Control Engineering Practice 82 (2019) 97–107

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

An implementation method for the supervisory control of time-driven systems applied to high-voltage direct current transmission grids

Miguel Romero-Rodríguez [a,*], Romain Delpoux [b,a], Laurent Piétrac [b,a], Jing Dai [c,a], Abdelkrim Benchaib [a,d], Eric Niel [b,a]

[a] *SuperGrid Institute SAS, F-69611 Villeurbanne, France*
[b] *Université de Lyon, CNRS, INSA-Lyon, AMPERE, F-69621 Villeurbanne, France*
[c] *Group of Electrical Engineering — Paris (GeePs), UMR CNRS 8507, CentraleSupélec, Univ. Paris-Sud, Univ. Paris-Saclay, Sorbonne Universités, UPMC Univ. Paris 06, F-91192 Gif-sur-Yvette, France*
[d] *GE Grid Solutions, F-91300 Massy, France*

ARTICLE INFO

Keywords:
Discrete event systems
Supervisory control theory
Control implementation
HVDC transmission systems
EMTP-RV
Computer programming languages

ABSTRACT

In recent years, the growth of renewable energy production has encouraged the development of new technologies, such as High-Voltage Direct Current (HVDC) networks, that enhance the integration of such energy sources to power transmission grids. However, this type of technology introduces new challenges in the way power transmission systems are controlled and operated, as faster and more complex control strategies will be needed in a domain which nowadays relies heavily on human decisions. In this context, Discrete Event Systems (DES) modeling and Supervisory Control Theory (SCT) are powerful tools for the development of a supervisory control to be deployed in the grid. This paper presents an application of the SCT to HVDC grids and proposes an implementation method for the resulting supervisors. The proposed method is capable of integrating decentralized and discrete-event controllers that interact with the continuous-time physical system. The language chosen for the implementation is C code, as it can be easily incorporated in power system simulation software, such as EMTP-RV. The method is validated by the simulation of the start-up of a point-to-point link in the EMTP-RV software.

1. Introduction

The integration of renewable energy sources to the existing electrical grids is a key issue in the domain of energy transportation. The development of large High-Voltage Direct Current (HVDC) networks that bring the power from remote renewable sources to load centers will increase the complexity of power transmission systems, thus introducing new challenges in the way these types of systems are controlled and operated (van Hertem & Ghandhari, 2010; Zhang, Li, & Bhatt, 2010). For instance, in traditional power transmission systems based on widespread Alternating Current (AC) technology, large turbo generators are connected to the grid. In consequence, the inertia of their rotating masses liberates energy that provides resistance against frequency disturbances, allowing the different frequency control actions to be deployed in a timescale from 1 to 2 s to 15 to 30 min after the disturbance (Rebours, Kirschen, Trotignon, & Rossignol, 2007). On the contrary, the lower energy stocked in HVDC systems provides less resistance against voltage disturbances. In consequence, the transient generated by the disturbance will not be compensated in time, and thus the control should react faster (in the order of 100 ms). In addition, new converter topologies such as the Modular Multilevel Converters (MMCs) introduce additional degrees of freedom for control that increase the complexity of grid operation. For all these reasons, the need for an automated and coordinated supervisory control system during grid operation will increase over the years.

In this context, Discrete Event Systems (DES) modeling and the Supervisory Control Theory (SCT), first proposed in Ramadge and Wonham (1987), offer a formal framework for the synthesis of supervisors ensuring that the system under control respects a set of behavioral specifications, imposed by the designer, within its physical limitations. Moreover, the use of an SCT-based modal approach, such as the one presented in Faraut, Piétrac, and Niel (2009), would allow to manage the transition between the different operating modes of an HVDC system: start-up, fault protection, power ramp, shut-down, etc.

Despite the need to ensure that the interaction between the components of complex power transmission networks (highly reconfigurable and composed of many interconnected components) does not impact negatively the behavior of the whole system, this problem has not been treated in the literature. In consequence, the authors proposed a method for the synthesis of a decentralized supervisory control system

* Corresponding author.
E-mail address: miguel.romerorodriguez@supergrid-institute.com (M. Romero-Rodríguez).

https://doi.org/10.1016/j.conengprac.2018.10.002
Received 18 July 2017; Received in revised form 26 March 2018; Accepted 3 October 2018
Available online xxxx

for HVDC grids in Romero Rodríguez, Delpoux, Piétrac, Dai, Benchaib, and Niel (2017). In the current paper, the aspects regarding the practical implementation of the theoretical supervisors for the start-up of a point-to-point link obtained in Romero Rodríguez et al. (2017) are addressed. Because it is desired to implement the supervisory control in power system's specific simulation software such as EMTP-RV (Mahseredjian, Dennetière, Dubé, Khodabakhchian, & Gérin-Lajoie, 2007) or its real-time simulation counterpart HYPERSIM (Do, Soumagne, Sybille, Turmel, Giroux, Cloutier, & Poulin, 1999), the implementation method presented here is based on common user oriented languages, such as C code.

A number of papers have contributed to the implementation of the supervisors obtained with the SCT over the last decades, for the most part related to the control of manufacturing systems based on widespread Programmable Logic Controllers (PLCs). Consequently, most of the related works in the literature try to adapt the SCT framework to the programming languages defined by the International Electrotechnical Commission in the IEC 61131-3 standard (International Electrotechnical Commission, 2003), especially to the most popular graphical languages: Ladder Diagram (LD) and Sequential Function Chart (SFC). While the methods developed in de Queiroz (2002), Fabian and Hellgren (1998), Gouyon, Pétin, and Gouin (2004), Lauzon, Mills, and Benhabib (1997), Leal, da Cruz, and Hounsell (2012) and Ramirez-Serrano, Zhu, Chan, Chan, Ficocelli, and Benhabib (2002) are all based on LD, those presented in Charbonnier, Alla, and David (1995) and Vieira, Santos, de Queiroz, Leal, Neto, and Cury (2017) are SFC-based. However, the particular syntax of those languages offers little portability for the proposed methods to be applied outside PLC-based environments.

In addition, because the future development of multi-terminal DC (MTDC) grids might imply a combinatorial explosion during the synthesis of the supervisors, a decentralized architecture that localizes the control is suitable. Thus, the implementation method requires that the information communicated between the different controllers should be taken into account, as opposed to previous contributions, where only centralized (Balemi, 1992; Cantarelli & Roussel, 2008) and modular (de Queiroz, 2002; Vieira et al., 2017) architectures with no communication between controllers were contemplated.

The remainder of this paper is organized as follows. Section 2 reviews the fundamentals of DES modeling and SCT. A case study is presented and a decentralized supervisory control for the start-up of an HVDC system is synthesized in Section 3. In Section 4, the proposed implementation method is presented and the simulation results obtained in the EMTP-RV software are shown. At last, conclusions are drawn in Section 5.

2. Background

This section reviews the basic notions of DES modeling, along with the fundamentals of SCT and the control architectures that can be derived from the synthesized supervisors.

2.1. Discrete event systems

A DES is a discrete-state, event-driven system which does not depend on time and whose state evolution depends entirely on the occurrence of asynchronous discrete events (Cassandras & Lafortune, 2008). Based on the property of controllability, it is possible to divide the event set Σ into two subsets, i.e. $\Sigma = \Sigma_c \cup \Sigma_u$, where Σ_c and Σ_u are respectively the set of controllable and uncontrollable events. The occurrence of an event in Σ_c (resp. Σ_u) can (resp. cannot) be prevented by a supervisor S. The concatenation of the events $\sigma_i \in \Sigma$ $(i = 1, \dots, n)$ forms finite sequences (or strings) which are all represented by the infinite set Σ^*, derived by the operation called Kleene-closure (*):

$$\Sigma^* = \{\varepsilon \cup \sigma_1 \cup \sigma_2 \cup \sigma_3 \cup \sigma_1\sigma_2 \cup \sigma_1\sigma_3 \cup \dots\}, \quad (1)$$

where ε is the empty string. Thus, a language L, which is a finite set of finite-length strings formed from events in Σ, is a subset of Σ^* ($L \subseteq \Sigma^*$). A language is said to be prefix-closed if any prefix $t \in \Sigma^*$ of any string $s \in L$ is also an element of L ($L = \overline{L}$), with $\overline{L}$ consisting of all the prefixes of all the strings in L:

$$\overline{L} := \{s \in \Sigma^* : (\exists t \in \Sigma^*)\, [st \in L]\}. \quad (2)$$

A deterministic automaton A can be defined as a six-tuple $A = (X, \Sigma, f, \Gamma, x_0, X_m)$, where X is the set of states, Σ is the finite set of events associated to A and $f : X \times \Sigma \to X$ is the partial transition function. This function can be extended to $f : X \times \Sigma^* \to X$ in a natural way. Moreover, $\Gamma : X \to 2^{\Sigma}$ is the active event function representing the set of all events σ for which a transition $f(x, \sigma)$ is defined at state x. Finally, $x_0 \in X$ is the initial state and $X_m \subseteq X$ is the set of marked states that represent the completion of a task.

We distinguish between the language $L(A)$ generated by A and the language $L_m(A)$ marked by A. While $L(A)$ represents all the strings s starting from the initial state and whose transition function f is defined at (x_0, s):

$$L(A) := \{s \in \Sigma^* : f(x_0, s)\} \text{ is defined}, \quad (3)$$

the language marked by A is formed by the strings s that start from the initial state and end at a marked state ($f(x_0, s) \in X_m$):

$$L_m(A) := \left\{s \in L(A) : f(x_0, s) \in X_m\right\}. \quad (4)$$

An automaton is said to be non-blocking when all its states are accessible from x_0 and co-accessible, that is, X_m can be reached from state x. Then, $\overline{L_m(A)} = L(A)$.

2.2. Supervisory control theory

The SCT was first proposed in Ramadge and Wonham (1987). Based on language theory and DES modeling, the SCT aims to synthesize a supervisor that ensures by construction that the behavior of the system (also called plant) under control remains admissible with respect to a set of specifications. The plant is modeled in the form of an automaton G and is independent of the control objectives as it represents the physical process. The designer then models in the same form the control specifications to be imposed on the uncontrolled plant in order to restrict its behavior within the subset $K \subseteq L_m(G)$. Then, conforming to the SCT, a non-blocking supervisor S exists such that $L_m(S/G) = K$ and $L(S/G) = \overline{K}$, with $K \subseteq L_m(G)$ and $K \neq \emptyset$, if and only if the controllability condition ($\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$) and the $L_m(G)$-closure condition ($K = \overline{K} \cap L_m(G)$) are respected. If K is not controllable, the largest sublanguage of K that is controllable, with $L_m(G)$-closure condition, can be computed. Formally, the supervisor S for the plant G is a function that maps each word of the language of G to the set of controllable events which are enabled after the occurrence of that word. Meantime, the set of feasible uncontrollable events cannot be disabled by the supervisor S. So for a string $s \in L(G)$, $S(s)$ is defined according to Cassandras and Lafortune (2008):

$$S(s) = [\Sigma_u \cap \Gamma(f(x_0, s))] \cup \left\{\sigma \in \Sigma_c : s\sigma \in \overline{K}\right\}. \quad (5)$$

In the first term of (5), the supervisor enables after string s all uncontrollable events that are feasible in G. In this way, a feasible uncontrollable event is never disabled. In the second term of (1), all the controllable events that extend s inside of K are allowed. The language marked by the closed-loop S/G is defined as follows:

$$L_m(S/G) := L(S/G) \cap L_m(G), \quad (6)$$

where $L_m(S/G) \subset L(G)$ is strictly contained in the language generated by G and it corresponds to the optimal behavior of G under the supervision of S. In a centralized or monolithic control architecture (Fig. 1), the automaton representing a supervisor is typically automaton S/G itself.