Systematic supervisory control solutions for under-load tap-changing transformers

A. Afzaliana,*, Ali Saadatpoorb, W.M. Wonhamb

aDepartment of Control Systems Engineering, Shahid Abbaspour University of Technology, P.O. Box 16765-1719, Tehran, Iran
bDepartment of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ont., Canada, M5S 3G4

Received 2 June 2006; accepted 15 November 2007
Available online 10 January 2008

Abstract

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 behavior are widely used in transmission systems to take care of instantaneous variations in the load conditions in substations. In this paper, 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.

r 2007 Elsevier Ltd. All rights reserved.

Keywords: Supervisory control; Discrete-event systems; Timed discrete-event systems; Hierarchical control structure; Under-load tap-changing transformers

1. Introduction

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 and communication protocols, and database management systems. The behavior of electrical power systems can be characterized by interactions between continuous dynamics and discrete events.

In the last two decades, DES 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 and 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 paper, retains the concept of maximally permissive supervision introduced in Brandin and Wonham (1994), allows the timed modeling of DES, admits subsystem composition, and admits forcing and disablement as means of control. Different synthesize methods have been developed and implemented as the software TCT (for untimed models) and TTCT (for timed models) (Wonham, 2006) 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 other software tools available for simulation and analysis of DES (Basile, Carbone, & Chiacchio, 2007).

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 DES can be designed to be hierarchically structured. Implementation of this approach to a control problem in electrical power systems is also discussed in the present paper.

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 (ULTC) transformers 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 disrupted 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).

DES theory has been applied to problems in electrical power systems (Afzalian & Wonham, 2006; Lee & Lim, 2004, Lin, Ho, & Lin, 2004; Prosser, Selinsky, Kwatny, & Kam, 1995). These applications include: supervisory control, modeling and analysis, and monitoring and diagnosis of power systems. A hierarchical DES supervisory control is synthesized in Yasar and Ray (2007) to coordinate the operation of twin engine propulsion system. The synthesis of a DES-based supervisory control for ULTC was introduced in Afzalian, Saadatpoor, and Wonham (2006), where the ULTC along with different specifications (control logics) were modeled as automata. The automatic voltage control of a tap-changer transformer can be regarded as a DES. 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).

After a brief review on DES supervisory approaches, this paper starts with the modeling of ULTC as an automaton. Control specifications in each mode of operation are also modeled as some finite automata. Then supervisory controllers are designed for the ULTC in Automatic and Auto/Manual modes of operation, as the first solution. 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 shutdown 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.

The contributions of the paper are summarized as follows:

1. DES modeling of a ULTC transformer and its control specification.
2. Evaluation of the required properties for the supervisory control system, i.e. controllability, non-blocking, and non-conflicting.
3. Systematic approaches are given to synthesize supervisory control solutions in the monolithic, modular and hierarchical structure, separately.

Section 2 reviews briefly the untimed and timed DES supervisory control as well as the hierarchical structure. The ULTC transformer is discussed in Section 3. The DES modeling of the plant and control logic, and the design and implementation of the supervisory control using untimed DES are discussed in Section 4. The two-level hierarchical DES supervisory control for the ULTC is given in Section 5. And finally, Section 6 discusses the TDES version of the solution.

2. Supervisory control of DES

The supervisory control problem for a DES 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, 2006).
2.1. Discrete-event models

A DES model is specified by: the set of states (including an initial state, and marker states which can be desired states in some applications), the set of events, and the state transition function of the system. Formally, a DES is represented by an automaton $G = (Q, \Sigma, \delta, q_0, Q_m)$ 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; $\Sigma$ is a finite set of events ($\sigma$) which is referred to as an alphabet, and finally $\delta$ is a transition mapping $\delta : Q \times \Sigma \rightarrow Q : \delta(q, \sigma) = q'$ which gives the next state $q'$ after an event $\sigma$ occurs. $G$ 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 (sequence of events). In other words it generates a language $L(G)$, consisting of strings of events which are physically possible in the plant.

A prefix of a string $s$ is an initial subsequence of $s$, i.e. if $r$ and $s$ are strings in $\Sigma^*$, $u$ 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, $\Sigma^*$ 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 $\bar{A}$ is defined which contains all the prefixes of each element of $A$. If $A = \bar{A}$, then the set $A$ is prefix-closed. If $A$ is not prefix-closed, then $A \subset \bar{A}$. The language $L(G)$ is the set of all event sequences which are physically possible in the plant. $L(G) = \{s | s \in \Sigma^*, \delta(q_0, s) \}$ is defined.

Clearly, $L(G)$ is a subset of $\Sigma^*$, 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 language which is denoted by $L_m(G)$ consists of all strings which reach the marker states. $L_m(G)$ is a subset of $L(G)$ and can be formally given as: $L_m(G) = \{s \in L(G) | \delta(q_0, s) \in Q_m\}$.

A DES is said to be non-blocking if $L_m(G) = L(G)$. This means that there will always exist a sequence of events which takes the plant from any (reachable) state to a 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 events set $G_3$ is the union of events in $G_1$ 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 specific language specification. It is often possible to meet these specifications in a minimally restrictive way, called optimal supervision in the DES literature.

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. $\Sigma_c$ is the set of controllable events which 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 ($\gamma$). Uncontrollable events ($\Sigma_u$) 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 set of sets, is defined as: $I = \{\gamma \in \text{Pwr}(\Sigma) | \gamma \supseteq \Sigma_u\}$. A supervisory control for the plant $G$ is any function $V : L(G) \rightarrow I$. 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 must be disabled 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:

(i) $\varepsilon \in L(V/G)$;
(ii) if $s \in L(V/G)$, $\sigma \in V(s)$, and $sa \in L(G)$, then $sa \in L(V/G)$;
(iii) no other strings belong to $L(V/G)$.

![Fig. 1. Block diagram of a supervisory control system.](image-url)
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 decided by the supervisor to be 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 \( L_m(V/G) = L(V/G) \). A language \( K \) representing some specification of a plant \( G \) is said to be controllable (with respect to \( G \)) if its prefix-closure \( \bar{K} \) does not change under the occurrence of uncontrollable events in \( G \). In other words, \( K \) is controllable if and only if \( \bar{K}\Sigma_u \cap L(G) \subseteq \bar{K} \), where \( \bar{K}\Sigma_u = \{s\sigma | s \in \bar{K}, \sigma \in \Sigma_u \} \). Therefore, the controllability condition on specification \( K \) only constrains \( \bar{K} \cap L(G) \). Based on this definition, to test the controllability of \( K \), one only needs to test its closure \( \bar{K} \).

The existence of an optimal (marking) non-blocking supervisory controller is proved in Wonham (2006). Let \( K \subseteq L_m(G) \), \( K \neq \emptyset \), and \( K = \bar{K} \). Then there exists a supervisory controller \( V \) such that \( L(V/G) = K \) if and only if \( K \) is controllable. The supervisory control of a DES 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 of 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. 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 and Wonham (2004) that finding a supervisor of minimal size is a NP-hard problem. Usually, a supervisor is looked for which is smaller than supremal supervisor (S) that does the job without satisfying any required behavior of the system. The TCT procedure, \textit{supreduce} (Plant, Supervisor, \textit{condat}()) procedure calculates a small equivalent implementation of the supervisor (S) such that the following conditions are satisfied:

\[
L(G) \cap L(S) = L(S) \quad \text{and} \quad L_m(G) \cap L_m(S) = L_m(S).
\]

The relation between languages generated by specification (\( K \)), plant (\( G \)), supremal supervisor (\( S \)), and the reduced supervisor (\( S_r \)) is given in Fig. 2.

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

1. Model the plant (components) as automata.
2. Model the specifications as DES and construct one DES, called \( \text{EDES} \), representing all the specifications together. This can be done by the “\( \text{meet} \)” operation in TCT.
3. Find the non-blocking supervisory controller using the “\( \text{supcon} \)” operation in TCT, i.e. \( \text{SUPER} = \text{supcon} (G, \text{EDES}) \).
4. There are some redundant constraints in \( \text{SUPER} \) which comes up with the controller with bigger size in number of states and/or number of transitions. To simplify the supervisor the command “\( \text{supreduce} \)” in TCT can be used. In this procedure some heuristics are employed to reduce the supervisor.

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

![Fig. 2. The reduced supervisor is significantly smaller in size than the supervisor, whereas it might generate a bigger language.](image-url)
2.3. Timed DESs

This section briefly reviews the TDES model proposed by Brandin and 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 the finite set of activities, $\Sigma_{act}$ is the finite set of events, a partial function $\delta_{act} : A \times \Sigma_{act} \rightarrow A$ is the activity transition function, $a_0 \in A$ is the initial activity, and $A_m \subset A$ is the set of marked activities. In order to construct a TDES model, timing information is introduced into $G_{act}$. Let $N$ denote a set of nonnegative integers. In $\Sigma_{act}$, each event $\sigma$ 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 $l_\sigma \leq u_\sigma$. Then the set of events is decomposed into two subsets $\Sigma_{spe} = \{\sigma \in \Sigma_{act} | u_\sigma \in N\}$ and $\Sigma_{rem} = \{\sigma \in \Sigma_{act} | u_\sigma = \infty\}$. 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 $T_\sigma$ is defined as

$$T_\sigma = \begin{cases} [0, u_\sigma] & \text{if } \sigma \in \Sigma_{spe}, \\ [0, l_\sigma] & \text{if } \sigma \in \Sigma_{rem}. \end{cases}$$

The TDES defined by Brandin and 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 $Q = A \times \prod_{\sigma \in \Sigma_{act}} [T_\sigma | \sigma \in \Sigma_{act}]$. 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 $q_0 = (a_0, \{t_{0,0} | \sigma \in \Sigma_{act}\})$, where

$$t_{0,0} = \begin{cases} u_\sigma & \text{if } \sigma \in \Sigma_{spe}, \\ l_\sigma & \text{if } \sigma \in \Sigma_{rem}. \end{cases}$$

The set $Q_m \subseteq Q$ is given by a subset of $A_m \times \prod_{\sigma \in \Sigma_{act}} [T_\sigma | \sigma \in \Sigma_{act}]$. The event set $\Sigma$ is defined as $\Sigma = \Sigma_{act} \cup \{\text{tick}\}$, where the additional event tick represents the passage of one time unit. The state transition function $\delta : Q \times \Sigma \rightarrow Q$ 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:

- $\sigma = \text{tick}$ and $\forall \tau \in \Sigma_{spe}: \delta_{act}(a, \tau)$; $t_\tau > 0$;
- $\sigma \in \Sigma_{spe}$ and $\delta_{act}(a, \tau) \neq 0$ and $0 \leq u_\tau - l_\sigma$;
- $\sigma \in \Sigma_{rem}$ and $\delta_{act}(a, \tau) = 0$ and $t_\sigma = 0$.

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

- if $\sigma = \text{tick}$ then $a' = a$ and for all $\tau \in \Sigma_{act}$,

$$t'_\tau := \begin{cases} t_\tau - 1 & \text{if } \delta_{act}(a, \tau) \neq 0, \\ t_\tau & \text{otherwise}, \end{cases}$$

- if $\sigma \in \Sigma_{act}$ then $a' = \delta_{act}(a, \sigma)$, $t'_\sigma = t_{\sigma,0}$, and for $\tau \in \Sigma_{act}$ if $\tau \neq \sigma$ then

$$t'_\tau := \begin{cases} t_\tau & \text{if } \delta_{act}(a', \tau) \neq 0, \\ t_{\tau,0} & \text{otherwise}. \end{cases}$$

Let $\Sigma^*$ be the set of all finite strings of elements in $\Sigma$, including the empty string $\varepsilon$. The function $\delta$ is extended to $\delta : Q \times \Sigma^* \rightarrow Q$ 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 $L_m(G) = L(G)$. As in untimed supervisory control, the set $\Sigma_{act}$ is partitioned into two subsets $\Sigma_u$ and $\Sigma_c$ of controllable and uncontrollable events. An event $\delta$ that can preempt the event tick is called a forcible event. The set of forcible events is denoted by $\Sigma_{for}$. A forcible event can be either controllable or uncontrollable. By forcing an enabled event in $\Sigma_{for}$ to occur, the event tick can be disabled. In this framework a supervisor repeatedly decides to disable or enable each event in $\Sigma_u \cup \{\text{tick}\}$.

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 $Elig_G(s) := \{\sigma \in \Sigma | s \in L(G)\}$. In order to define the notion of controllability a language $K \subseteq L(G)$ is considered to define: $Elig_K(s) := \{\sigma \in \Sigma | s \in K\}$. $K$ is controllable with respect to $G$ if, for all $s \in K$,

$$Elig_K(s) \supseteq \begin{cases} Elig_G(s) \cap (\Sigma_u \cup \{\text{tick}\}), \\ Elig_G(s) \cap \Sigma_{for} = \emptyset, \\ Elig_G(s) \cap \Sigma_u, \\ Elig_G(s) \cap \Sigma_{for} \neq \emptyset. \end{cases}$$
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 (2006) 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 conclude the properties of the whole. Usually, 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 coverage 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 have some common properties that are independent of their specific content (Zhong & Wonham, 1990).

The DES supervisory control can be designed to be hierarchically structured. Fig. 3 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. 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 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 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 center of a complex system, such as electric power distributions 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 for low to 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 facilitate an effective high-level control.
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. $\Sigma^*$ is the set of finite strings $s$, for which the extended transition map $\delta : Q \times \Sigma^* \rightarrow Q$ 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 $supC(E_l \cap L(G_l))$. The following notation is used for this supervisor: $M^1 := supC(M)$.

The refined information flow through the “report” channel which consists of significant event $s$ can be represented by the language $T$. Thus the “report” can be modeled as a causal map $\theta : L_l \rightarrow T^*$ with following properties: $\theta(\varepsilon) = \varepsilon$, $\theta(s\sigma) = either \theta(s) or \theta(s)\tau,$ for some $\tau \in T$, $s \in L_l$, and $\sigma \in \Sigma$.

An abstract model for the plant in the high level can be given as an automaton $G_h$ that generates a language $L_h := \theta(L_l) \subseteq T^*$. The high-level controller $C_h$ that observes only the state of $G_h$ must be able to make meaningful control decision. Following steps and related TCT procedures were proposed to formulate the suitable control structure (Wonham, 2006):

1. Adopting the usual supervisory structure having the same type as in $G_l$ (Supcon(...)).
2. Refining the state structure of $G_l$ (Recode(.)).
3. Extending the high-level event alphabet $T$ (Vocalize(...)).
4. Finding the corresponding structure for $G_h$ (Higen($G_l$)).
5. Partitioning this extension into controllable and uncontrollable subsets to provide manager the ability of setting up specifications in terms of controllable events. This was 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($G_l$)).
6. Designing 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 is optimistic is respect to the effectiveness of the command-control process. But if $E_h$ is not larger than operator realization, i.e. $\theta((\theta^{-1}(E_l))^*) = E_h$, 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 in DES model of the lower plant, in other words, enhancing 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 computing the Hiconsis($G_l$) procedure.

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 or 115 to 44, 27.6, or 13.8 kV (Kundur, 1994).

Whereas many articles considered ULTC as a nonlinear element in the power system model 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 (Kundur, 1994; Ohtsuki, Yokoyama, & Sekine, 1991; Otomega, Sermanson, & Cutsem, 2003) as well as in manufacturers’ catalogues (e.g. GE Consumer Industrial, 2005) in different detail. When the voltage is not “normal” (outside a desired limit), the controller changes tap ratio after a time delay to recover the voltage, i.e. bring it back into its dead-band. The delay time 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. 4 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 is 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 modeling of the plant

As shown in Fig. 4, 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, \( \Delta V = V_o - V_i \), is the voltage deviation and ID the insensitivity degree, which is defined as the maximum admissible variation of the voltage before originating a command to change the tap.

Voltmeter reports following events associated with the load voltage (Fig. 5a):

- Voltmeter initialized (ev11)
- Report \( |\Delta V| > ID \) and \( \Delta V \) is negative (ev10)
- Report \( |\Delta V| < ID \) (voltage recovered) (ev12)
- Report \( |\Delta V| > ID \) and \( \Delta V \) 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. 5b):

- 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 five steps. Events associated with the TAP-CHANGER are (Fig. 5c):

- 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.

4.2. Control specifications

The control logic for a ULTC transformer is normally provided by the manufacturer and/or by the designer. A control logic which is given (GE Consumer Industrial, 2005) by GE company is employed in this paper. The control logic is modeled as some 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 (Kim & Lee, 2005; Thukaram, Jenkins, Khincha, Yesuratnam, & Kumar, 2004). 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

(a) If the voltage deviation $|\Delta V| > \Delta D$ and $\Delta V$ is negative (ev10) then the timer will start and when it “times out”, i.e. reaches its maximum (ev27) a “tap increase command” (ev33) will be made and the timer will be “reset” (ev23).
(b) If the voltage deviation $|\Delta V| > \Delta D$ and $\Delta V$ is positive (ev14) then the timer will start and when it “times out” i.e. reaches its maximum (ev27) 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 change 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 s and therefore the lowering tap command (ev31) happens instantaneously.

Fig. 6 shows the DES model of the control specification in the Automatic mode. It actually implements all above logics in a single DES. The automatic voltage controller of a tap-changer transformer can be regarded as a DES. 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).

(II) Incorporating operator override (Manual mode) In 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. 7a):

- 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 following events:

2. An abnormal situation such as, failed tap up/tap down.

Fig. 6. DES model of the control logic (specification) for ULTC in Automatic mode.

Fig. 7. (a) An automaton for the operator. (b) DES model for control specification in the Auto/Manual mode. The transition 43 from * represents similar transitions from all states to the "manual operation".
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. 7b shows the DES model for control specification in the Auto/Manual mode.

4.3. Design of the DES supervisor

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

(I) Automatic mode

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

\[
\begin{align*}
\text{SUPER1} &= \text{Supcon(PLANT1,SPEC1)} (78,171) \\
\text{CONDAT1} &= \text{Condat(PLANT1,SUPER1) Controllable.} \\
\text{SIMSUP1} &= \text{Supreduce(PLANT1,SUPER1,CONDAT1)} (22,92;\text{slb} = 20)
\end{align*}
\]

SIMSUP1 is the reduced order supervisor with 19 states and 60 transitions. The reduced order supervisory control and the control data are shown in Fig. 8.

![Diagram](image-url)

Fig. 8. The reduced order supervisor (a) and the control data (b) in the Automatic mode.

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. 7b shows the DES model for control specification in the Auto/Manual mode.

4.3. Design of the DES supervisor

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

(I) Automatic mode

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

\[
\begin{align*}
\text{SUPER1} &= \text{Supcon(PLANT1,SPEC1)} (78,171) \\
\text{CONDAT1} &= \text{Condat(PLANT1,SUPER1) Controllable.} \\
\text{SIMSUP1} &= \text{Supreduce(PLANT1,SUPER1,CONDAT1)} (22,92;\text{slb} = 20)
\end{align*}
\]

SIMSUP1 is the reduced order supervisor with 19 states and 60 transitions. The reduced order supervisory control and the control data are shown in Fig. 8.
Auto/Manual mode

The operator override is incorporated in the model by the control specification shown in Fig. 7b. Using this specification and the new plant model which is synchronized by the “Operator” automata, the supervisory control is synthesized.

\[
\begin{align*}
\text{SUPER2} & = \text{Supcon}(\text{PLANT2}, \text{SPEC2}) \ (198,831) \\
\text{CONDAT2} & = \text{Condat}(\text{PLANT2}, \text{SUPER2}) \ \text{Controllable}. \\
\text{SIMSUP2} & = \text{Supreduce}(\text{PLANT2}, \text{SUPER2}, \text{CONDAT2}) \ (12,54; \text{slb} = 11) \\
\text{SIMCD2} & = \text{Condat}(\text{PLANT2}, \text{SIMSUP2}) \ \text{Controllable}. \\
\text{MPS} & = \text{Sync}(\text{PLANT2}, \text{SIMSUP2}) \ (198,831) \ \text{Blocked\_events} = \text{None} \\
\text{true} & = \text{Isomorph}(\text{MPS}, \text{SUPER2}; \text{identity})
\end{align*}
\]

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 is shown in Fig. 9.

4.4. Verification of the results

It is guaranteed by the theorems and procedures of the supervisory control (Wonham, 2006) which are employed in this paper, that the supervisor is non-blocking and meet 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 in the automatic mode of operation by inspecting their behaviors when an increase in the voltage is reported.

Suppose that ev14 has happened. Therefore, a tap decrease is required until the voltmeter reports that the load voltage is recovered, i.e. |ΔV|<ID (ev12). The event sequence applied by the proposed supervisory control is highlighted in the DES shown in Fig. 8a. The trajectory includes states 0, 2, 5, 1, 8, and finally 6, which is a marker state. The events which are disabled by the supervisory control in each state are given in Fig. 8b.

At state (0), events 21, 31, and 33 are disabled (Fig. 8b). 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 (2) by the supervisor.

At state (2), the timer starts. When it “times out”, i.e. reaches its maximum (ev27) the system goes to state (5).

At state (5), 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 (1).
At state (1), if the tap-down is successful (ev32), the supervisor sends the plant to state (8), where the timer will be reset (ev23) first, and then the system goes to state (6).

State (6) is a marker state where the voltmeter is activated to read new voltages. If the voltmeter reads ev12, i.e. the voltage is recovered, then the system stays in this state, unless the voltmeter reads ev10 or ev14. If ev10 happens, the trajectory would go through states 8, 13, 7, 1, and 0 (the dotted line in Fig. 8a).

Similarly, one can follow the supervisor actions on this trajectory.

5. Hierarchical solution

High-level management executes a “Stop” command only after occurrence of abnormal behavior in the plant, such as a specific number of tap up/down failures, to shutdown the regulation mechanism of the tap-changer. As described in Section 2, 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. The DES model of SUPER1 is shown in Fig. 8a.
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. 10b). 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. A supervisory control system synthesized again after adding the DES models for the manager and the shut-down logic to the plant (SUPER3) whose reduced order version is shown in Fig. 11.

\[
\text{SUPER3} = \text{Supcon(PLANT3,SPEC3)} (100,228) \\
\text{CONDAT3} = \text{Condat(PLANT3,SUPER3)} \text{ Controllable.} \\
\text{SIMSUP3} = \text{Supreduce(PLANT3,SUPER3,CONDAT3)} (29,123;slb = 28)
\]

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

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

Reasonably, a small abstraction model (Fig. 12a) of the low-level controlled behavior is achieved (3 states vs. 29 states).

3. The specification shown in Fig. 12b, is used to shut the system down after three 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. 13.

\[
\text{OC_P} = \text{Outconsis(RVSUP)} (119,252) \\
\text{HC_P} = \text{Hiconsis(RVSUP)} (123,268) \\
\text{false} = \text{Isomorph(HC_P,OC_P)} \\
\text{true} = \text{Isomorph(HC_P,OC_P)} (123,268)
\]

Fig. 10. DES models: (a) manage, (b) system shutdown specification.
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 it is shown in Fig. 13, 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. These technical specifications can be different from one vendor to another, and the differences can simply be considered in the DES models.

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

6. TDES supervisory control for ULTC

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

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. For adding time 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 five 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 5 s 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. 14. 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.
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):
(a) If the voltage deviation $|\Delta V| > \Delta V$ is negative (ev10) then the timer will start and when it times out, i.e. the time delay in occurrence of ev31 elapses a "tap increase" event (ev33) will occur and the timer will reset.
(b) If the voltage deviation $|\Delta V| > \Delta V$ is positive (ev14) then the timer will start and when it times out a "tap decrease" (ev31) will occur and the timer will reset.
(c) 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.
(d) If the voltage exceeds the value set for "Quick Lowering" (ev16), then the lowering tap command without delay (ev35) happens instantaneously.

Fig. 15 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 above specifications, these events should be defined as "forcible" events (Section 2).
(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]).

Fig. 16. TTG of the control specifications in Auto/Manual mode. The transition 43 from * represents similar transitions from all states to the “manual operation” state.

Fig. 17. TATG of the supervisory controller for Auto/Manual mode of operation.
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 (Fig. 15). 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. 16 shows the TDES model (TTG) for the control specification in Auto/Manual mode.

### 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.

\[
\begin{align*}
\text{SUPER1} &= \text{Supcon}(\text{PLANT1}, \text{SPEC1}) \ (198,300) \\
\text{MINSUPER1} &= \text{Minstate}(\text{SUPER1}) \ (52,79)
\end{align*}
\]

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. 16. 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:

\[
\begin{align*}
\text{SUPER2} &= \text{Supcon}(\text{PLANT2}, \text{SPEC2}) \ (231,543) \\
\text{MINSUPER2} &= \text{Minstate}(\text{SUPER2}) \ (56,130) \\
\text{PMINSUP} &= \text{Project}(\text{MINSUPER2}, \text{‘tick’}) \ (26,53)
\end{align*}
\]

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

---

**Control data**

Control data are displayed as a list of supervisor states where disabling occurs, together with the events that must be disabled there. If TICK is disabled f signifies the events which are forced.

<table>
<thead>
<tr>
<th>State</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31 33 35 41</td>
</tr>
<tr>
<td>1</td>
<td>31 33 35 41 43</td>
</tr>
<tr>
<td>2</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>3</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>4</td>
<td>11 31 33 35 41 5</td>
</tr>
<tr>
<td>5</td>
<td>0 11 31 33 41 35f</td>
</tr>
<tr>
<td>6</td>
<td>11 31 33 35 41 7</td>
</tr>
<tr>
<td>7</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>8</td>
<td>11 41 43 9</td>
</tr>
<tr>
<td>9</td>
<td>11 31 33 35 41 43</td>
</tr>
<tr>
<td>10</td>
<td>11 31 33 35 41 11</td>
</tr>
<tr>
<td>11</td>
<td>11 31 33 35 41 13</td>
</tr>
<tr>
<td>12</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>13</td>
<td>11 31 33 35 41 15</td>
</tr>
<tr>
<td>14</td>
<td>31 33 35 41 43</td>
</tr>
<tr>
<td>15</td>
<td>11 31 33 35 43</td>
</tr>
<tr>
<td>16</td>
<td>11 31 33 35 41 17</td>
</tr>
<tr>
<td>17</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>18</td>
<td>11 31 33 35 41 19</td>
</tr>
<tr>
<td>19</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>20</td>
<td>0 11 31 33 41 35f</td>
</tr>
<tr>
<td>21</td>
<td>0 11 31 35 41 33f</td>
</tr>
<tr>
<td>22</td>
<td>0 11 33 35 41 31f 23</td>
</tr>
<tr>
<td>23</td>
<td>11 31 33 35 41 35</td>
</tr>
<tr>
<td>24</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>25</td>
<td>11 31 41 35</td>
</tr>
<tr>
<td>26</td>
<td>11 31 33 35 41 35f 27</td>
</tr>
<tr>
<td>27</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>28</td>
<td>11 31 33 35 41 29</td>
</tr>
<tr>
<td>29</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>30</td>
<td>31 33 35 41 31</td>
</tr>
<tr>
<td>31</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>32</td>
<td>31 33 35 41 33</td>
</tr>
<tr>
<td>33</td>
<td>11 31 33 35 41 33</td>
</tr>
<tr>
<td>34</td>
<td>31 33 35 41 43 35</td>
</tr>
<tr>
<td>35</td>
<td>11 31 33 35 43</td>
</tr>
<tr>
<td>36</td>
<td>11 31 33 35 41 37</td>
</tr>
<tr>
<td>37</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>38</td>
<td>11 31 33 35 41 39</td>
</tr>
<tr>
<td>39</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>40</td>
<td>0 11 31 33 41 35f 41</td>
</tr>
<tr>
<td>41</td>
<td>0 11 31 35 41 33f</td>
</tr>
<tr>
<td>42</td>
<td>0 11 33 35 41 31f 43</td>
</tr>
<tr>
<td>43</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>44</td>
<td>11 31 33 35 41 45</td>
</tr>
<tr>
<td>45</td>
<td>11 41 43</td>
</tr>
<tr>
<td>46</td>
<td>11 41 43 47</td>
</tr>
<tr>
<td>47</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>48</td>
<td>11 31 33 35 41 49</td>
</tr>
<tr>
<td>49</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>50</td>
<td>11 31 33 35 41 51</td>
</tr>
<tr>
<td>51</td>
<td>11 31 33 35 41</td>
</tr>
<tr>
<td>52</td>
<td>11 31 33 35 41 53</td>
</tr>
<tr>
<td>53</td>
<td>0 11 31 35 41 33f</td>
</tr>
<tr>
<td>54</td>
<td>0 11 33 35 41 31f 55</td>
</tr>
<tr>
<td>55</td>
<td>11 41 43</td>
</tr>
</tbody>
</table>

---

Fig. 18. Control data of the TDES supervisory controller for Auto/Manual mode.
the supervisor for Auto/Manual mode is shown in Fig. 17. The control data of the supervisor are shown in Fig. 18 as a list of supervisor states where disabling occurs, together with the events that must be disabled there.

7. Conclusions

In this paper, different solutions based on supervisory control of DES were proposed and implemented for a control problem in electrical power systems. The voltage regulation problem by ULTC was first modeled in terms of plant components and control specification. Controllability of the specification was evaluated and supervisory controllers were designed in different modes of operations and also in two-level hierarchical structure using the TCT software program. It is guaranteed by the synthesize 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 summarized plant model in the high level was controlled by another supervisor, or manager, to handle the ULTC in failure situations.

The synthesize study shows that hierarchical supervisory control structure can be applied as a solution to the control problem in electrical power substations. Protective system designers in 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 some 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.

Appendix

A quick review on some of the TCT commands which are used in this paper:

- 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.

References


