

Afzalian, A. & Wonham, W. M. (2006). Discrete-event system supervisory controller design for an electrical power transmission Network, *14th Iranian Conference on Electrical Engineering (ICEE'06*), 2006, Tehran, Iran.

Afzalian, A. & Noorbakhsh, M. (2008). PLC implementation of decentralized supervisory control for dynamic flow controller. *2008 IEEE International Conference on Control Applications (CCA'08)*, pp. 522-527, September 3-5, 2008, San Antonio, Texas (USA).

Brandin, B.A. & Wonham, W.M. (1994). Supervisory control of timed discrete-event systems, *IEEE Transactions on Automatic Control*, Vol. 39, No. 2, pp. 329-342.

Dietrich, P.; Malik, R.; Wonham, W.M. & Brandin, B.A. (2001). Implementation considerations in supervisory control. In B. Caillaud, P. Darondeau, L. Lavagno, X. Xie, editors, *Synthesis and Control of Discrete Event Systems*. Kluwer Academic Publishers. 2001, pp. 185-201.

Fink, L.H. (1999). Discrete events in power systems, *Discrete Event Dynamic Systems,* Vol. 9, No. 4, pp. 319-330.

Fabian, M. & Hellgren, A. (1998). PLC-based implementation of supervisory control for discrete event systems, *Proceedings of the 37th IEEE conference on Decision & Control,* pp. 3305-3310, 1998, Tampa, Florida, USA.

GE Consumer Industrial, M. (2005). DTR - Digital Tap Changer Controller Instruction Manual GEK-106305A: 7-12.

Hellgren, A.; Lennartson, B. & Fabian, A. (2002). Modeling and PLC-based implementation of modular supervisory control, *Proceedings of the 6th International Workshop on Discrete Event System (WODES'02),* pp. 371-376.

Hiskens, I.A. & Sokolowski, P.J. (2001). Systematic modeling and symbolically assisted simulation of power systems, *IEEE Transactions on Power Systems*, Vol. 16, No. 2, pp. 229-234.

Kim, G.W. & Lee, K.Y. (2005). Coordination control of ULTC transformer and STATCOM based on an artificial neural network, *IEEE Transactions on Power Systems*, Vol. 20, No. 2, pp. 580 - 586.

Kundur, P. (1994). *Power System Stability and Control*, McGraw-Hill.

Leal, A..; Cruz, D. & Hounsell, M. (2009). Supervisory control implementation into programmable logic controllers, *Proceedings of the 14th IEEE international conference on Emerging technologies & factory automation,* Palma de Mallorca, Spain, pp. 899-905.

Leduc, R.J. & Wonham W.M. (1995). PLC implementation of a DES supervisor for a manufacturing test bed. *Proceeding of Thirty-Third Annual Allerton Conference on Communication, Control and Computing*, pp. 519-528, University of Illinois.

Leduc, R.J. (1996). *PLC Implementation of a DES Supervisor for a Manufacturing Test bed: an Implementation Perspective*. M.A.Sc Thesis, Dept. of Electl. & Cmptr. Engrg., Univ. of Toronto, January 1996.

Lee, M.S. & Lim, J.T. (2004). Restoration strategy for power distribution networks using optimal supervisory control, *IEE Proceedings Generation, Transmission and Distribution*, Vol. 151, No. 3, pp. 367-372.

Liu, J. & Darabi, H. (2002). Ladder logic implementation of Ramadge-Wonham supervisory controller, *Proceedings of the 6th International Workshop on Discrete Event System (WODES'02),* pp. 383-389.

Lin, S.Y.; Ho, Y.C. & Lin, C.H. (2004). An ordinal optimization theory-based algorithm for solving the optimal power flow problem with discrete control variables, *IEEE Transactions on Power Systems*, Vol. 19, No 1, pp. 276-286.

Manesis, S. & Akantziotis, K. (2005). Automated synthesis of ladder automation circuits based on state-diagrams. *Advances in Engineering Software*, 36, pp .225–233.

Music, G. & Matko, D. (2002). Discrete event control theory applied to PLC Programming, *Automatica*, Vol 43, 1−2, pp. 21−28.

Noorbakhsh, M. & Afzalian, A. (2007a). Implementation of supervisory control of DES using PLC. *15th Iranian Conf. on Electrical Engineering (ICEE'07)*, (in Farsi), Tehran, Iran.

Noorbakhsh, M. & Afzalian, A. (2007b). Design and PLC Based Implementation of Supervisory Controller for Under-load Tap-Changer. *Proc. of the 2007 IEEE Int. Conf. on Control, Automation and Systems (ICCAS'07)*, pp. 901-906, Seoul, Korea.

Noorbakhsh, M. (2008). *DES Supervisory Control for Coordination of Under-Load Tap-Changing Transformer (ULTC) and a Static VAR Compensator (SVC)*. M.A.Sc Thesis, Dept. of Elect. & Cmptr. Eng., Shahid Abbaspour University of Technology, (in Farsi), Tehran, 2008.

Noorbakhsh, M. & Afzalian, A. (2009). Modeling and synthesis of DES supervisory control for coordinating ULTC and SVC. *2009 American Control Conf. (ACC' 09)*, pp. 4759-4764, Saint Louis, Missouri USA.

Ohtsuki, H.; Yokoyama, A. & Sekine, Y. (1991). Reverse action of on-load tap changer in association with voltage collapse, *IEEE Transactions on Power Systems*, Vol. 6, No. 1, pp. 300-306.

Otomega, B.; Sermanson, V. & Cutsem, T.V. (2003). Reverse-logic control of load tap changers in emergency voltage conditions, *IEEE Power Tech Conference Proceedings*, Vol. 1, Bologna.

Prosser, J.; Selinsky, J.; Kwatny, H. & Kam, M. (1995). Supervisory control of electric power transmission networks, *IEEE Transactions on Power Systems*, Vol. 10, No. 2, pp. 1104-1110.

Queiroz, M. & Cury, J. (2002). Synthesis and implementation of local modular supervisory control for a manufacturing cell, Proceedings of the *6th International Workshop on Discrete Event System (WODES'02)*, pp. 377-382.

Ramadge, P. J. G. & Wonham, W. M. (1987). Supervisory control of a class of discrete event processes, *SIAM Journal on Control and Optimization*, Vol. 25, No. 1, pp. 206 - 230.

Ramadge, P.J.G. & Wonham, W.M. (1989). The control of discrete event systems, *Proceedings of the IEEE,* Vol. 77, No. 1, pp. 81-98.

Simon, H. A. (1962), The architecture of complexity, *Proceedings of the American Philosophical Society*, Vol. 106, No. 6, pp. 467-482.

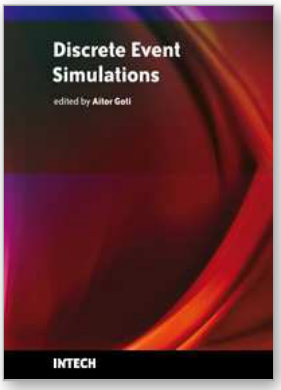
Su, R.; Wonham, W. M. (2004). Supervisor reduction for discrete-event systems, *Discrete Event Dynamic Systems,* Vol. 14, No. 1, pp. 31-53.

Thukaram, D.; Jenkins, L.; Khincha, H.P.; Yesuratnam, G. & Kumar, B.R. (2004). Monitoring the effects of on-load tap changing transformers on voltage stability, *International Conference on Power System Technology*, Vol. 1, pp. 419–424.

Vieira, A.D., Cury, J.E.R. & Queiroz, M. (2006). A model for PLC implementation of supervisory control of discrete event systems, *IEEE Conference on Emerging Technologies and Factory Automation, ETFA '06.*, pp. 225 – 232.

Wonham, W. M. (2009). *Supervisory Control of Discrete-Event Systems*, The University of Toronto, available from: http://www.control.utoronto.ca/DES.

Zhong, H. & Wonham, W.M. (1990). On the consistency of hierarchical supervision in discrete-event systems, *IEEE Transactions on Automatic Control*, Vol. 35, No. 10, pp. 1125-1134.



Discrete Event Simulations

Edited by Aitor Goti

ISBN 978-953-307-115-2
Hard cover, 330 pages
Publisher Sciyo
Published online 18, August, 2010
Published in print edition August, 2010

Considered by many authors as a technique for modelling stochastic, dynamic and discretely evolving systems, this technique has gained widespread acceptance among the practitioners who want to represent and improve complex systems. Since DES is a technique applied in incredibly different areas, this book reflects many different points of view about DES, thus, all authors describe how it is understood and applied within their context of work, providing an extensive understanding of what DES is. It can be said that the name of the book itself reflects the plurality that these points of view represent. The book embraces a number of topics covering theory, methods and applications to a wide range of sectors and problem areas that have been categorised into five groups. As well as the previously explained variety of points of view concerning DES, there is one additional thing to remark about this book: its richness when talking about actual data or actual data based analysis. When most academic areas are lacking application cases, roughly the half part of the chapters included in this book deal with actual problems or at least are based on actual data. Thus, the editor firmly believes that this book will be interesting for both beginners and practitioners in the area of DES.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821