DECISION AID FUNCTION FOR RESTORATION OF TRANSMISSION POWER SYSTEMS AFTER A BLACKOUT

THÈSE N^{O} 1702 (1997)

PRÉSENTÉE AU DÉPARTEMENT D'ÉLECTRICITÉ

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

Tatjana KOSTIĆ

Master of Technical Sciences in Electrical Engineering, Belgrade University, Yougoslavie
de nationalité yougoslave

acceptée sur proposition du jury:

Prof. A. Germond, directeur de thèse
Prof. R. Bacher, corapporteur
Dr P. Bornard, corapporteur
Dr Y. Izui, corapporteur
Prof. C.-C. Liu, corapporteur
Prof. A. Rufer, corapporteur

Lausanne, EPFL
1997

Abstract

This thesis, based on a project realised in cooperation with Électricité de France (EDF), proposes a new concept for a Decision Aid Function FOr Restoration (DAFFOR) of transmission power systems after a blackout. DAFFOR is an interactive computer tool which provides the operators in power system control centres with guidance concerning the actions to execute during the restoration, in real-time conditions. In other words, it takes into account the real-time state of the power system, including the unforeseen events that may happen during the restoration.

Since time is a limiting factor and the decision making is a highly combinatorial problem, a knowledge-based system is proposed in order to solve it. The restoration process can be decomposed into two main stages. The first one, **skeleton creation**, consists of starting the production units and connecting some transmission devices in order to energise a strong network. The second stage, **load pickup**, aims to supply the consumers. In DAFFOR, EDF's strategy for the first restoration stage has been implemented, and a new strategy for the load pickup stage has been proposed and implemented in the form of rules. The above restoration strategies represent DAFFOR's knowledge, which has been enhanced with a number of heuristics.

DAFFOR consists of two kernels: the Reasoning kernel and the Real Time Update kernel. The **Reasoning kernel** has the task of assisting the operator during the restoration process and is the interactive guidance part of DAFFOR. It can either suggest a control action to execute on the power system to the operators or assess a control action provided by the operators. The control action is suggested with respect to operating limits (over- and under-voltages, frequency excursions and overloads) and according to knowledge (restoration strategy and heuristics). The feasibility of an action is tested within an internal dynamic simulator, which also takes into account the time necessary to physically execute an action (e.g., telephone a person in the field). The Reasoning kernel can adapt its operation via data generated by the **Real Time Update (RTUpd) kernel**. The RTUpd kernel steadily reads real-time power system data from System Control and Data Acquisition (SCADA) function and those entered by the operators (if unavailable from SCADA). It generates a coherent data set, which is the only real-time information available to the Reasoning kernel, and the message which indicates to the Reasoning kernel how to continue its operation. In addition to the real-time data, the RTUpd kernel has two feedback inputs internal to DAFFOR: a coherent data set generated in the previous data processing by the RTUpd kernel itself, and a simulated data set generated by the Reasoning kernel (i.e., its internal dynamic simulator). With these three inputs, the RTUpd kernel generates the current image of the power system, and identifies unforeseen events. Thanks to the RTUpd kernel, the Reasoning kernel may keep up with the dynamic evolution of the power system.

The stand-alone prototype of DAFFOR has been tested with data provided by EDF, and shown very good efficiency. At present, it is about to be coupled with the EDF's operator training simulator in order to test its real-time functionality.

This work also proposes an original method aimed at the determination of a strategy for the load pickup stage. A genetic algorithm has been developed which generates the optimised sequences of manoeuvres for different initial states of the power system for the second restoration stage. It uses the dynamic simulator as its evaluation function. The obtained results have shown that some additional manipulations should be done in order to deduce generic rules for the load pickup strategy. At present, the obtained sequences are classified in a decision tree, which permits the most adequate sequence for the initial state to be chosen.

Résumé

La recherche liée à cette thèse a été effectuée dans le cadre d'une étroite collaboration entre le Laboratoire des réseaux électriques de l'EPFL et l'Electricité de France. Elle propose un concept original ayant permis d'élaborer une fonction d'aide pour la reconstitution (DAFFOR) d'un réseau de transport suite à une panne majeure. DAFFOR est un outil informatique interactif permettant de guider les opérateurs dans la sélection des actions à exécuter pendant la reprise de service en temps réel. En d'autres termes, il prend en considération l'état du réseau en temps réel et en particulier les aléas susceptibles de survenir pendant la reprise de service.

La reprise de service est un problème fortement combinatoire et requiert une durée aussi courte que possible. Ainsi, on propose un système basé sur la connaissance afin de le résoudre. Le processus de reconstitution est décomposé en 2 phases principales. La première, appelée **création de l'ossature**, consiste à démarrer les unités de production et à mettre en service les lignes de transport de sorte à constituer un réseau suffisamment rigide. La seconde, appelée **reprise de charge**, a pour but d'alimenter les consommateurs. Dans DAFFOR, la première phase consiste essentiellement en une stratégie actuellement adoptée par EDF. En revanche, la seconde a fait l'objet d'une stratégie de reprise de service proposée à titre de contribution originale dans cette thèse. L'ensemble a été implémenté sous forme de règles et renforcé par un certain nombre d'heuristiques.

DAFFOR consiste en 2 noyaux: le noyau de raisonnement ("Reasoning kernel") et le noyau de mise à jour ("Real Time Update kernel"). Le **noyau de raisonnement** est un guide interactif ayant pour tâche d'assister l'opérateur durant le processus de reconstitution. Il peut soit suggérer à l'opérateur une action (manoeuvre) à exécuter dans le réseau, soit évaluer une action proposée par l'opérateur. Les actions sont suggérées par DAFFOR tout en respectant les limites d'exploitation et en accord avec la base de connaissance. La faisabilité d'une action est testée à l'aide d'un simulateur dynamique qui prend également en compte le temps nécessaire pour exécuter physiquement cette action. Le noyau de raisonnement peut ajuster son fonctionnement selon les données générées par le noyau de mise à jour. Le **noyau de mise à jour** lit les données en temps réel provenant du SCADA et celles introduites manuellement par l'opérateur. Il génère d'une part un ensemble de données cohérent qui représente la seule information en temps réel disponible pour le noyau de raisonnement et d'autre part les messages indiquant à ce noyau de raisonnement comment poursuivre sa tâche. En plus des données temps réel, le noyau de mise à jour a deux autres ensembles de données générés de façon interne dans DAFFOR: un ensemble de données cohérent issus du précédent traitement effectué par le noyau de mise à jour lui-même et un ensemble de données simulé issus du noyau de raisonnement (c'est à dire son simulateur dynamique). A l'aide de ces trois type de données, le noyau de mise à jour crée l'état courant du réseau et identifie les aléas. Grâce au noyau de mise à jour, le noyau de raisonnement suit l'évolution dynamique du réseau.

Le prototype de DAFFOR a été testé avec des données provenant d'EDF et a fait preuve d'une bonne efficacité. A présent, il est sur le point d'être couplé au simulateur d'entraînement d'EDF afin de tester sa fonctionnalité en temps réel.

Ce travail propose également une méthode originale ayant pour but de déterminer une stratégie de reprise de charge. Elle consiste en un algorithme génétique produisant des séquences d'actions optimisées pour différents états initiaux du réseau préalablement à la phase de reprise de charge. Elle utilise le simulateur dynamique comme fonction d'évaluation. Les résultats ont montré que des traitements supplémentaires devraient être faits afin de déduire des règles génériques pour la reprise de charge. Actuellement, les séquences obtenues sont classées selon un arbre de décision. Celui-ci permet d'identifier la séquence la plus adéquate pour un état initial spécifié.

Acknowledgements

The idea of this project emerged during my 15-month stay at the Electric Power Systems Laboratory (LRE) of the Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland. I was able to come here and start the work on my M.Sc. thesis thanks to the financial support of the Swiss Federal Commission for Grants, the recommendations of Prof. Milan Ćalović and Prof. Nikola Rajaković from Belgrade University, Federal Republic of Yugoslavia, and to Prof. Alain Germond who made me welcome in his Laboratory (LRE).

My sincere gratefulness is dedicated to Prof. Ćalović, who accepted to be the mentor for my M.Sc. work in spite of the particular circumstance that I carried out the research in Switzerland. My further thanks go to Prof. Rajaković, who accepted to be a member of the committee, and to Prof. Germond, who was the mentor "on-site".

This doctoral thesis is the continuation of my M.Sc. work and was realised under the supervision of Prof. Alain Germond. I am greatly indebted to him for giving me the opportunity to start the researcher career in his laboratory and for accepting to supervise this thesis. His high scientific competence and human generosity guided me through this research. His famous enthusiasm and motivation were contagious - they used to render me enthusiastic and motivated too, even when the tunnel seemed too dark. Also, I wish to thank him for introducing me to many interesting people from the international power system community.

My further thanks go to Prof. Martin Hasler (EPFL) for chairing this thesis committee and to Prof. Rainer Bacher (Swiss Federal Institute of Technology, Zürich), Dr. Pierre Bornard (Électricité de France, Paris), Dr. Yoshio Izui (Mitsubishi Electric Co., Amagasaki), Prof. Chen-Ching Liu (University of Washington, Seattle) and Prof. Alfred Rufer (EPFL) for honouring me by accepting to be the committee members.

I am grateful to the Swiss National Foundation for Energy Research (NEFF) and to the R&D Division of Électricité de France (EDF) for having sponsored this work.

This project was realised in close collaboration with the group of Patrick Pruvot who provided us with all the necessary information on EDF's restoration strategy and test data.

I wish to thank Dr. Rachid Cherkaoui for always being available for the discussions on electric power systems theory and praxis, and for his moral support. He kindly accepted to play the role of the operator in some validation scenarios for DAFFOR (Section 4.4), and to translate the Abstract in French, thus leaving me several more hours for writing this document.

The research on genetic algorithms (Chapter 6) probably would have never been started without my old friend and colleague, Dr. Thierry Cornu. He told me the story about the small robots designed at EPFL, which learned how to get out of the labyrinth thanks to genetic algorithms. After he explained the idea of that method to me, I imagined "my" restoration *control actions recombining in order to produce sequences of actions, and the next morning I started to read the legendary Goldberg's book.*

A few months later, I met Dr. Juan Alba and I showed him my first results. He came to the idea to classify the sequences within a decision tree. We decided to study together the feasibility of the combined approach, which was realised later on, during Juan's stay at the LRE. I wish to thank him for the hours of fruitful discussions and team work and, above all, for his friendship.

I have greatly appreciated the excellent atmosphere of the LRE during the last six years. Bad times were made easier thanks to people like Ms. Andrée Moinat, Dr. Antonio Piras, Dr. Thomas Czernichow, Dr. Farhad Rachidi, Prof. Michel Ianoz, Ms. Marcela Lenz, Dr. Christian Tinguely, Eric Hounkpatin, Dr. Dagmar Niebur, Dr. Pierre Zweiacker, Dr. Silvia Guerrieri, José-Luis Bermudez, Carlos Batlle and Mike Chapman (whom I thank also for reading and correcting the manuscript). My warm thanks go to all of them, not only as colleagues but as friends.

Finally, my infinite thankfulness is dedicated to my parents, who always served me as good examples and encouraged me to study. There were periods when, together with my sister, they sacrificed a lot of their comfort in order to enable me to be educated by my own choice. This work is the expression of my boundless love and gratitude for all three of them.

Tatjana Kostić

Lausanne, July 1997

Contents

Chapter 1

Introduction

The restoration of a transmission power system from a blackout is a complex real-time control problem. Operators in control centres are, fortunately, not faced with this situation very often because power systems are designed to prevent a total system collapse (e.g., protection devices which isolate a faulted transmission line, shed load, or isolate thermal units). However, a disturbance may cause some cascade events which result in a degradation of frequency and / or voltage conditions in the power system. As the transmission power system is meshed, the consequences of this degradation are spread all over the system. Usually, the system is divided into islands (by protection devices) in order to preserve at least some parts of the system. However, if there is no balance between production and consumption in the islands, they are likely to be blacked-out.

The degradation of the conditions in a power system may be too fast (a few seconds) for its protection to play its role, which leads to a critical loss in the consumer supply, i.e., a blackout. The following are some of the reported examples of blackouts:

- The blackout that the French power system experienced on December 19, 1978 [Che80]. The initial event was the tripping of a 400kV transmission line due to an overload which spread over the network and blacked-out more than 75% (i.e., 29'000MW) of load. The total load was restored after 10 hours (55% after 3 hours, 90% after 6 hours).
- The blackout that the southern Swedish power system experienced on December 27, 1983 [Kea87]. The initial event was a busbar fault in a vital substation near Stockholm which

caused a cascade tripping of lines and cut off the load centres in the south from hydro production and imports. About 67% (i.e., 11'400MW) of load was left unsupplied. The total load was restored after 7 hours (80% after 2 hours).

- The total blackout that the Hellenic power system experienced in March 1989 [Fou97]. Similar to the Swedish case, the initial event was a busbar fault in a vital substation which caused a cascade tripping of lines and cut off loads, which were restored after 3 hours.

The above examples show that blackouts, unfortunately, happen. Since this is not the case very often, there are few operators who are familiar from their own experience with the control of the power system in restoration. Since the powerful operating training simulators (OTSs) have become available, operators are usually trained to be prepared for restoration [IEE93]. In addition, power system utilities usually develop a restoration plan [Adi94, Anc95] which provides the operators with guidelines for control in restoration. However, these plans are usually concentrated on the first restoration stage, in which the power system is the most vulnerable. The second restoration stage, in which the main objective is to supply the consumers once the power system becomes more robust, has been paid little attention in literature.

The conditions of control during restoration are very difficult. The operators are under stress due to the amount of alarms and clients' calls, and due to the responsibility they have in "rebuilding" the power system quickly in order to decrease the time of no energy supply to the consumers. In addition, while performing restoration control actions, they must respect many operating constraints concerning the power system elements and their permissible operating points (time-, frequency-, voltage-, security-related requirements). Since the power system is a dynamic system, and the restoration procedure is carried out in real time, it can be subjected to unforeseen disturbances which may put in question the prepared-in-advance restoration guidelines. All these factors lead to the conclusion that an operators' decision-support tool, integrated in the existing energy management system (EMS) environment, would be welcome [IEE94c].

The subject of this work is such a decision support tool, called **D**ecision **A**id **F**unction **FO**r **R**estoration (DAFFOR). The proposed DAFFOR has the required functionality of guidance in real time. Transmission power system restoration is a kind of problem for which there is no universal mathematical algorithm. In DAFFOR, the guidance for the restoration procedure is implemented through a combined knowledge-based and algorithmic approach, each of them being well suited for the representation of different elements of the restoration problem: the restoration strategy and the heuristics are translated into rules and facts, while the power system models (topology, network, production and load) are implemented in an internal dynamic simulator. The real-time functionality of DAFFOR is reached through the original concept of real-time update and the flexible structure of the guidance part which enable DAFFOR to keep up with the evolution of the power system during restoration.

The proposed DAFFOR is the result of collaboration with the industrial partner, Électricité de France (EDF), which provided data and a part of the "knowledge" implemented in DAFFOR.

In the following section is given a brief overview of the reported works, which are in a similar direction as ours, i.e., knowledge-based systems for power system restoration.

1.1 Knowledge-based systems

A pioneering work in the domain of the application of knowledge-based systems (KBSs) to power systems is that of Sakaguchi and Matsumoto [Sak83]. They have proposed a rule-based method which finds paths of energisation towards sub-transmission network loads after a local disturbance. This KBS contained only 16 rules, but its global importance is the introduction of the concept of non-algorithmic methods in the minds of members of different power system-related communities: utilities, manufacturers and universities.

Wollenberg and Sakaguchi in [Wol87] define the so called **cognitive barrier**, which means the operator's inability for answering all information gathered by the EMS during abnormal power system conditions, and state that one of the principal motivations for introducing KBSs in the EMS environment is an attempt to overcome this cognitive barrier. They also forecast the suitability of such systems in real-time control problems such as alarm processing, switching operations or restoration control, and for operator training.

The KBSs are characterised by a clear separation between the domain's knowledge and inference. They are manipulating symbolic information (as opposed to numeric) and are suitable for representing less deterministic problems, where there is no standard algorithm. As Dillon says in [Dil88], "...the knowledge in the knowledge base represents the manner in which the problem solver tackles the problem, rather than being a description of the problem".

The motivation for building the KBS in the EMS environment is based on its ability to act fast on the information about the disturbances which occurred in the network: (1) quick information gathering, and (2) examining and applying many more rules than the operator could do during the same time. In order to use all the advantages of the KBS, and that it serves as an operator's aid in abnormal operating conditions, the following requirements should be satisfied:

- The KBS should produce the results in a very short time, which is smaller or at least equal to the time the operator would devote for the same reasoning;
- The coupling with the existing EMS software should be carried out, where the serious problem represents the communication between the KBS and the real-time power system, i.e., its dynamic database. Some software and hardware issues concerning the coupling between KB and algorithmic methods are discussed in [Hol92], and in [Ban95], an Operator Decision Environment (ODE) architecture is proposed for different KBS applications;
- Initially, the knowledge base of the KBS which is integrated in the EMS has about 60% to 80% of all possible cases that could occur during the operation. This is why the knowledge base should be created in such a manner that it can be easily modified, either its rules (e.g., new experiences) or facts (e.g., new power system structure).

The increasing interest in the KBS-methodology application to power systems in the 80's resulted in the creation of a dedicated conference, Symposium on Expert System Application to Power Systems, held for the first time in August 1988. In 1992/93, several surveys on the present development of KBSs were published: for Europe [Ger92], for Japan [Tan92], and the global state of the art [Liu92], [Mat92] and [CIG93b]. Most applications that have reached the operational, on-line, level concerned alarm processing, substation monitoring and sub-transmission/distribution network fault diagnosis and restoration: this confirmed the expectation of Wollenberg and Sakaguchi from 1987.

1.1.1 Knowledge-based systems for restoration

Many works have been reported on the restoration of distribution and sub-transmission power systems, and the following brief overview is far from being exhaustive.

Some of the prototype **distribution** system restoration KBSs can be found in [Liu88], and in [Koj89a] (which is the continuation of [Sak83]). [Wad89] reports on a system installed in the Himeji Sales Office (Kansai Electric Power Co.) since August 1988. Two distribution system restoration KBSs have been integrated as an enhanced function of the SCADA: [Mor89] has been operating since March 1990 at the Fukuoka Central Control Centre (Kyushu Electric Power Co.), and [Hig94, Nag91] in the Shinsonezaki Control Centre (Kansai Electric Power Co.) since March 1993. A recently reported KBS for fault analysis and restoration [Par95, Par97] has been installed at a sub-control centre in Seoul for field testing.

Another group of KBSs concern the **sub-transmission** level networks. One of the first on-line systems was CRAFT (Costumer Restoration And Fault Testing) [Mar91, Tom87], operating in the Puget Sound Power and Light Company since 1990. [Shi96, Tak88] have reported on an on-line fault diagnosis and restoration KBS installed in the Chubu Electric Power Co. in 1994, which is also coupled to the OTS. The KBS for radial sub-transmission network restoration [Hot90, Kur93] was field tested for 18 months starting in April 1989 at the Kobe Dispatching Centre (Kansai Electric Power Co.). Another KBS in Chubu Electric Power Co. [Kob94, Shi91, Shi92, Shi94] has been tested with the OTS at the Okazaki Dispatching and Control Centre. Finally, the Restoration Assistant (RA) [Kir91] system, which generates the switching sequences for path establishing during the first restoration stage, has been integrated with the OTS of the Consolidated Edison Company of New York, in October 1994 [Raf96].

As far as the **transmission** power system restoration is concerned, most of the reported works have reached only the prototype level, and usually in universities and research institutes. In our opinion, the main reason is that serious incidents in the power system occur quite rarely, so utilities consider that a tool which would be used rarely is not worth the necessary investments. However, we emphasise that the main motivation for building a KBS for power system restoration is exactly the fact that, since major incidents do not occur very frequently, operators usually have little experience in operating the system in such a state - the KBS is designed to aid the operator while making decisions.

A prototype [Dij88] has been developed to demonstrate that a KB approach is suitable for choosing and implementing restoration strategies. It can handle restoration strategies with or without islands. The prototype has been tested on a Dutch utility 380kV/150kV system.

Another prototype KBS [Kak88, Kak91] considers only the first restoration stage, and generates the switching sequences inside the substations.

There are two prototype KBSs for transmission power system restoration that have been developed in cooperation with EDF. The one reported in [Dar89, Fan91] focuses on the first restoration stage after a total blackout, and generates the complete sequence of restoration actions until the islands are synchronised. The other prototype, called MARS [Mon92], is aimed at generation of the restoration procedure after a local disturbance in the transmission network.

Recently, a prototype has been described in [Gai96] which uses an object-oriented approach for implementing a power system restoration knowledge base. It has been tested with Taiwan Power Company data.

All of the above prototypes use the load flow (AC, decoupled or DC), and sometimes the sensitivity matrix, for operation limit checking.

The development of a KBS for bulk power system restoration has been sponsored by the Electric Power Research Institute (EPRI). The first results of this work have been reported in [Lio95, Liu93], which concentrate on generation capability dispatch and tie-line utilisation in restoration. The prototype system has been tested with Philadelphia Electric Company data. At present, there are no numerical modules coupled to the prototype. The reasoning is based on pre-stored system data and power plant characteristics. In the framework of the same project, a concept of generic restoration actions (GRAs) has been proposed [Fin95], which can be used to construct the basic restoration building blocks (RBBs) according to restoration guidelines, which usually differ from one utility to another. Furthermore, the conceptual GRAs have been modelled by a scheduling method based on Petri nets [Wu97]. The method determines the sequence of restoration actions and the estimated time from the starting time. This is an ongoing project with the objective of integration of the prototype KBS within EPRI OTS.

A KBS for restoration after a medium size disturbance has been reported in [Cor93]. This system generates the restoration plan based on on-line EMS data. It has been tested with the OTS and is installed off-line in Iberdrola's Power Dispatch Centre in Bilbao (Spain). It performs the check of operating limit violations with an AC power flow. The important point about this system is the fact that consistency analysis and measurement estimation are performed to a certain extent since it is supposed that the state estimator output is not available in restoration.

A KBS for restoration after local disturbance has been described in [Koj89b, Miy94]. This system serves for the determination of faulty equipment and overload relief by adjusting the generation and the network switching operations. It checks the operating limit violations with an AC power flow. Since November 1993 the system has been installed in the Seibu System Load Dispatching Office (Tokyo Electric Power Co.) and connected to SCADA for field testing.

Last, but not least, a good example of different development stages of a KB application is the training simulator with expert system for restoration, reported by Krost, Rumpel et al. in numerous publications. The prototype knowledge-based system with high explanatory facilities has been described in [Kro89a, Kro89b]. The second step of the project was the creation of an interface to the power system database, in which the power system elements are represented with the intuitive Grid Data Language [Kro91, Rum92b]. After that, the power system models (long-term dynamic model and power flow) have been integrated in the simulator [Rum92a]. Finally, the intelligent OTS has been installed and tested since 1993 at the Stadtwerke Duisburg AG [Kön94]. It has been also used by the operators for every-day operation power flow computation.

1.2 Objectives of this thesis

This thesis has four objectives:

1) To propose a conceptual design for an interactive computer tool which is capable of helping operators in making decisions during the restoration of a transmission power system in real time, and which is modular so that any module can be easily replaced by another one (having the same function);
2) To propose the guidelines for the second restoration stage (load pickup stage) with application to the French transmission power system;
3) To develop and implement the stand-alone prototype of the tool and to validate it with data provided by EDF; and
4) To develop and implement the real-time prototype which is to be coupled to EDF's OTS in order to be validated.

1.3 Plan of the thesis

Chapter 2 concentrates on the problem of power system restoration. It introduces the general concept and definitions of restoration, such as plans, strategies, control problems, operators' preparation, etc. It also gives an overview of relevant publications in this domain and finishes with a particular case, considered in this work: EDF's power system and restoration strategy.

Chapter 3 enumerates the functional requirements which must be met in order to satisfy objective (1) from Section 1.2. According to them, the architecture for DAFFOR is proposed. It contains two kernels: the Reasoning kernel, which is responsible for guidance and interactivity, and the Real Time Update kernel which is responsible for keeping the Reasoning kernel informed about the evolution of the power system in real time.

Chapter 4 represents about half of the thesis, since it deals with the stand-alone prototype of DAFFOR, i.e., the Reasoning kernel. It starts with the formulation of the restoration problem and proceeds with the description of the modules, which form the Reasoning kernel, and interactions between them. One of these modules is based on knowledge, and contains the restoration strategy: EDF's strategy for the first restoration stage, and a strategy for the second restoration stage proposed in this work (objective 2). The validation results of the stand-alone prototype are presented and discussed at the end of the chapter (objective 3).

The real-time extension to the stand-alone prototype is the subject of Chapter 5. After describing the Real Time Update kernel's function and operation, it will be shown how the proposed concept of DAFFOR is supposed to work in real time (objectives 1 and 3). Finally, the validation architecture will be presented and discussed (objective 4).

Chapter 6 deals with, to our knowledge, an original approach to determination of the load pickup strategy (objective 2). At the very beginning of this thesis, a genetic algorithm-based method has been examined, which was expected to quickly give a solution for the load pickup problem. The first results of this approach are shown and discussed, along with reasons to postpone research in this direction.

Finally, Chapter 7 highlights the essential points of this work in terms of a conclusion, and identifies the directions for further research.

Chapter 2

Transmission power system restoration

This chapter introduces the general concept and definitions of power system restoration, and reviews the relevant publications.

A power system is operated under three categories of constraints: equality (power balance) constraints $G(x,u,z)=0$, operating constraints $H(x,u,z)\geq 0$ and security constraints $S(x,u,z)\geq 0$, with the vector functions $G(\bullet)$, $H(\bullet)$and $S(\bullet)$of the vector arguments x, u and z (state, control and disturbance vectors, respectively). The first diagram of the power system security states, in which the sates are defined with respect to the above constraints, has been proposed in [DyL67]. Following this publication, [Cih69] added the emergency state and today's well known state diagram is shown in Figure 2.1. When the system is in the normal state, the control is preventive with the aim of permanently maintaining the system in this state. Otherwise, the goal of the control is to bring the system back to the normal state as soon as possible.

In the **normal state** all three of the sets of constraints are satisfied: the load is supplied under the normal voltage conditions and frequency, there is no overload on the system transmission devices and the system is (n–1) secure. If a disturbance occurs, the system moves to the **alert state** where the load supply is still satisfied and the complete system is in service, but the security (n–1) is violated. The system can be brought to the normal state by means of the preventive control, but if another disturbance occurs, the protection is likely to operate and bring the system into the **emergency state** which is characterised by operating constraints violations (voltages and frequency outside of permitted limits, overloaded transmission devices). If the cause of the disturbance is not too serious, the system may be brought to the normal state either directly or through the alert or restoration state, by means of corrective control.

A serious disturbance may lead to the breaking of the system into isolated islands and/or load shedding, and then the system is in the **in-extremis state**, in which none of the constraints are satisfied. The restoration control actions are then applied to the system. The system will pass through the **restoration state**, where the operating constraints are satisfied but not all the load is supplied, in order to reach the normal state, either directly or through the alert state.

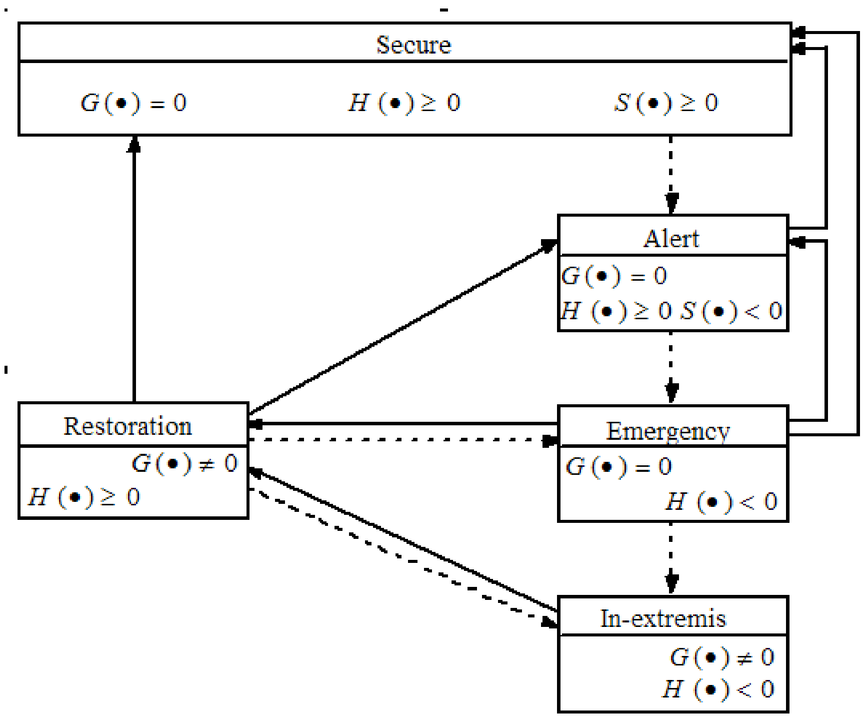


Figure 2.1 Power system operating states. Dashed and bold lines represent moving from one system state to another due to disturbances and control actions, respectively.

The initial event, which causes the system to move from the normal secure to one of the abnormal states, may be one of the following [Dav75]:

- natural occurrence (e.g., fault caused by lightning),
- false operation (e.g., malfunctioning equipment or protection device), or
- human factor (e.g., false manoeuvre).

The last two kinds of events have been shown as dangerous because they are not predictable and they can cause a major disturbance in the power system (e.g., the cascade failures). Consequently, the power system may reach the emergency or in-extremis operating state. In order to avoid the system disintegration some corrective actions can be undertaken by the operators. Also, a certain number of protection devices may be triggered intentionally in order to isolate the faulty part of the system and/or to shed some loads. If the fault influence can be localised in this manner, the restoration is partial and can sometimes be executed automatically. However, if despite the corrective actions a great part of the system has been involved, the existing automatic restoration functions are locked [EDF86, Kea87], and the operators must start the total restoration procedure.

In this work, the case considered is one in which a major disturbance has caused a large scale incident in the transmission network so that the restoration must be applied by the operators in the control centre. The objective of the restoration procedure is to supply the maximum of the unserved load as soon as possible, while meeting all the operating constraints.

An excellent compilation of many restoration-related publications can be found in [Anc95]. Some of those publications are referred to in the following sections, where the most important problems which arise about the restoration are enumerated and briefly discussed.

Since it is almost impossible to forecast all the possible disturbances and predict their consequences on the system, the electrical utilities usually define restoration plans. They are founded on an appropriate restoration strategy which depends strongly on the type of the system. An overview of the main restoration strategies will be given in Section 2.1.

However, there are a number of restoration problems that are common for all power systems. These kinds of problems are due to characteristics and interactions among the power system elements, which are addressed in Section 2.2.

The issues concerning the human factor and its role during the restoration are discussed in Section 2.3.

Finally, Section 2.4 introduces the restoration strategy of the French national utility, Électricité de France, which has been considered in this thesis.

2.1 Restoration strategies

The control criteria in the restoration state are the maximisation of the load supply and the minimisation of the time required to achieve complete service restoration, with the security and operating constraints satisfied. Depending on the electrical and geographic characteristics of the power system, two principal restoration strategies can be distinguished:

1. sequential restoration of the entire system skeleton, and
2. parallel restoration of the asynchronous electrical islands.

The common characteristic for both strategies is the choice of the source of the initial power, called a black-start unit[1]. The black-start units are usually those with gas or hydraulic turbines because of their ability to be coupled on the network in short time. In the very beginning of the restoration, it is essential to provide a quick supply for the auxiliary consumption of the thermal units which failed their load reject[2] in order to make them available for the further

1 A black-start unit has the capability to start-up a generating unit without any power supply from the power system.

2 Load reject is the action of isolating a generating unit or plant from the system load and leaving the unit to supply only houseload.

restoration steps. Time is the critical factor because of the very severe operating constraints the thermal units have. The first restoration sub-goal is to synchronise as many production units as possible, in the short time interval, in order to increase the regulation capacity of the system being restored.

2.1.1 Sequential restoration of system skeleton

This strategy is usually applied to power systems with production and load concentrated in two geographically remote parts of the system, as it is the case with the transmission network of Sweden [Kea87]. It consists of energising the high voltage transmission lines with the aim of fast power supply for the auxiliaries of the thermal units which failed their load reject. When energising the network, some load must be picked up in order to maintain the system stability. In this manner, a global system skeleton is created.

2.1.1.1 Limitations

The system requirements in applying this strategy are related to:

- the initial power supply unit which must be able to satisfy the load necessary for maintaining an acceptable voltage level, and
- the available compensation equipment on the transmission lines which should provide a satisfactory reactive power compensation, and therefore the improvement of the voltage profile in the low loaded network.

2.1.2 Parallel restoration of asynchronous electrical islands

This strategy is more widely used than the previous one for the reason of operating system stability. In order to define the electrical islands, the geographic location of each production unit and load must be taken into account, in the sense that each island must have at least one black-start unit. The scenario for creating the islands is usually determined in advance and is often executed automatically. The borders of the islands usually correspond to those of the control regions.

In the framework of this strategy, there are two global approaches depending on the speed of the load supply. In order to analyse the differences between them, the restoration state from Figure 2.1 has been decomposed into several restoration stages.

2.1.2.1 Pick up loads in islands first

During the restoration of an island, the maximum possible load is supplied (see Figure 2.2). The load level is limited by the production capacities that are in service in the island. When at least two islands are restored in this manner, they are coupled in order to form a new one, and so forth.

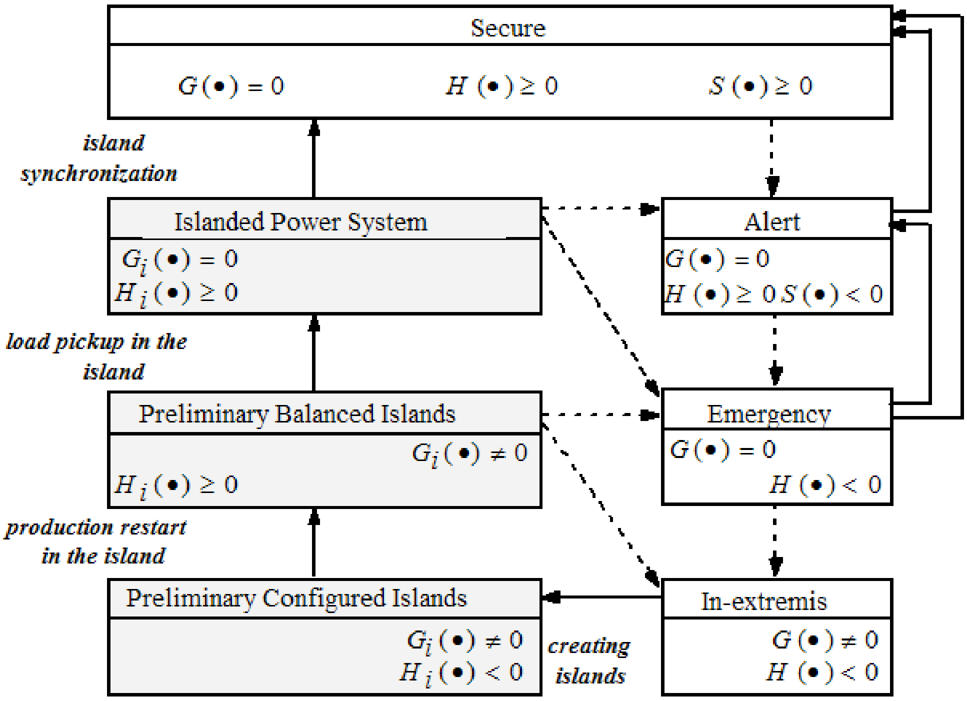


Figure 2.2 Parallel restoration with the priority of load supply. Dashed and bold lines represent moving from one system state to another due to disturbances and control actions, respectively. Grey boxes refer to islands, and white ones to the whole power system.

When the islands are coupled by switching on a tie-line, the problem of voltage magnitude and/or frequency mismatch can occur between the connecting nodes because of the relatively high amount of load which is already being supplied in the two restored islands. This approach has been reported as restoration policy for the Mexican power network [Gut87].

2.1.2.2 Synchronise islands first

During the restoration of an island, only a portion of the load is supplied (see Figure 2.3), that which is sufficient to satisfy the technical minimum of thermal units that are in service and to stabilise the voltages in the island. When at least two islands are restored in this manner, they are coupled in order to form a new one, and so forth. Then the load is picked up for the whole system.

This approach seems to be the most common one. It represents the restoration strategy adopted in many utilities: ENEL (Ente Nazionale per l'Energia Elettrica, Italy) [Del96, Mar84], EDF (Électricité de France) [EDF94a], PEPCo (Philadelphia Electric Power Company, USA) [Kaf81], Swiss Utilities [Wel92], Hydro-Québec (Canada) [Mor87], Hellenic Public Power Corporation [Fou97], etc.

2.1.2.3 Limitations

Some of the limiting factors in applying parallel restoration strategy for both of the above-mentioned approaches are the following:

- the number of available black-start units,

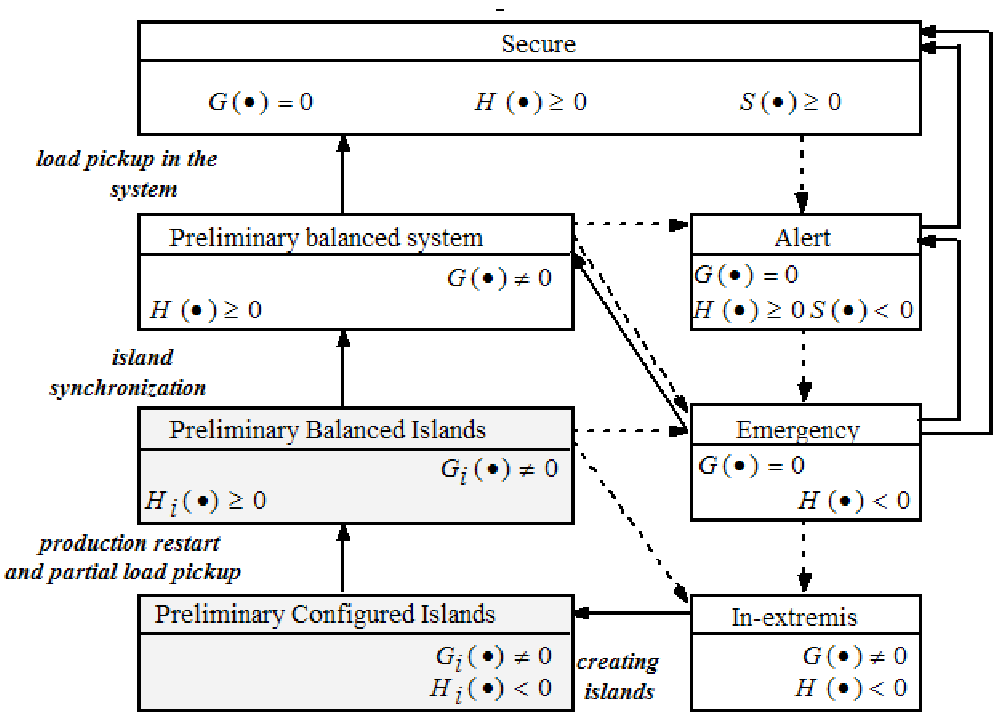


Figure 2.3 Parallel restoration with the priority of island synchronisation. Dashed and bold lines represent moving from one system state to another due to disturbances and control actions, respectively. Grey boxes refer to islands, and white ones to the whole power system.

- the ability to match generation and load within prescribed frequency limits,
- the existence of tie points capable of measuring synchronisation with neighbouring subsystems,
- the number of operator teams available in the control centres which are able to operate the electrical islands simultaneously, etc.

2.2 Problems and control in restoration

The power system that has been subject to a major disturbance is in a state which is quite different from the normal operating one because the disturbance will cause operation of the protection devices and the faulty part of the system will be isolated. If the initial event caused the cascading, it is likely that a large segment of the system, or even the whole system, be blacked out because of the voltage and/or frequency collapse. The isolated system is out of service and some of the characteristics of such a system are the following:

- long unloaded transmission lines to be energised,
- uncertainty about the load that will be found after energising the network,
- several subsystems (islands) to be operated simultaneously,
- small number of coupled units (i.e., low inertia) in each subsystem, etc.

The restoration procedure is carried out with the aim of supplying the consumers as soon as possible. To do this, the production must first be restarted and then the network must be strengthened in order to maintain secure and stable power system operation while providing power of good quality to the consumers. The restoration procedure can therefore be divided into two global stages:

I A stage of **production restarting and network integration**, wherein the main control concerns are about voltage profile, unit restart, switching operations and prime movers response to a sudden load pickup; and

II A stage of **load supply**, where more attention is paid to the real and reactive power balance, overloads and load uncertainty.

The above-mentioned classification of the control issues is not exclusive. For instance, the voltages must also be maintained within the prescribed limits during the load supply stage, and power balance must be respected during the first restoration stage. The problems are only classified according to the probability of their occurrence, which depends strongly on the nature of the power system and the interactions among its elements.

In the following discussion, the typical restoration problems will be briefly discussed, then the means of the power system control will be enumerated, and finally some guidelines concerning the control during the restoration will be given.

2.2.1 Typical restoration problems

2.2.1.1 Ferranti effect

The Ferranti effect is the phenomena that occurs when energising an unloaded transmission line. The sustained voltage generated on the receiving end of the line is higher then the voltage on its sending end. For short overhead lines, voltage increase is almost negligible (on a 400kV 200km line, the receiving voltage is higher than the sending voltage by 2,2%). However, for longer lines the voltage increase can be very high (on a 400kV 500km line, the voltage increase is by 15,5%). This problem can be managed by decreasing the source's exciter voltage (taking care to not violate the under-excitation limit) which should yield acceptable voltages on the receiving end of the line.

The voltage increase causes a reactive power production increase that should be compensated. Some of the measures that can be undertaken to decrease the reactive power produced on the line are the following:

- absorb capacitive load;
- switch off all the reactive power sources, including the shunt capacitor banks;
- set taps of the transformers working in parallel at different positions in order to increase the circulation current and the reactive losses;
- energise only the lines which carry an important load.

2.2.1.2 Switching over-voltages

Over-voltages occur as the consequence of energising large power system segments or switching on the capacitive elements. They are of a transient nature and usually are not a limiting factor in a restoration procedure. The experiments reported in [IEE92c] show that the switching over-voltages are not dangerous if the steady state voltage in the system is less than $1.2U_{nom}$.

2.2.1.3 Ferroresonance

Ferroresonance is a long-term oscillatory low-damped phenomenon which may appear in an electric circuit when the equivalent capacity is in series or in parallel with the equivalent non-linear inductive element. In the former case, it is referred to as series ferroresonance while in the latter case it is called parallel ferroresonance.

When a transformer is energised in the early restoration stages, the unloaded overhead line (capacitive load) and the transformer (non-linear inductive element) are in series and form the oscillatory circuit that is likely to trigger series ferroresonance. It can occur when the circuit's proper frequency equals the nominal network frequency and is manifested by a sudden current jump in the transformer. This current may cause an over-voltage on the transformers' ends [Ger75] which is much higher than the source voltage and the transformer insulation might be damaged.

In order to damp ferroresonance oscillations when energising an unloaded transformer (i.e., to modify the U–I characteristic of the oscillatory circuit), some of the following measures may be undertaken:

- loading the transformer by connecting a sufficient load to the network;
- switching on the transformer together with its closing resistance.

Some heuristic rules have been established based on the results of numerical simulations:

- in order to damp the ferroresonance over-voltage on a satisfying level ($1.1U_{nom}$), the necessary 230kV load should be around 1.8MW per kilometre of a 500kV overhead line [IEE92c];
- an energised transformer should be loaded with at least 10% of its nominal power [Mor87].

2.2.1.4 Black-start capabilities

The very first control actions in a restoration procedure concern the production restart. It is necessary to be familiar with the geographic location and the nature of the production units. At this restoration stage, the black-start units are of the greatest interest because they will produce power for the auxiliaries of the thermal units without black-start capabilities. The black-start units are usually those with combustion turbines (cold restart takes about 10-15 minutes [IEE87a, IEE94a]), hydroelectric units and pumped-storage units[3]. The priority for

[3] In the EDF's (Électricité de France) restoration strategy [EDF94a], the use of the nuclear units for blackstart is primordial because of predominantly nuclear production in the country. Consequently, the nuclear units are equipped with low frequency isolation schemes [IEE95a] which enable a considerable success in load reject. In the USA, on the contrary, the Nuclear Regulatory Commission [IEE95b] forbids the use of nuclear units for blackstart purposes.

the supply is always given to the thermal units' auxiliaries, and in particular to nuclear plants for security reasons (another reason is their high nominal power and inertia time constant, which guarantee more secure load pickup). Usually some load must be picked up in these early restoration stages in order to maintain the real power balance which is constrained with the technical minimum production of the thermal units running in the system.

2.2.1.5 Thermal units start coordination

During the restoration, some critical intervals for the thermal units' restart must be taken into account. Namely, some thermal units may be restarted after only the *minimum critical time* (cold start), while the others must be synchronised on the network before the *maximum critical time* (hot start). Obviously, the priority in the restart schedule is given to the latter units: if the synchronisation fails, they will not be available during a long time interval (up to 36 hours). Hence, it is desirable to know several critical intervals of the prime movers, such as:

- interval between the shut down and the restart,
- interval between the restart and the synchronisation,
- interval between the synchronisation and the technical minimum, and
- interval between the technical minimum and the full load.

In the planning process, some utilities [IEE87b, IEE92a] create diagrams of the available real and reactive power as a function of time, which depend on the prime movers' response to the sudden load pickup (i.e., the inertia time constant). In this manner the operators are provided with simple guidelines based on the approximate off-line analysis.

2.2.1.6 Overloads

Overload of the transmission lines and transformers should be avoided during the restoration because the system (or the island) that is being restored is not as robust as the system in normal operating conditions. Therefore, the overloads could cause an opening of the breakers on some elements which transmit the power, which may result in the failure of the restoration procedure and bring the power system back in the in-extremis state. However, the transmission equipment is constructed in such a manner that it can support a certain level of overload during some period, which gives some time to the operators to undertake the control actions which will eliminate the overload.

2.2.1.7 Load uncertainty

It is impossible to forecast the load in the system (island) with full certainty. There are several factors which influence the load variations (with respect to the load before the blackout), such as:

- load level (peak or low load),
- total actual supplied load,

- power factors and types of the consumers,
- duration of the outage, etc.

Some field test results for mixed urban and rural load, reported in [Mar84], show that after a long disconnection (5 minutes), the restored load stabilised in 1 second at 110%, while after a short disconnection (2 seconds) it stabilised in 5 seconds at 90% of the old value. The transient peak was between 130% and 150%.

2.2.1.8 Load pickup step

When energising the path to non-black-start units, some load must be picked up for stability reasons. The load must match the technical minimum production of the thermal units running in the system. Each load pickup causes a certain decrease in frequency. If the inertia in the system is small (small number of units that are running), the load must be supplied in small increments. There are several empirical rules that have been obtained from field tests and simulations, in order to keep the frequency decrease within 0.5Hz:

- a load pickup step should not exceed 5% of the present system (island) generation [Kaf81] ;
- a load pickup step should not exceed *P*% of the total nominal power of the units running in the system; if there is no support from an external network, *P*=10, otherwise *P*=50 [Wel92].

2.2.1.9 Switching operations

Usually all the circuit breakers in the blacked out area are automatically opened before the restoration starts [Kea87, Mar84]. In this manner, the operators know exactly the state of their system before they proceed to the service restoration. However, the weaknesses of the method are that the closing of the breakers is a very time consuming operation, and that it creates a great burden on the stored energy. For these reasons, some utilities have only a pre-defined number of breakers open [EDF86] in order to avoid the redundant switching operations.

Switching operations are carried out with the aim of reconnecting the network and supplying the loads, and (dis)connecting the compensation devices. When energising the lines in the early restoration stages, the priority is given to those lines which link the black start units with the thermal plants whose auxiliaries need to be supplied for the production to restart. For these purposes, some utilities have defined the plan of switching sequences during the early restoration stages [EDF86, Mar84]. These switching sequences are prepared in advance and are executed either manually or automatically by remote control.

2.2.1.10 Under-frequency protection

The admissible variations of the system frequency in the normal operation state are within 1% range around the nominal frequency, so that the under-frequency relays are set according to this. Experience shows that during restoration, the frequency excursions are much larger. The results based on the analysis of 19 major disturbances [Dav75] show that the frequency variations in the restoration were within (96÷106) % of the nominal frequency. That is why during the restoration, these relays should be either locked or set according to the new operating conditions [IEE88, IEE95a].

2.2.1.11 Synchronisation of electrical islands

If the power system has been split into electrical islands, they should be synchronised according to the utility's restoration strategy. The choice of the substation where the islands are to be coupled should be based on the existence of synchro-check relays, capable of testing the phase angle difference across an open circuit breaker. Frequency, voltage magnitude and voltage angle differences must be reduced to within acceptable limits before closing the breaker.

2.2.1.12 Tie line closing

When a segment of the interconnection is blacked out, tie lines are usually automatically opened by the protection devices. Most utilities prepare the restoration plan based on their own black-start and control capacity. In this case, tie lines are closed only when the blacked out system becomes stable and restored enough for activation of the load frequency and power exchange control (AGC). However, if a blacked out system happens not to have any black-start capability, the power might be sent by the neighbouring utilities via a tie line. In order to safely close the tie line, several conditions must be verified [Lio95]: voltage transients, tie line capacity, enough loads along the energising path to absorb the extra power that may be sent, etc. Once the blacked out system manages to restart its production, the flow across ties are kept close to zero until the return of the AGC control.

2.2.1.13 State estimation

State estimation is one of the functions that is usually implemented in energy management system (EMS) of control centres, and is necessary for providing the operators and other EMS functions with the coherent power system state. The state estimator uses redundant measurement readings and the switch status in the power system in order to produce a coherent and reliable state. During the restoration procedure, or at least its very beginning, the telemetered information about the power system that is gathered in the EMS is usually sparse. This prevents the standard state estimator from producing a reliable power system state. Therefore, a restoration-dedicated state estimator should be available in order to be able to run different EMS functions during the restoration.

2.2.2 Means of control in power systems

2.2.2.1 Voltage and reactive power (V/Q) control

There are several devices for the voltage and reactive power control in a power system. Exciters of the generators, synchronous condensers, static VAR compensators and regulating transformers are devices that may be controlled on a defined interval of values, while shunt condenser and reactor banks, and series condensers are incremental devices (may be switched on/off). Long transmission lines may be used as shunt condenser banks.

Generator exciters, synchronous condensers and static VAR compensators are automatically controlled devices whose response depends on the actual voltage state of the network. Their controllers react on the difference between the input signal (generator output voltage) and the set-point, and act on the produced reactive power in order to compensate for the change. The response of these devices is generally very fast so that the compensation is efficient.

As for regulating transformers, their tap position change can be done automatically or manually and their response is much slower (10÷90 seconds [Wal86]). If the tap is fixed to a given position, the transformer is no longer in regulating mode.

Shunt condenser and reactor banks, and series condensers are incremental devices. In order to modify the produced reactive power they change their operating status. Long transmission lines belong to this group because of their capacitive nature. It should be noted that series condensers affect not only the voltage, but the real power distribution as well.

2.2.2.2 Frequency and real power (f/P) control

The main method for frequency and real power control in a power system is to act on the production units' output power; other ways are load shedding, pump storage input or output and phase-shift transformers (for the control of circulating real power). It is obvious that the devices used for real power control are fewer than that of the reactive power since there are fewer real power sources than reactive power sources.

The primary frequency control is carried out automatically in all system states. The automatic secondary f/P control is usually disabled in restoration after a major disturbance, and is carried out manually until the system becomes robust enough to be coupled with its neighbouring systems.

Load shedding is an extremely undesirable control action, but may be applied in the case of a major disturbance in order to prevent power system disintegration [Cou93]. Otherwise, load shedding can occur as the result of an under-frequency relay's triggering.

In the normal system operation, the turbine/pump storage units are used as generators only in the cases of peak load, and in pumping mode during low load demand. However, in the restoration state they are one of the essential power sources (i.e., black-start units) for starting the restoration procedure.

Contrary to all the devices mentioned above, which control the real power balance in the network, the tap position change of the phase-shift transformers is carried out in order to control the real power flow in a loop containing this transformer.

2.2.3 Control in restoration

2.2.3.1 Early restoration stages

During the early restoration stages in the systems with operating thermal units, it is necessary to pick up some load in order to ensure balance with the technical minimum generation of those units. A small number of running units implies a low inertia in the system (i.e., small capacity of the prime movers' response). Therefore, the load should be picked up in small increments. This can slow the restoration procedure, but provides smaller frequency deviations in the system.

Since the low loaded high voltage lines are being energised, the voltages in the network increase as well as the reactive power. Therefore, all the compensation devices producing reactive power (e.g. shunt condenser banks) should be switched off, and those which absorb the reactive power (e.g. shunt reactor banks) should be switched on. The voltage set-point of the units should be set to the value close to the under-excitation limit. The transformers should be connected together with some resistant load to avoid ferroresonance, and the taps of the regulating transformers should be locked during the first restoration stage. Finally, the lines should be energised according to the minimum reactive power production criterion.

As was mentioned in the previous section, the automatic secondary frequency control is usually disabled in the early restoration stages[4]. However, every load pickup will result in an overall frequency decrease. It is therefore necessary to adjust the power set-point of units manually as the amount of the restored load increases, in order to replace the secondary frequency control as long as it is disabled.

2.2.3.2 Late restoration stages

In later restoration stages, as the number of synchronised units increases, the inertia in the system also increases, and the load pickup step may be higher. The automatic secondary control is also likely to be enabled again. Then the control issues of main concern are the load pickup steps that correspond to the available spinning reserve of the units that are running and that do not cause the overloading of the transmission equipment.

If the taps of the regulating transformers have been locked at the beginning of the restoration procedure, they should be released when the restored load becomes more important. Also, the voltage set-point of generators should be set to a value close to the nominal one. It might be necessary to switch off some compensation devices absorbing the reactive power and switch on those that produce it. The goal is to increase the voltages in the system which becomes normally loaded. Another action is to energise the double circuits since this increases voltage.

[4] An approach to using a "secondary" AGC function inside the island, in parallel with the "main" AGC function in the healthy part of the system has been proposed in [Ros94]. According to the authors' answers to discussion, the approach could be used if only the islanding occurred (without the blacked out areas), in order to match the load demand with the production in the island.

2.3 Human factor in restoration

The operators in control centres play the main role in re-establishing the normal operation of a power system that has undergone a major disturbance. As the restoration state is not an everyday situation and the control requirements are quite different than those in the normal secure system operation, the operators must be appropriately prepared to cope with it. Some of the important issues concerning the human factor in restoration will be addressed next.

2.3.1 Preparing for restoration

2.3.1.1 Planning

Power utilities usually define restoration plans to be used by the operators when the system suffers a major disturbance. To do this, different numerical simulations are carried out [CIG90, IEE94b] by means of various computer programs: power flow analysis, transient stability analysis, voltage stability analysis, electro-magnetic transient program (EMTP), long-term dynamic programme, etc. The restoration plan is defined on the basis of the analysis of the results of these programmes and jointly with the operators' experience in controlling their power system. In order to follow the changes in the system equipment and load demand, the restoration plan update is an essential task in the utilities. Detailed description of the restoration planning procedure and the restoration plan requirements can be found in [Adi94].

An algorithmic approach for restoration planning has been proposed in [Hua91, Hua95]. The restoration is defined as a non-linear optimisation problem, where the minimisation criterion are the bus voltage deviations from their prescribed values. These deviations are to be minimised over a number of restoration stages. However, this approach considers only the successive steady state conditions in the power system, neglecting its dynamic behaviour.

2.3.1.2 Operators training

Since major disturbances in the power system do not occur very often, the operators have little occasion to face on-line restoration. In order to prepare operators for restoration conditions, it is essential to regularly organise training drills [IEE82, IEE91, IEE92b]. Reference [IEE93] describes the wide variety of training methods currently used in existing system restoration training programmes. The issues concerning the preparation, conduction and evaluation of the training for restoration are discussed in detail in [IEE96a, IEE96b].

The most advanced method of the training are the drills with the operator training simulators (OTSs). One of the first dynamic OTSs has been developed by Control Data Corporation and demonstrated at a PICA Conference in 1977 [Lat77]. Another large-scale OTS, capable of simulating a major disturbance, was developed in TEPCo (Tokyo Electric Power Co., Japan) [Sus86a, Sus86b] and has been in use since 1984. The next reported OTS was that of the EDF (Électricité de France) [Jea88], installed and running in EDF's Operators Training Centre in Caen since 1988. PECo (Philadelphia Electric Company, USA) has been the host utility for the EPRI OTS. Some restoration training results and evaluations are presented in [Chu91a, Chu91b]. In the Netherlands, a group of Dutch Electricity Generating Board operators have been trained for restoration on the standalone training simulator developed at Gerhard-Mercator University, Duisburg. The experiences and results of the training sessions are reported in [Omm94].

All of these reports show an increasing concern in the electrical utilities for their operators' training for restoration.

2.3.2 Acting in restoration

The operators in control centres have the energy management system (EMS) available, which provides information about the actual state of different system variables. They have also different types of EMS functions that can be used in order to facilitate the assessment of the system and the decision making. The operators carry out analysis on the basis of the above information and, after making a decision, act on the system either directly, by means of the EMS, or by asking the on-site colleagues to execute the control actions.

2.3.2.1 Monitoring and assessment

The operator's task is, first of all, to discover the disturbances in the system's operation. An important aid in this task is given by the visualisation of all the elements of the system and their operating status (on/off), as well as by the information about the main system variables (voltage level, load level, branch charge, network configuration etc.). Having all the necessary information, the operators can analyse the actual system state using the available EMS functions and find out the level of the disturbance in the system operation. According to the current state of the system and defined restoration strategy, the operators make step-by-step decisions on restoration actions.

2.3.2.2 Coordination among control centres

The production/transmission electrical utilities usually have the following control structure:

- one main control centre, having the information on the very high voltage network (400kV in Europe, 500kV in the USA and Japan) which covers the territory controlled by the utility,
- several area/region control centres, each responsible for the control of its area/region on the transmission and sub-transmission level (over 35kV), and
- production control centres, located at the power generation sites.

Each of these control centres should have a well defined role in the utility's restoration plan. The main control centre is usually the coordinator of the restoration procedure. It is responsible for general restoration control. The concrete restoration actions are carried out by the operators in area/regional control centres. The control can be direct, by means of EMS functions (e.g., remotely operated switching devices), or indirect, by the request for actions to be executed by the production control operators (e.g., modifications to the power set-point of a production unit) or by the teams in the power system substations (e.g., manual locking or changing of the tap position for the TCUL transformers). Consequently, the restoration procedure depends a great deal on the coordination among the entire team.

2.3.2.3 Communication channels

The strong need for coordinated action in restoration implies the need for appropriate and reliable communication devices between different control centres and the teams in the field. When a major disturbance occurs, the communication lines are usually overloaded because of client calls, which interfere with the communication between the control centres and the sub-stations and the production sites. Hence it is necessary to plan special communication devices (e.g., high frequency links, special telephone lines, optical fibres) which are reserved to the staff only and available at any moment, in order to enable a satisfying coordination in the operation control at all levels.

After a major disturbance in the power system, some information may not be available to the EMS, but could be communicated from another control centre by telephone, for example. This can help the operators to better assess the actual power system state, and decide more accurately about the actions to take.

2.4 EDF's restoration strategy

The work on the Decision Aid Function FOr Restoration of the transmission power systems (DAFFOR), reported in this thesis, is based on the project defined in cooperation with the French National Utility, Électricité de France (EDF). The interest shown by EDF in developing such a decision support tool has been based mostly on the fact that about 75% of the installed production capacity in the system is represented by nuclear units, which impose very severe operating constraints in restoration. For the same reason, EDF's restoration strategy has some peculiarities compared to the strategies of other power utilities, which were generally reviewed in Section 2.1. It is therefore convenient first to globally describe the French transmission-production power system, with its peculiarities concerning the restoration, and the organisation of EDF's power system control before proceeding to the restoration strategy itself. The data used for the development and validation of DAFFOR has been provided by EDF.

2.4.1 The French power system

The transmission/production power system controlled by EDF covers the whole of France on three main voltage levels: 400kV, 225kV and 63kV (the distribution level is out of the scope of this work). The power is transmitted through:

- overhead lines (400kV, 225kV and 63kV),
- underground cables (225kV and 63kV),

- auto-transformers (400/225 kV/kV),
- transformers (400/63 kV/kV and 225/63 kV/kV), and
- tie-lines with neighbouring countries[5] (400kV).

The circuit breakers of some transmission and compensation equipment are provided with special protection devices called AMUs[6]. They have two functions:

- they automatically open or close the breaker they are attached to when the absence of voltage is detected (e.g., after a disturbance), and
- if initiated by a single action from the control centre, they are programmed to command the reclosing or reopening of the breakers as soon as the voltage appears on their ends (this function may be disabled, as will be discussed in Section 2.4.3).

The production units are connected to the transmission network by step-up transformers:

- nuclear units (to 400kV or 225kV network),
- classical thermal units (to 400kV, 225kV or 63kV network), and
- hydraulic units (to 225kV or 63kV network).

The thermal units (both nuclear and classical) are equipped with isolating protection schemes which enable the load-reject capability of these units.

The loads are connected to the transmission network at any voltage level either directly or by transformers with tap changer under load (TCUL).

2.4.2 EDF's operational organisation

The French power system is operated from the National Control Centre (NCC), seven Regional Control Centres (RCC) and several Production Control Centres (PCC) located at production sites. The NCC has real-time information about the entire 400kV network, while each RCC has real-time information only for its own area, i.e. for all voltage levels from 63kV to 400kV. The coordination among RCCs and PCCs in the normal operation, as well as during the restoration, is the task of the NCC.

When a disturbance occurs, the NCC, as coordinator, gathers the information from all the RCCs and can therefore assess the overall impact of the disturbance on the power system. Depending on the severity of the outage, the NCC is responsible to declare either a local or a general incident [EDF86], as follows:

Local incident is declared if the disturbance has resulted in the blackout of an area of the network, but there is still a strong neighbouring network[7] which can provide the secure power supply of the blacked-out area thanks to the automatic restoration.

---

5 The French system is a part of the UCPTE interconnection.
6 Derived from the French name "**A**utomates à **M**anque de tension (**U**)"
7 A strong network may be either a healthy part of the network (i.e., which did not suffer the effects of the disturbance) or an external network (e.g., interconnection). The network is said to be strong if it covers a

General incident is declared if the disturbance has resulted in the blackout of one or more areas of the network, and the application of the automatic service restoration or actual generation plan is estimated not adapted and likely to worsen the situation, even if there is a strong neighbouring network.

If the disturbance has resulted in a local incident, the automatic service restoration may be executed by AMUs (i.e., automatic reclosure of the corresponding circuit breakers). Note that the automatic service restoration must be initiated by the operators with a single manual command from the RCC. However, if the incident is general, the automatic restoration function of the AMUs is completely disabled in order to leave the decision about the restoration actions to the operators.

In this work, only the case of a general incident is considered. Therefore, the automatic breakers' reclosure/reopening function of AMUs is supposed to be disabled and, consequently, the restoration strategy is to be applied to the power system by the operators.

* * *

As mentioned in Section 2.2, the restoration procedure can be divided into two principal stages. In the first restoration stage, the power system skeleton is built by synchronising the available power plants and energising transmission devices between them. Some loads may need to be picked up in order to maintain the satisfactory voltage and frequency conditions in the skeleton network. If the system is islanded, the above-mentioned apply to separate islands, after which they are synchronised together. Then the second restoration stage starts, in which the main part of the load is to be picked up and the transmission capacity is to be strengthened by energising the second circuits.

Before proceeding to the description of the restoration strategy itself, the description of the power system state after the general incident is given. This state is the result of the action of AMUs and the production units' protection devices.

2.4.3 Preparation for restoration after general incident

An important feature of EDF's global restoration strategy is the action of the programmable protection equipment AMU. AMUs are installed at certain points in the French network. Since they operate the corresponding breakers when the absence of voltage is detected (e.g., after clearing the fault induced by a disturbance), the network is brought into a particular topological configuration in a planned manner, as opposed to opening all the circuit breakers after a disturbance.

The points of installation of AMUs were defined with respect to several objectives [EDF86]. The division of the network, resulting from the action of AMUs, must be fine enough to let the operators entirely control the service restoration, but the number of switching devices that

large geographic zone, is stable in frequency and voltage, and has an important spinning reserve (a few thousand megawatts) are open should be minimised. The network must be divided so that the restoration after either a general or a local incident is facilitated. Finally, the programming of AMUs must be based on simple rules since their action may be superimposed by those of the operators.

The actual AMUs' programming and location in the French network reflect the above objectives. Since the case of a local incident is not considered in this work, only the location of AMUs in the network is discussed, not their automatic restoration function.

When the absence of voltage is detected, AMUs act as follows:

400kV level

- open both CBs of the line,
- close CB of inductive shunt compensators,
- open 400kV CB of all but one autotransformer (or 400kV/63kV transformer if there is no autotransformer) per substation,
- open 225kV CB of all autotransformers and 63kV CB of all transformers.

225kV level

- open both CBs of cables,
- open 63kV CB of all 225kV/63kV transformers,
- open 225kV CBs of transformers in antenna only,
- open CB of capacitive shunt compensators,
- open CB of loads,
- create one or more 225kV *load zones* (lz225) per 400kV substation so that:
    - lz225 has no more than 300km of lines,
    - lz225 has no more than 600MVA of installed transformation capacity,
    - there is no loop among 400kV substations through the 225kV level,
    - open only one CB of lines when creating lz225,

63kV level

- open CB of capacitive shunt compensators,
- create one or more 63kV *load zones* (lz63) per 225kV substation so that:
    - lz63 has no more than 200km of lines,
    - lz63 has about 50MW of connected load,
    - there is no loop among 225kV substations through the 63kV level,
    - open only one CB of lines when creating lz63.

The topology of the power system after the operation of AMUs is illustrated by a sample network shown in Figure 2.4. There may be several types of configurations for 63kV load zones (lz63):

- an lz63 may contain a single load (e.g., lz63_1, lz63_2 and lz63_4) or several loads (e.g., lz63_3),

- an lz63 may be embedded in another zone of the same type (e.g., lz63_2 is embedded in lz63_3),
- an lz63 may be between 2 zones of the same type (e.g., lz63_4 is connected to lz63_1 and lz63_3).

In any case, an lz63 leans on a single 225kV substation (e.g., lz63_1 and lz63_3 lean on substations S1 and S2, respectively), which means that there are no loops among 225kV substations via the 63kV level network. In addition, the closing of a single breaker is sufficient to pick up an lz63 (e.g., CB1 and CB2 for lz63_1 and lz63_3, respectively).

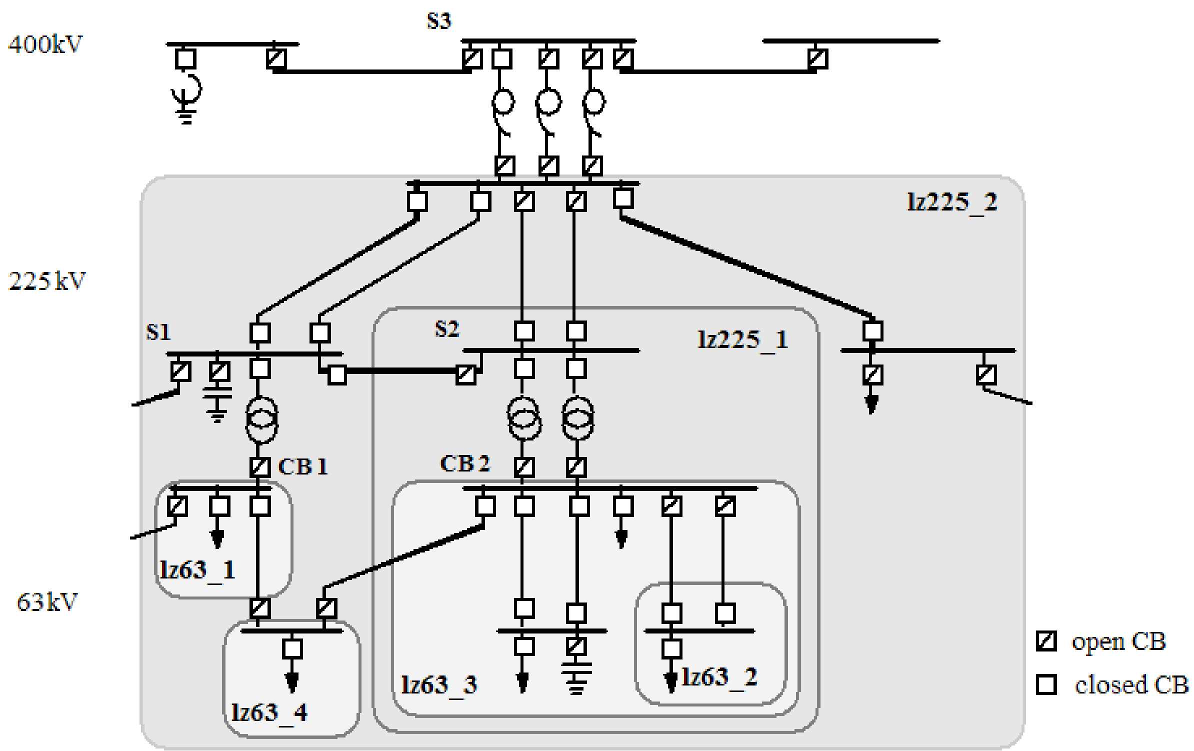


Figure 2.4 Sample network after the breakers are operated by AMUs

Since similar considerations apply also to 225kV level load zones (lz225), it can be concluded that the number of CBs that are opened in the absence of voltage is minimised. Inside 225kV and 63kV load zones, almost all CBs are left closed (except for those of 225kV cables, 225kV loads and all shunt condensers). This fact reflects the intention to improve the time necessary for service restoration by:

- decreasing the number of necessary switching operations,
- decreasing the probability of false manoeuvres,
- decreasing the probability of loops through lower voltage levels,
- maintaining connected load zones with small amounts of megawatts, each of which can be picked up by a single switching operation.

It should be noted that the production units are not represented on Figure 2.4, since there are no AMUs attached to the circuit breakers of production units and their step-up transformers. Since the French power system contains a great number of thermal units, and in particular nuclear ones, EDF's long term policy was to equip them with isolating protection schemes. These protection devices enable the load reject capability of thermal units, which is essential when the service is to be restored.

Figure 2.5 depicts the scheme of a thermal unit with a somewhat simplified representation of its auxiliaries. A thermal unit (nuclear or classical) is said to succeed the load reject if it tripped to the houseload (ICb open) and both internal power sources are available. In this case, the unit sends the telemetry point called TS1, which is visible at the synoptic map in the control centres. **The unit which emitted TS1 is ready to be coupled to the network, and therefore may be considered as a black-start unit.** From now on, it will be referred to as the "TS1 unit".

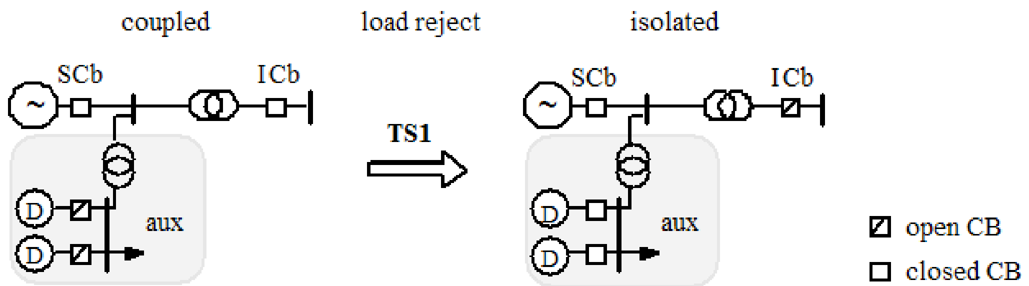


Figure 2.5 Both nuclear and classical thermal units may emit the TS1 signal. SCb stands for shutdown circuit breaker, and ICb for isolating circuit breaker.

A nuclear unit is said to partially succeed the load reject if it tripped to the houseload but one of the internal power sources is not available (i.e., the operation is not secure). In this case, shown in Figure 2.6, the unit sends the telemetry point called TS2, which is visible at the synoptic map in the control centres. **The nuclear unit which has emitted TS2 requires that its auxiliaries be supplied from the network as soon as possible, for security reasons.** From now on, it will be referred to as the "TS2 unit".

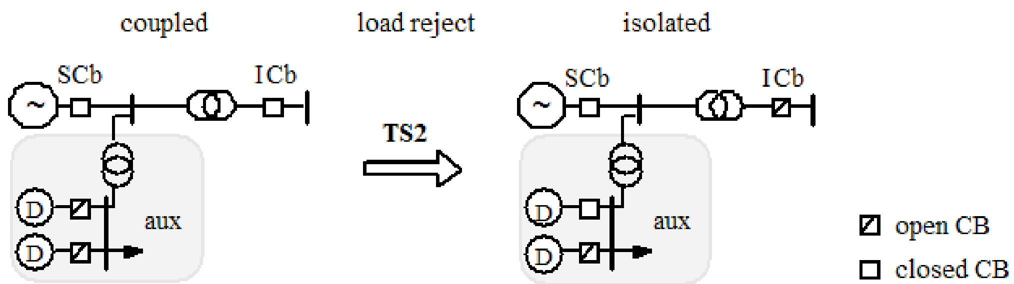


Figure 2.6 Only nuclear units may emit the TS2 signal. SCb stands for shutdown circuit breaker, and ICb for isolating circuit breaker.

A thermal unit (nuclear or classical) is said to fail the load reject if it has been shut down by its protection. The auxiliaries of a shutdown unit should be supplied from the network as soon as possible, in order to enable the unit's hot start. Otherwise, it may become unavailable for about 36 hours.

Finally, the available (e.g., not under annual maintenance period) hydraulic and combustion units are always considered as black-start units because they may be coupled very quickly (5 to 10 minutes).

* * *

A severe disturbance results in the operation of the protection devices as explained above. The operators from RCCs inform the NCC about the state of the area under their control. When the operators from the NCC conclude that the disturbance has produced a general incident, they send the message to all the other actors in control, i.e., operators in RCCs and PCCs. In turn, they will undertake the necessary actions in order to lock the automatic service restoration function of AMUs (from RCC) and to emit the corresponding telemetry points (PCC). Then they will proceed to the restoration of the French power system.

2.4.4 First restoration stage: Production restart and network integration

As most of power system utilities, EDF has paid particular attention to development of the restoration plan for the first restoration stage. There are at least two reasons:

1) the restoration must be guided by the natural order of control actions (e.g., a shutdown thermal unit can not be synchronised before its auxiliaries are supplied; the auxiliaries can not be supplied if there is no energised link to them; the link can be energised only if there is voltage on its end, given by a black-start unit), and
2) during this stage, the power system is in its most vulnerable state with respect to violation of operating limits in general.

In the case of the French network, another important fact is the large number of nuclear units. Since nuclear units impose additional technical and security constraints in the restoration, they have a particular consideration in EDF's restoration strategy.

As introduced in Section 2.4.3, the TS2 units and the shutdown nuclear units represent the priority from the point of view of the power supply in the very beginning of the restoration. Since any nuclear unit might emit the TS2 signal or be shut down in the blackout, several scenarios for supply of their auxiliaries have been prepared in advance and are tested in the field at least once every three years.

The scenarios for the supply of the nuclear units' auxiliaries are represented on the instruction sheets called FIDEOs[8]. A FIDEO sheet shows the preferential energising path from a black-start unit (source unit) towards a nuclear unit which needs the fast supply of its auxiliaries (goal unit), and describes in detail the order of switching operations in order to restore the path from the source to the goal unit. A sample FIDEO sheet can be found in [Dar89]. It should be noted that AMU devices, discussed in Section 2.4.3, are programmed in such a manner that the paths defined in FIDEO sheets are left closed in order to minimise the number of switching manoeuvres when applying the scenarios.

[8] Deduced from the French name "**FI**che **D**escriptive d'**E**nchaînement des **O**pérations"

Every nuclear production site has at least three scenarios associated with it: one on-site scenario (where the auxiliaries of the goal unit are supplied by a TS1 unit from the same production site) and two off-site scenarios (where the source unit is a hydraulic unit, a combustion unit or a TS1 unit from another site). The scenarios are ranked according to the order of priority, which is based on the speed and the facility of execution.

2.4.4.1 Present strategy: Elementary sub-networks

The first objective of EDF's present strategy [EDF94a] is to restore elementary sub-networks by fast supply of the auxiliaries of TS2 units and shutdown nuclear units, either according to the scenarios described above or from a strong neighbouring network (if there is one), whichever operation is estimated faster. The thermal units must be brought to their technical minimum as soon as possible, and therefore the matching quantity of consumption must be picked up.

The means of voltage control in the sub-networks are, by order of priority: (1) connection of inductive shunt compensators, (2) load pickups, and (3) temporary decrease of the units' voltage set-point. When two sub-networks are synchronised, the double circuits between them are to be energised. The final objective is to couple the elementary sub-networks in a national skeleton network.

The present strategy has two principal weak points: the number of elementary sub-networks that are to be controlled simultaneously and the time necessary to connect them into national skeleton network.

On one hand, the number of sub-networks is likely to be greater than the number of operators' teams which does not facilitate already stressful conditions of the control in restoration. In addition, the elementary sub-networks have quite low frequency and voltage control capacity, which decreases the operators' control margin. Finally, the more sub-networks there are, the higher is the probability of coupling problems.

On the other hand, the time necessary for the supply of the nuclear units' auxiliaries is quite long (about 20 minutes) because of the complexity of manoeuvres in the production site that are necessary to couple the auxiliaries to the network. In addition, the current strategy prohibits any other manoeuvre in the sub-network during the application of scenarios for supply of auxiliaries. As reported in [EDF93a], the application of the scenarios takes about 77% of the time necessary to create a national skeleton network, while energising the lines and load pickups take 18% and 5%, respectively.

2.4.4.2 Improvements to present strategy: Regional skeleton networks

Because of the inconvenience cited above, several studies were carried out by EDF in order to find a way to improve the current restoration strategy. The first studies examined the possibility of immediately energising a compensated 400kV network of about 2'000km of lines, by coupling a single black-start unit on it. It was shown that the network thus created would have very bad performance from the point of view of voltage stability and frequency, and that it could not cope with such contingencies as a unit shutdown or a line fault.

The following studies [EDF92, EDF93b, EDF93c] were oriented towards an approach combining some elements of the current strategy and the compensated 400kV network strategy. The strategy proposed by this approach would be first to energise *radial 400kV skeleton networks,* each of which would be under the responsibility of an RCC, and then to couple them in order to obtain a secure national skeleton network. The regional 400kV skeleton networks should be defined in advance in such a manner that the most important production units at the 400kV voltage level (mostly nuclear units) be connected by single circuits.

The feasibility of this new strategy has been tested for two regional skeleton networks and the results reported in [EDF94b] show important improvements in time and operational organisation compared to those of the present strategy. **Therefore, the strategy for the first restoration stage which will be considered in this work is the improved present strategy, based on regional 400kV skeleton networks (RSN), described in [EDF94b]**.

The applicability of this strategy depends on the existence of at least one TS1 unit in the structure defined as the RSN to be energised. If there is no such unit, the present strategy (based on scenarios for the nuclear units' auxiliaries supply, i.e., FIDEOs) is to be applied.

Following are the principal rules which generally describe the adopted strategy for a single RSN creation that is to be applied by the operators of the RCC on the territory of whom the RSN is located geographically:

1) If there are no TS1 units on the RSN, apply the current, FIDEO-based strategy; otherwise, apply the new, RSN-based strategy;
2) Decrease the voltage set-points of production units;
3) Connect all available inductive shunt compensators;
4) If there is a strong network close to the RSN, first energise paths toward the TS1 units and then synchronise them with the strong network;
5) (a) If there are both a TS2 unit and a TS1 unit on a production site, supply the auxiliaries of the TS2 unit (i.e., synchronise the TS2 unit);

   (b) If there is a TS2 unit on the production site without a TS1 unit, energise a path from some TS1 unit towards the auxiliaries of the TS2 unit and supply them (i.e., synchronise the TS2 unit);

   (c) As soon as a TS2 unit is synchronised, it is considered as a TS1 unit;
6) If there is a shutdown thermal unit, energise a path from some TS1 unit towards the auxiliaries of the shutdown unit;
7) If for either 5) or 6) the operators find that there is a scenario for the auxiliaries' supply which provides a faster and more secure result, apply the scenario available on the FIDEO sheet;

8) If during the integration of RSN (steps 5, 6 and/or 7) there is the risk of over-voltages, pick up some load in order to avoid them;
9) If some load is picked up (step 8), increase the power set-point of a synchronised thermal unit in order to keep the frequency within the permitted limits. By priority, chose a coupled unit which has not yet reached its technical minimum production;
10) As soon as the paths towards the auxiliaries of the shutdown units on the RSN are energised, couple the auxiliaries simultaneously on all production sites with shutdown units. Any load pickup during this coupling is prohibited. The shutdown units then become stand-by, and could be synchronised as soon as their thermal circuits' conditions permit the secure synchronisation;
11) A RSN is said to be created as soon as the auxiliaries of all shutdown thermal units belonging to the RSN are coupled (i.e., step 10 is achieved).

During the creation of the RSNs, there is little communication between the NCC on one side and the RCCs on the other. The creation of a RSN is the responsibility of the corresponding RCC. However, the NCC may ask an RCC to execute certain actions in order to help another RCC. For example, if there is a TS2 unit whose auxiliaries can not be quickly supplied from a TS1 unit belonging to the same RSN, the NCC may ask a neighbouring RCC to energise the path towards the TS2 unit.

When two neighbouring RSN are restored, they may be synchronised. The objective of the RSN-based strategy is the same as that of the FIDEO-based strategy - an energised national skeleton structure - but the manner of obtaining it is different.

The differences between the RSN-based strategy and the present strategy are as follows:

- The scenarios for the supply of the thermal units' auxiliaries are no longer the basis of the strategy. However, they may be applied if the operators estimate that the result will be reached faster and in a more secure manner;
- The problem of voltage control is partially solved by decreasing the voltage set-point of units before starting the restoration of RSNs. Loads are picked up only if the voltage in the RSN cannot be decreased by connecting the inductive shunt elements. The consequence is that an RSN is less loaded than an elementary sub-network (from the current strategy), so that coupling with another RSN is easier because of the smaller voltage angle difference;
- Quickly reaching the technical minimum production of the thermal units is no longer a priority, since it is admitted that they can operate securely for a certain time with the lower power set-point. However, if it is necessary to maintain the balance with the picked up load, these units are the priority candidates for increasing their power set-point;
- The auxiliaries of the shutdown thermal units belonging to an RSN are coupled *simultaneously* as soon as the paths to all of them are energised. This measure results in important decrease in time compared to the time necessary to couple auxiliaries one by one, as in the current strategy;
- The number of the RSNs corresponds to the number of RCCs, i.e., each operating team is responsible for a single sub-network. In addition, the number of RSNs is smaller than the

number of elementary sub-networks (from the current strategy), so that the probability of synchronisation problems is decreased;

- All the connections inside the RSNs as well as inside the national skeleton network are single circuits, in order to avoid the excessive reactive power generation in the lightly loaded network. The double circuits will be energised later.

The guidelines presented in this section represent the strategy based on the creation of regional skeleton networks. This strategy, resulting from the studies done by the EDF's engineers, has been implemented in the form of rules in DAFFOR. The criteria for the choice of a black-start unit or the unit that should have the power set-point increased, the choice of a path to be energised, etc., will be described in detail in Section 4.2.3.2.

2.4.5 Second restoration stage: Load pickup

When most production units are coupled and connected in a skeleton network, there is little load that is supplied. In addition, this network is not robust enough because the double circuits are still open and any contingency might lead to the disintegration of the system. Therefore, the second restoration stage consists mainly in strengthening the network and picking up the loads until the complete restoration of service in the power system.

After reviewing the large amount of literature concerning service restoration at the transmission level, we have concluded that the load pickup sub-problem has not received much attention. Although most of the papers in this area briefly discuss it, no general strategies or procedures are widely accepted (or at least they are not reported). It seems that electrical utilities consider the load pickup stage as not very critical, and the operators in control centres are left to themselves to make decisions about control actions according to their own experience.

Because of the small probability that a total blackout occurs, the operators are likely to reason guided by intuition and the experience acquired in operating the system in its normal state. This reasoning may lead to control actions that put in question the whole restoration reached up to the moment (e.g. an overly fast load pickup). There is thus a need for a better determined procedure for the load pickup stage.

Since EDF has not defined a strategy for the second restoration stage, one of the objectives of this thesis was to propose one. For this purpose, two different approaches were studied: a Genetic algorithm-based approach and a Rule-based approach.

2.4.5.1 Load pickup strategy: Genetic algorithm-based approach

As there is little field experience in restoration and in particular in the load pickup sub-problem, the first idea was to:

- try to automatically generate different sequences of control actions for the load pickup stage, from different initial power system states resulting after the first restoration stage, and

- analyse these sequences in order to deduce rules which could represent a strategy for the load pickup restoration stage in a given power system.

Determination of the acceptable control action sequence that leads to the load pickup in the second restoration stage is a highly combinatorial multi-constraint optimisation problem. Genetic Algorithms (GAs) have been shown to be a quite successful technique for solving this kind of problem. Therefore, an approach based on a GA has been used in order to generate the *optimised load pickup sequences* for different initial states of the power system, with respect to operating limits.

The first results have shown that optimised sequences obtained with the proposed GA-based method were satisfactory, but that it is not easy to **quickly** deduce simple rules from the obtained optimised sequences. Therefore, the studies on the GA-based approach had to be postponed because of the lack of time. Nevertheless, this approach is, to our knowledge, an original one, and the work done up to now is thus presented in this thesis. Chapter 6 concentrates entirely on this approach, where promising results will be shown, the directions for the future research will be enumerated and the possibility of its integration in DAFFOR will be discussed.

2.4.5.2 Load pickup strategy: Rule-based approach

The skeleton network which results from the first restoration stage is lightly loaded, little meshed, with voltages close to the upper limit values and with no double circuits energised. The objective of the restoration is to supply the loads as soon as possible and as securely as possible. Considering the (n–1) security of the power system during the restoration, at least until most of the service is restored, makes little sense. Therefore the expression "as securely as possible" simply means that the restoration control actions should be chosen so that the power system variables are kept "far away" from their limit values.

At the very beginning of the load pickup restoration stage, there are several contradictions between the mentioned objectives (i.e., speed of load pickup vs. security as defined above):

- since the voltages in the skeleton network are likely to be close to the upper limit values, energising the unloaded lines (to extend the network) or closing lines among the energised substations (to strengthen the network) may cause over-voltages;
- the small number of production units that are coupled in the skeleton network implies small inertia of masses, and an overly fast load pickup might cause stability problems because of the prime movers' response;
- if the skeleton network has not been strengthened along with load pickups, the overloads may appear on the transmission devices.

In addition to control actions which physically restore the service in the skeleton network (load pickups and an extension and strengthening of the network), there are actions which must be executed in order to keep the frequency and the voltages inside the operating limits. For example, since the secondary frequency/power (f/P) control is disabled during the restoration, this function must be provided "manually", by the operators, in order to keep a balance between the production and the consumption.

As the restoration procedure progresses, certain thermal units that were shutdown will become available for coupling to the network. This coupling should be done as soon as the thermal circuits of units permit it, which will increase the robustness of the network from the point of view of both frequency and voltage conditions.

As a result of the above considerations and numerous simulations, we have proposed a strategy to apply to the French power system during the load pickup restoration stage. This strategy has been implemented in DAFFOR in the form of rules, and will be presented and justified in Sections 4.2.3.3 and 4.2.3.4.3, respectively.

Chapter 3

DAFFOR - Functional description

In the previous chapter, an attempt was made to present different issues concerning the transmission power systems' restoration. The objective was to show: (1) how restoring the service in a power system after a major disturbance is complex, and (2) what are the difficulties the operators in the control centres must cope with in real time.

The objective problems in restoration that were enumerated concern the nature of the power system components and their interactions. These issues were discussed first in general, and then for the case of the French power system in particular. To these network-related factors must be added such subjective factors as:

- operators' preparedness for the control in restoration (the level of familiarity with the restoration plan, restoration training experience and, maybe, experience from a past blackout),
- stressful situation imposed by the restoration objective itself (the load must be restored as soon as possible with respect to the operating constraints), and
- personal ability to handle the unusual quantity of alarms, client calls etc. in parallel with decisions about the control actions.

The power system contains many different elements that can be operated (production units, loads, transformers' taps, switching devices). For some of them, only one control action at a time is possible (e.g., switching devices), while for the others there are several actions (e.g., production units). In addition, the restoration paths are numerous because of the meshed structure of the transmission power system. These facts result in the combinatorial nature of the restoration process, which raises the difficulty of choosing the right restoration action at the right moment.

Finally, the restoration procedure is carried out in real time. Even if a good restoration plan has been defined, with a quite exhaustive set of possible initial conditions, there might always be an unforeseen event during the restoration which brings into question the further application of the restoration plan.

All these points lead to the conclusion that an on-line guidance tool, capable of adaptation to the real-time evolution of the power system, would be highly useful to the operators in control centres, as assistance during the restoration. In the discussion about the decision assistance tools, Shahidehpour and Kirschen wrote in [IEE92b]: "... Like all good assistants, they should require as little instructions as possible, stay out of the way of the operator unless called upon, and last but not least, leave the final decision to the operator".

The Decision Aid Function FOr Restoration (DAFFOR), proposed in this work, is conceived with the aim of satisfying these attributes. The functional requirements DAFFOR should meet are enumerated in Section 3.1, and according to them, the proposed architecture is presented in Section 3.2.

3.1 Functional requirements

3.1.1 Guidance and interactivity

During the restoration, the operators reason in terms of control actions. At each step, they try to choose the action which, according to the restoration strategy and their experience, fits best the current state of the power system and which would not result in operating limit violations. Therefore, DAFFOR must be capable of:

- suggesting one control action at a time, if asked by the operators,
- assessing a control action suggested by the operators and answering if it is appropriate for the current power system state, and
- validating and proving that the suggested / assessed control action will not cause any violation of the operating limits if executed on the power system.

Finally, in order for the operators to believe in their "computer assistant", the latter one must be capable of explaining any result it generates.

3.1.2 Real-time operation

The restoration procedure is carried out in real time. As any system controlled in real time, the power system may not respond as expected to the control actions, all the more so since the system is in an extremely unusual state. The guidance tool, supposed to run in real time, must be "informed" about the current power system state at any moment. Therefore, DAFFOR must be capable of:

- taking into account the evolution of the power system variables,
- identifying the unforeseen events as they occur during the restoration process,
- providing an answer according to the current state of the power system, and
- giving the answer quickly.

3.1.3 Use in the real environment

As all functions supposed to run in the energy management system (EMS) environment, DAFFOR must pass through several development stages: (1) development of a stand-alone prototype, (2) coupling with a power system dynamic simulator, (3) integration within the operator training simulator (OTS) environment (if there is one), and (4) full integration into the EMS environment.

Having this in mind, it is obvious that the stand-alone prototype must already be conceived in the manner which will guarantee the following:

- easy coupling with a power system simulator, for first validations,
- easy implementation in a real environment (OTS and EMS), for further validations and real-time use, and
- suitability for the actual working habits of the operators, for real-time use.

3.2 Proposed architecture

According to the above objectives, DAFFOR is structured as shown in Figure 3.1. DAFFOR is divided into two principal units which exchange data: the Reasoning kernel is responsible for assistance and interaction with the operators, while the Real Time Update (RTUpd) kernel provides a coherent picture of the real-time state of the power system.

Man Machine Interface, equally represented in Figure 3.1 is out of the scope of this work. However, it is important to note the clear decoupling between DAFFOR's interfaces: on one hand, there is a data interface with EMS, and on the other, the user interface. This makes possible the easy coupling with external functions - the only adjustments to be done are at the level of DAFFOR's interface variables.

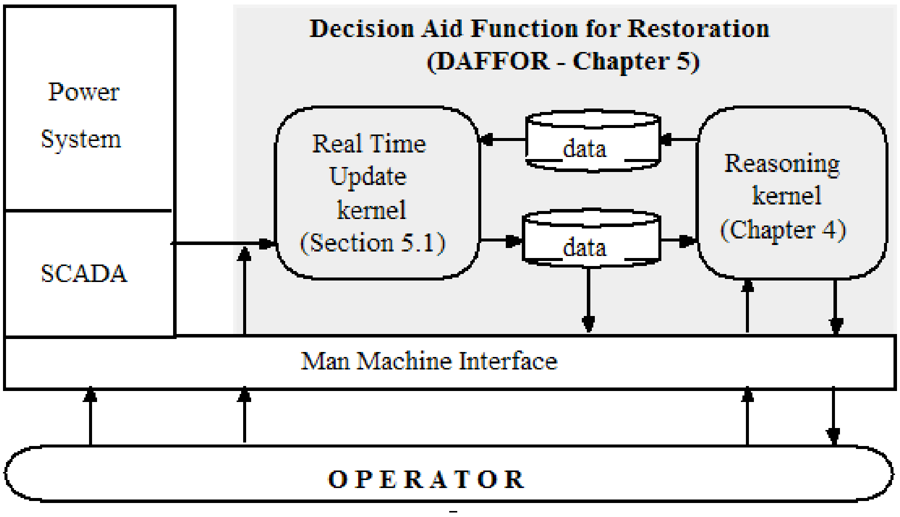


Figure 3.1 Overall structure of DAFFOR integrated in a control centre

3.2.1 Reasoning kernel

The Reasoning kernel guides the restoration process in real time by suggesting one control action at a time. The action is suggested according to data generated by the RTUpd kernel and the restoration strategy implemented in the Reasoning kernel itself. It can also assess a control action proposed by the operators. A combined knowledge-based and procedural approach was chosen for these purposes.

Chapter 4 is dedicated to the Reasoning kernel which may be considered as the stand-alone prototype of DAFFOR (first development stage mentioned in Section 3.1.3).

3.2.2 Real time update (RTUpd) kernel

The Real time update kernel is DAFFOR's real-time data interface. It steadily receives data coming from different sources, performs comparisons among them and generates a coherent power system state description at a given moment. This state description is the only real-time information made available to the Reasoning kernel.

Chapter 5 introduces first the RTUpd kernel, continues with the description of the real-time operation of DAFFOR, and concludes with the validation architecture (second development stage mentioned in Section 3.1.3).

Chapter 4

Reasoning kernel

At the beginning of this chapter, let us recall that the restoration procedure is carried out by the operators in control centres. They decide in real-time about the control actions that should be applied to the power system, and execute them as long as the service in the system is not completely restored (i.e., all the consumers are supplied). The function of DAFFOR is to be the operators' decision support in restoration. The subject of this chapter is DAFFOR's interactive assistance part: the Reasoning kernel.

The Reasoning kernel is the part of DAFFOR that received particular attention during the design and the development, since it represents the operators' assistant. It can be considered as an **off-line or stand-alone prototype** of DAFFOR, and can be run without real-time data (see the remarks on the development stages in Section 3.1.3). In this chapter the stand-alone version of DAFFOR is described in detail. The real-time extensions are the subject of Chapter 5.

Before entering into a description of the Reasoning kernel itself, the factors that have been taken into account for its design will first be explained.

Consider a power system which has suffered a major disturbance. Suppose that the protection devices have operated and that the system is defined by an initial steady state $s(0)$. The first executed restoration action, denoted as a_1, will naturally produce a dynamic response of the system, and the system should stabilise in a steady state denoted as $s(a_1)$. The dynamic response of the system, as well as the resulting steady state, should not have any violation of operating limits (e.g., over-voltage). Therefore, action a_1 should be chosen as a function of the initial state $s(0)$ in such a manner that the transition of the power system to the resulting state $s(a_1)$ be as smooth as possible. This last state is then considered as an initial one for the next restoration action, a_2, which brings the system into state $s(a_2)$, and so forth until the power system is restored.

If the steady state of the completely restored system is denoted by $s(a_{Nall\_act})$, the restoration procedure may be seen as the transition of the power system from states $[s(a_0)\ldots s(a_{r-1})\ldots s(a_{Nall\_act-1})]$ to states $[s(a_1)\ldots s(a_r)\ldots s(a_{Nall\_act})]$ due to control actions $[a_1\ldots a_r\ldots a_{Nall\_act}]$, respectively. The restoration may be represented by the following system of transition expressions:

$$a_1 = f(s(0)) \rightarrow s(a_1)$$

$$a_2 = f(s(a_1)) \rightarrow s(a_2)$$

$$\ldots$$

$$a_{Nall\_act} = f(s(a_{Nall\_act-1})) \rightarrow s(a_{Nall\_act})$$

which can be generalised thus:

$$a_i = f(s(a_{i-1})) \rightarrow s(a_i),\ \mathrm{i}=(1, 2,...,r,\ldots,Nall\_act). \qquad (4.1)$$

The transition expression given by (4.1) implies that action a_r, which is to be applied to the system whose current state is $s(a_{r-1})$, should be chosen in such a manner that neither power system response nor resulting steady state $s(a_r)$ have any violation of operating limits. Since the restoration objective is to bring the power system into the steady state $s(a_{Nall\_act})$, in which the complete service is restored, it is necessary to find a complete sequence of N_{all_act} control actions defined thus:

$$Seq(1, N_{all\_act}) = [a_1\ldots a_r\ldots a_{Nall\_act}] \qquad (4.2)$$

At each stage of the restoration, the operators first consider the current power system state and try to identify possible restoration actions in that moment with respect to the operating limits. Then they choose the most appropriate one, according to their experience and the defined restoration strategy, and decide whether to apply it to the power system.

As the operators' decision support tool, DAFFOR should be capable of "reasoning" in a similar manner, i.e., suggesting one control action at a time (as opposed to suggesting several actions and leaving the choice to the operators). This fact depicts a double concern:

- providing the operators with a suggestion which fits their own manner of operating the power system under restoration (action by action), and
- making the DAFFOR capable of following and adapting to the dynamic evolution of the power system (state by state).

The DAFFOR's Reasoning kernel emulates the operators' way of thinking. Given the adopted step-by-step operation and the interactive-advisory function, the Reasoning kernel can perform the following functions for step *r*:

1) find action a_r according to the restoration strategy, taking into account the current power system state $s(a_{r-1})$, and providing that operating limits will be respected if the action is performed on the system;
2) suggest action a_r to the operators;
3) test action a_r suggested by the operators;
4) suggest new action a_r or test another action a_r, if asked by the operators;
5) start the search for the next action a_{r+1} (i.e., move to the next step r+1) only after the operators accept to execute the action a_r on the power system.

The Reasoning kernel operates in either search or validation mode. In search mode, its function is to suggest a control action to the operators (point 2), and in validation mode it is to assess a control action suggested by the operators (point 3). The output control action, whether found by the Reasoning kernel, or suggested by the operators and validated by the Reasoning kernel, is proven to not result in any violation of operating limits if executed on the power system (point 1). In addition, the Reasoning kernel is capable of suggesting or testing new control actions if the operators are not satisfied with the current result (point 4).

Note that the operations described in points 1 through 4 concern action a_r: it will be searched or tested with respect to the current system state $s(a_{r-1})$. In fact, the Reasoning kernel performs in real time all or some of these operations, following the operators' requests. As long as the operators do not decide to execute action a_r on the power system (point 5), the Reasoning kernel assumes that the current system state $s(a_{r-1})$ does not change[9]. Therefore, the symbol for the action it operates on (search or testing) keeps the same index, *r*.

The Reasoning kernel consists of three modules, as shown in Figure 4.1:

- The Dialogue&control (D&C) module represents the Reasoning kernel's (and DAFFOR's) user interface. It offers to the user (step 1) possibilities of choosing the operation mode (search or validation), of accepting or rejecting the suggested action, of asking for the explanation and of ending the session. Depending on the user's choice (step 2), the D&C module activates the corresponding reasoning task (step 3) and presents the operators with the result of the overall reasoning (step 6).
- The Knowledge base (KB) module contains rules and facts, describing the restoration strategy and the manner in which it is applied to the power system. According to them, the KB module searches for a control action (e.g., change of operating status for lines; change of power and voltage set-point for units; pick up of a load, etc.). Each control action (either found during the search in the KB module in search mode or suggested by the operators in validation mode) is tested within the Dynamic simulator (step 4).

[9] It will be shown in Chapter 5 how the Reasoning kernel takes into consideration the changes in the current system state.

- The Dynamic simulator generates the response of the power system (step 5) to a control action (step 4) in terms of the evolution of frequency, voltages, production level, power flows, losses etc., as well as the statistics about the violation of operating limits: overload, over-voltage, under-voltage and frequency excursions. It also contains a number of tests which filter the control actions that can be executed at a given moment from the set of all possible control actions. These tests are either deterministic (e.g., if a line is connected, the action that would try to connect it again is disabled) or heuristic (e.g., the maximum quantity of a load that can be picked up at a given moment is defined with respect to the current spinning reserve).

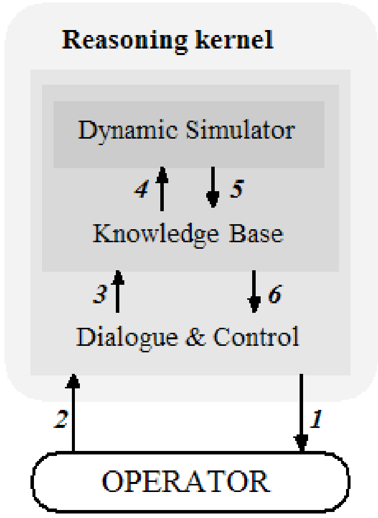


Figure 4.1 Reasoning kernel, the interactive advisory part of DAFFOR

The above brief introduction to the Reasoning kernel and its modules gives a general idea of the Reasoning kernel's functionality. To better understand the role of each element and the interactions between them, the discussion that follows has a bottom-up structure. We start with the description of elementary functions, and continue gradually by adding "building blocks" around them. Section 4.1 thus introduces the Dynamic simulator, which represents the procedural (numerical) part of the Reasoning kernel. It models the power system elements, dynamic behaviour and operating limits, and is used to test a single control action, given the current power system state.

Section 4.2 deals with the Knowledge base module, which is the symbolic part of the Reasoning kernel (and DAFFOR), containing the strategy for the two restoration stages discussed in Sections 2.4.4.2 and 2.4.5.2. The strategy is implemented in terms of facts and rules. The Knowledge based module performs a heuristic search for a control action, given the current power system state. It uses the Dynamic simulator to test the action it finds before suggesting it to the operators.

The Dialogue&control module is presented in Section 4.3. It reflects the intent to provide the operators with one control action at a time, which guarantees a viable service restoration. The operation of this module is independent of the adopted restoration strategy and power system in question - it can be regarded as an operating environment for the other modules of the Reasoning kernel.

Finally, the global functionality of the Reasoning kernel is summed up in Section 4.4. Some restoration scenarios for the French power system and the New England test network on which the stand-alone DAFFOR (i.e., the Reasoning kernel) has been tested will be presented and discussed.

Note that steps 1 through 6 from Figure 4.1 are consistently kept in all the figures representing DAFFOR's modules, with the following meaning:

- steps 1 and 2 represent the DAFFOR's user interaction facility,
- steps 3 and 6 perform the DAFFOR's internal task scheduling, and
- steps 4 and 5 are the coupling facility between the DAFFOR's numerical and symbolic parts.

4.1 Dynamic simulator (DS)

The Dynamic simulator (DS) is an ANSI C [Dar91] computer programme which has been developed to facilitate the analysis of a transmission power system in different operating states, and in particular during service restoration [Kos94]. It can simulate the response of the whole power system, or its islands, for different control actions.

The DS may be run either as an independent interactive program (waiting for the user to enter control actions one by one), or as a function that can be called from another programme (input being generated by the caller, and output being returned to the caller).

In the Reasoning kernel, the DS is called as function in order to simulate a single control action at a time, as shown in Figure 4.2. Referring to the transition expression (4.1), relating a control action to power system states before and after its execution, the DS operates as follows:

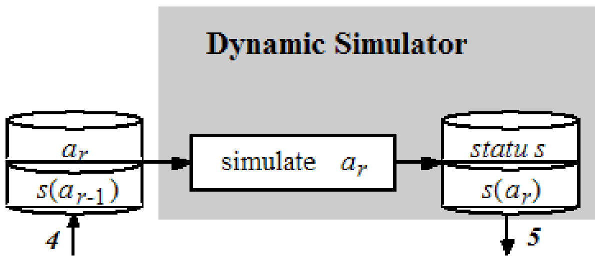


Figure 4.2 Operation of the Dynamic simulator as function

- it takes as input control action a_r and the current power system state $s(a_{r-1})$ (step 4),
- it simulates the execution of action a_r, and
- it outputs (step 5) the power system state $s(a_r)$, expected after action a_r, and the status of the simulation flow and result, *status* (e.g., whether the resulting state is steady, whether there have been some operating limits violations, etc.).

Since it is necessary to simulate many different actions, the programme speed has been of main concern. That is why, contrary to usual simulators with detailed models for all network components, only the necessary minimum was implemented in the DS. The mechanical aspect of production units (prime mover, speed governor) is simulated *dynamically* thanks to a long-term dynamics model. The electrical model of production units (generator, exciter, exciter governor) has not been implemented, but the steady state values of system voltages and reactive power of units are yielded *statically*, in discrete time intervals, from the power flow computation. Thanks to decoupled modelling of mechanical and electrical aspects of the power system, a great amount of processing time is saved during the simulation, while keeping the satisfactory precision.

Power system elements represented in the DS are grouped in the following classes of elements:

- a unit may be thermal (classical and nuclear), hydraulic or equivalent,
- a load may be consumer load, thermal unit's auxiliary or shunt compensation device (inductive or capacitive), and
- a transmission element may be a line, a transformer with tap changer under load (TCUL), a step-up transformer or an autotransformer.

The mathematical and topology-related models for the above classes of elements have been chosen as a function of two parameters: (1) available data (provided by EDF), and (2) application of the DS (restoration in real time). First an overview of the models will be given, followed by the overview of DS commands which correspond to restoration control actions. After these basic concepts are introduced, the manner in which DS simulates actions taking into account the real-time operation will be described. Finally, the DS outputs will be discussed.

4.1.1 Mathematical models

From the dynamic point of view, the power system is represented with the long-term dynamics model where all the production units are connected to a single node, i.e., there is a single average frequency common to all electrically connected nodes. The units are represented by turbine and governor models. The DS simulates, in fact, the automatic, primary frequency control.

Since the electrical aspect of production units has not been implemented, the voltages and the reactive production or absorption of units are computed thanks to the fast decoupled power flow model every time the power system has reached steady state. The production (PV) nodes are characterised with a voltage set-point and reactive power limit values. Therefore, if the reactive power limit of a PV node is violated, it will be considered as a load (PQ) node as long as the violation is not eliminated. The power flow implemented in the DS contains also the models of TCUL transformers. The tap position may be fixed to a given value, but there is also a heuristic method for automatic tap position change. Finally, transmission lines and transformers are represented with the equivalent symmetrical π-scheme.

This is a brief overview of the basic models the DS contains. The complete description of the above models and their implementation in the DS can be found in [Kos94]. The rest of this section focuses on new added models closely related to the restoration problem.

4.1.1.1 Production units

Consider a thermal unit whose power set-point at moment t_0 equals P_0. When the power set-point is changed to P_{set}, the desired power output is not reached immediately, but following a certain gradient of power in time, i.e., by time t_{set}, as shown in Figure 4.3. The loading gradient of the thermal unit is known as the percentage x of the unit's rated power per minute:

$$grad = x \times P_{nom} \text{ [MW/min]}. \tag{4.3}$$

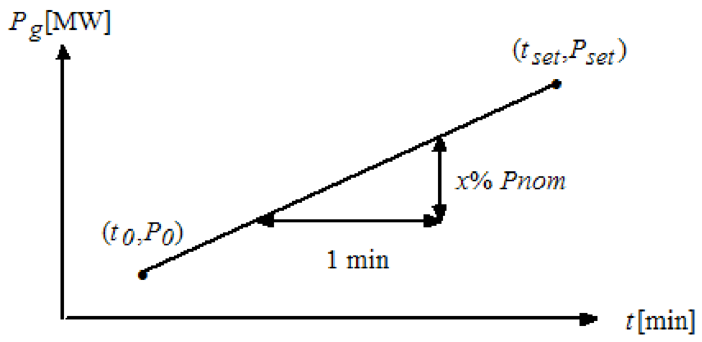


Figure 4.3 Loading of a thermal unit

In the DS, the loading gradient (4.3) is expressed as the power increase ΔP per second:

$$\Delta P = (1/60) \times x \times P_{nom} \text{ [MW]}. \tag{4.4}$$

Therefore, when simulating the loading of a thermal unit in the DS, the unit's current power set-point P_0 is increased by ΔP every second, as long as the new set-point P_{set} is not reached. Since the usual loading gradient of a thermal unit is 5% of the unit's rated power per minute, x=5 in equations (4.3) and (4.4).

The hydraulic units can be loaded very quickly from 0 to their rated power output P_{nom} (order of a minute). Therefore, in the DS, x=100 in equations (4.3) and (4.4).

There is no model for the loading of equivalent units implemented in the DS since they represent the production of the external network and can not be operated intentionally during the simulation. Depending on the restoration scenario (with or without external support), these units are either kept in the same state as before the blackout or are not considered at all. In any case, their power set-point never changes and it is consequently not necessary to consider the loading.

In the DS, the possibility of modifying the unit's power set-point is implemented in order to remedy the absence of the automatic secondary frequency/power (f/P) control which is locked in restoration. Modification of the unit's power set-point in the DS translates into moving the static characteristic of the unit's speed governor in the plane P–f along the frequency axis.

Remember that the DS simulates primary frequency control, which is active even in restoration. However, it may be desirable to temporarily lock the speed governors of some units (i.e., force the units to output constant power, independently of the frequency fluctuations). To take into account the operation mode of units from the point of view of frequency control, the DS associates one of the following three values for control status of each production unit:

1) primary frequency control locked (i.e., constant output power equals the set-point value P_{set}) - this applies to all but equivalent units,
2) primary frequency control enabled (i.e., variable output power), and
3) unit under loading (i.e., variable output power between P_0 and P_{set}, as shown in Figure 4.3) - this applies to all but equivalent units.

4.1.1.2 Loads

In the power system, most consumer loads are connected at distribution level (e.g., 6kV). Several distribution loads can then be grouped and represented by a single load at the transmission level (e.g., 63kV). In order not to lose the information about the load shedding/supply priority, a load P_{l_init} represented in the transmission network is usually divided into a number N_{shed} of portions p_i such that:

$$\sum p_i = 1,\ i=1,\ldots,N_{shed}. \tag{4.5}$$

The portions (4.5), the number of portions N_{shed} and the initial load P_{l_init} are used in the DS to define three kinds of levels for a consumer load, which are all shown in Figure 4.4 for N_{shed}=5.

4.1.1.2.1 Shedding levels

The first shedding level $P_{shed}(1)$ is that part of the load which has the lowest supply priority. Therefore, if for any reason some load is to be shed, the load $P_{shed}(1)$ will be disconnected first. If this is not enough, second shedding level $P_{shed}(2)$ will be disconnected, and so on. The portion of the load which has the highest supply priority, (P_{l_init}–$P_{shed}(4)$), will be shed the last.

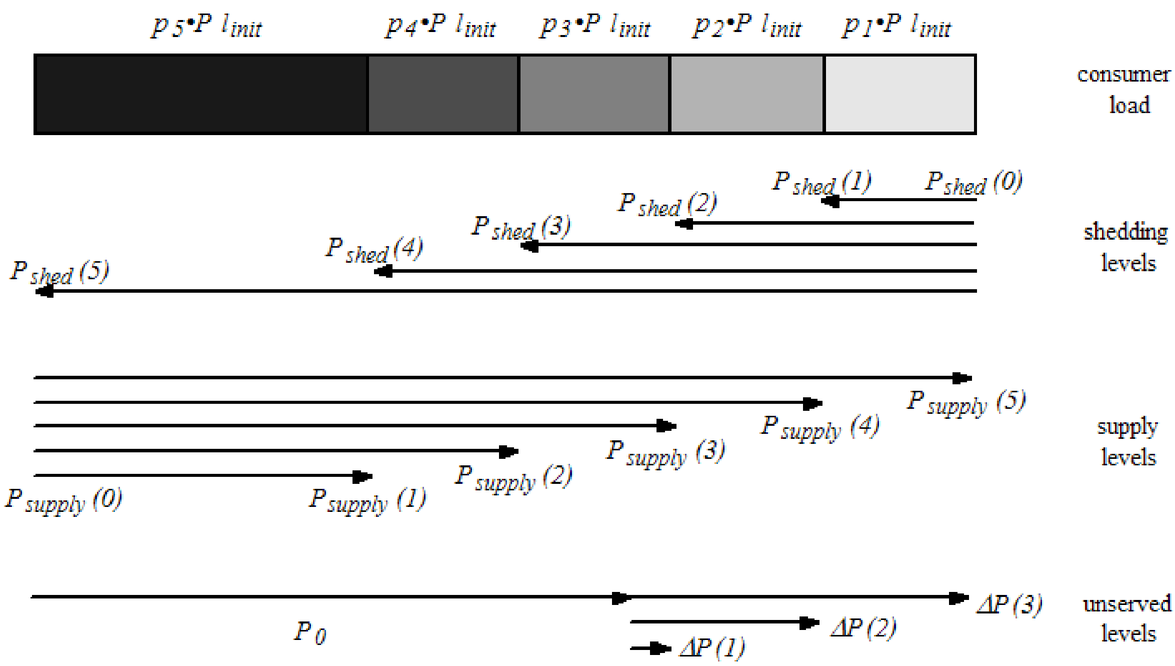


Figure 4.4 Consumer load divided into N_{shed}=5 portions and the corresponding shedding, supply and unserved levels

The shedding levels are defined thus:

$$P_{shed}(m) = 0,\ m=0$$

$$P_{shed}(m) = P_{l\_init} \times \sum P_i\ ,\ i=1,\ldots m,\ m=1,\ldots,N_{shed}. \tag{4.6}$$

4.1.1.2.2 Supply levels

In the case of restoration after a major disturbance, most consumers are unserved and should be resupplied as soon as possible. Therefore, when a consumer load is to be picked up, the first attempt will be to supply it completely, i.e., $P_{supply}(5)=P_{l\_init}$. If this is not possible for some reason, the next try will be to pick up the second supply level $P_{supply}(4)=(P_{l\_init}–P_{shed}(1))$, and so on. Note that the order of supply is inversed order of shedding. The supply levels are defined as a function of the shedding levels (4.6) as follows:

$$P_{supply}(m) = P_{l\_init} - P_{shed}(N_{shed} - m),\ m=0,\ldots,N_{shed}. \tag{4.7}$$

4.1.1.2.3 Unserved levels

Suppose that portion P_0 of the load has already been supplied, which is between supply levels 2 and 3, i.e., $P_{supply}(2)\leq P_0\leq P_{supply}(3)$. From Figure 4.4 it can be seen that there are 3 unserved levels left. When this load is to be picked up, the first attempt will be to supply the highest unserved level $\Delta P(3)$. If it is not possible for some reason, the next try will be to pick up one level less, $\Delta P(2)$, and so on. The unserved levels are defined as a function of the already supplied portion of load P_0 as follows:

$$P_{supply}(m) \leq P_0 < P_{supply}(m+1),\ m=0,\ldots,N_{shed}-1$$

$$\Delta P(i) = P_{supply}(m+1) - P_0,\ i=1,\ldots,N_{shed}-m. \tag{4.8}$$

Note that the maximum value index i may take, $(N_{shed}–m)$, represents the number of unserved levels.

* * *

Consumer loads in the DS are represented by constant power models, (P_l+jQ_l). The load before the blackout, denoted as $(P_{l\_init}+jQ_{l\_init})$, is assumed to be known and will be called the initial load. After the blackout, a consumer load may be picked up entirely or in portions. In the latter case, reactive power Q_l of a consumer load is considered to be a linear function of its real power P_l. Therefore, when a portion ΔP of a consumer load is picked up during the restoration, the corresponding reactive load ΔQ is computed in the DS as:

$$\Delta Q = (Q_{l\_init}/P_{l\_init}) \times \Delta P,\ P_{l\_init}\neq 0. \tag{4.9}$$

If the portion of consumer load already supplied is (P_0+jQ_0), the consumer load model can then be written as:

$$P_l+jQ_l = (P_0 +\Delta P) + j(Q_0 + Q_{l\_init}/P_{l\_init}) \times \Delta P. \tag{4.10}$$

For loads other than consumers (i.e., shunt compensation devices and the thermal units' auxiliaries), the notion of shedding levels does not apply: they cannot be picked up in portions, only entirely. Their model therefore represents the special case of (4.10) and can be written:

$$(P_l,Q_l) = (P_{l\_init},Q_{l\_init}) \text{ if connected; } (0,0) \text{ if disconnected} \quad (4.11)$$

Note that (4.11) applies also to a consumer load if it is picked up entirely.

4.1.2 Topology considerations

The power system consists of such equipment as production units, transformers, overhead lines, compensation devices, etc. All of them are coupled in a certain manner inside the substations in order to supply the consumers, which are coupled to the power system at substation level too. Inside a substation there are busbar sections and switching devices: busbar couplers and isolators, circuit breakers and isolators of units, transformers etc. The operating status of switching devices (open/closed) defines the operating status of power system equipment, and consequently the topology of the power system at a given moment. Topological description of the system considered at the level of substations will be called **substation topology**.

For a number of power system analyses (e.g., security, power flow), a somewhat simplified representation of substations is sufficient. In fact, given the operating status of the busbar couplers and isolators, all the busbar sections that are connected electrically (i.e., through closed switching devices) can be considered as a single electrical node. Therefore, the power system may be represented by the graph whose nodes correspond with the electrical nodes at a given moment, and branches with the transmission equipment. The production units and the loads are represented as nodes' injections. This simplified representation of the power system will be called **nodal topology**.

As most power system analysis programmes, the DS has **nodal** image of the power system. Remember, however, that it is to be used for restoration in real time which implies (1) high computation speed, and (2) capability to handle the real-time operating status of switching devices (i.e., substation topology representation). In order to satisfy these requirements, the nodal models have been kept in the DS and a set of functions, which translate the substation topology into the nodal one, has been developed.

4.1.2.1 Conversion from substation to nodal topology

The conversion functions translate substation topology models, contained in a data file provided by EDF, into nodal topology models, implemented in the DS. The conversion starts with the identification of electrical nodes and continues with the definition of the nodes for branches and injections. It terminates with the determination of operating and availability status of branches and injections depending on the operating and availability status of switching devices associated to transmission equipment, and production units and loads, respectively.

4.1.2.1.1 Node identification: Substation→Node

Busbar sections in a substation are coupled through busbar couplers and isolators. Each busbar section is a potential electrical node. Therefore, the maximum number of nodes per substation equals the number of busbar sections in the substation.

To illustrate the identification of electrical nodes, refer to Figure 4.5 in which a sample network with two substations, S1 and S2, is shown. A consumer load is coupled in substation S1 by circuit breaker CB1, and a transformer to substations S1 and S2 by circuit breakers CB2 and CB3, respectively. Substation S1 has two busbar sections, BS1 and BS2, connected through busbar couplers BC1 and BC2, while substation S2 has single busbar section BS3. The maximum number of electrical nodes in each substation corresponds to the number of busbar sections it contains (i.e., two for S1, one for S2).

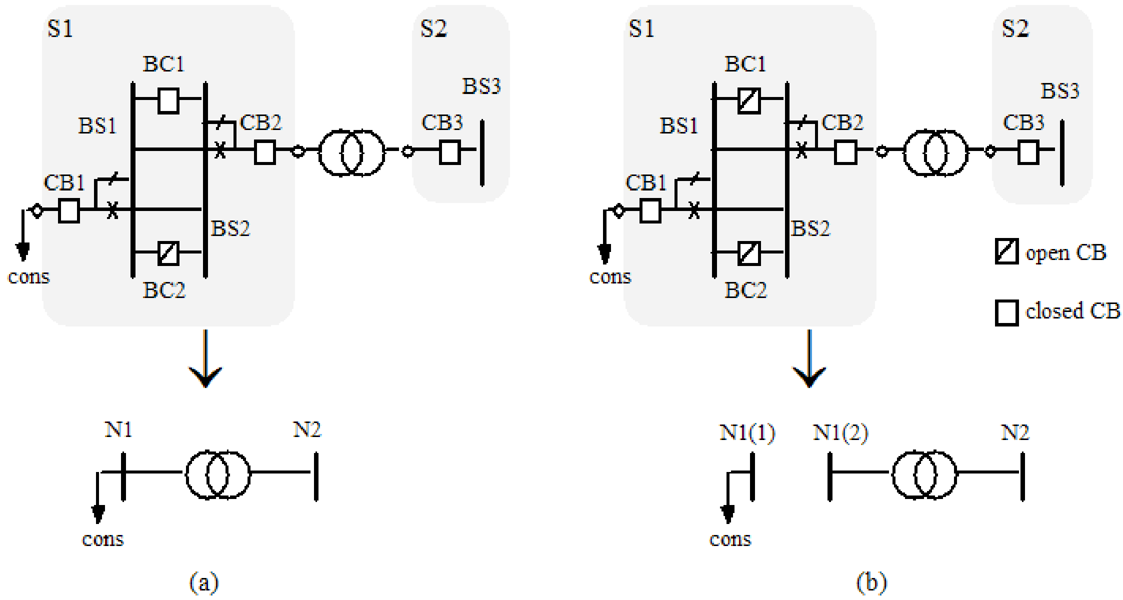


Figure 4.5 Conversion from substation to nodal topology model. S stands for substation, BS for busbar section, BC for busbar coupler, CB for circuit breaker, N for electrical node and "cons" for consumer load. Substation S1 may be replaced by (a) one electrical node, or (b) two electrical nodes.

To identify electrical nodes in a substation with more than one busbar section, it is necessary to consider the operating status of busbar couplers. For the example shown in Figure 4.5.a, busbar coupler BC1 is closed and busbar sections BS1 and BS2 are electrically connected. Therefore, the substation S1 and all its switching devices may be replaced by a single electrical node, N1. However, if both busbar couplers BC1 and BC2 are open, there are two electrical nodes, N1(1) and N1(2), as shown in Figure 4.5.b. Substation S2 is always replaced by a single node, N2, since it contains only the busbar section BS3.

4.1.2.1.2 Associating nodes to branches and injections

After the electrical nodes are identified, it is necessary to associate one node to each injection and two nodes to each branch. For the example given in Figure 4.5.a, the consumer load is associated with node N1, and the transformer with nodes N1 and N2. However, what happens if the secondary side circuit breaker of the transformer, CB3, is open? In this case, there is no electrical node that could be associated to the transformer on the secondary side.

In order to enable the node definition for the disconnected branch or injection (which represent the transmission equipment, or production unit or load, respectively, whose circuit breaker is open), it is necessary to introduce some fictitious elements, as shown in Figure 4.6.

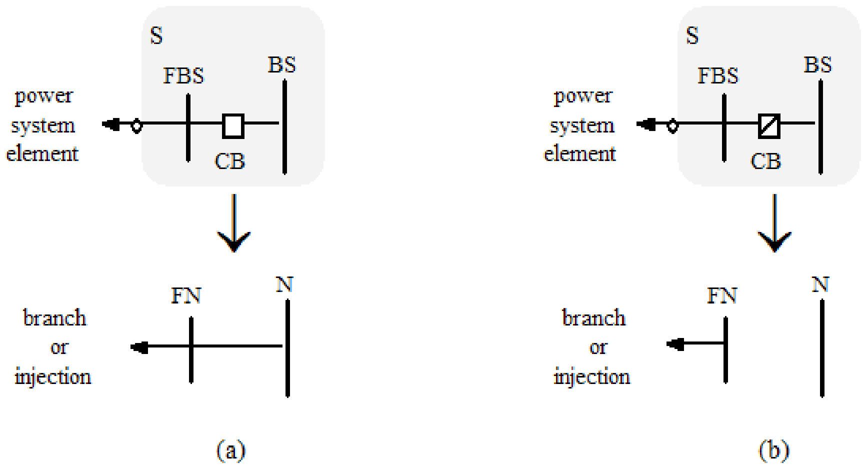


Figure 4.6 Introduction of fictitious elements in both topology models. Original models' elements are substation S, power system element (line, transformer, unit, load) and its circuit breaker CB, and electrical node N. Fictitious elements are busbar section FBS and node FN.

Introduction of a fictitious busbar section FBS between a power system element (line, transformer, unit, load) and its circuit breaker CB translates into the introduction of a fictitious node FN between the element model (branch, injection) and electrical node N. If the element's circuit breaker CB is closed (Figure 4.6.a), the fictitious node FN is electrically connected with node N so that the element can be associated with node N. On the contrary, if the element's circuit breaker CB is open (Figure 4.6.b), the fictitious node FN will be associated to the element.

This model applies to all circuit breakers associated to transmission equipment, production units and loads. It allows electrical nodes to always be associated with power system elements, independently of the operating status of their circuit breakers.

4.1.2.1.3 Determination of operating and availability status of branches and injections

The last step which the conversion functions perform is the determination of:

1) the operating status of branches and injections, based on the operating status of switching devices; for certain elements, it may have other values than on and off, and
2) the availability status of branches and injections, based on the availability status of both switching devices and power system elements; for all types of elements, it takes either true or false.

Follows the description of the last step done by topology conversion functions, by power system element type.

4.1.2.1.4 Production unit→Injection

There are four types of production units in the DS: classical thermal, nuclear thermal, hydraulic and equivalent units. The topology conversion models are shown in Figure 4.7.

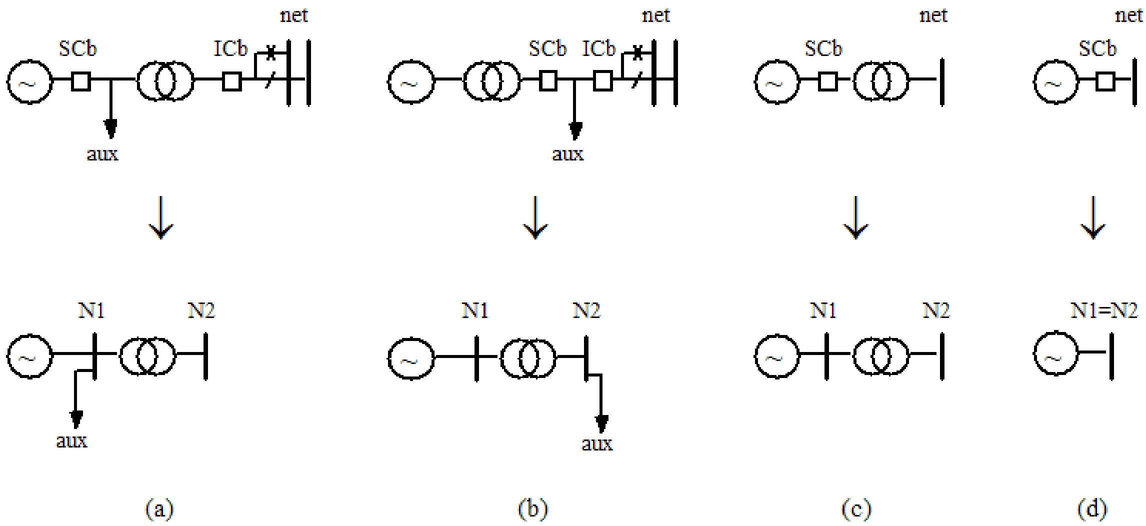


Figure 4.7 Conversion from substation to nodal topology for (a) classical thermal and certain nuclear units, (b) other nuclear units, (c) hydraulic units and (d) equivalent units. SCb stands for shutdown circuit breaker, ICb for isolating circuit breaker and "aux" for the thermal units' auxiliaries. "net" represents the substation where the unit is coupled to the network. N1 and N2 are electrical nodes.

Depending on the operating status of the shutdown circuit breaker (SCb) for all types of units and the isolating circuit breaker (ICb) for thermal units only, a unit may be:

1) coupled to the network - for thermal units, both SCb and ICb are closed, and for other types of units, SCb is closed;
2) shutdown - for thermal units, both SCb and ICb are open, and for other types of units, SCb is open;
3) stand-by - (thermal units only) the unit's auxiliaries are supplied from the network, i.e., SCb is open and ICb is closed (the thermal unit must pass through this operating state before its synchronisation to the network), and
4) isolated - (thermal units only) the unit is not connected to the network but it supplies its own auxiliaries, i.e., SCb is closed and ICb is open (the thermal unit becomes isolated after the successful load reject; refer to the discussion about load reject in Section 2.4.3).

The values that can be assigned to the operating status of the production units are summarised in Table 4.1.

A unit is said to be available[10] if:

it is not shutdown for maintenance purposes,

and its step-up transformer is not under maintenance,

[10] The availability status has a physical sense for all but equivalent units. However, It is also associated with the latter ones, which is useful when certain restoration scenarios are to be considered (e.g., if the scenario is to restore the system without external support, the availability status of equivalent units is set to false).

and its auxiliaries are not under maintenance (for thermal units only),

and its shutdown circuit breaker is not faulty,

and its isolating circuit breaker is not faulty (for thermal units only),

and there is at least one non-faulty isolator and busbar section in the substation in which it is coupled to the network (for thermal units only).

Table 4.1 Operating status of production units

	Thermal Units (Fig. 4.7.a and 4.7.b)				Other Units (Fig. 4.7.c and 4.7.d)	
	coupled	isolated	stand-by	shutdown	coupled	shutdown
SCb	closed	closed	open	open	closed	open
ICb	closed	open	closed	open		

If any of the above conditions are not satisfied, the unit is declared unavailable, i.e., no control action could be performed on it until it becomes available.

4.1.2.1.5 Load→Injection

There are four types of loads in the DS: consumer loads, inductive and capacitive shunt compensators and the thermal units' auxiliaries. Topology conversion for auxiliaries is handled along with their production units (Figures 4.7.a and 4.7.b). The other three types of loads share the same topology conversion model, shown in Figure 4.8.

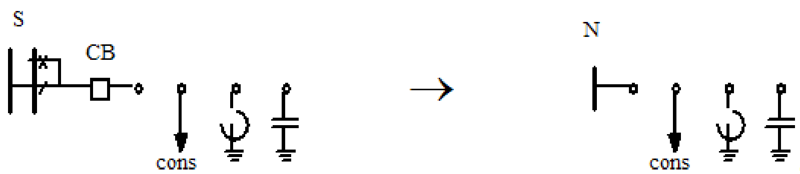


Figure 4.8 Conversion from substation to nodal topology for all loads, except for the thermal units' auxiliaries (they are considered along with thermal units). S stands for substation, CB for load circuit breaker, N for electrical node and "cons" for consumer load.

The values that can be assigned to the operating status of loads are summarised in Table 4.2.

Table 4.2 Operating status of loads

	Auxiliaries (Fig 4.7.a and 4.7.b)	
	disconnected	connected
SCb,ICb	both open	otherwise

	Other Loads (Fig. 4.8)	
	disconnected	connected
CB	open	closed

A load is said to be available if:

it is not under maintenance (except for consumer loads),

and its circuit breaker is not faulty (except for auxiliaries),

and there is at least one non-faulty isolator and busbar section in the substation in which it is coupled to the network (except for auxiliaries).

If any of the above conditions is not satisfied, the load is declared unavailable, i.e., no control action could be performed on it until it becomes available.

4.1.2.1.6 Transmission equipment→Branch

There are four types of transmission equipment in the DS: lines, step-up transformers, TCUL transformers and autotransformers. The topology conversion model for all of them is the one shown in Figure 4.9.

Depending on the operating status of the circuit breakers on the ends of a line or transformer, CB1 and CB2, the transmission element may be:

1) connected - both circuit breakers CB1 and CB2 are closed,
2) disconnected - both circuit breakers CB1 and CB2 are open, or
3) stand-by - a line or transformer has this operating status if one circuit breaker is open and another is closed (there may be a line or a transformer which is not coupled in a substation, but directly connected with another line, as its derivation; the derivation element has no circuit breaker at the point of coupling, and therefore it can be either connected or stand-by, but never disconnected).

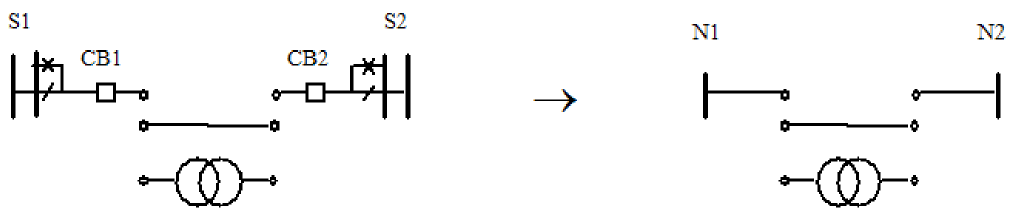


Figure 4.9 Conversion from substation to nodal topology for all transmission equipment. S stands for substation, CB for circuit breaker and N for electrical node. Index "1" is related to the sending end of a line (primary side of a transformer) and "2" to the receiving end of a line (secondary side of a transformer) .

The values that can be assigned to the operating status of transmission equipment are summarised in Table 4.3.

Table 4.3 Operating status of transmission equipment

	Transmission Equipment (Fig. 4.9)			
	connected	stand-by-1	stand-by-2	disconnected
CB1	closed	open	closed	open
CB2	closed	closed	open	open

Transmission equipment is said to be available if:

it is not under maintenance,

and neither of its circuit breakers is faulty,

and there is at least one non-faulty isolator and busbar section in both substations in which it is coupled.

If any of the above conditions is not satisfied, the transmission equipment is declared unavailable, i.e., no control action could be performed on it until it becomes available.

* * *

Up to now, the conversion from substation to nodal power system representation has been described. The topology conversion functions create the nodal image of the power system along with the operating and availability status of power system elements, which are appropriate for use in the DS. From now on, the discussion on DS functionality and operation will be based on nodal topology models.

Apart from the topology models for power system elements, the DS considers two more topology related concepts. Namely, after the nodes have been identified and the operating status of power system elements determined, the nodes, the branches and the injections may be grouped into zones according to different criteria. Whatever the criteria is, a zone represents the set of nodes which are linked through connected or stand-by branches. Since each node may be assigned a zone, the same may be done with the injections and branches since one or two nodes, respectively, are associated with them.

The DS considers two types of zones: electrical zones, closely related to injections that model production units, and load zones, defined on the basis of injections representing consumer loads.

4.1.2.2 Electrical and load zones

An electrical zone (EZ) in the DS corresponds to what is called a power system island and is defined as follows:

> *All the nodes in the network that are electrically linked (through stand-by or connected branches) represent an EZ if there is at least one node associated with a* ***production unit****.*

An elementary EZ contains a single node to which the production unit is connected. Therefore, the maximum number of EZs in a power system equals the number of production units.

The notion of a load zone (LZ) in the DS has been introduced with respect to EDF and its restoration strategy (refer to Section 2.4.3) as follows:

> *All the nodes in the network that are electrically linked (through stand-by or connected branches) but not energised represent an LZ if there is at least one node associated with a* ***consumer load****.*

An elementary LZ contains a single node to which the consumer load is connected. Therefore, the maximal number of LZs in a power system equals the number of consumer loads.

4.1.3 Commands

The DS is used to simulate the restoration control actions. A simplified syntax has been defined to translate the control actions into commands, understandable by the DS. The DS interprets a command by executing an appropriate set of functions. From all available DS commands[11], only those that are likely to be used within DAFFOR will be described below.

A DS command is the text string of at most four words, each of which has certain function:

1) the first word defines the power system element type (i.e., production unit, load or transmission equipment),
2) the second word defines the kind of action that can be simulated (e.g., synchronise production unit, pick up load, etc.),
3) the third word must be a valid name of the power system element whose type is defined by the first word, and
4) the fourth word is a numerical constant *c*, necessary only for some commands (e.g., pick up *c* MW of load).

Before the execution of a command in the DS, different tests are performed in order to determine its validity and applicability. For example, if a line is connected, the command that would try to connect it is invalid. Another example could be the attempt to change the power set-point of an equivalent unit, which is impossible. The conditions contained in this kind of test may be considered as deterministic. The second set of tests contains conditions based on heuristics and operating limits. For instance, the maximal amount of consumer load that can be picked up at a given moment depends on the spinning reserve of the units running in the same electrical zone. **In order for the DS to execute a command, all the conditions related to the command must be satisfied. If this is the case, the command is considered as valid, and its execution in the DS is the simulation of the corresponding control action execution on the power system.** In the following sections, these conditions are designated as **c.xx**.

The conditions are mostly divided in the same manner as commands, i.e., by power system element. Therefore, they will be introduced along with the explanation of the corresponding command. However, some conditions are common to all the commands (e.g., a command is valid if the element in question is available, with the availability status defined in Sections 4.1.2.1.4-6), or are based on the same heuristics. Before proceeding to the enumeration of the DS commands, two heuristics concerning consumer load pickup will be discussed.

4.1.3.1 Heuristics related to load demand increase

The main objective of the restoration procedure is to re-establish the supply to all unserved consumer loads. This is in general performed by successive load pickups, i.e., by control

[11] There are also commands that are useful when the DS is run as an independent user-interactive programme such as: total or partial blackout simulation, tracing and printing information by type of element, help facility, etc.

actions executed directly on loads. Remember, however, that the French power system would be divided after a general incident so that the load zones are created, as explained in Section 2.4.3. In fact, there are groups of consumer loads which are connected, but not supplied, and a single line energisation or production unit synchronisation may lead to the pickup of these loads. In this case, the load pickup is done indirectly, i.e., by an action executed on a power system element other than the consumer load.

Whenever a control action results in consumer load pickup, either directly or indirectly, special attention must be paid to at least two factors: (1) the uncertainty about the amount of load that will be supplied in reality, and (2) the current spinning reserve in the electrical zone, which should be sufficient to answer the load demand increase. In order to take these factors into account, the following two heuristics are implemented in the DS:

h.1 Verify that there will be no violation of operating limits (over- and under-voltages, overloads, frequency excursions) if the load demand after the action is executed happens to be 20% higher or lower than expected, and

h.2 The load pickup increment should not be higher than 10% of the current spinning reserve in the electrical zone.

The above heuristics are implemented in the DS as conditions associated to commands which result in load demand increase - the conditions must be satisfied in order for the command to be valid. Since the first heuristic concerns the dynamic response of the power system, its implementation will be explained in Section 4.1.5 which deals with the dynamic simulation. The conditions corresponding with the second heuristic will be introduced along with the related DS command.

4.1.3.2 Production units

For thermal units (refer to Figures 4.7.a and 4.7.b), there is a constraint on the modification of their operating status which states that it is impossible to operate SCb and ICb circuit breakers simultaneously. Therefore, the possible "paths" of moving a thermal unit from one operating status to another are as follows:

shutdown ↔ stand-by ↔ coupled

shutdown ← isolated ↔ coupled

Synchronisation of a unit: `$ gen on <unit>`

This command is used to change the operating status of a `unit` to coupled. It may be applied to a shutdown hydraulic or equivalent `unit`, or to a thermal `unit` which is isolated or in stand-by.

An isolated thermal unit is one that succeeded the load reject, i.e., a unit which supplies its own houseload. As discussed in Section 2.4.3, the success of load reject may be complete (TS1 nuclear or classical thermal units, shown in Figure 2.5) or partial (TS2 nuclear units, shown in Figure 2.6). The TS1 units are ready for synchronisation at any moment, and therefore considered as black-start units. However, the TS2 units, although isolated, must not be synchronised before their auxiliaries are supplied from the network. This can be formulated in a condition as follows:

c.1 If the isolated nuclear `unit` is a TS2 unit, it can be synchronised only to the energised network, i.e., where at least one production unit has already been coupled.

Each production unit is associated with a minimum time interval t_{min} before which it can not be coupled to the network. In general case, this time interval is on the order of a few hours for thermal units, and for other types of units it is on the order of a few minutes. The minimum time interval consideration applies to stand-by thermal units (nuclear and classical) and to shutdown hydraulic and equivalent units. The condition may then be formulated as follows:

c.2 If the restoration has started at moment t'_0, the `unit` may be synchronised to the network only after time interval t_{min}, i.e., $t' \geq t'_0 + t_{min}$.

When a black-start `unit` is to be coupled (isolated thermal, or shutdown hydraulic or equivalent unit), there may be a load zone on the closed path that will be supplied as soon as the path is energised by the black-start `unit`. If the load zone demand (which is the sum of all the connected consumer loads given by equation (4.11)) is denoted as $P_{l\_zone\_init}$, according to heuristic **h.2**, the black-start `unit` may be coupled if its spinning reserve R_{spin} satisfies the following condition:

c.3 $$R_{spin} \geq 10 \times P_{l\_zone\_init}.$$

<u>Shutdown of a unit: `$ gen off <unit>`</u>

This command is used to change a `unit`'s operating status to shutdown. It may be applied to a coupled hydraulic or equivalent `unit`, or to a thermal `unit` which is isolated or stand-by.

<u>Auxiliaries supply of a unit: `$ gen sby <unit>`</u>

This command is used to simulate either the supply of a `unit`'s auxiliaries from the network, or the intermediate step towards the `unit`'s shutdown. Therefore, it may be applied only to a thermal `unit` which is either shutdown or coupled, respectively.

In order to supply the auxiliaries of a shutdown thermal unit, there must be an energised network, which is translated into condition as follows:

c.4 A shutdown thermal `unit` can change its operating status to stand-by only if there is an energised network which can supply its auxiliaries.

<u>Isolation of a unit: `$ gen isl <unit>`</u>

This command is used to simulate the load reject by disconnecting a `unit` from the network, so that the `unit` supplies its own auxiliaries. Therefore, it may be applied only to a thermal `unit` which is coupled.

Locking primary frequency control of a unit: `$ gen fix <unit>`

This command is used to simulate the locking of the primary frequency control of a `unit` (see Section 4.1.1.1). Therefore, it may be applied only to a thermal or hydraulic `unit` which already participates in frequency control.

Unlocking primary frequency control of a unit: `$ gen aut <unit>`

This command is used to simulate the activation of the primary frequency control of a `unit` (see Section 4.1.1.1). Therefore, it may be applied only to a thermal or hydraulic `unit` which has the primary frequency control locked.

Loading of a unit (to default level): `$ gen aset <unit>`

This command is used to simulate the increase of the power set-point of a `unit` to the next default set-point level.

For a thermal `unit`, the default power set-point levels are [30; 75; 90] % of its rated power P_{nom}, and a hydraulic `unit` has single default power set-point level, 90% of its rated power P_{nom}. The load increase per second is given by equation (4.4), where x=5 for thermal and x=100 for hydraulic `unit`.

Before changing the current power set-point of a production unit, it is necessary to verify that the maximum set-point level P_{set_max} has not already been reached. Therefore, this command can be applied to a thermal or hydraulic `unit` whose current power set-point P_0 satisfies the following condition:

c.5 $$P_0 \leq P_{set\_max}.$$

Loading of a unit (to desired level): `$ gen set <unit> <p>`

This command is used to simulate the increase or decrease of the power set-point of a `unit` to `p` MW. The only difference with the command `aset` is the new power set-point value which is no longer defined by a default set-point level, but introduced with the command's argument `p`.

When the current power set-point P_0 of a production unit is to be changed to `p` MW, it is necessary to verify that the argument `p` has a valid value, i.e., that it is between 0 and the unit's rated power P_{nom}, and that it is different from the current set-point P_0. Therefore, this command can be applied to a thermal or hydraulic `unit` if the command's argument `p` satisfies the following conditions:

c.6 $$0 \leq \mathtt{p} \leq P_{nom} \text{ AND } \mathtt{p} \neq P_0.$$

Voltage set-point change (by default value): `$ gen avol <unit>`

This command is used to simulate the increase of the voltage set-point of a `unit` by a default value ΔV pu, specified as a parameter for the power system in question, usually (0.01-0.02) pu.

The voltage set-point of the `unit` may take any of values from the interval [V_{set_min}, V_{set_max}]. Before changing the current voltage set-point of the `unit`, it is necessary to verify that the upper set-point limit V_{set_max} will not be exceeded. Therefore, this command can be applied to a thermal or hydraulic `unit` whose current voltage set-point V_0 satisfies the following condition:

c.7 $$V_0 \le V_{set\_max} - \Delta V.$$

Voltage set-point limits are parameters with default values V_{set_min}=0.95pu and V_{set_max}=1.05pu.

Voltage set-point change (by desired value): `$ gen vol <unit> <v>`

This command is used to simulate the increase (`v`>0) or the decrease (`v`<0) of the voltage set-point of a `unit` by `v` pu. The only difference with the command `avol` is the increment value which is no longer defined by default, but introduced with the command's argument `v`.

When the current voltage set-point V_0 of a production unit is to be modified for `v` pu, it is necessary to verify that the argument `v` has a valid value, i.e., that the resulting new set-point voltage does not exceed the permitted minimum and maximum limits, V_{set_min} and V_{set_max}, respectively. Therefore, this command can be applied to a thermal or hydraulic `unit` if the command's argument `v` satisfies the following condition:

c.8 $$(V_{set\_min} - V_0) \le \mathrm{v} \le (V_{set\_max} - V_0) \text{ AND } \mathrm{v} \ne 0.$$

* * *

In Table 4.4 are summarised the conditions that the DS tests in order to determine if a command for a production unit is valid. For example, the command `$ gen aset <unit>` is valid for an **available coupled hydraulic** `unit` **participating** in frequency control provided that condition **c.5** is satisfied.

4.1.3.3 Loads

Connecting a load: `$ loa on <load>`

This command is used to simulate the connection of a `load`, i.e., to change its operating status to connected. It may be applied to a disconnected consumer or shunt compensation `load`.

When `load` is a consumer load, the result of this action should be the total `load` pickup (4.11). According to heuristic **h.2**, the action will be valid for any consumer load whose initial real power P_{l_init} satisfies the condition on the current spinning reserve of the electrical zone R_{spin}, which should cover the load demand, as follows:

Table 4.4 Conditions for DS commands on production units ("*" stands for the condition based on heuristic **h.1**)

DS command	unit type	availability status	operating status	frequency control	other conditions
`gen on`	nuclear	available	isolated		**c.1** and **c.3***
	class. thermal	available	isolated		**c.3***
	thermal	available	stand-by		**c.2**
	other	available	shutdown		**c.2** and **c.3***
`gen off`	thermal	available	isolated or stand-by		
	other	available	shutdown		
`gen sby`	thermal	available	shutdown		**c.4**
	thermal	available	coupled		
`gen isl`	thermal	available	coupled		
`gen fix`	thermal	available	isolated or coupled	enabled	
	hydraulic	available	coupled	enabled	
`gen aut`	thermal	available	isolated or coupled	locked	
	hydraulic	available	coupled	locked	
`gen aset`	thermal	available	isolated or coupled	enabled	**c.5**
	hydraulic	available	coupled	enabled	**c.5**
`gen set`	thermal	available	isolated or coupled	enabled	**c.6**
	hydraulic	available	coupled	enabled	**c.6**
`gen avol`	thermal	available	isolated or coupled		**c.7**
	hydraulic	available	coupled		**c.7**
`gen vol`	thermal	available	isolated or coupled		**c.8**
	hydraulic	available	coupled		**c.8**

c.9 $$P_{l\_init} \leq 0.1 \times R_{spin}.$$

Disconnecting a load: `$ loa off <load>`

This command is used to simulate the disconnection of a `load`, i.e., to change its operating status to disconnected. It may be applied to a connected consumer or shunt compensation `load`.

Load level change (by desired value): $ loa set <load> <p>

This command is used to simulate the pickup (p>0) or the shedding (p<0) of p MW of a consumer load.

Referring to the consumer load model given by equation (4.10), argument p represents in fact the increment ΔP. The minimum value it can take is $-P_0$, in which case the load is completely shed (the result is exactly the same as if command $ loa off <load> were applied). The maximum value for the argument p is $(P_{l\_init}-P_0)$, in which case the load is completely supplied. However, according to heuristic **h.2**, argument p may be limited by the current spinning reserve of the electrical zone R_{spin}, which should cover the load increase ΔP. Therefore, the command is valid if the argument p satisfies the following condition:

c.10 $$-P_0 \leq \mathrm{p} \leq min((P_{l\_init}-P_0),(0.1 \times R_{spin})) \text{ AND } \mathrm{p} \neq 0.$$

Picking up a load (by unserved levels): $ loa aset <load> <n>

This command is used to simulate the pick up of n unserved levels of consumer load.

Equations (4.8) give the number and size of unserved levels as a function of the currently supplied portion of the load, P_0. Therefore, the command's argument n could take any of values $i = 1,\ldots,(N_{shed}-m)$. According to heuristic **h.2**, it is necessary to test the unserved levels against the current spinning reserve R_{spin} in the electrical zone that supplies the load increase $\Delta P(i)$ as follows:

$$\Delta P(i) \leq 0.1 \times R_{spin},\ i = (N_{shed}-m),\ldots,1. \qquad (4.12)$$

As soon as (4.12) becomes true, the current index i represents the maximum number of unserved levels, denoted by i_{max}. In this case, the corresponding unserved level $\Delta P(i_{max})$ is the load increment ΔP from equation (4.10). If (4.12) is not satisfied for any value of i, no unserved level can be picked up and i_{max}=0. In conclusion, the command is valid for the consumer load if the argument n satisfies the following condition:

c.11 $$1 \leq \mathrm{n} \leq i_{max} \text{ AND } \mathrm{n} \neq 0.$$

* * *

Table 4.5 contains summarised conditions that the DS tests first in order to filter the valid commands on loads. For example, the command $ loa on <load> is valid for an **available disconnected consumer** load provided that condition **c.9** is satisfied. Note that there is no command that can be applied directly to the thermal units' auxiliaries. This type of load is handled along with the thermal units.

Table 4.5 Conditions for DS commands on loads ("*" stands for the condition based on heuristic **h.1**)

DS command	load type	availability status	operating status	other conditions
`loa on`	shunt	available	disconnected	
	consumer	available	disconnected	**c.9***
`loa off`	shunt or consumer	available	connected	
`loa set`	consumer	available		**c.10***
`loa aset`	consumer	available		**c.11***

4.1.3.4 Transmission equipment

Energisation of a branch: `$ bra on <branch>`

This command is used to change the operating status of a `branch` from disconnected or stand-by to connected. It may be applied to a stand-by or disconnected line, autotransformer or TCUL transformer.

When a branch is energised in reality, both ends of the branch are tested on the voltage magnitude difference, voltage angle difference and frequency difference. If any of these differences is over the given threshold (Δv, $\Delta\varphi$ and Δf, respectively), the branch cannot be energised. In the DS, the tests that must be performed to define the validity of the command depend on the result of `branch` energisation, which may be threefold: coupling, strengthening and extension.

If the `branch` is to connect nodes N1 and N2 which are in different electrical zones, the result of energisation is **coupling**. In the DS, the voltage angle for a node is in relation to the power flow slack node for a given electrical zone; since two electrical zones have different slack nodes, the reference for voltage angles of nodes N1 and N2 is different, so there is no physical meaning in testing their difference. On the contrary, the voltage magnitude difference $V(\mathrm{N1})-V(\mathrm{N2})$ and the frequency difference $f(\mathrm{N1})-f(\mathrm{N2})$ must satisfy conditions **c.12** and **c.13**, respectively:

c.12 $$|V(\mathrm{N1})-V(\mathrm{N2})| \leq \Delta v$$

c.13 $$|f(\mathrm{N1})-f(\mathrm{N2})| \leq \Delta f.$$

If the `branch` connects nodes N1 and N2 which are in the same electrical zone, the result of energisation is **strengthening**. In the DS, the frequency is computed by the long-term dynamic model which assumes that the frequency is equal in all nodes in an electrical zone. Therefore, the frequency is equal in both nodes N1 and N2. On the contrary, the voltage magnitude difference $V(\mathrm{N1})-V(\mathrm{N2})$ and the voltage angle difference $\varphi(\mathrm{N1})-\varphi(\mathrm{N2})$ must satisfy conditions **c.12** and **c.14**, respectively:

c.14 $|\varphi(N1)-\varphi(N2)| \leq \Delta\varphi$.

The thresholds Δv, $\Delta\varphi$ and Δf are input parameters, and in the DS they have the following default values: Δv =0.1pu, $\Delta\varphi$=30° and Δf=0.1Hz.

If the `branch` is to energise the path from node N1, belonging to electrical zone EZ(N1), to node N2 which does not belong to any electrical zone, the result of energisation is **extension**. Since node N2 does not belong to an electrical zone, it may belong to a load zone that will be supplied as soon as the path is energised by the `branch`. If the load zone demand (which is the sum of all connected consumer loads given by equation (4.11)) is denoted as $P_{l\_zone\_init}$, according to heuristic **h.2**, the `branch` may be connected if the spinning reserve R_{spin} of electrical zone EZ(N1) satisfies the following condition:

c.15 $R_{spin} \geq 10 \times P_{l\_zone\_init}$.

Disconnection of a branch: `$ bra off <branch>`

This command is used to change the operating status of a `branch` from stand-by or connected to disconnected. It may be applied to a stand-by or connected line, autotransformer or TCUL transformer.

* * *

Table 4.6 contains summarised conditions that the DS tests first in order to filter the valid commands on transmission equipment. For example, the command `$ bra on <branch>` is valid for an available disconnected or stand-by line which results in coupling, provided that conditions **c.12** and **c.13** are satisfied. Note that there is no command that can be applied directly to the units' step-up transformers. This type of transmission equipment is handled along with thermal and hydraulic units.

4.1.3.5 Conclusion

In the previous sections, the DS commands have been described one by one. A number of conditions are associated with each command: all the related conditions must be satisfied in order for the command to be valid, and therefore executed in the DS. Regarded macroscopically, the set of valid commands corresponds to **possible control actions for a power system characterised with a state at a given moment**.

Table 4.6 Conditions for DS commands on transmission equipment ("*" stands for the condition based on heuristic **h.1**)

DS command	transmission equipment type	availability status	operating status	result	other conditions
`bra on`	line or autotransformer or TCUL transformer	available	disconnected or stand-by	coupling	**c.12** and **c.13**
				strengthening	**c.12** and **c.14**
				extension	**c.15***
`bra off`	line or autotransformer or TCUL transformer	available	connected or stand-by		

4.1.4 Time consideration

From the point of view of operation in real time, each control action requires time to be performed. Consider as an example a line that is to be energised between two islands. In real time operation, this might imply the following: (1) the operator must first contact a person in the field, (2) this person will close the line circuit breaker in his/her substation, and (3) the circuit breaker in the substation on the second end of the line will be closed by the synchro-check equipment as soon as coupling conditions permit it. These three real-time operations will take some time, after which the line will really be connected. From that moment, two islands are coupled and some transient events will appear in the system.

In the general case, for action a_r, the time of manoeuvres will be denoted as $T_{man}(a_r)$, and the time of the power system response as $\tau_{ps}(a_r)$. Their sum will be called total response time, denoted as $\tau_{resp}(a_r)$ and defined as:

$$\tau_{resp}(a_r) = T_{man}(a_r) + \tau_{ps}(a_r). \tag{4.13}$$

Total response time is shown on the **real-time axis** t in Figure 4.10.a. Power system state $s(a_{r-1})$ from the transition expression (4.1) coincides with instant $t_{a\_request}$ in which the action is requested, while the steady state $s(a_r)$ is expected to be reached at instant t_{exp_ss}.

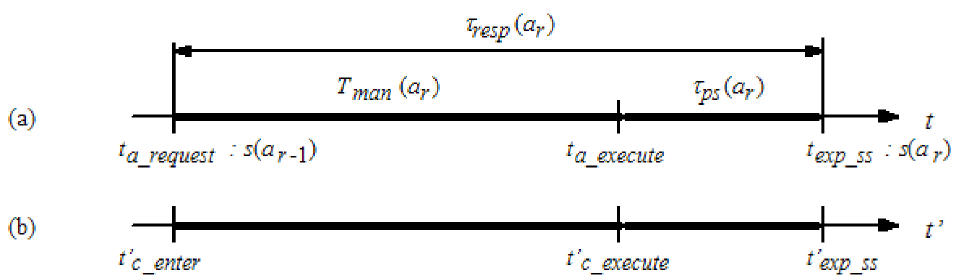


Figure 4.10 Real-time operation timing and corresponding power system states represented on (a) real-time axis t, and (b) simulated time axis t'

The DS simulates in the similar manner both real-time operation and power system dynamic response. The time consideration in the DS is shown on the **simulated time axis** t' in Figure 4.10.b.

4.1.4.1 Time of manoeuvres

The DS command which translates control action a_r is entered at time t'_{c_enter}. The time $T_{man}(a_r)$ of manoeuvres by which the control action a_r is carried out depends on the type of the action. In the DS, this time is a parameter associated to a type of command. Default values are given in Table 4.7. The command is executed in the DS after time $T_{man}(a_r)$ has expired, i.e., at instant $t'_{c\_execute}$.

Table 4.7 Time of manoeuvres by type of control action

a_r	$T_{man}(a_r)$ [minutes]
coupling between two islands	5
starting and coupling a hydraulic unit from shutdown	4
coupling a black-start thermal unit	7
other actions	2

4.1.4.2 Time of power system dynamic response

After the command had been executed at time $t'_{c\_execute}$, the DS simulates the dynamic response of the power system, as shown in Figure 4.11. The evolution of the system mean frequency and the output active power of production units are computed until the steady state is reached. This state is, in fact, a quasi-steady state because it has been obtained with power losses dating from the state preceding the command execution (due to the long-term dynamic model). Next, the power flow is run to compute system voltages, units' reactive power and actual power losses of the network. Finally, the long-term dynamics computation is run again to take into account the updated power losses, until the real steady state is reached, at time t'_{exp_ss}.

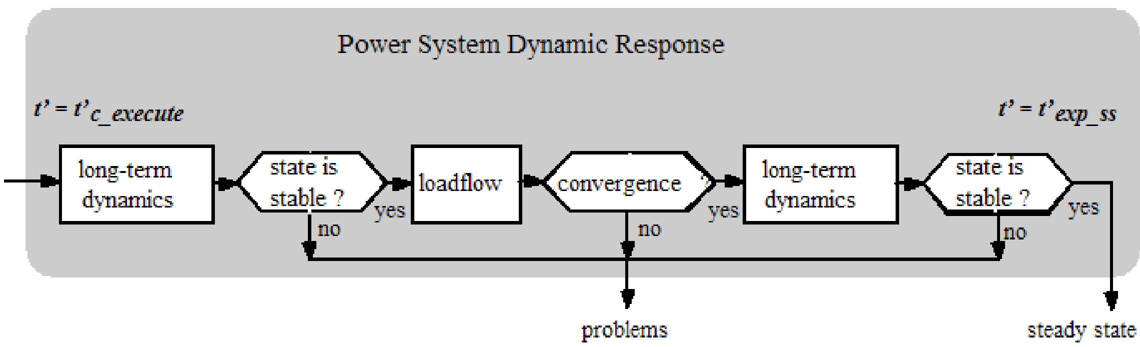


Figure 4.11 Simulation of dynamic response of the power system in the Dynamic simulator

If the power system state has been found unstable during the dynamic simulation, or if the power flow has not converged, the simulation is stopped and the information about the failure is returned.

4.1.5 Simulating an action

The basic concepts concerning the DS introduced so far permit us to extend Figure 4.2 by the flow chart of the DS shown in Figure 4.12. Action a_r and states $s(a_{r-1})$ and $s(a_r)$ correspond to those given by transition expression (4.1), and time-related items (t'_{c_enter}, $T_{man}(a_r)$, $t'_{c\_execute}$ and t'_{exp_ss}) with those from Figure 4.10.b.

Consider a command, translating control action a_r, which is entered at time t'_{c_enter} (step 4). The DS first determines the validity of the command on the basis of the current power system state $s(a_{r-1})$. If the command is invalid, this information is saved in the simulation *status* and the input state $s(a_{r-1})$ is left unmodified. Otherwise, the time of manoeuvres $T_{man}(a_r)$ is determined and the simulation time is increased by this value. Depending on the type of action a_r, it might be necessary to take into account the load uncertainty according to heuristic **h.1**, stated in Section 4.1.3.1 and referred to by an asterisk in Tables 4.4 through 4.6.

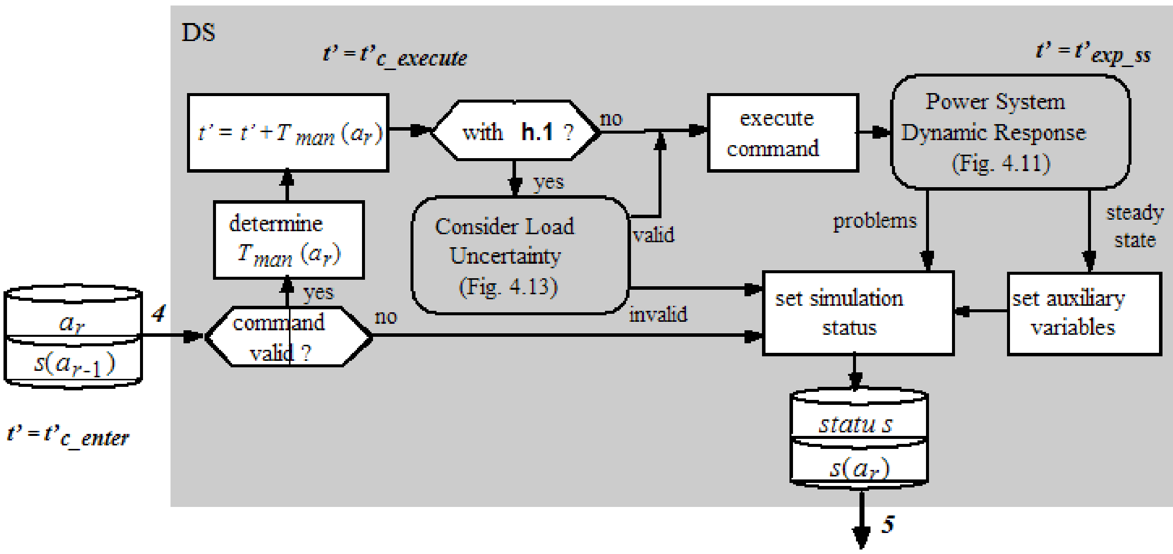


Figure 4.12 Flow chart of the Dynamic simulator

If the current command execution does not result in direct or indirect load demand increase (without **h.1**), it is executed at time $t'_{c\_execute}$ and the dynamic response of the power system is simulated. If the state has been unstable or if the power flow has not converged, this information is saved in the simulation *status*, and the state $s(a_r)$ which is returned in this case is not a steady one. Otherwise, the steady state $s(a_r)$ is reached at simulated time t'_{exp_ss}. Some auxiliary variables from $s(a_r)$ and some items from *status* are set before being returned as output of the DS (step 5). Precision on auxiliary variables and simulation status will be given in Section 4.1.6.

However, if the current command has as a result some consumer load pickup, additional validation, based on heuristic **h.1**, must be performed before the command is executed. The manner in which the DS takes into account the load uncertainty is shown in Figure 4.13.

Consider that the current command is `$ gen on <unit>`, which translates the action of coupling a black-start hydraulic `unit`. Consider also that the coupling will result in the pickup of a load zone of $P_{l\_zone\_init}$ MW. Referring to Table 4.4, it can be seen that the command is valid if conditions **c.2** and **c.3** are satisfied, but that the load uncertainty must be also taken into account before the command is executed. Therefore, the expected load is first increased by 20%, the command is executed and the dynamic response of the power system is simulated. If the steady state is reached with no violation of operating limits, the same operations are performed for the case in which the load is lower for 20% than the expected one.

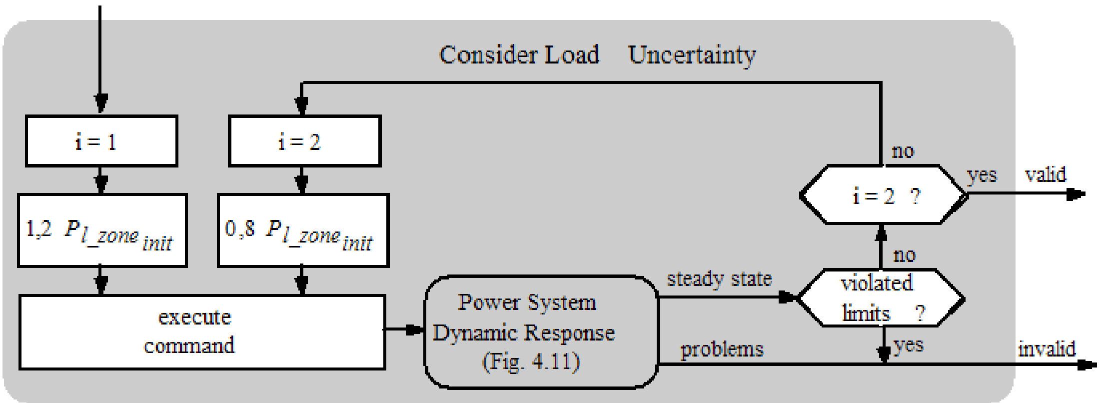


Figure 4.13 Implementation of heuristic **h.1** in the Dynamic simulator

If there are either operating limits violations or stability and convergence problems, the simulation is stopped and the command is declared invalid. The information about the failure is saved in the simulation *status* (see Figure 4.12) and the input state $s(a_{r-1})$ is left unmodified. On the contrary, if the steady state without operating limits violations has been reached for both $1.2P_{l\_zone\_init}$ and $0.8P_{l\_zone\_init}$, the current command is declared valid and therefore executed as any other command.

4.1.5.1 Example

To illustrate the operation of the DS, the connection of a load was simulated for the New England test network, given in Appendix A. Initially, load `ln38` of 158MW was disconnected. Figure 4.14 shows the evolution of the frequency and the real power output of production units, simulated by the DS, as power system response to the pickup of load `ln38`.

The command `$ loa on ln38` was entered at simulated time t'_{c_enter}=00:08:14. It has first been validated (i.e., successful tests on conditions from Table 4.5), and the time of manoeuvres was determined to be 2 minutes (see Table 4.7). Then the simulated time was increased and the appropriate functions were executed to connect the load at moment $t'_{c\_execute}$=00:10:14.

The dynamic simulation started at $t'_{c\_execute}$=00:10:14, and after about 38 seconds of simulated time, the quasi-steady state was reached. At that moment, the power flow computation was performed, and it yielded, among other variables, updated power losses. They were taken into account in a new run of the long-term dynamic model, and the real steady state was reached by time t'_{exp_ss}=00:11:07.

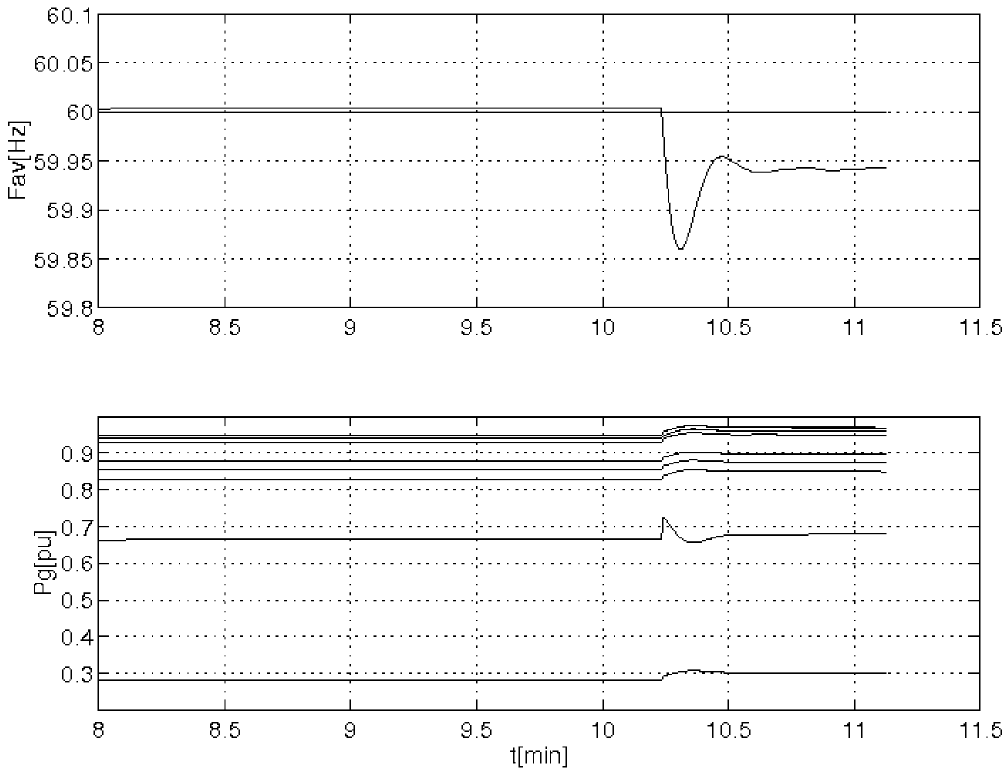


Figure 4.14 Frequency and units' output power evolution after connecting load of 158MW, obtained by the Dynamic simulator

4.1.6 Dynamic simulator output

The DS receives both the current power system state $s(a_{r-1})$ and action a_r as input. The action a_r is executed, and the DS returns the resulting state $s(a_r)$ and the simulation *status* as output. Section 4.1.3 dealt with "action" and its translation into a command which can be understood by the DS. This section focuses on the notion of "power system state" as it is defined in the DS, and on the kind of information contained in the simulation *status*.

4.1.6.1 Power system state

The DS recognises 3 classes of power system elements and 3 classes of topology related concepts, as follows:

1) transmission equipment (lines, autotransformers, step-up transformers, TCUL transformers),
2) production units (nuclear thermal, classical thermal, hydraulic and equivalent),
3) loads (consumer loads, shunt reactors, shunt condensers, the thermal units' auxiliaries),

4) electrical nodes,
5) electrical zones, and
6) load zones.

Since the DS is implemented in the programming language C, each of the above classes has been defined with a C-structure type, and all elements of the same type (e.g., all production units) form an array of C-structures of the corresponding type. The six arrays are then grouped into a complex C-structure, which represents the power system. In addition, the C-structure defining the power system state contains fields such as dimensions of the arrays, simulated and real time, total initial consumer load and the amount of load currently supplied. Original type definitions for C-structures representing power system and its elements, as well as the structure containing different simulation parameters (with their default values), can be found in Appendix B.

The current values in the structures' fields completely describe the state of the power system in a given moment. By embedding all the relevant information in the power system and its elements into one type, a state may be declared as a single C-variable. This fact facilitates the access to any information in the power system and enables us to pass its state as single argument to any C-function. Since the DS is used as a function in the Reasoning kernel, this is exactly the way it receives the current state $s(a_{r-1})$ on which it simulates action a_r in order to issue the resulting state $s(a_r)$.

Not all the fields of the complex power system structure are modified in the DS. Values contained in the fields such as the name of a power system element, rated power of a production unit or impedance of a line never change.

On the contrary, the operating and availability status of a power system element, power and voltage set-point of a production unit, amount of consumer load that has already been supplied, spinning reserve in an electrical zone, voltage in an electrical node, etc., are the fields whose values are modified in the DS since they change in time. In fact, they are computed in functions that execute a command and simulate the dynamic response, shown in Figure 4.12.

4.1.6.1.1 Operating limits

During the simulation of the dynamic response of the power system, the evolution of frequency, voltages and power flows are tested against their operating limits. In case an operating limit is violated, the information is saved in the appropriate field of the power system structure.

Operating limits are parameters of the simulation and are set with respect to the power system. The following operating limits are considered in the DS:

- frequency f_i of electrical zone i should not exceed steady state frequency limits:

$$F_{SSmin} \leq f_i \leq F_{SSmax} \qquad (4.14)$$

which represent the percentage of the power system nominal frequency (their default values can be found in Appendix B);

- voltage V_i in node i should satisfy the following inequality:

$$V_{imin} \leq V_i \leq V_{imax} \tag{4.15}$$

which represent the percentage of the nodes nominal voltage (their default values can be found in Appendix B);

- power S_i through transmission equipment i should not exceed the maximum admissible value S_{imax}:

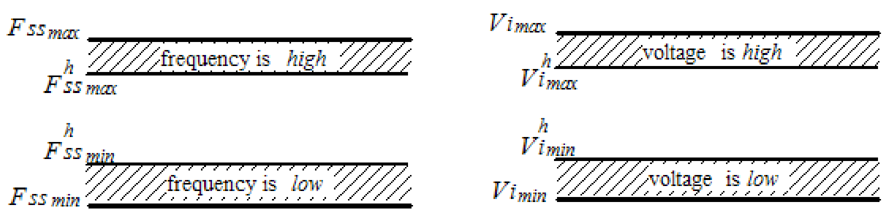
$$S_i \leq S_{imax} \tag{4.16}$$

for transformers, S_{imax} is maximum apparent power, and for lines, $S_{imax}=\sqrt{3}V_{inom}I_{inom}$, where V_{inom} and I_{inom} are the line's nominal voltage and maximum admissible current per phase, respectively.

4.1.6.1.2 Auxiliary variables: Heuristic limits

The DS is to be used for transmission power system restoration, during which the frequency and voltage excursions may be significant, and it is important to determine, for instance, when the frequency is *high* or the voltage is *low*. Auxiliary variables that are set just before the steady state is returned from the DS (see Figure 4.12) reflect this kind of criteria, which are of heuristic nature. The idea is to introduce a "measure of the distance" of the power system state from its operating limits. If the power system happens to be in a state which is "close" to some operating limits, a warning can be issued and an appropriate control action can be performed in order to "move the system away" from its limits.

The operating limits for electrical zone frequency (4.14) and node voltage (4.15) are shown in Figure 4.15. The above-mentioned measure of distance can be understood as pair of heuristic limits: (F^h_{SSmin},F^h_{SSmax}) for frequency and(V^h_{imin},V^h_{imax}) for voltage. The heuristic limits must be stricter than the original operating limits, and are defined empirically as functions of the characteristics of the power system in question.



(a) (b)

Figure 4.15 Operating and heuristic limits for (a) electrical zone frequency, and (b) electrical node voltage

In the DS, the heuristic limits for frequency and voltage are defined as follows:

$$F^h_{SSmax} = F_{SSmax} – F\_HIGH \times (F_{SSmax} – F_{SSmin}) \quad (4.17)$$

$$F^h_{SSmin} = F_{SSmin} + F\_LOW \times (F_{SSmax} – F_{SSmin}) \quad (4.18)$$

$$V^h_{imax} = V_{imax} – V\_HIGH \times (V_{imax} – V_{imin}) \quad (4.19)$$

$$V^h_{imin} = V_{imax} + V\_LOW \times (V_{imax} – V_{imin}) \quad (4.20)$$

where the coefficients *F_HIGH*, *F_LOW*, *V_HIGH* and *V_LOW* are the parameters whose default values can be found in Appendix B.

If the steady state has been reached in simulation (see Figure 4.12), the auxiliary variables are set with respect to equations (4.17-4.20), according to following heuristics:

h.3 Frequency f_i in electrical zone i is said to be *high* if $f_i > F^h_{SSmax}$.

h.4 Frequency f_i in electrical zone i is said to be *low* if $f_i < F^h_{SSmin}$.

h.5 Voltage V_i in electrical node i is said to be *high* if $V_i > V^h_{imax}$.

h.6 Voltage V_i in electrical node i is said to be *low* if $V_i < V^h_{imin}$.

4.1.6.2 Simulation status

The information contained in the simulation status variable *status* is a kind of summary for the caller of the DS. It concerns mainly the power system state (whether it is stable and whether there were violations of operating limits), but also the validity of the command that has been introduced.

The adopted criteria in the DS for both the steady and unstable states of the power system are as follows:

- the power system is said to be in *steady state* if the frequency is constant during 20 consecutive integration steps, the integration step being 0,05 seconds (these are default values), and
- power system is said to be in an *unstable state* if the frequency is outside ±2% of the nominal frequency (50 Hz), or if the frequency does not stabilise during 10 minutes of simulated time after an action is executed.

Simulation status contains also condensed information on eventual operating limit violations due to the execution of control actions. During dynamic simulation, inequalities (4.14-4.16) are tested in the DS and corresponding fields of power system elements to which they apply are appropriately set (e.g., an over-voltage detected in node *i* is saved in the corresponding field of the structure representing node *i*). Simulation status has three Boolean fields, each corresponding with one of three types of operating limits, defined in Section 4.1.6.1.1. At the end of the simulation, if there is at least one operating limit violation, the corresponding field in the simulation status is set to true. In this manner, the calling function may check only the array from the power system state variable to which the type of violation may apply.

Finally, if the DS has found that the entered command is invalid, a message about the reason is saved in the simulation status.

4.1.7 Conclusion

The subject of Section 4.1 was the Dynamic simulator (DS) which is called as a function in the Reasoning kernel. It performs the simulation of the dynamic response of the power system on control action a_r. Action a_r is applied to the power system, which is characterised with state $s(a_{r-1})$. During the simulation, the power system transits from state $s(a_{r-1})$ to state $s(a_r)$, due to action a_r. One call to the DS corresponds exactly to the transition expression (4.1) when $i=r$.

There are four important points to stress about the DS:

1) it considers several deterministic and heuristic criteria in order to declare an action as valid or invalid;
2) it takes into account both the time of manoeuvres necessary to carry out an action in real time and the response time of the power system to the action;
3) it keeps track of the violations of operating limits; and
4) it is written in ANSI C programming language and is thus fast and portable.

4.2 Knowledge base (KB) module

The introduction of this chapter stated that DAFFOR can suggest a control action to the operators (search mode) and assess an action suggested by the operators (validation mode). Both the search for an action to suggest and the assessment of the operators' action are performed in the Knowledge base (KB) module.

The KB module contains mostly the qualitative, or symbolic, knowledge about the restoration strategy and the way to apply it to the power system after a blackout. It also has quantitative knowledge on the current power system state (defined in Section 4.1.6.1) and the types of restoration control actions (defined in Sections 4.1.3.2 through 4.1.3.4). All that knowledge enables the KB module to perform a heuristic search among control actions which are valid for the current power system state (defined in Section 4.1.3.5) in order to choose the one to suggest to the operators (search mode). However, before it suggests the action, it uses the Dynamic simulator (DS) to verify what should happen in the power system if the action is executed, i.e., to get 'quantitative' feedback on its 'qualitative' knowledge, as shown by steps 4 and 5 in Figure 4.16.

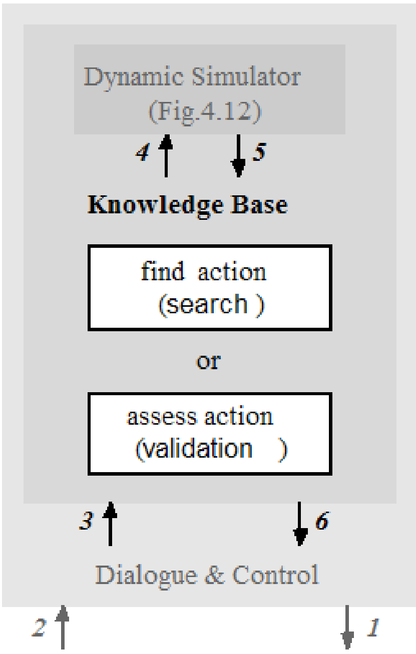


Figure 4.16 The Knowledge base module coupled with the Dynamic simulator

The operation of the KB module differs from one mode to the other as follows:

- In search mode, the KB module performs a heuristic search for the action to suggest, according to the restoration strategy and to the current power system state. The action the KB module finds is first simulated by the DS and is returned from the KB module only if it results in a steady state with no violations of operating limits. As long as such an action is not found, the KB module searches for another one.

- In validation mode, the KB module does not perform any search for an action, since the action is provided by the operators. The only function of the KB module in this case is to pass the action and the current power system state to the DS, to assess the power system response on the basis of simulation *status*, and to return both the resulting assessment and the power system state.

The KB module always generates some information on its operation, which is stored in the following variables:

1) *sol_found* contains information on whether the solution has been found in the KB module, and
2) *info* contains information on the overall operation of the KB module; it is a report about the inference and serves as the explanatory facility of DAFFOR.

The operation of the KB module in validation mode is quite simple since it does not imply either the heuristic search or the restoration strategy, but only the interaction with the DS, given the current power system state and the control action. Furthermore, this interaction represents an "elementary function" applied every time the search gives some result when the KB module operates in search mode. Therefore, the general operation of the KB module in validation and search mode will be described before addressing the search process, the restoration strategy and the heuristic criteria in detail. Finally, the implementation of the KB module will be discussed.

4.2.1 Functionality

Independently of the mode (search or validation), **the KB module focuses on a single control action, denoted as a_r, which should be the solution of the transition equation (4.1) for the given state $s(a_r\text{-}1)$, i.e., the r^{th} action in sequence (4.2). Action a_r is said to be a solution of (4.1) if it moves the power system from the initial state $s(a_r\text{-}1)$ to a steady state $s(a_r)$ without violations of the operating limits (4.14-4.16); otherwise, action a_r is not a solution**.

4.2.1.1 validation mode

In validation mode, the KB module is expected to assess action a_r provided by the operators when the current power system state is $s(a_{r-1})$. In other words, it must answer the question: "Is action a_r a solution of (4.1)?" To do that, the KB module uses the DS, as shown in Figure 4.17.

The KB module receives the action a_r provided by the operators and the current power system state $s(a_{r-1})$ (step 3.1), and launches the simulation of action a_r (step 4). After the DS issues the resulting state $s(a_r)$ and the simulation *status* (step 5), the KB module analyses them in order to set the assessment results for action a_r as follows:

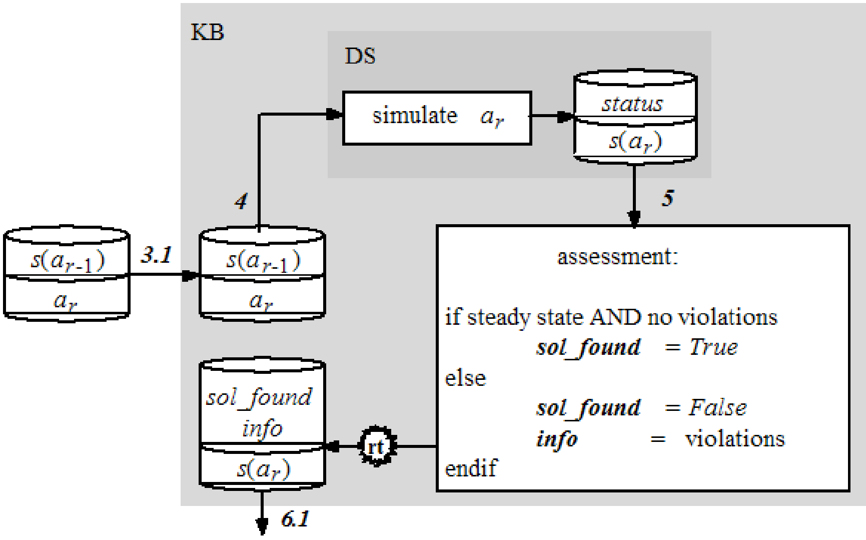


Figure 4.17 Validation represents the testing of an action within the Dynamic simulator (the star after the block "assessment" has to do with operation in real-time)

- *sol_found* is set to true or false depending on the content of the simulation status, *status*. If the resulting state $s(a_r)$ is steady and if, during the simulation, there has been no violation of the operating limits, action a_r is a solution of equation (4.1) and *sol_found* is set to true; otherwise, it is set to false. By setting *sol_found* to true in validation mode, the KB module "approves" action a_r suggested by the operators.
- *info* is filled only if the KB module disapproves action a_r which the operators have suggested. It reports the reasons for such a conclusion, i.e., the information on the operating limits violations, instability or the power flow divergence caused by action a_r, whichever is the case.

The outputs of the KB module are both the assessment results (*sol_found*, *info*) and the power system state $s(a_r)$ (step 6.1). The star at the exit of assessment box has to do with the real-time operation and will be discussed in Section 5.2.4.2 in the next chapter; at present, it can be ignored.

Note that, in validation mode, action a_r is provided by the operators (step 3.1), but in the general case, the source of action may be different (e.g., the action may be found in the heuristic search). Consequently, the validation mode, shown in Figure 4.17, can be considered as the "general simulation and analysis facility" of the KB module, and thus be used as an assessment function for the actions resulting from the heuristic search when the KB module operates in search mode.

4.2.1.2 search mode

In search mode, the KB module is expected to suggest to the operators a restoration action a_r when the current power system state is $s(a_{r-1})$, which must be a solution of (4.1). To do that, the KB module performs a search for an action a_r (among the actions that are valid for the current state $s(a_{r-1})$), based on the restoration strategy and some heuristics. A detailed description of the search will be given in Section 4.2.2. At present, let us describe the general operation of the KB module in search mode, shown in Figure 4.18.

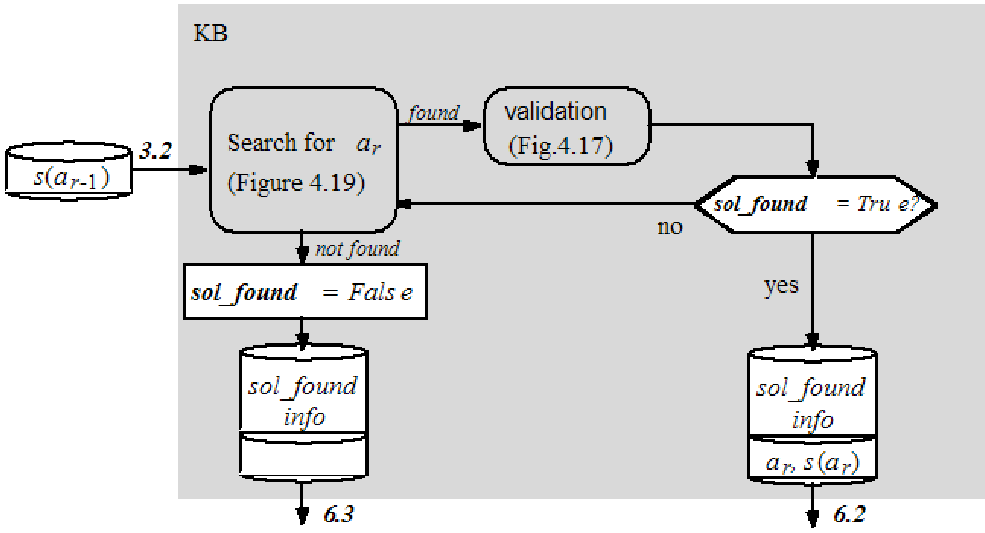


Figure 4.18 Operation of the KB module in search mode

The KB module receives the current power system state $s(a_{r-1})$ (step 3.2) and then starts the heuristic search for action a_r. When the KB module finds a valid action a_r, it first launches the assessment (as in validation mode) to see whether the action it found represents a solution of (4.1), and then proceeds as follows:

- if action a_r is not a solution (*sol_found*=false), the KB module continues the search for another action a_r that could be a solution of (4.1) for the same initial state $s(a_{r-1})$;
- as soon as action a_r is assessed as a solution (*sol_found*=true), the KB module returns both action a_r and the resulting state $s(a_r)$, as well as the assessment results *sol_found* and *info* (step 6.2);
- if the KB module has exhausted all the search paths and the action a_r could not be found, it sets *sol_found* to false; since no action has been found, there is neither an action nor a state to return, only assessment results *sol_found* and *info* (step 6.3).

Note that the variable *info* is filled inside the "validation"-box whenever *sol_found* is set to false (refer to Figure 4.17).

It can be said that action a_r, found by the KB module and which will be suggested to the operators, guarantees that the resulting state $s(a_r)$ will be stable and reached without operating limit violations. If such an action cannot be found, the KB module prefers not to suggest anything and to leave the operators, for instance, to ask for validation of their own action a_r.

Note here that the failure of the search does not mean that all valid actions for the current state $s(a_{r-1})$ have been assessed, since the search is not exhaustive[12]. On the contrary, the KB module examines only some of the currently valid actions according to the restoration strategy and heuristic criteria, i.e., its knowledge. Therefore, if the search fails, it usually means that the KB module does not have enough knowledge, or that the knowledge it contains must be refined or re-organised.

[12] Because the transmission power system restoration is such a highly combinatorial problem, it is difficult to imagine that an exhaustive search could be performed satisfactorily in real-time conditions.

4.2.2 Search process

Transmission power system restoration is a highly combinatorial problem, since many actions may be taken at a given moment; referring to equations (4.1-4.2), for the power system described with current power system state $s(a_{r-1})$, there are usually several equally valid restoration control actions which may be solutions of (4.1). The goal of the search process performed by the KB module is to choose one of them, action a_r, according to the restoration strategy, and with the aid of certain heuristic criteria. The current power system state $s(a_{r-1})$ and the set of all valid actions for that state are the KB module's *quantitative* knowledge, while the restoration strategy and the heuristic criteria represent its *qualitative*, or *symbolic* knowledge.

4.2.2.1 Principle

The following is a description of the elementary operations performed in the KB module during the search process. The objective is to show the principle of searching for action a_r as a function of the current power system state $s(a_{r-1})$ and according to the KB module's knowledge, in order to draw the box "Search for a_r " from Figure 4.18.

4.2.2.1.1 Current goal

The qualitative knowledge in the KB module (strategy and heuristics) is organised in a definite number N_{goal} of goals g_i=1,…, N_{goal} (e.g., decrease frequency, strengthen network, supply consumer load, etc.). The KB module determines which goal g_i will be set as the current one as a function of the current power system state $s(a_{r-1})$. For instance, if the frequency is *high* (refer to **heuristic h.3** in Section 4.1.6.1.2), the current goal will be to "decrease frequency".

4.2.2.1.2 Subset of valid actions

Goal g_i may be reached by applying a certain type of control action a_r (e.g., frequency may be decreased by picking up some load). This control action must be chosen from the set of all actions which are valid for the current power system state $s(a_{r-1})$ (this set may be determined thanks to **heuristic h.2** and conditions **c.1** through **c.15** already discussed in Section 4.1.3). The KB module determines the subset of all valid actions which are of the desired type (e.g., actions which result in consumer load pickup). Since the subset of actions is determined as a function of the current goal g_i, it will be denoted by A_{gi}. Consequently, action a_r should be one of the actions from this subset, i.e., $a_r \in A_{gi}$.

4.2.2.1.3 Assigning Priorities

In order to choose the most adequate action $a_r \in A_{gi}$, the KB module will first order actions in subset A_{gi} according to certain criteria. For instance, a **heuristic** could state that the action which results in the highest amount of load supplied is to be given the highest priority, 1. The KB module therefore associates a priority to each action in A_{gi}, which translates in ordered subset A'_{gi}, i.e., $A_{gi} \rightarrow A'_{gi}$. The first element in the ordered subset A'_{gi} is assigned priority 1 (in this example, the valid load pickup action that supplies the highest amount of load), the second element has a priority of 2, and so on.

4.2.2.1.4 Testing

The KB module "names" a_r as the first action from the ordered subset A'_{gi} since it has obtained the highest priority, i.e., $a_r = A'_{gi}(1)$. Then it starts the assessment of action a_r to see whether it is the solution (i.e., the "validation"-box from Figure 4.18). If action a_r is a solution, the search process stops, and action a_r is returned. Otherwise, the KB module deletes action a_r from the ordered subset A'_{gi} and repeats the same procedure for the "new" first action in the reduced subset A'_{gi}.

4.2.2.1.5 Changing current goal

If no action from the ordered subset A'_{gi} has been found as a solution, A'_{gi} is left empty. Since there are no more actions to test for the current goal g_i, the KB module will set the next goal, g_{i+1}, as the current one (e.g., to strengthen the network).

As long as no solution has been found (*sol_found*=false) and there are some goals g_i left ($i<N_{goal}$), the KB module repeats the operations described in Sections 4.2.2.1.1 through 4.2.2.1.5. If the last goal could not be reached (*sol_found*=false and $i=N_{goal}$), the KB module failed to find a solution.

* * *

Now all the above operations, which the KB module performs to find action a_r as a function of the current power system state $s(a_{r-1})$ and according to its knowledge, can be represented by the flow chart shown in Figure 4.19.

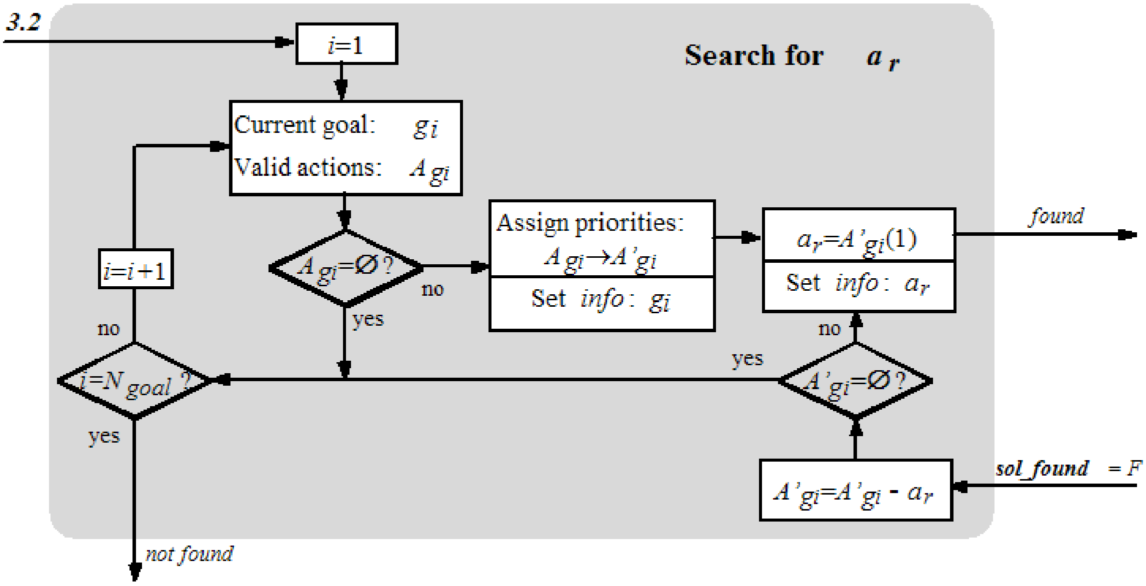


Figure 4.19 Flow chart of the search process in the KB module

When the KB module starts the search for action a_r, it sets goal g_1 as the current one. If it does not match the current power system state $s(a_{r-1})$, the subset of valid actions for goal g_1 will be empty, i.e., $A_{g1}=\emptyset$. Therefore the KB module simply moves to the next goal, g_2.

Suppose now that the current goal g_2 matches current state $s(a_{r-1})$. The KB module will determine the subset of valid actions A_{g2}. This subset might be an empty set (e.g., currently there is no valid action for goal g_2), meaning goal g_2 cannot be reached. If this is the case, the KB module moves to the next goal, g_3. If subset A_{g2} is not empty, its elements (actions) will first be assigned priorities which results in an ordered subset of valid actions A'_{g2}, and the current goal, g_2, is saved in the variable *info*. Then the assessment-loop starts on ranked elements in A'_{g2}. The first element in A'_{g2} (which is the best ranked one) is supposed to be the action a_r. It is first saved in the variable *info* and then assessed as shown in Figure 4.18. If action a_r is the solution (*sol_found*=true), it is returned from the KB module. Otherwise (*sol_found*=false), action a_r is deleted from subset A'_{g2}, making the second-ranked action the first element in the reduced subset A'_{g2}. The assessment-loop continues until (1) the solution has been found, or (2) the subset is reduced to an empty set ($A'_{g2}=\emptyset$).

If there are no actions left in A'_{g2} (i.e., $A'_{g2}=\emptyset$), the KB module moves to the next goal, g_3. This process continues until (1) the solution has been found (in the assessment-loop), or (2) there are no more goals to reach ($g_i=g_{Ngoal}$) and the solution (action a_r) has not been found.

4.2.2.2 Conclusion

Given the current power system state $s(a_{r-1})$, the KB module performs a search for action a_r, guided by the restoration strategy and different heuristic criteria, i.e., qualitative knowledge. This knowledge is organised in a definite number of goals g_i=1,…, N_{goal}, which are ranked by priority. This means that each time the KB module starts the search it checks whether the first goal matches the current power system state. If the first goal matches the current state, the KB module will determine the subset of valid actions which permit the goal to be reached, rank them and try to find the solution (action a_r) among them; otherwise, the first goal is skipped and the KB module will simply move to the second one, and so on. The tests which enable the KB module to determine whether a goal matches the current state are simple in order for **the KB module to be able to quickly "focus its attention" on the goal which is the most appropriate for the current power system state**.

When a goal is found to be the most appropriate one, the KB module ranks the actions that permit to reach the goal by priority, and tests them in a "best-first" manner. **Even if the solution cannot be found for the current goal, and provided that the current goal is not the last one, the KB module can still move to the next goal in order to search for the solution**.

In the terminology of the Search Theory, the KB module internally performs the best-first *depth* search, as illustrated by the example in Figure 4.20.a. It starts with the first goal and concentrates on the corresponding valid actions, by order of priority. If none of them is a solution, the KB module moves to the second goal. Since there are no actions to concentrate on, it proceeds to the third goal, in which it finds solution a_r as the second-ranked action.

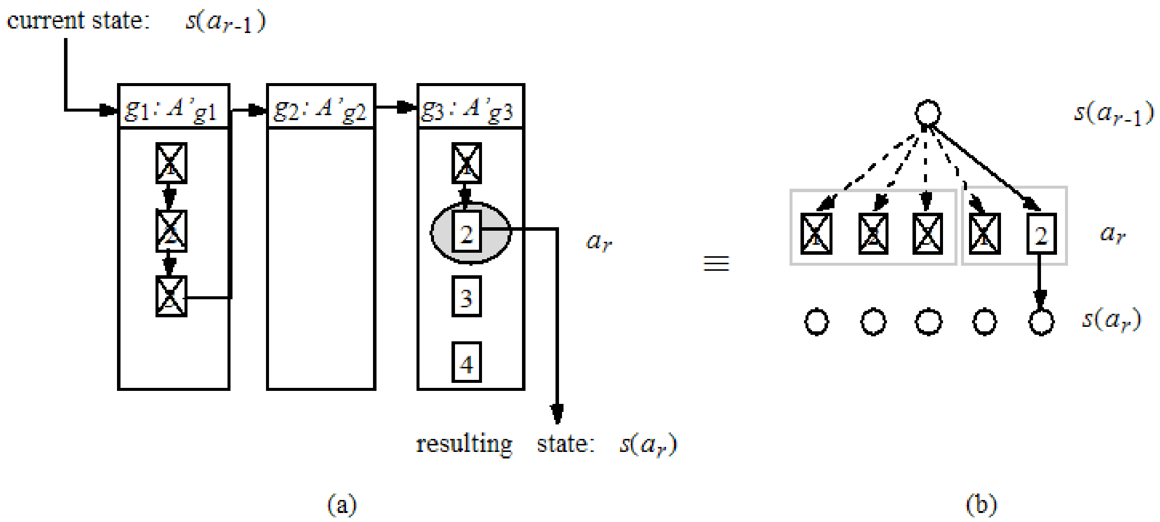


Figure 4.20 The search performed by the KB module is always the best-first search. Regarded from the inside (a), it is searched in depth, but from the outside (b), it is searched in breadth. Circles represent power system states, and rectangles represent control actions with the corresponding priority number.

Regarded from the outside of the KB module, it performs the best-first *breadth* search, as shown in Figure 4.20.b. In the light of the global restoration problem, the KB module searches for a single action a_r which can be a solution of the transition expression (4.1) for the single power system state $s(a_{r-1})$.

4.2.3 Qualitative knowledge: Strategy and heuristics

The qualitative, or symbolic, knowledge in the KB module consists of facts and rules, which are convenient for the representation of the restoration strategy (which is itself a set of facts and rules). Let us recall here that EDF has a well defined strategy for the first restoration stage (see Section 2.4.4.2). This strategy is based on creation of the regional skeleton networks

(RSNs) which are to be restored and synchronised. After that, the second restoration stage starts, in which the focus of attention is the consumer supply. For this latter one, EDF has not defined its strategy, **and one of the objectives of this work is to propose a strategy for the load pickup stage**.

The following sections show how the RSN-based strategy has been translated in facts and rules, and what kind of load pickup strategy has been proposed for the second restoration stage.

4.2.3.1 Determination of restoration stage

It is necessary to determine the restoration stage in order to know which strategy to apply. The rules are as follows:

r.1 If there are still some shutdown thermal units belonging to the RSN, the RSN is not restored and an **RSN-based strategy** is to be applied. Otherwise, a **load pickup strategy** is to be applied.

r.2 If an **RSN-based strategy** is to be applied, all available TS1 thermal units and hydraulic units *belonging to the RSN*, and all available equivalent units representing external network are defined as black-start units.

In the first restoration stage (creation of RSNs), there are two sub-stages. As was mentioned in Section 2.4.4.2, first the path must be energised to each shutdown thermal unit belonging to the RSN, and only after that are their auxiliaries to be supplied. This is formulated in rule **r.3** as follows:

r.3 If the paths to all shutdown thermal units belonging to the RSN are energised, and the nodes to which their auxiliaries would be connected belong to the same electrical zone, the **auxiliaries are to be supplied**. Otherwise, the **paths are to be energised**.

In conclusion, there are two restoration stages: (1) the first stage is the creation of the RSNs and it is divided into two sub-stages (the energisation of the RSNs and the supply of the auxiliaries of the shutdown thermal units in the RSN), and the second stage concentrates on the load restoration.

* * *

Since the knowledge in the KB module is organised into a number of goals (discussed in Section 4.2.2.2), there are three groups of goals, denoted as *G1*, *G2*, *G3* and shown in Figure 4.21. Each of them corresponds to one of the above defined restoration stages or sub-stages as follows:

- Goals in group *G1* apply when the RSN is to be energised (RSN-based strategy);
- Goals in group *G2* apply after the RSN is energised, when the auxiliaries of the shutdown thermal units belonging to the RSN are to be supplied (RSN-based strategy). A certain number of goals belong to both groups *G1* and *G2*, and have the same priority compared to other goals in groups *G1* and *G2*.

- **Goals in group** `G3` **represent the load pickup strategy proposed in this work. Normally, they apply after the RSN had been created. In addition, they have been organised in such a manner that they can apply also if the solution cannot be found through the goals in group** `G1` **or** `G2`**.**

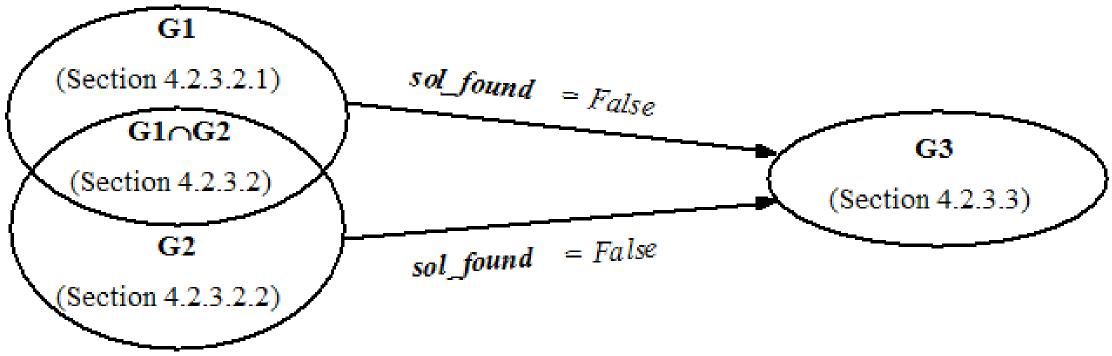


Figure 4.21 Groups of goals corresponding to restoration stages and sub-stages

In summary, the KB module focuses on the group of goals which apply to the current restoration stage (defined in rules **r.1** and **r.3**), i.e. *G1* or *G2* or *G3*. If the solution has not been found by applying the RSN-based strategy (goals *G1* or *G2*), the KB module will move to the last group of goals *G3*, which permits the solution that is not in contradiction with the RSN-based strategy to be found.

In the following sections, the above three groups of goals are described. Inside each group, the goals are introduced according to their priority (see Section 4.2.2.2). For each goal, there are two or more rules, which are also introduced by priority, as follows:

- The first rule associated with a goal tests whether the goal matches the current power system state. Referring to Figure 4.19, this test is represented by condition A'_{gi}=Ø in the outer loop. If the goal does not match the current state, the KB module moves to the next goal. Otherwise,
- The other rules associated to the goal encode the heuristics which permit the means (restoration control action) to reach the goal to be found, i.e., to find the solution. Referring again to Section 4.2.2.2, these rules are responsible for selecting, ranking and testing valid actions, i.e. they carry out the inner loop of the search process shown in Figure 4.19. If the solution has not been found after the last rule (condition A'_{gi}=Ø is true), the KB module moves to the next goal, and so on. As soon as a solution is found, the KB module stops the search and returns the solution (output *found* in Figure 4.19).

It is important to note that, in addition to the qualitative knowledge (strategy and heuristics), the rules also contain the quantitative knowledge (current power system state and valid actions). The interaction with the Dynamic simulator (DS), shown by steps 4 and 5 in Figures 4.16 and 4.17, is accomplished by the rules. A possible solution (i.e., a control action) deduced from a rule is passed to the DS as one of its commands (see Section 4.1.3). Therefore, in the rules which permit the solution to be found, the corresponding DS command will also be given.

4.2.3.2 Goals and rules for the RSN-based strategy (G1 *and* G2*)*

There are four goals that are common for both sub-stages when the RSN is to be created (i.e., these four goals belong to both the groups of goals *G1* and *G2*).

Goal 1: Check unit's reactive absorption limit

r.4 If there is no coupled unit which has reached its reactive absorption limit, move to the next goal.

r.5 Rank coupled units which have reached their reactive absorption limit by reactive power (minimal first), and increase the voltage set-point. The DS command is `$ gen avol <unit>`.

Goal 2: Synchronise TS2 unit in RSN

r.6 If there is no TS2 unit belonging to the RSN which can be coupled, move to the next goal.

r.7 Rank TS2 units belonging to the RSN which can be coupled by voltage in the node of coupling (minimal first), and synchronise. The DS command is `$ gen on <unit>`.

Goal 3: Synchronise black-start unit

r.8 If there is no black-start unit which can be coupled, move to the next goal.

r.9 Rank black-start units by voltage in the coupling node (minimal first), and synchronise. The DS command is `$ gen on <unit>`.

Goal 4: Increase frequency

r.10 If there is no electrical zone in which the frequency is *low* (heuristic **h.4** and (4.18)), move to the next goal.

r.11 First, rank electrical zones in which the frequency is *low* by frequency (minimal first). Then, for each zone rank the coupled units by spinning reserve (maximal first), and increase the power set-point. The DS command is `$ gen aset <unit>`.

If a solution has not been found at this point, the KB module moves to the next goal either in group *G1* (if the sub-stage is energisation of the RSN) or in group *G2* (if the sub-stage is the supply of the auxiliaries of the shutdown thermal units in the RSN).

4.2.3.2.1 Energisation of the RSN (group of goals G1*)*

Goal 5: Decrease frequency

r.12 If there is no electrical zone in which the frequency is *high* (heuristic **h.3** and (4.17)), move to the next goal.

r.13 First, rank electrical zones in which the frequency is *high* by frequency (maximal first). Then, for each zone rank the nodes with the unsupplied consumer loads by voltage (maximal first). Finally, for each node rank consumer loads by the unserved amount of megawatts (maximal first) and pick up loads by the unserved level (4.8) as stated in (4.12). The DS command is `$ loa aset <load> <n>`.

Goal 6: Connect inductive shunt compensator

r.14 If there is no inductive shunt compensator which can be connected, move to the next goal.

r.15 Rank inductive shunt compensators which can be connected by voltage in the connection node (maximal first), and connect. The DS command is `$ loa on <load>`.

Goal 7: Energise RSN path from black-start to TS2 unit

r.16 If there is no TS2 unit belonging to the RSN, move to the next goal.

r.17 First find all nodes that belong to electrical zones containing black-start units, and extract those nodes which are on the border of the zone – the set of these nodes will be called *n1*. Then find all nodes to which would be coupled the auxiliaries of TS2 units in the RSN – the set of these nodes will be called *n2*. Now find the shortest paths from each node in *n1* to each node in *n2* according to the minimal number of branches to energise (or the minimum electrical distance). If there are several branches between two nodes give the priority to the one which is in stand-by over those that are completely disconnected. Keep a single (the shortest) path per node in *n2*. Finally, rank the selected paths by the number of branches to connect (minimal first), and energise the fist branch from the path. The DS command is `$ bra on <branch>`.

Goal 8: Energise RSN path from black-start to shutdown thermal unit

r.18 If there is no shutdown thermal unit belonging to the RSN, whose coupling node to the network is not energised yet, move to the next goal.

r.19 This rule is similar to **r.17**. The only difference is that the *n2* contains unenergised nodes to which would be coupled the auxiliaries of the shutdown thermal units in the RSN.

Goal 9: Energise path between black-start units

r.20 If all the black-start units are in the same electrical zone, move to the next goal.

r.21 This rule is similar to **r.17**. The only difference is that sets *n1* and *n2* contain border nodes of electrical zones which contain black-start units.

Goal 10: Pick up load in RSN

r.22 If there is no unserved consumer load belonging to the RSN, move to the next goal.

r.23 First rank the nodes with the consumer loads by voltage (maximal first). Then, for each node rank the consumer loads by $tan(\varphi)$ (4.9) (maximal first). Finally, rank loads by their amount of unserved megawatts (maximal first) and pick up loads by their unserved level (4.8) as stated in (4.12). The DS command is `$ loa aset <load> <n>`.

If the solution has not been found at this point, the KB module moves to the first goal in group *G3* (Section 4.2.3.3).

4.2.3.2.2 Supply of auxiliaries (group of goals G2*)*

Goal 11: Supply auxiliaries of the shutdown thermal unit in RSN

r.24 If there is no shutdown thermal unit in the RSN whose auxiliaries can be supplied, move to the next goal.

r.25 Rank shutdown thermal units by voltage in the node where their auxiliaries would be coupled (maximal first), and supply auxiliaries. The DS command is `$ gen sby <unit>`.

EDF's strategy states that all the auxiliaries of the shutdown thermal units in the RSN should be supplied more or less simultaneously once the RSN is energised. In addition, the strategy prohibits any consumer load pickup which could interfere with the supply of the auxiliaries.

However, it may happen that the auxiliaries cannot be supplied simultaneously (i.e., the solution has not been found after Goal 11), so that some other solution should be searched for. This alternative solution must not be in contradiction with EDF's RSN-based strategy. Therefore, we have proposed to first search for an action which is certain not to result in any consumer load pickup, as follows:

Goal 12: Energise path between black-start units

r.20 (see Goal 9 in Section 4.2.3.2.1)

r.21 (id.)

If a solution has been found after Goal 12, the number of electrical zones (i.e., islands) in the RSN is likely to decrease. This renders the RSN more robust and the supply of the auxiliaries can therefore continue more quickly. However, if a solution has not been found at this point (e.g., the RSN is already a single electrical zone; no branch in the RSN can be energised because of the over-voltage problems), the KB module moves to the first goal in group *G3* (Section 4.2.3.3).

4.2.3.3 Goals and rules for load pickup strategy (G3)

If the KB module activates this group of goals and rules *because a solution could not be found in group G2*, any action which results directly or indirectly in consumer load pickup is forbidden. For the purposes of readability, the goals or rules to which this limitation applies will be marked by an asterisk "*", without stating the condition explicitly; the asterisk is to be ignored otherwise.

Goal 13: Check unit's reactive absorption limit

r.4 (see Goal 1 in Section 4.2.3.2)

r.5 (id.)

Goal 14: Synchronise unit

r.26 If there is no unit in the network which can be coupled, move to the next goal.

r.27* Rank all units which can be coupled by voltage in the coupling node (minimal first), and synchronise. The DS command is `$ gen on <unit>`.

Goal 15: Increase frequency

r.10 (see Goal 4 in Section 4.2.3.2)

r.11 (id.)

Goal 16*: Decrease frequency

r.12 (see Goal 5 in Section 4.2.3.2.1)

r.13 (id.)

Goal 17: Increase voltage

r.28 If there is no electrical node in which the voltage is *low* (heuristic **h.6** and (4.20)), move to the next goal.

r.29 Find electrical zones which contain nodes with *low* voltage. For each zone, find all disconnected branches which can extend the zone without load pickup. If there is no such branch, move to the next rule.

Otherwise, rank branches by voltage in the node from which the zone would be extended (maximal first), and connect. The DS command is `$ bra on <branch>`.

r.30 Find electrical zones which contain nodes with *low* voltage. For each zone, find all inductive shunt compensators which can be disconnected. If there is no such compensator, move to the next rule.

Otherwise, rank inductive shunt compensators by voltage in the node in which they are coupled (minimal first), and disconnect. The DS command is `$ loa off <load>`.

r.31 Find electrical zones which contain nodes with *low* voltage. For each zone, find all capacitive shunt compensators which can be connected. If there is no such compensator, move to the next rule.

Otherwise, rank capacitive shunt compensators by voltage in the node in which they would be coupled (minimal first), and connect. The DS command is `$ loa on <load>`.

r.32 Find electrical zones which contain nodes with *low* voltage. For each zone, find all production units which can have their voltage set-point increased. If there is no such unit, move to the next goal.

Otherwise, rank units by voltage in the node in which they are coupled to the network (minimal first), and increase the voltage set-point. The DS command is `$ gen avol <unit>`.

Goal 18: Supply auxiliaries of shutdown thermal unit

r.33 If there is no shutdown thermal unit in the network whose auxiliaries can be supplied, move to the next goal.

r.34 Rank shutdown thermal units by voltage in the node where their auxiliaries would be coupled (maximal first), and supply the auxiliaries. The DS command is `$ gen sby <unit>`.

Goal 19: Energise path from coupled to TS2 unit

r.35 If there is no TS2 unit in the network, move to the next goal.

r.36* This rule is similar to **r.17**. The only difference is that *n1* contains border nodes of zones with all coupled units *in the network*, and *n2* contains nodes to which TS2 units **from the whole network** would be coupled.

Goal 20: Energise path from coupled to shutdown thermal unit

r.37 If there is no shutdown thermal unit whose coupling node to the network is not yet energised, move to the next goal.

r.38* This rule is similar to **r.17**. The only difference is that *n1* contains border nodes of zones with all coupled units **in the network**, and *n2* contains unenergised nodes to which would be coupled shutdown thermal units **from the whole network**.

Goal 21: Synchronise two islands

r.39 If there is no branch that can couple two electrical zones, move to the next goal.

r.40 Rank branches by voltage difference between the coupling nodes (minimal first), and connect. The DS command is `$ bra on <branch>`.

Goal 22: Increase transmission capacity towards consumer loads

r.41 If there is no autotransformer or TCUL transformer that can strengthen the network, move to the next goal.

r.42 Rank transformers by rated power (maximal first), and connect. The DS command is `$ bra on <branch>`.

Goal 23*: Pick up load

r.43 If there is no unserved consumer load that can be picked up, move to the next goal.

r.44 First rank the nodes with the unserved consumer loads by voltage (maximal first). Then, for each node rank the consumer loads by their amount of unserved megawatts (maximal first) and pick up loads by their unserved level (4.8) as stated in (4.12). The DS command is `$ loa aset <load> <n>`.

Goal 24: Strengthen network

r.45 If there is no branch that can strengthen the network, move to the next goal.

r.46 Rank branches by maximal voltage on their ends (minimal first), and connect. The DS command is $ bra on <branch>.

Goal 25*: Supply load zone

r.47 If there is no branch that can extend the network and pick up a load zone, move to the next goal.

r.48 Rank branches by the amount of load they would supply (maximal first), and connect. The DS command is $ bra on <branch>.

Goal 26: Extend network

r.49 If there is no branch that can extend the network, **the solution has not been found**.

r.50* Rank branches by voltage in the extension node (minimal first), and connect. The DS command is $ bra on <branch>.

4.2.3.4 Synthesis and discussion

In this section, the above goals will be put together per restoration stage or sub-stage. In other words, the corresponding restoration strategies, extended with some heuristics, will appear as they are implemented in the KB module.

4.2.3.4.1 Energisation of RSN

Let us begin by discussing Figure 4.22, which shows by priority the goals that translate EDF's strategy based on the creation of the RSNs (extended with some heuristic criteria), and in particular the energisation of the RSNs.

When the initial state for the restoration is a total blackout, there will probably be no unit which has reached the reactive absorption limit, and Goal 1 will be skipped. However, as the restoration progresses (i.e., connection of unloaded lines), such a unit might appear, and the KB module will first search for an action that can change this state. This goal is based on heuristics, and has a general scope (it is the same as Goal 13).

RSN-based strategy: energization of RSN

Check unit's reactive absorption limit	(g_1)
Synchronize TS2 unit in RSN	(g_2)
Synchronize black-start unit	(g_3)
Increase frequency	(g_4)
Decrease frequency	(g_5)
Connect inductive shunt compensator	(g_6)
Energize RSN path from black-start to TS2 unit	(g_7)
Energize RSN path from black-start to shutdown thermal unit	(g_8)
Energize path between blackstart units	(g_9)
Pick up load in RSN	(g_{10})
~~Check unit's reactive absorption limit~~	(g_{13})
Synchronize unit	(g_{14})
~~Increase frequency~~	(g_{15})
~~Decrease frequency~~	(g_{16})
Increase voltage	(g_{17})
Supply auxiliaries of shutdown thermal unit	(g_{18})
Energize path from black-start to TS2 unit	(g_{19})
Energize path from black-start to shutdown thermal unit	(g_{20})
Synchronize two islands	(g_{21})
Increase transmission capacity towards consumer loads	(g_{22})
Pick up load	(g_{23})
Strengthen network	(g_{24})
Supply load zone	(g_{25})
Extend network	(g_{26})

Figure 4.22 The EDF's strategy for the creation of RSNs (first sub-stage), extended with the heuristics. g_i corresponds to definition from Section 4.2.2.1.1, and represents "Goal i" from Sections 4.2.3.2 and 4.2.3.3.

Goal 2 applies only if there is a TS2 unit in the RSN which can be synchronised, i.e., if the node to which its auxiliaries are to be coupled is energised. Therefore, there must be either a coupled TS1 unit in the same production site as the TS2 unit, or an energised path to the auxiliaries of TS2 unit. Since Goal 1 does not focus on the coupling of a unit in the first search process, so Goal 2 will be skipped. Later on, as soon as one of the above conditions is satisfied, the coupling of a TS2 unit will have the highest priority.

Goal 3 is the one which will certainly apply in the very first search process, because a black-start unit must be coupled first, which will permit paths to other units in the RSN which need the supply for their auxiliaries to be energised. Note that, if a black-start unit is a TS1 unit and there is a TS2 unit in the same production site, the KB module will activate Goal 2 in the next search process.

Goals 4 and 5 are based on heuristics and have general scope (they are same as Goals 15 and 16, respectively). They serve to check whether it is necessary to do something in order to improve the frequency in the network. For instance, when the consumer loads start to be picked up, the frequency in the network will decrease. Since the automatic frequency/power (f/P) secondary control is locked in restoration, it is necessary to keep the frequency within admissible limits manually. Goal 4 permits the power set-point of a unit to be increased (see Section 4.1.1.1). When the power set-point of a unit has been increased in the beginning of the restoration, there are usually a few units that are coupled in the RSN. Therefore, the increase of the power set-point may result in a very high frequency increase. Goal 5 permits some load to be picked up in order to decrease the frequency.

Goal 6 permits inductive compensation devices in the RSN to be connected, if they exist, before energising the lines.

Goals 7 and 8 reflect the requirement stated in EDF's strategy to first energise paths towards TS2, and then towards shutdown thermal units in the RSN. Note that:

- when the coupling node of a TS2 unit becomes energised, in the next search process, the KB module will first activate Goal 2;
- as soon as the path towards the **last** shutdown thermal unit in the RSN is energised, the RSN is said to be energised, i.e., the **first sub-stage of the RSN-based strategy is terminated**;
- if in the second sub-stage of the RSN-based strategy the coupling node of a shutdown thermal unit in the RSN happens to become unenergised (e.g., the protection opened a line in the energising path), the KB module will again activate this group of goals.

If the paths cannot be energised towards all shutdown thermal units in the RSN, and if there are at least two islands in the RSN, Goal 9 permits paths between the islands to be energised in order to reinforce the robustness of the RSN.

Goal 10 is activated if none of Goals 1 through 9 can apply, and permits to pick up some load in the RSN. Note that it often happens that the unloaded lines cannot be closed because of overly high voltage; therefore, picking up some load will improve voltage conditions in the RSN.

If even Goal 10 does not permit the solution to be found, the KB module will focus on the next group of goals (Goals 13 through 26), which apply not only to the RSN but to the whole network. According to our experience from simulations for three different power systems, this has never been the case. However, it has happened that the KB module activated the last group of goals because it failed to find the solution in the sub-stage of the RSN-auxiliaries' supply. Since the similar discussion would apply in both cases, the goals belonging to the load pickup strategy will be discussed in the next section.

4.2.3.4.2 Supply of auxiliaries in RSN

Figure 4.23 shows by priority the goals that translate EDF's strategy based on the creation of the RSNs (extended with some heuristic criteria), and in particular the supply of the auxiliaries of shutdown thermal units in the RSNs. The constraint imposed by this strategy is that no consumer load pickup should interfere with the supply of auxiliaries.

In this restoration stage, Goal 1 may apply, since a number of lines are energised and the RSN is lightly loaded. Therefore, the KB module will first search for an action that can move the unit in question from its limit.

If a TS2 unit could not be synchronised in the first sub-stage (e.g., because of voltage problems), Goal 2 makes the coupling of a TS2 unit remain a high priority task. Similar discussion applies to Goal 3.

RSN-based strategy: supply of auxiliaries

	Check unit's reactive absorption limit	(g_{1})
	Synchronize TS2 unit in RSN	(g_{2})
	Synchronize black-start unit	(g_{3})
	Increase frequency	(g_{4})
	Supply auxiliaries of shutdown thermal unit in RSN	(g_{11})
	Energize path between blackstart units	(g_{12})
	~~Check unit 's reactive absorption limit~~	(g_{13})
*	Synchronize unit	(g_{14})
	~~Increase frequency~~	(g_{15})
*	~~Decrease frequency~~	(g_{16})
	Increase voltage	(g_{17})
	Supply auxiliaries of shutdown thermal unit	(g_{18})
*	Energize path from black-start to TS2 unit	(g_{19})
*	Energize path from black-start to shutdown thermal unit	(g_{20})
	Synchronize two islands	(g_{21})
	Increase transmission capacity towards consumer loads	(g_{22})
*	~~Pick up load~~	(g_{23})
	Strengthen network	(g_{24})
*	~~Supply load zone~~	(g_{25})
*	Extend network	(g_{26})

Figure 4.23 EDF's strategy based on the creation of the RSNs (second sub-stage) extended with the heuristics. g_i corresponds to definition from Section 4.2.2.1.1, and represents "Goal i" from Sections 4.2.3.2 and 4.2.3.3.

When the KB module activates Goal 4 for the first time in the auxiliaries' supply stage, it is not likely to match the current power system state (since the last action in the RSN-energisation stage was certainly an action that closes a line, which does not decrease the frequency). However, later on, when some auxiliaries are supplied, the frequency will decrease (since the auxiliaries are also loads), and it might be necessary to improve the frequency in the RSN by increasing the power set-point of a coupled unit.

Finally, Goal 11 is the principal one in this stage – it permits auxiliaries of the shutdown thermal units in the RSN to be supplied. As discussed in Section 4.2.3.2.2, it may happen that the auxiliaries cannot be supplied more or less simultaneously, but the KB module is still expected to find a feasible solution, which is not in contradiction with the constraints imposed by EDF's strategy. Therefore, it is worthwhile to reinforce the RSN itself, if it contains at least two islands, before trying to find a solution on the level of the whole network. Goal 12 permits paths between two islands in the RSN to be energised, which leads to the decrease of the number of islands and, consequently, improves the regulation capacity of the RSN.

However, if the solution could not be found even after Goal 12, the KB module will activate the last group of goals, which belong to the load pickup strategy. Goal 13 will always be skipped in this case because it is the same as Goal 1. Goal 14 allows a unit from the whole network to be synchronised (e.g., a TS1 or hydraulic unit). There is a limitation to Goal 14 in this case (marked by "*"): the valid actions are only those which do not result in a load zone pickup (see Section 4.1.3.1). If there is no such unit, the KB module moves to the next goal.

In this case, Goals 15 and 16 are always skipped: Goal 15 because it is the same as Goal 4, and Goal 16 (marked by "*") because it is forbidden to pick up a consumer load in the sub-stage of the supply of auxiliaries.

In this restoration stage, the voltage in the network is rather high. However, Goal 17 might apply since the voltage may be low in coupling nodes of the shutdown units' auxiliaries which are supplied by a unit *at the same production site* (because the coupled unit has its voltage set-point set to minimum value before the restoration starts).

Goal 18 permits the auxiliaries of any shutdown thermal unit in the network to be supplied. When this goal is activated for the first time in this stage, it will be skipped since there is no energised path out of the RSN.

When the KB module moves to Goal 19 because it could not find the solution in the RSN auxiliaries supply sub-stage, the objective is to extend the RSN towards other TS2 units in the network (if they exist) without picking up any load zone on the way (marked by "*"). If such a unit exists, the priority is to couple it as soon as possible which will also enhance the robustness of the RSN (it will have one additional coupled unit). Therefore, in the later search process, it will be easier to continue with the supply of auxiliaries in the RSN itself.

Goal 20 also results in an extension of the RSN, but towards other shutdown thermal units in the network. When activated in this restoration stage, the same limitation on load zone pickup applies as in Goal 19 (marked by "*"). The objective here is to energise paths to other shutdown thermal units. As soon as this is the case, Goal 18 will apply.

From our experience in simulations with three different power systems, the last goal that has ever been activated in this stage was Goal 20. Note, however, that all the following goals which concern the consumer load pickup are marked by "*", and therefore either prohibited completely (Goals 23 and 25) or limited to those actions which do not result indirectly in a load zone pickup (Goal 26).

4.2.3.4.3 Load pickup strategy

Figure 4.24 shows by priority the goals that translate the load pickup strategy proposed in this work. It has already been shown how it permits a solution to be found which is not in contradiction with EDF's RSN-based strategy when the latter cannot be applied (in the same manner, it reflects general restoration guidelines for the networks in which there is no notion of the RSN at all). In this section, we concentrate on the case where the RSN has successfully been created and the load pickup stage itself may start.

Every time the KB module activates this group of goals, the coupled units will first be checked on their reactive absorption limits by Goal 13. It is a general purpose goal and is the same as Goal 1. It has been numbered differently for purposes of readability, in order to see it clearly as the highest priority goal in the load pickup strategy.

The next priority is given, by Goal 14, to the synchronisation of units in the whole network as soon as this is possible. At the beginning of this stage, the units to be coupled will be the hydraulic ones, and as the restoration advances, the shutdown thermal units will have their auxiliaries supplied and will also become ready for coupling.

Goals 15 and 16 are (as is Goal 13) general purpose goals, which focus on frequency conditions in the network. They are the same as Goals 4 and 5, respectively, and have already been discussed. They have been numbered differently (as has Goal 13) to clearly see their priority in the load pickup strategy.

Load pickup strategy

Check unit's reactive absorption limit	(g_{13})
Synchronize unit	(g_{14})
Increase frequency	(g_{15})
Decrease frequency	(g_{16})
Increase voltage	(g_{17})
Supply auxiliaries of shutdown thermal unit	(g_{18})
Energize path from black-start to TS2 unit	(g_{19})
Energize path from black-start to shutdown thermal unit	(g_{20})
Synchronize two islands	(g_{21})
Increase transmission capacity towards consumer loads	(g_{22})
Pick up load	(g_{23})
Strengthen network	(g_{24})
Supply load zone	(g_{25})
Extend network	(g_{26})

Figure 4.24 The strategy for the load pickup stage proposed in this work. g_i corresponds to the definition from Section 4.2.2.1.1, and represents "Goal i" from Section 4.2.3.3.

A new heuristics-based goal, Goal 17, serves for improving voltage conditions in the network. The pickups of consumer loads will in the beginning of this stage lead to the improvement of the voltages in the network, which are rather high just after the RSN had been created. However, load pickups will continue to make the voltages in the network decrease. Therefore, we propose to do something before the voltages reach their limit values, i.e., Goal 17 will be activated as soon as there is a node in the network which reaches the heuristic limit given by (4.20).

If no unit can be coupled, and if the power system is not close to its operating limits, the KB module will focus on Goal 18, which permits to supply the auxiliaries of the shutdown thermal units, if there are any. This will make possible the coupling of such units as soon as their thermal circuits conditions permit the synchronisation. In other words, in some later search, these units could be coupled through Goal 14.

Goals 19 and 20 permit the energisation of the paths towards any TS2 and shutdown thermal unit in the network, respectively. The higher priority is given to TS2 units which, as soon as the voltage appears in the node of their coupling to the network, could be synchronised through Goal 14. Similar discussion applies to shutdown units, which will later have their auxiliaries supplied (Goal 18) and then be ready for coupling (Goal 14). Note here that:

- if there is a load zone on the energisation path, it will be supplied, and
- the initial RSN is continuously being extended.

The next priority (Goal 21) is given to the synchronisation of electrical islands in the network, if they exist. Otherwise, Goal 22 results in the increase of transmission capacity of the network, towards the consumer loads.

Goal 23 permits the consumer loads which are not in load zones to be picked up, i.e., those loads that are shed after the blackout. The KB module will try to pick up the maximal amount of megawatts in the node which has the highest voltage.

If no consumer load can be picked up, it is probable that the second circuits in the network should be connected, i.e., the overall transmission capacity of the network should be improved. This is the objective of Goal 24.

If there are some load zones that have not been supplied along with the energisation of the paths (Goals 19 and 20), this will be done through Goal 25.

Finally, if none of the above goals matches the current power system state (e.g., all the loads in the currently energised network are supplied), the network is to be extended (Goal 26). However, **if the network cannot be extended, and there are still some unserved loads in the network (i.e., Goal 26 does not provide the solution), it is said that the KB module failed to find the solution, since there are no other goals to activate**.

* * *

The common characteristic for all the stages is that there are heuristic criteria concerning the operating limits, which are translated into goals with the highest priorities. In other words, the current power system will first be tested on operating limits, and, if it is not "too close" to its limits, the strategy itself will be applied. Otherwise, the KB module will try to search for an action which "moves" the power system away from its limits. It can be seen that the goal "Decrease voltage" does not exist, which should be activated when there is a node whose voltage is *high*, i.e., the goal based on heuristic limits (4.19). The action in this case would be to pick up some load. The voltages in the network are high in the first restoration stage, but the priority is to energise the RSN and not to pick up loads. Since some load in the RSN will be picked up anyway if no energisation is possible (see Goal 10), it was not necessary to explicitly define the goal "Decrease voltage".

Finally, it is important to note that **the proposed load pickup strategy permits a solution to be found not only when EDF's RSN-based strategy cannot be completed, but also when it cannot be applied at all (i.e., when there is no black-start unit in the RSN)**. The load pickup strategy considers the whole network and encodes, by priority, the strategic goals.

4.2.4 Implementation

The choice of the implementation tool for the KB module was guided by several criteria:

1. the nature of the problem to be coded, i.e., symbolic knowledge representing the restoration strategy and the heuristics;

2. the ease of implementation from the point of view of the developer, i.e., the speed of building the prototype;
3. the availability of debugging facilities for both the development and the testing of the prototype;
4. the ease of modifications and maintenance of the code, and
5. the ease of interfacing with existing applications (in particular the Dynamic simulator, which is written in C programming language).

The first two criteria brought us to the conclusion that the easiest way would be to use a rule-based tool, with the integrated inference engine. Therefore, the evaluation of 10 commercial rule-based tools has been made [EDF94c], and the choice has been fixed on the Neuron Data product NEXPERT *OBJECT*™ (NXP).

NXP is a development tool for knowledge-based applications, which completely satisfies criteria 3, 4 and 5. It is an object-oriented [Dil94] tool with a powerful inference engine. It provides the developer with the Graphical User Interface (GUI) which permits the visualisation of rules and objects, cross references and inference status at any moment, and to modify any of them on-line. NXP is, in fact, a library of C functions with the GUI, which facilitates the embedding or interfacing with an existing application.

Therefore, the KB module has been implemented mostly in NXP, with some C-functions called from the rules. For instance, the Dynamic simulator is a C-function called from rules. Another example is a C-function in which different path-search algorithms have been implemented according to [Sed83] (depth-first, breadth-first, Prim's algorithm for minimum spanning tree and Dijkstra's algorithm for weighted shortest path): this function is called from rules associated to goals which focus on the path energisation. Finally, all the power system state data in the KB module (represented by classes and objects) are initialised with the corresponding values from the power system state C-structure (see Section 4.1.6.1) and kept consistent through calls to C-functions.

From the point of view of the developer, we have found NXP to be an excellent tool for developing and testing the prototype of the KB module. However, for a real-time application, our conviction is that NXP should be completely replaced with a procedural programming language, such as C or C++, for the purposes of speed and portability.

4.2.5 Conclusion

The KB module's function is to assess an action suggested by the operators (in validation mode) or to find itself an action to suggest to the operators (in search mode). If the current power system state is $s(a_{r-1})$, the action in question is denoted as a_r in both modes. In order for the KB module to set the variable *sol_found* to true, action a_r must be a solution of the transition equation (4.1) for $i=r$, i.e., it must lead the power system from its current state $s(a_{r-1})$ to a new steady state $s(a_r)$ with no violations of the operating limits. Input and output of the KB module are shown in Figure 4.25.

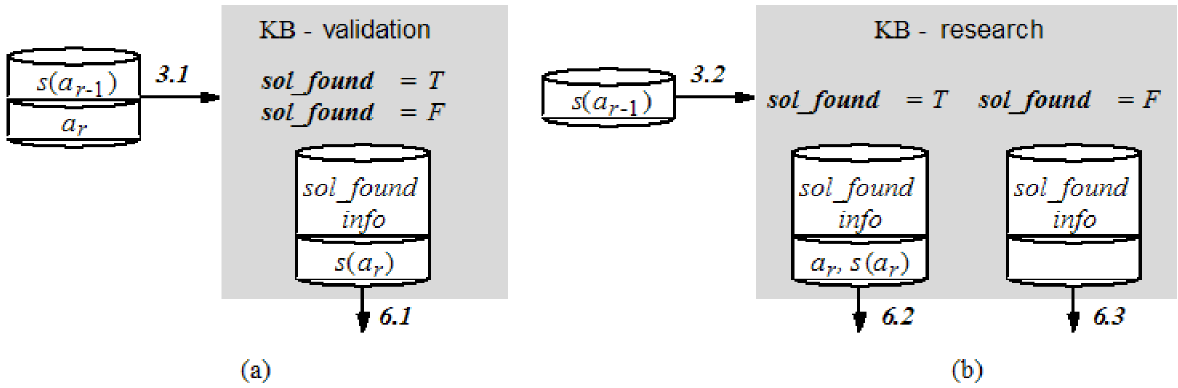


Figure 4.25 Input and output of the KB module when operating in (a) validation mode, and (b) search mode

4.2.5.1 Input and output

In validation mode (Figure 4.25.a), the KB module only tests whether action a_r, provided by the operators (step 3.1), is a solution. If this is the case, the KB module approves action a_r by setting *sol_found* to true; otherwise, action a_r is disapproved and *sol_found* is set to false. The report on the performed assessment is saved in variable *info*. Both *sol_found* and *info* are returned from the KB module, together with the power system state $s(a_r)$ resulting from action a_r, whether this latter be the solution or not (step 6.1).

In search mode (Figure 4.25.b), the KB module is expected to find action a_r which is a solution. To do that, the KB module performs a heuristic best-first search in breadth, described in Section 4.2.2. The search is guided by the restoration strategy and the heuristic criteria, which represent the symbolic knowledge. During the search, variable *info* is regularly filled with information on the search process. If the search is not successful, the KB module sets the variable *sol_found* to false and returns it together with *info* (step 6.3); otherwise, *sol_found* is set to true and returned in step 6.2 with the variable *info*, the solution (action a_r) and the resulting state ($s(a_r)$).

4.2.5.2 Search and knowledge

The symbolic knowledge used for the heuristic search is organised in strategic goals, each of which has a certain priority. This enables the KB module to focus the search on the first goal it encounters and which matches the current power system state.

EDF's strategy for the first restoration stage has been translated in goals and extended with some heuristic goals. The strategy for the second restoration stage (load pickup stage) has been proposed in this work, and also translated in strategic and heuristic goals. The organisation of goals, which reflects the load pickup strategy, and the principle of search are such that the proposed strategy may apply in the following three cases:

1) after the first restoration stage has been successfully completed according to EDF's strategy based on the creation of regional skeleton networks (RSNs);

2) when the first restoration stage cannot be completed with the RSN-based strategy, i.e., either the RSN cannot be energised or the auxiliaries of shutdown thermal units in it cannot be supplied at once; and
3) when there is no black-start unit in the RSN, which puts into question the application of the RSN-based strategy.

In addition, the proposed strategy can be applied to a power system which has no notion of the RSNs, i.e., as a general purpose restoration strategy. An example will be shown and discussed in Section 4.4.3.4.

4.3 Dialogue&control (D&C) module

The Dialogue&control (D&C) module is independent of the restoration strategy and of the power system in question. It is the uppermost layer of the Reasoning kernel, as shown in Figure 4.26. By "putting" Figure 4.26 into the "Reasoning kernel"-box in Figure 3.1, which shows the entire DAFFOR, the following remarks can be made:

1. The D&C module is the only "intermediate" between the operators and the reasoning itself (i.e., the KB module);
2. The D&C module performs data exchange between the Reasoning and the Real Time Update kernel, inside DAFFOR. The stars *rt* in Figure 4.26 show the points where the interactions between the two kernels exist. Since in this chapter the focus of attention is the stand-alone DAFFOR, i.e., the Reasoning kernel, these interactions will be ignored until Section 5.2.4.2 in the next chapter.

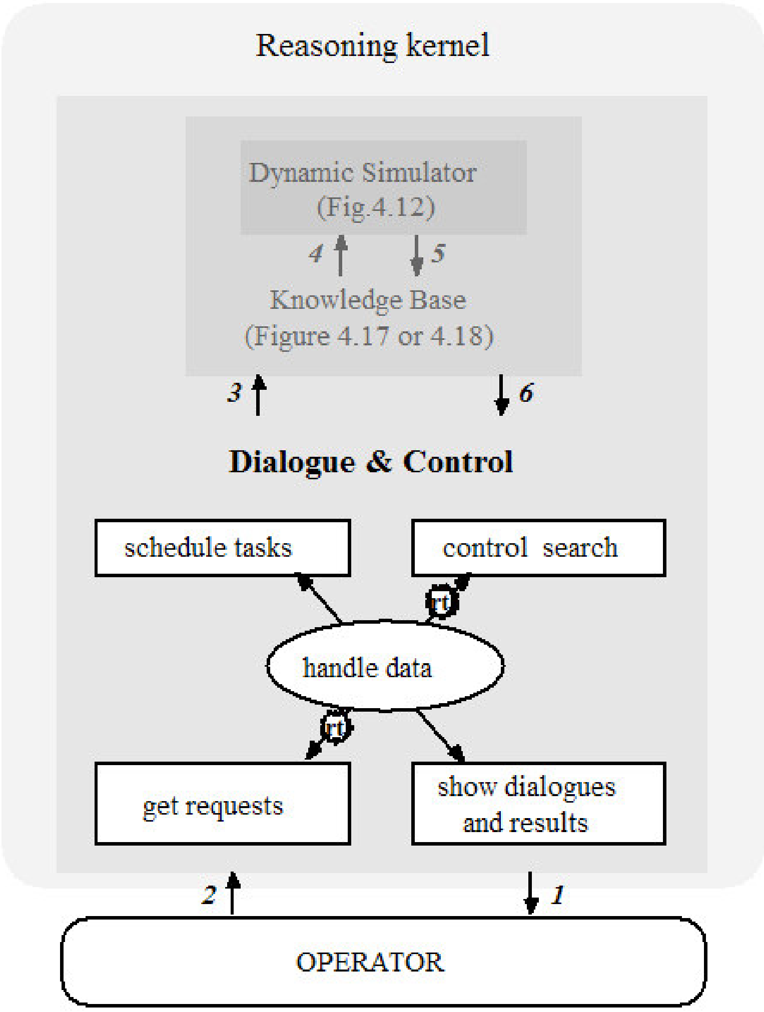


Figure 4.26 The D&C module as coordinator between the users and the reasoning (the stars towards boxes "get requests" and "control search" have to do with real-time operation, *rt*)

Different functions of the D&C module are represented in Figure 4.26, and they are all dependent on each other:

- On the user side, the D&C module shows the appropriate set of dialogue options and the results of the reasoning (step 1), and accepts the users' requests (step 2);
- On the reasoning side, it schedules the tasks corresponding to users' requests (step 3), and controls the process of search for an action (step 6);
- It provides and keeps the data which is used by other modules of the **Reasoning** kernel consistent, and represents DAFFOR's internal data interface in real-time operation, i.e., between the Reasoning and the **Real time update** kernel (stars around data handling function); the latter one will be introduced in Chapter 5.

Therefore, **the D&C module can be considered globally as DAFFOR's operating environment and user interface**. It is responsible for the coordination of different functions, internal to DAFFOR, with the aim of meeting functional requirements enumerated in Sections 3.1.1 and 3.1.2. Here we concentrate on those that concern interactivity and guidance:

- in search mode, the capability of suggesting a control action as a function of the current power system state and according to the restoration strategy,
- in validation mode, the capability of assessing a control action suggested by the operators, and
- in both modes, the capability of showing "why" and "how" DAFFOR arrived at the current conclusion.

In the following sections, first the basic functionality of the D&C module will be described, which also represents DAFFOR as it appears on the user side in both stand-alone and real-time operation. After that, some extensions will be given to the search control function of the D&C module, which enhance the guidance capability of DAFFOR and the related data handling. Finally, the implementation issues will be discussed.

4.3.1 DAFFOR's interactivity

The D&C module provides 5 dialogue options, each of which **always has the same meaning for the operators**:

- by choosing the option search, the operators ask DAFFOR to suggest an action;
- by choosing the option validation, the operators ask DAFFOR to (1) show valid actions, and then (2) to assess the one they suggest;
- by choosing the option accept, the operators tell DAFFOR that they will execute the current action on the power system;
- by choosing the option explain, the operators ask DAFFOR to explain its current conclusion; and,
- by choosing the option end, the operators exit DAFFOR.

These 5 options provide the operators with a simple, straightforward "vocabulary" which is easy to learn. As is shown in the following sections, the above options are combined in 4 dialogues, which represent the strict minimum for both the operators' interaction effort and DAFFOR's appropriate functioning.

The interactivity attribute of DAFFOR will be shown by describing the basic operations the D&C module performs in both validation and search modes, with particular attention given to the interaction with the operators and the corresponding data handling.

4.3.1.1 validation mode

Before any interaction, the D&C module must create the data set which describes the current power system state (see Section 4.1.6.1), denoted as $s(a_{r-1})$. This is represented in the "initialisation"-box in Figure 4.27. Since it is up to the operators to choose the mode, the D&C module shows the **initial dialogue in step 1.0**.

If the operators' request is validation (step 2.1), the D&C module first determines which control actions are valid (see Sections 4.1.3.2 through 4.1.3.5) for the current power system state. Then it shows the valid actions to the operators (step 1.1), who are expected to choose one of them (step 2.2)[13]. According to the transition expression (4.1), the D&C module denotes the chosen action as a_r and schedules its assessment for the current state $s(a_{r-1})$ (step 3.1). Then the KB module starts the operation in validation mode, as described in Section 4.2.1.1. When it returns (step 6.1), the D&C module checks whether the solution has been found, and shows the appropriate **result-dialogue**, which contains conclusion (i.e., result) and possible options as follows:

[13] If an action is invalid at a given moment, there is a reason (e.g., if line B9 is connected, it cannot be reconnected). By showing valid actions only, the D&C module shows the operators the actions that DAFFOR can assess, and therefore decreases the time of useless interaction.

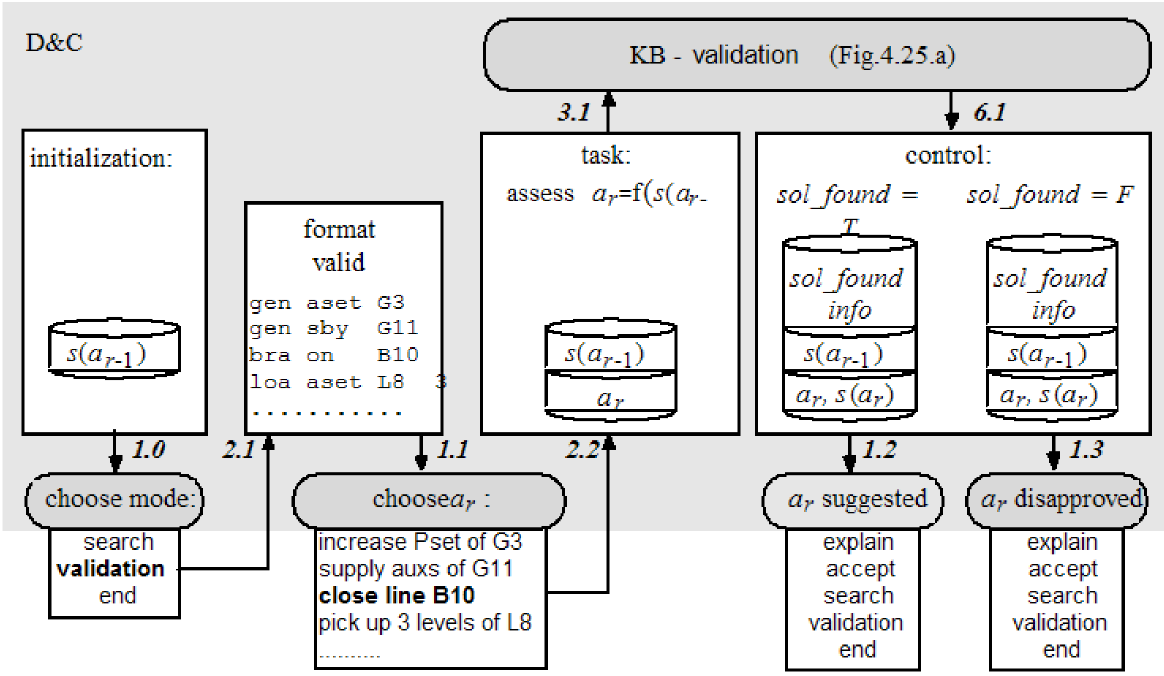


Figure 4.27 Basic functionality of the D&C module (and DAFFOR) in validation mode

- If a solution has been found, the conclusion is that action a_r has been approved. Through the **result-dialogue 1.2**, DAFFOR suggests to the operators to execute action a_r on the power system, and then waits for an answer.
- Otherwise, the conclusion is that action a_r has been disapproved. Through the **result-dialogue 1.3**, DAFFOR suggests to the operators NOT to execute action a_r on the power system, and then waits for an answer.

First of all, the operators may ask for explanation (explain), in which case the D&C module will first display the content of the variable *info* (filled in by the KB module), and then the same dialogue without option explain.

By choosing the option accept, the operators inform DAFFOR that **they will execute** action a_r on the power system whether it has been approved or not. Note that the decision is left to the operators. Therefore, the D&C module will set state $s(a_r)$ (resulting from action a_r) as the current one and reset all the other data it contains. The current state $s(a_r)$ would then figure in "initialisation"-box (instead of $s(a_{r-1})$), and the D&C module shows the initial dialogue in step 1.0.

If the **operators do not intend to execute** action a_r on the power system, they should decide whether they want DAFFOR to assess another action (validation) or to search itself for an action to suggest (search). In any case, the D&C module first disables action a_r (and keeps it disabled as long as the current state is $s(a_{r-1})$), then resets all the data it contains except the current state $s(a_{r-1})$, and then continues in the chosen mode (validation or search) as if the option were chosen from the initial dialogue in step 1.0.

Note that "reject" option does not exist in DAFFOR. If it had existed, then when chosen it would have led to an "initialisation"-box and then to an initial dialogue in step 1.0. In that case, there would have been one more interaction with the operators. This additional interaction is useless, since the mode-related options (validation and search) chosen from a result-dialogue (step 1.2 or 1.3) have exactly the same effect: they make the D&C module perform some internal data handling, while keeping the same meaning for the operators.

4.3.1.2 search mode

Let us consider now the case when the operators choose the option search from the initial dialogue, as indicated by step 2.3 in Figure 4.28. According to transition expression (4.1), the D&C module denotes a_r as the action to be found and schedules the search for it as a function of the current state $s(a_{r-1})$ (step 3.2). Then the KB module starts the operation in search mode, as described in Section 4.2.1.2. When it returns (steps 6.2 and 6.3), the D&C module checks whether the solution has been found, and shows the appropriate **result-dialogue**, which contains the conclusion (i.e., result) and possible options as follows:

- If a solution has been found, the conclusion is that action a_r has been found. Through **result-dialogue 1.2**, DAFFOR suggests to the operators to execute action a_r on the power system, and then waits for an answer.
- Otherwise, the conclusion is that action a_r has not been found. Through **result-dialogue 1.4**, DAFFOR tells to the operators that it has nothing to suggest.

Comparing result-dialogues in Figures 4.27 and 4.28, the following conclusions can be made:

- If a solution has been found, the same result-dialogue 1.2 is shown to the operators, independently of the mode; and
- If DAFFOR has failed to find a solution (as opposed to the case when DAFFOR has disapproved action a_r suggested by the operators), there is neither the search nor the accept option in result dialogue 1.4. In fact, the KB module has exhausted its knowledge (as described in Sections 4.2.2 and 4.2.3) without finding action a_r. It is therefore not worth asking for the search again. Since there is no action a_r, the operators have no DAFFOR suggestion to accept. All they can do is think of an action that DAFFOR can assess.

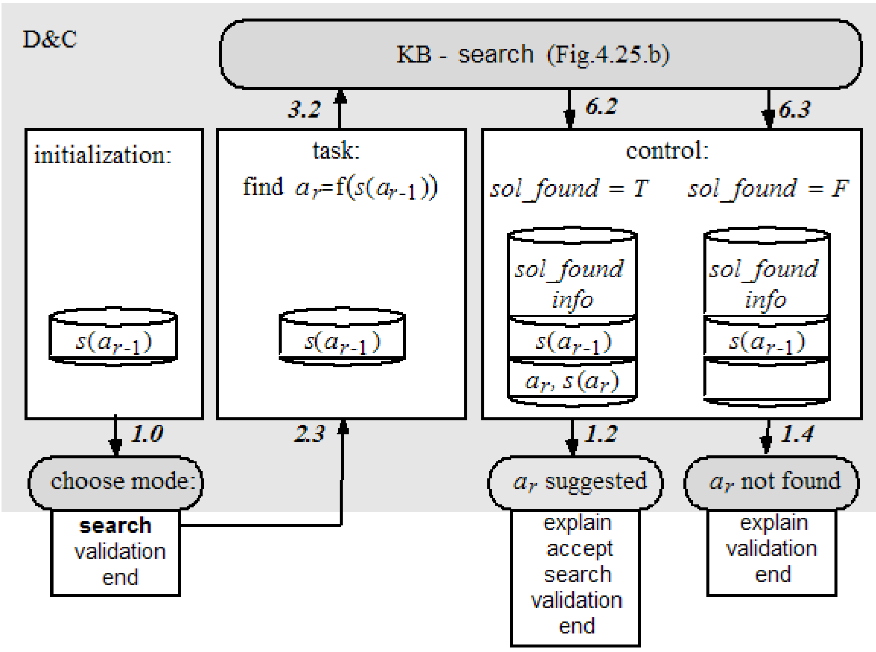


Figure 4.28 Basic functionality of the D&C module (and DAFFOR) in search mode

The descriptions of all the dialogue options, given in Section 4.3.1.1, apply in exactly the same manner for the above dialogues: each option keeps the same meaning for the operators, and the D&C module performs the same internal operations.

* * *

Let us now concentrate on step 6.3 in Figure 4.28, when there is nothing to suggest since the search for an action in the KB module has failed. Some factors which may cause this failure were briefly discussed at the end of Section 4.2.1.2, and in particular the content and the quality of the knowledge on which the search is based.

The following section shows how the D&C module controls the search process, performed in the KB module, in order to (1) enhance the reliability of the suggestion DAFFOR gives to the operators, and (2) decrease the probability of the appearance of the extremely undesirable conclusion "nothing to suggest".

4.3.2 DAFFOR's guidance capability

One of the requirements DAFFOR should satisfy is to always deal with a **single** restoration action (see Section 3.1.1). Therefore, the KB module also focuses on a **single** action (see Section 4.2.1): given the current power system state $s(a_{r-1})$, its objective is to assess (in validation mode) or to find (in search mode) action a_r, which is a solution of a **single** transition expression (4.1), i.e., the r^{th} element of the complete sequence of restoration actions (4.2). Since the search the KB module performs is in breadth (refer to Figure 4.20.b), it can be said that the depth of the search is 1.

However, the global goal is to completely restore the power system. Therefore, given the current power system state $s(a_{r-1})$, the KB module should be capable of approving (validation) or finding (search) action a_r, which is not only the solution of a single transition expressions (4.1), but which guarantees that solutions also exist for the remaining (N_{all_act}–(r+1)) expressions. In other words, the action a_r which the KB module approves (validation) or finds (search) should be the first element of a sequence of actions $Seq(r,N_{all\_act}-r)=[a_r|a_{r+1}\ldots a_{Nall\_act}]$ which leads the power system from the current state $s(a_{r-1})$ to a completely restored state $s(a_{Nall\_act})$, without violations of operating limits. To obtain such a sequence, the KB module should perform 1 assessment and (N_{all_act}–(r+1)) single-action searches in validation mode, or (N_{all_act}–r) single-action searches in search mode. In this case, the depth of the search would be (N_{all_act}–r).

A search of depth (N_{all_act}–r) has, however, some disadvantages. Towards the end of the restoration procedure (i.e., when only a few actions remain to restore the system completely), this kind of search could work well since its depth is small. On the contrary, at the very beginning of the restoration, when r=0, the search for a complete sequence of N_{all_act} should have to be performed. This kind of search is not acceptable for real-time operation because of the time it might take. In addition, if some unforeseen events occur, the obtained sequence should have to be adapted in some manner to the new state of the power system.

4.3.2.1 Increasing reliability of solution

The problem with the minimal-depth search (depth=1) is that the current solution might not be reliable in the light of the global restoration, while the problem with the maximal-depth search (depth=N_{all_act}–r) is the required computation time. The solution reliability increases, while the speed of finding the solution decreases proportionally with the depth of the search. Since both reliability and speed requirements are dependent on this depth, a compromise solution has been adopted as follows:

> *The depth of the search will be a parameter denoted as* N_{act}, *such that* $1<N_{act}<<N_{all\_act}$. *The objective of the search in this case is to find a sub-sequence of* N_{act} *control actions. Given the current power system state* $s(a_{r-1})$, *the sub-sequence should start with action* a_r:
>
> $$SubSeq(r, N_{act})=[a_r|a_{r+1}\ldots a_{r+(Nact-1)}] \qquad (4.21)$$
>
> *If such a sub-sequence is found, it guarantees to bring the power system from its current state* $s(a_{r-1})$ *to the state* $s(a_{r+(Nact-1)})$ *without operating limit violations.*

As reliability and speed requirements are contradictory, the parameter N_{act} (depth of the search, or length of the sub-sequence) should be set appropriately after a number of tests in order to meet both of them to a certain extent.

4.3.2.2 Role of D&C module in search control

Since the operation of the KB module is based on the assessment or the search for a **single** action (given the current power system state), the function of the D&C module is to control the search process of depth $N_{act}>1$ by launching the KB module in an appropriate mode, keeping track of its result, and handling the data.

In Figures 4.27 and 4.28, the basic functionality of the D&C module has been shown. The search control function represented on them performs only the testing of the result from the KB module (steps 6.1, 6.2 and 6.3), and is valid **only** for the search of depth $N_{act}=1$. Since the same remark applies to data resulting from steps 6.1, 6.2 and 6.3, the search control function and data handling must be reconsidered in light of the search of depth $N_{act}>1$. The other functions in the D&C module do not depend on the search depth.

Therefore, **the basic functionality of the D&C module, shown in Figures 4.27 and 4.28 remains the same, except for the search control and data handling functions**, which will be extended in the following discussion.

To illustrate how the D&C module controls the search of depth $N_{act}>1$, consider first the ideal case, i.e., when the KB module manages to find sub-sequence (4.21). Figure 4.29 shows the extended search control function and data handling of the D&C module for the search of depth $N_{act}=3$. Given the current state $s(a_{r-1})$, the objective is to find sub-sequence $SubSeq(r,3)=[a_r|a_{r+1}\ a_{r+2}]$. Steps 6.1 and 6.2 are identical to those from Figures 4.27 and 4.28, respectively, for the case where the solution has been found in the KB module (*sol_found*=true).

In both validation and search mode, the solution obtained from steps 6.1 and 6.2 at the first level (l=1) is action a_r. The D&C module saves the current solution and its resulting power system state $s(a_r)$, and schedules the search for the second action in $SubSeq(r,3)=[a_r|a_{r+1}\ a_{r+2}]$ with the recently obtained state $s(a_r)$ (step 3.2 at level l=2).

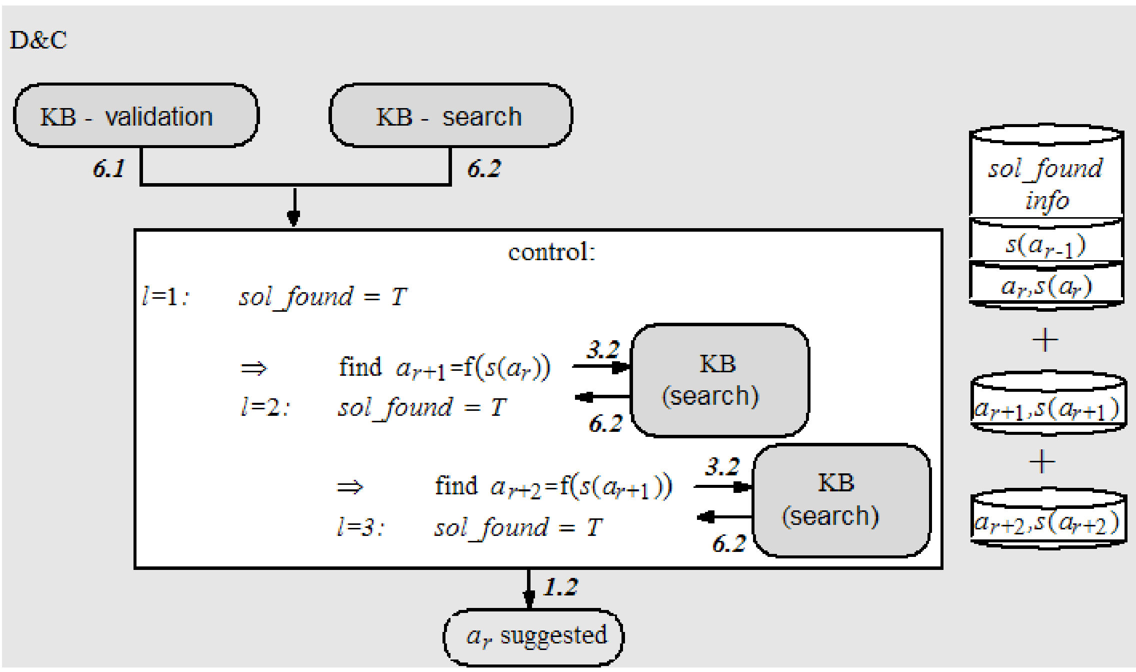


Figure 4.29 Extended search control functionality of the D&C module for the search of depth $N_{act}=3$ and content of data through different search levels l

When the KB module returns action a_{r+1} (step 6.2 at level l=2), the D&C module saves it along with the resulting state $s(a_{r+1})$, and once again schedules the same operations for the third action in $SubSeq(r,3)=[a_r|a_{r+1}\ a_{r+2}]$ (step 3.2 at level l=3).

Finally, when the KB module returns action a_{r+2} (step 6.2 at level l=3), the D&C module concludes that sub-sequence $SubSeq(r,3)=[a_r|a_{r+1}\ a_{r+2}]$ of N_{act}=3 has been found. Since the objective of the search has been reached, the head of the sub-sequence (action a_r) can be suggested to the operators through result-dialogue 1.2, as in Figures 4.27 or 4.28.

There is an important point concerning the overall conclusion of the D&C module when sub-sequence (4.21) of length N_{act} has been found, independently of the mode (validation or search):

> *If the KB module manages to find sub-sequence (4.21), the conclusion is that DAFFOR suggests to the operators to execute on the power system the first action from sub-sequence (4.21), and the D&C module shows* **result-dialogue 1.2** *(as in Figures 4.27 and 4.28 for the case* sol_found=*true).*

Let us discuss now how the D&C module controls the search as a function of the mode (validation or search) when the KB module fails to find the solution at any level l, $l \le l \le N_{act}$, while searching for sub-sequence (4.21) of length $N_{act} \ge 3$ for the current power system state $s(a_{r-1})$.

4.3.2.2.1 validation mode

If the KB module has not already found the solution after the first call, i.e., *sol_found*=false at level l=1, the D&C module concludes that the KB module has disapproved action a_r suggested by the operators, so that it cannot be the head of any sub-sequence (4.21). Therefore, it is not worthwhile continuing the search. The control function of the D&C module performs exactly the same operation as shown in Figure 4.27 for the case *sol_found*=false: it shows **result-dialogue 1.3**.

When the KB module has not found the solution at the second level (*sol_found*=false at level l=2), the D&C module concludes that the KB module first approved action a_r suggested by the operators (l=1), and then searched for action a_{r+1} (l=2). It has used all its knowledge without finding an action a_{r+1} that could follow action a_r, and it is therefore not worth continuing the search. Since sub-sequence (4.21) has not been found, action a_r suggested by the operators is disapproved, and **result-dialogue 1.3** is to be shown, as in Figure 4.27.

If the KB module fails to find the solution at the third level (*sol_found*=false at level l=3), the D&C module concludes that an action (a_{r+1}) exists that had been found by the KB module at the previous search level (l=2). Since the KB module had found action a_{r+1} through the heuristic best-first search and among several valid actions (see Section 4.2.2.2), it is probable that there is another valid action that could "replace" action a_{r+1}, i.e., the KB module still can try another search path. Therefore, the D&C module concludes that it should "move" the search process in the KB module up one level (from l=3 to l=2), i.e., it should **backtrack** through action a_{r+1}. If N_{act}>3, a similar discussion applies to any level l>3.

The backtracking will be explained in Section 4.3.2.3. At this point, it should be noted that, in validation mode, the D&C module never backtracks on the first action since it has been suggested by the operators, and therefore should not be replaced by another one.

There is an important point concerning the overall conclusion of the D&C module in validation mode:

> *If the KB module fails to return the solution at the first or second level of the search (*l*<3), the conclusion is that DAFFOR disapproves the operators' action, and the D&C module shows* **result-dialogue 1.3** *(as in Figure 4.27 for the case* sol_found=*false).*

If the solution has not been found at any level l≥3, the D&C module moves the search process in the KB module up one level (i.e., it backtracks through the last solution found) instead of showing result-dialogue 1.3.

4.3.2.2.2 search mode

If the KB module has not already found the solution after the first call, i.e., *sol_found*=false at level l=1, the D&C module concludes that the KB module has used all its knowledge without finding an action a_r that could be the head of some sub-sequence (4.21). Therefore, it is not worth continuing the search. The control function of the D&C module performs the same operation as shown in Figure 4.28 for the case *sol_found*=false: it shows **result-dialogue 1.4**.

If the KB module fails to find the solution at the second level (*sol_found*=false at level l=2), the D&C module concludes that an action (a_r) exists that had been found by the KB module at the previous search level (l=1). Therefore, the D&C module can **backtrack** through action a_r. A similar discussion applies to any level l≥2.

There is an important point concerning the overall conclusion of the D&C module in search mode:

> *If the KB module fails to return the solution at the first level of the search (*l*<2), the conclusion is that DAFFOR has nothing to suggest, and the D&C module shows* **result-dialogue 1.4** *(as in Figure 4.28 for the case* sol_found=*false).*

If the solution has not been found at any level l≥2, the D&C module moves the search process in the KB module up one level (i.e., it backtracks through the last solution found) instead of showing result-dialogue 1.4.

4.3.2.3 Enhancing the efficiency of search by backtracking

When the KB module fails to return the solution at level l (i.e., the l^{th} action from sub-sequence (4.21)), the D&C module moves the search process in the KB module up one level when l≥3 (in validation mode) or l≥2 (in search mode). The objective is to make the KB module try another search path, i.e., find another action that could be a solution for the given power system state. This process is called backtracking.

To illustrate how the search control function of the D&C module performs the backtracking in general, let us reconsider that the current power system state is $s(a_{r-1})$ and that the objective of the search is to find sub-sequence (4.21) of length $N_{act}\geq 3$. Figure 4.30 shows a part of extended search control function with backtracking occurring at level l=3. Since $l\geq 3$, the following description applies to both the validation and search mode.

Suppose that the KB module failed to return the solution a_{r+2} at the third level (*sol_found*=false in step 6.3 at level l=3). As the last solution is action a_{r+1} found by the KB module (and not suggested by the operators), the D&C module concludes that the search can be moved from level l=3 to level l=2, i.e., it can backtrack through action a_{r+1}. Therefore, the D&C module performs the following operations:

1. It disables action a_{r+1} **for the state as a function of which it has been found**, $s(a_r)$, and keeps it disabled for the KB module as long as this state is the current one. This will make the KB module try another search path, since it will consider action a_{r+1} as invalid. Note that the KB module can later find action a_{r+1} as a solution at some other search level with another current state. In other words, action a_{r+1} could figure in the final sub-sequence at any position but the second (since it has been disabled for search at level l=2).
2. Then the D&C module resets the solution that had been obtained at level l=2 (action a_{r+1} and its resulting state $s(a_{r+1})$).
3. Finally, it schedules the search for a **new action**, a_{r+1}, as a function of the current state for that level, $s(a_r)$ (step 3.2 at level l=2).

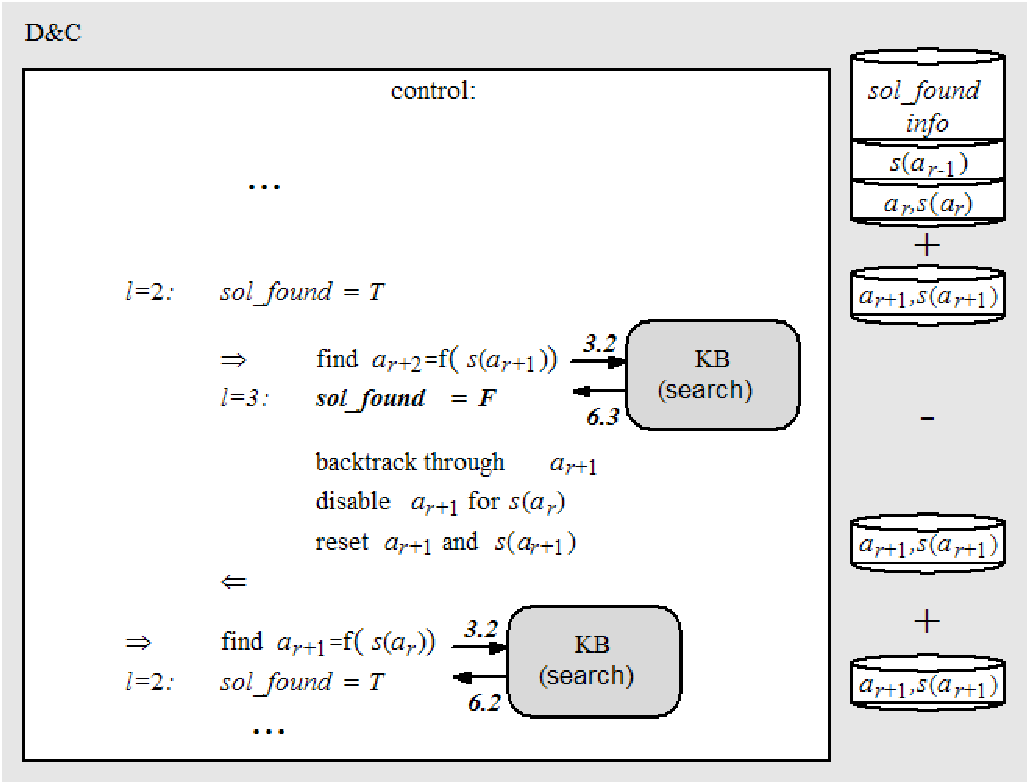


Figure 4.30 Part of extended search control function of the D&C module and content of data before and after backtracking which occurs at level l=3

If the mode is validation and the KB module fails to return the solution at level l=2, according to the discussion in Section 4.3.2.2.1, the D&C module cannot backtrack anymore. Therefore, it follows that action a_r suggested by the operators, has been disapproved. On the contrary, if the mode is search, according to the discussion in Section 4.3.2.2.2, the D&C module can still backtrack through action a_r. Only if the search fails at level l=1 will the conclusion be that DAFFOR has nothing to suggest.

4.3.2.3.1 Discussion on backtracking

The backtracking has been implemented in the D&C module in order to enhance the efficiency of the search process performed in the KB module. This search process is heuristic and therefore not optimal. Among several possible search paths, the KB module chooses the first one, and the backtracking, when it occurs, makes it explore other search paths.

The backtracking mechanism has been extremely useful in DAFFOR's development stage. In fact, the frequent occurrence of backtracking used to signal incorrect strategy representation in the KB module, and thus served as debugging information. In the same sense, it can be also useful if DAFFOR is used for restoration planning, since the occurrence of backtracking may be significant for re-examination of the strategy itself.

The main disadvantage of backtracking is the time it requires. Since it is a recursive process, it may be time consuming in case the depth of the search, N_{act}, is high. By choosing N_{act}=3, minimal backtracking is done in both validation and search mode. If N_{act}>3 and if backtracking occurs frequently, the time it takes is not acceptable in real-time conditions.

In our opinion, the backtracking mechanism should be used intensively in the testing stage of DAFFOR for the above mentioned advantages. This would permit the knowledge in the KB module to be refined in such a manner that the backtracking almost does not occur. Consequently, it could be completely disabled in real-time operation of DAFFOR.

4.3.2.4 Data handling

The extension of the data handling function of the D&C module when it controls the search of depth N_{act} remains. Section 4.3.1.1 discussed in detail the meaning of all the dialogue options which appear in result-dialogues 1.2, 1.3 and 1.4 (shown in Figures 4.27 and 4.28) along with operations the D&C module performs on data when any option is chosen. The only extension to what has been said in Section 4.3.1.1 concerns the case when the operators decide to execute the suggested action on the power system, i.e., when they choose **option accept from result dialogue 1.2**.

The D&C module shows this result-dialogue when sub-sequence (4.21) of length N_{act} has been found. This case has been represented in Figure 4.29 when sub-sequence $SubSeq(r,3)=[a_r|a_{r+1}\ a_{r+2}]$ of length $N_{act}=3$ has been found for the current power system state a_{r-1}. Data contained in the D&C module in this case is not only data obtained after the first level of the search (as in Figures 4.27 and 4.28), but there are two more actions (a_{r+1} and a_{r+2}) and their resulting states ($s(a_{r+1})$ and $s(a_{r+2})$, respectively). Since the actions and the states have been found in the search process, they may be kept and serve as a departure point in the next search if the operators decide to ask for DAFFOR's suggestion from the next initial dialogue 1.0 (search mode). In other words, the current power system state will be $s(a_r)$, and the goal of the search will be to find sub-sequence $SubSeq(r+1,3)=[a_{r+1}|a_{r+2}\ a_{r+3}]$. Since the first two actions had already been found, the KB module would only have to search for the last action of the sub-sequence, a_{r+3}, as a function of the power system state $s(a_{r+2})$. If it fails to find action a_{r+3}, the D&C module backtracks through action a_{r+2} in exactly the same manner as explained in Section 4.3.2.3.

On the contrary, if the operators decided to suggest an action themselves, only the state $s(a_r)$ is necessary. As the D&C module cannot predict which mode the operators will choose from the next initial dialogue, it also saves actions a_{r+1} and a_{r+2} for the case in which the option search would be chosen.

In the general case, the only extension for the data handling function of the D&C module is the fact that the obtained sub-sequence is saved, without the first action (i.e., $SubSeq(r+1, N_{act}-1)=[a_{r+1} \ldots a_{r+N_{act}}]$) if the operators choose **option accept from result dialogue 1.2**.

4.3.3 Implementation

The D&C module has been implemented in both the C programming language and NEXPERT *OBJECT*™ (NXP). Since the KB module is mostly implemented in the latter, the easiest and quickest way to obtain a prototype was to do the same with the D&C module. In addition, NXP provides a graphical user interface which is convenient for the interactivity function of the D&C module (i.e., DAFFOR). At present, only data handling functions are implemented in C, and are called from the rules when necessary.

As soon as the Man Machine Interface (MMI) becomes available in an industrial version (see Figure 3.1), the D&C module should be completely implemented in C, which will certainly increase both the execution speed and the portability.

4.3.4 Conclusion

The D&C module is the front-end of the Reasoning kernel, and globally of DAFFOR, towards the operators. It coordinates different tasks (internal to DAFFOR) with the operators' requests in such a manner that DAFFOR can perform according to the guidance and interactivity requirements stated in Section 3.1.1.

The search control function of the D&C module aims to provide the operators with a reliable solution for the current power system state. By controlling the search not only for a single restoration action, but also for sub-sequence (4.21) of $N_{act}>1$ actions, the solution offered to the operators (i.e., the first action from the sub-sequence) is reliable to a degree of ($N_{act}-1$). The backtracking mechanism renders the search process more efficient and decreases the probability that DAFFOR has nothing to suggest. In addition, it can be used for improving the quality of the knowledge contained in the KB module.

The D&C module performs all the control inside DAFFOR, and provides operators with the result concerning a single action at a time, as stated in guidance and interactivity requirements. There are at all 4 dialogues: 1 initial and 3 result-dialogues, as shown in Figure 4.31. These dialogues contain the necessary minimum of options for both operators' response effort and DAFFOR's correct operation.

First of all, the D&C module creates a data set representing the current power system state $s(a_{r-1})$. Then it shows the initial dialogue 1.0, asking the operators to choose the mode of operation (validation or search) and performs different tasks according to the operators' choice in order to issue a result. The result may be threefold, and is presented to the operators through result dialogues, along with the possible dialogue options:

1. If sub-sequence (4.21) has been found in any mode (i.e., **solution is found), DAFFOR suggests to the operators to execute action ar on the power system** through result dialogue 1.2;
2. If sub-sequence (4.21) has not been found in validation mode (i.e., validation **failed), DAFFOR suggests to the operators NOT to execute action ar on the power system** through result dialogue 1.3; and,
3. If sub-sequence (4.21) has not been found in search mode (i.e., search **failed), DAFFOR informs the operators that it has no action ar to suggest for the current power system state** through result dialogue 1.4.

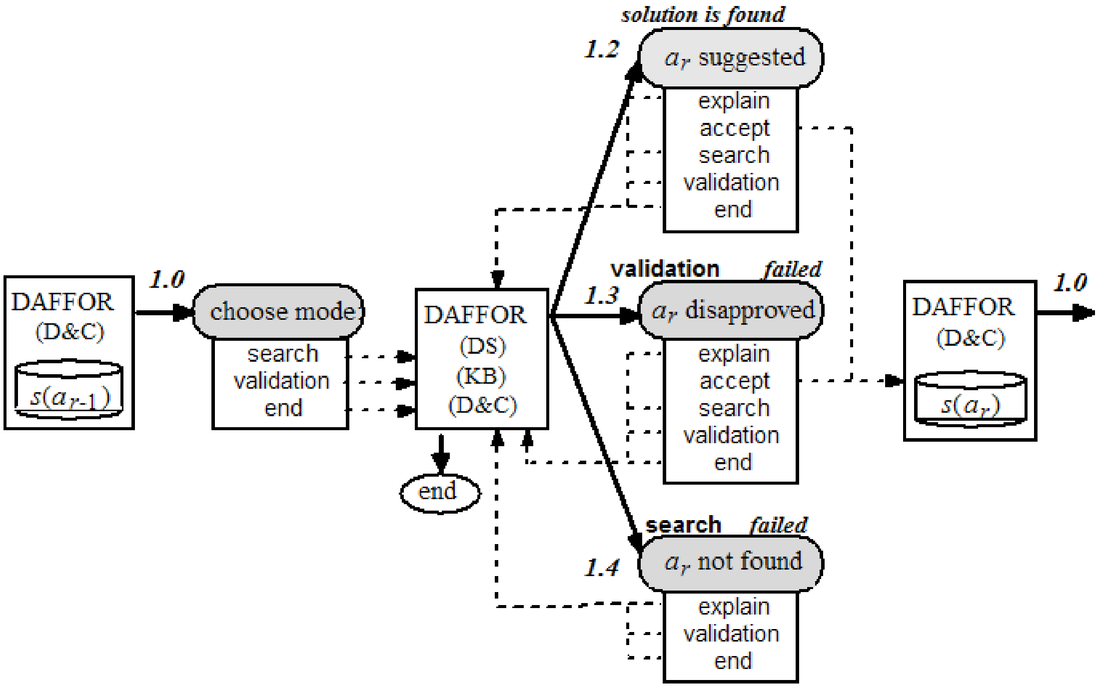


Figure 4.31 Dialogues shown to the operators. D&C, KB and DS stand for the Dialogue&control module, the Knowledge base module and the Dynamic simulator.

The operators may ask why and how DAFFOR obtained the result (option explain), or ask DAFFOR to assess or suggest another action a_r for the same current state $s(a_{r-1})$ (options validation and search, respectively), or decide to execute action a_r on the power system by choosing option accept.

Only after option accept will the D&C module change the current power system state from $s(a_{r-1})$ to $s(a_r)$ (resulting from action a_r) **and show again initial dialogue 1.0**. In other words, DAFFOR will be ready to assess (validation) or to search for (search) action a_{r+1} as a function of the **new** current state, $s(a_r)$. This fact permits us to introduce the term "reasoning cycle r".

4.3.4.1 Reasoning cycle

The reasoning cycle r characterises the time interval of variable length during which the Reasoning kernel (i.e., stand-alone DAFFOR) assesses or searches for action a_r as a function of the current power system state $s(a_{r-1})$, presents and explains its conclusion to the operators, or restarts the assessment or the search for another action a_r if asked by the operators (all these operations are coordinated by the D&C module). Consequently, the duration of the reasoning cycle depends on at least two factors: (1) the time necessary for DAFFOR to issue a conclusion on action a_r and (2) the number of times the operators reject the conclusion and ask for another assessment or search (there is a third factor, related to the operation with real-time data, which may interrupt the reasoning cycle r, as will be explained in Section 5.2.1.1).

Referring to Figure 4.31, the reasoning cycle r can be defined most concisely as follows:

> *The reasoning cycle* r *is the time interval between two consecutive appearances of initial dialogue 1.0, the first being when the D&C module sets* s(a_{r-1}) *as the current power system state for the whole Reasoning kernel, and the second when the D&C module sets* s(a_r) *as the current power system, which results from* accepted *action* a_r.

When talking about action a_r accepted from result-dialogue 1.2, it is understood that it represents the head of sub-sequence (4.21) obtained through the search of depth N_{act}>1. The resulting state $s(a_r)$ is obtained by simulating action a_r in the DS. In order to emphasise the fact that power system state $s(a_r)$ is **simulated** (and not obtained from the **real** power system), it can be denoted by $Sim(a_r)$:

$$Sim(a_r) = s(a_r) \tag{4.22}$$

Now Figure 4.31 can be simplified and the operation of the Reasoning kernel can be represented on the real-time axis t as in Figure 4.32. It can be said that reasoning cycle r:

- **starts** when initial dialogue 1.0 is shown to the operators, who are expected to indicate whether the Reasoning kernel should assess or search for action a_r as a function of the current power system state $s(a_{r-1})$,
- **lasts** as long as the operators do not accept to execute action a_r on the power system by choosing option accept from result-dialogues 1.2 or 1.3 (whether it has been, respectively, suggested or disapproved by the Reasoning kernel), and
- **ends** when the D&C module changes the current power system state for the Reasoning kernel from $s(a_{r-1})$ to $s(a_r)$. State $Sim(a_r)= s(a_r)$ is said to be the result of the terminated reasoning cycle r. After that, the next reasoning cycle (r+1) may start by showing initial dialogue 1.0 to the operators.

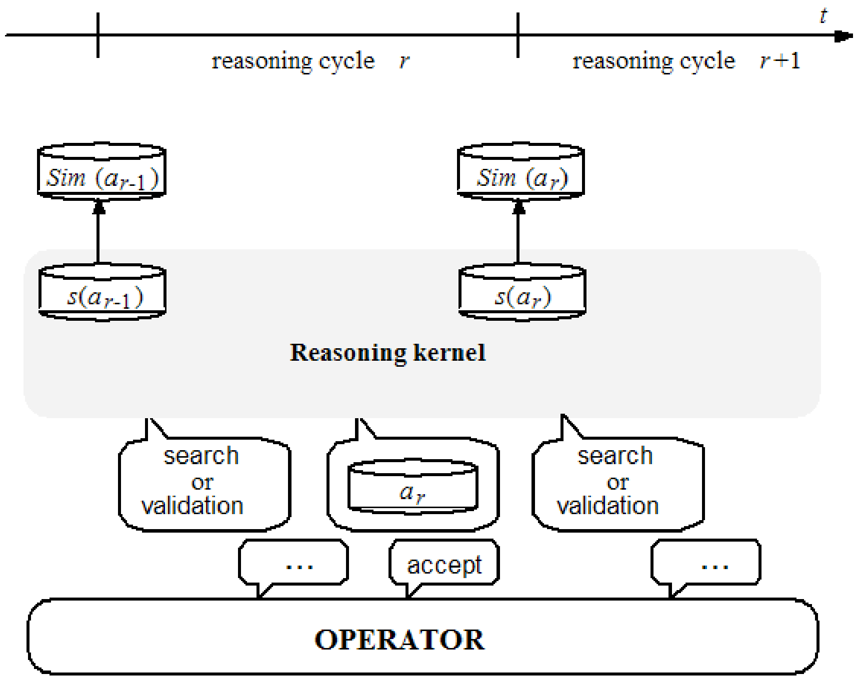


Figure 4.32 The terminated reasoning cycle r results in the simulated state $Sim(a_r)$

The notion of the reasoning cycle *r* permits us to "abbreviate" all the operations and interactions, inside the Reasoning kernel and with the operators that have been described in detail through Sections 4.1, 4.2 and 4.3. Indeed, the whole discussion of the Reasoning kernel and its modules has been concentrated on action *a* with index *r*, and not haphazardly.

4.3.4.2 A Step towards real-time DAFFOR

The notion of the reasoning cycle *r* is essential for real-time DAFFOR, since it represents the only information on what the operators do and, as a function of that, how the power system should respond.

For instance, if there is a terminated reasoning cycle *r* (i.e., operators have chosen option accept), the result of this reasoning cycle is the power system state $Sim(a_r)$, shown by step 7' in Figure 4.33, which is expected to "occur" after the operators execute action a_r on the power system, shown by step 7 in the same figure (since these two operations are almost simultaneous, they are denoted with the same number, 7). Referring to Figure 3.1, which represents real-time DAFFOR, it can be concluded that data from the Reasoning kernel towards the Real Time Update kernel is exactly this simulated state $Sim(a_r)$.

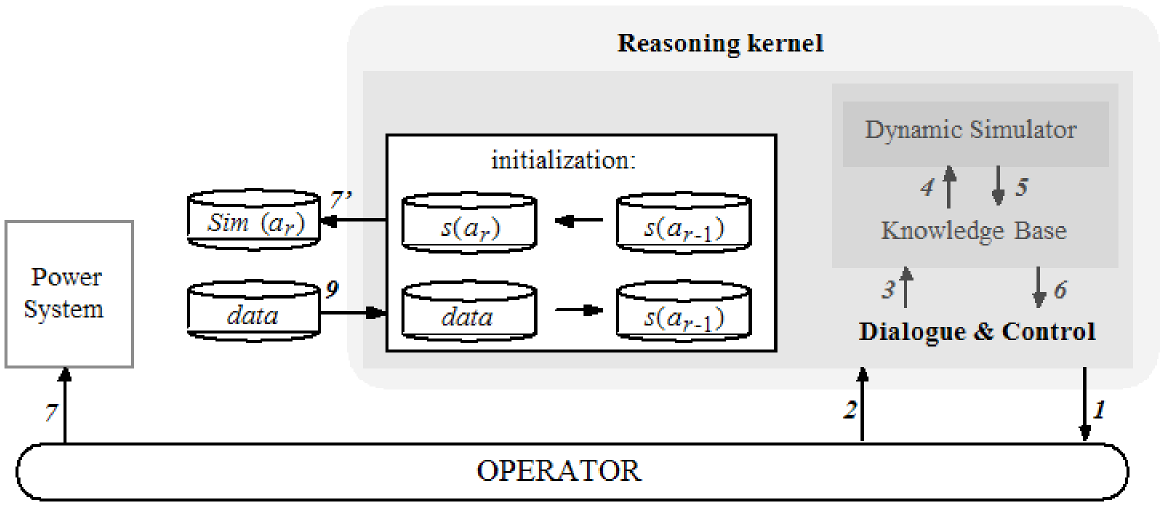


Figure 4.33 The D&C module handles data in real-time DAFFOR

Since the D&C module handles data for the whole Reasoning kernel, its function is to change the current power system state from $s(a_{r-1})$ to $s(a_r)$ as soon as action a_r is accepted by the operators (see Section 4.3.1.1). Also, its function is to set the current state in the Reasoning kernel ($s(a_{r-1})$) according to real-time *data*, generated by the Real Time Update kernel (step 9).

In conclusion, the data link between the two kernels is provided by the D&C module's data handling function, which will be described in detail in Section 5.2.4.2.

4.4 Off-line reasoning

In Sections 4.1 through 4.3, the role of each Reasoning kernel's "building block" and the interactions among them have been described for assessment or search for a single action from sequence (4.2). Then in Section 4.3.4.1 the notion of reasoning cycle was introduced as an "alias" for all the above.

This section presents the overall functionality of the Reasoning kernel as the stand-alone prototype of DAFFOR, i.e., how it guides the whole restoration of a transmission power system by assessing or searching for consecutive actions from sequence (4.2) with respect to transition expression (4.1), according to restoration strategies described in Section 4.2.3. In other words, it will be considered that the blackout happened in the power system and that DAFFOR should guide the restoration through reasoning cycles $r=1,2,\ldots,N_{all\_act}$, until the total load in the power system is restored.

First the content of the so called *blackout scenario* will be presented, which defines the initial state for all the simulations that have been carried out. Then the test power systems used for simulations will be introduced and described. Before presenting and discussing the complete restoration procedures (issued by DAFFOR), some of DAFFOR's facilities stated in the previous sections will be demonstrated with examples.

4.4.1 Blackout scenario (initial state)

The blackout scenario is a text file which enables the setting of different power system variables that define the initial state from which the restoration starts:

- the load level;
- the availability of the external support (for power systems where the equivalent for the external network exists);
- for production units, there are 4 variables that should be defined: (1) belonging to the regional skeleton network (RSN), (2) telemetry status, (3) availability status, and (4) the minimum time interval before which the unit cannot be coupled (note that items (2) and (4) apply only to thermal units, although they may be defined in the file for all units); and
- the availability of each element of the transmission equipment.

Figure 4.34 shows a sample blackout scenario file for the New England test network (given in Appendix A). This kind of scenario permits us to test the stand-alone DAFFOR's capability of restoring the power system from different initial states.

```
  Scenario_4
#
# Load level
#
  100.
#
# External support
#
  0
#
# UnitName    InSkel TS     Avail  Tmin[min]
#
  gen1        0      *      0        0.0000
  gen2        1      ts1    1      240.0000
  gen3        0      *      1      230.0000
  gen4        0      *      1      220.0000
  gen5        1      ts1    1      240.0000
  gen6        0      *      1      210.0000
  gen7        0      *      1      200.0000
  gen8        1      ts1    1      240.0000
  gen9        0      *      1      190.0000
  gen10       0      *      1      180.0000
#
#
# BranchName   Avail
#
  n11n12      1
 ..............
  n12n10      1
```

Figure 4.34 A sample scenario file for the New England test network. The load to restore is equal to the nominal (100%). There is no support from the external network. Three units are defined to be in the RSN as TS1 units (i.e., black-start units). All the real units are available. The minimum time before coupling is taken into account only for the shutdown thermal units (those that do not emit any telemetry, i.e., field TS is "*"), and it varies from 180 to 230 minutes. Finally, all the branches are available.

Regardless of the blackout scenario and according to EDF's restoration strategy (described in Section 2.4.4.2), the following settings are performed:

- the voltage set-points of the black-start units are set to their minimum value of 0.95pu;
- the taps of TCUL transformers are locked at the position which provides the minimum voltage on the secondary side (lower voltage level); and
- all available inductive shunt compensators are connected.

From the restoration viewpoint, two types of power systems will be distinguished (according to EDF's definition): a **regional** power system consisting of several voltage levels (i.e., as seen from a Regional Control Centre), and a **national** covering a single high voltage level (i.e., as seen from a National Control Centre). The consumer loads in the latter are represented as injections connected directly to the high voltage nodes, and possibly through transformers or autotransformers from the lower voltage levels.

There are two more settings that are performed for the case of a *national* power system: all the transmission equipment and all the loads are initially disconnected.

4.4.2 Test networks

Three test power systems have been used for the validation of the stand-alone DAFFOR: (1) a part of the French national network, (2) a French regional network, and (3) the New England test network. For the purposes of the confidentiality of data, the names of power system elements in the former two networks have been modified. The location of electrical nodes does not correspond to geographical location of substations.

4.4.2.1 A part of the French national network

The nodal topology shown in Figure 4.35 has been computed by topology conversion functions, discussed in Section 4.1.2.1, from the original data provided by EDF (which contains a substation topology model).

The characteristics of this network are as follows:

- 31 nodes;
- 10 units: 2 classical thermal (G1 and G2), and 8 nuclear, with a total capacity of 9180MW (auxiliaries' consumption deduced);
- 31 consumer loads for a total of 4406MW;
- 29 400kV lines, 6 400/225kV/kV autotransformers and 10 step-up transformers;
- 4 capacitors for a total of 292MVar.

Different dynamic simulation parameters for this network can be found in Appendix C.

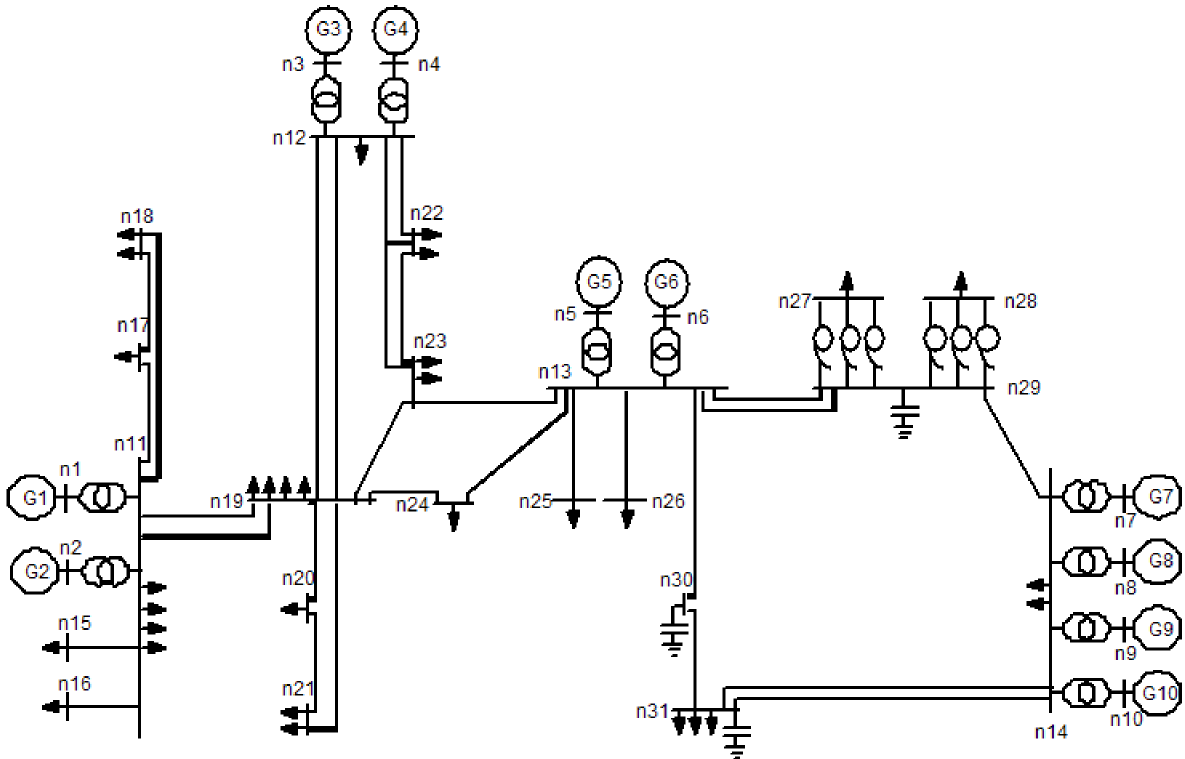


Figure 4.35 A part of the French national network

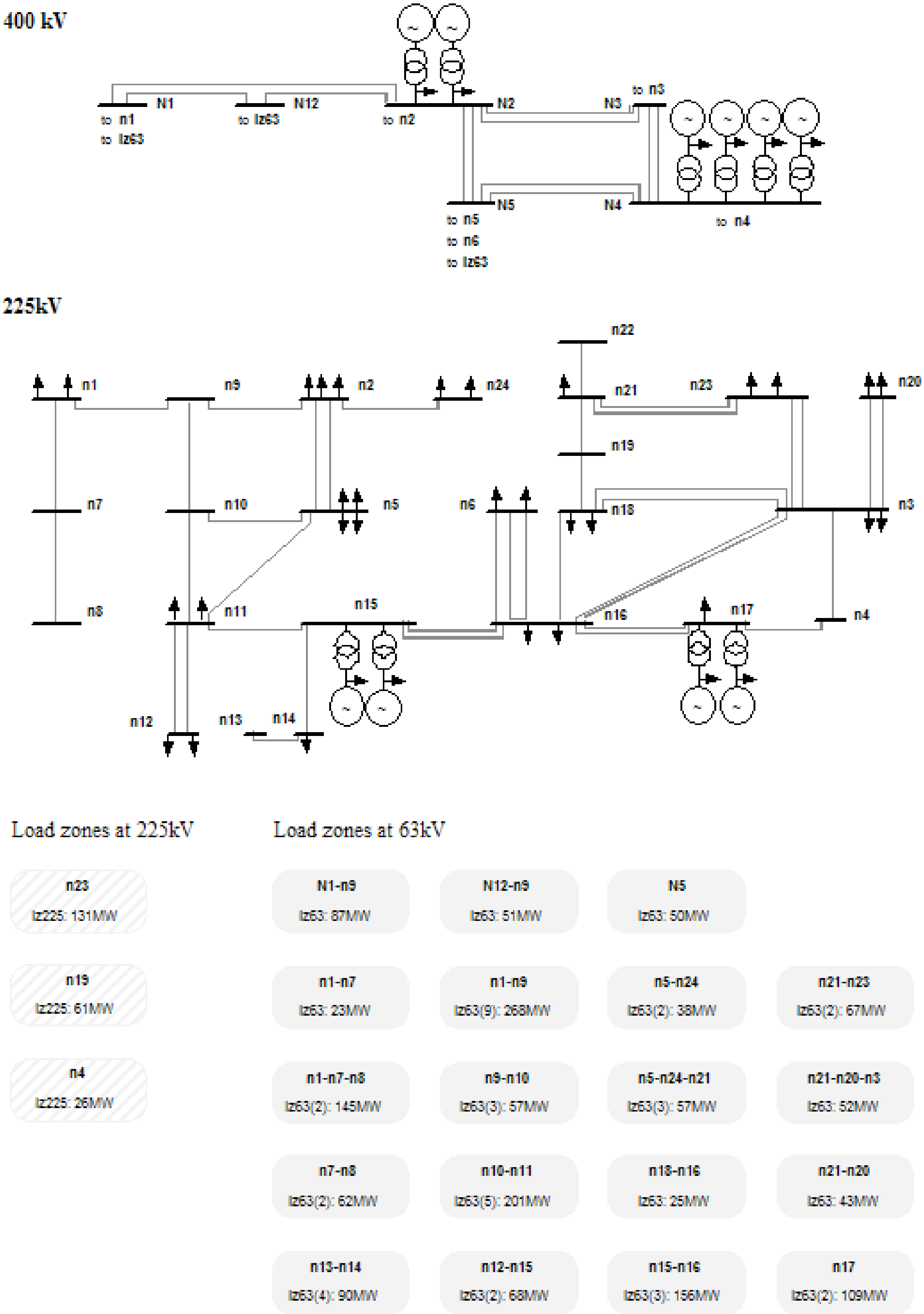


Figure 4.36 A simplified French regional network

4.4.2.2 Simplified French regional network

The nodal topology shown in Figure 4.36 has been computed by topology conversion functions, discussed in Section 4.1.2.1, from the original data provided by EDF (which contains substation topology model). The initial topology is that obtained after the network has been divided by the operation of AMUs (see Section 2.4.3). This network has 3 voltage levels: 400kV, 225kV and 63kV.

400kV network: all the lines are open on both ends (they are drawn with dashed lines). There are no consumer loads. There are 6 nuclear production units; arrows represent the location of their auxiliaries. "to *n**" means that there is one or more autotransformers toward the node *nx*, and "to *lz63*" means that there are one or more transformers with tap changer under load (TCUL) toward load zone *lz63*.

225kV network: has been simplified (the closed lines between nodes and the respective nodes have been replaced by a single node). All the lines shown in the figure are open either on one or both ends (they are drawn in dashed lines). There are 4 classical thermal units; arrows between the unit and the step-up transformer represent the location of the units' auxiliaries. The consumer loads shown by arrows in nodes are all disconnected. There are 3 load zones at the 225kV level: they represent those consumer loads that are connected, but unserved (e.g., as soon as node *n23* is energised, the load zone of 131MW will be picked up). For the purposes of readability, the TCUL transformers toward load zone *lz63* are not represented.

63kV network: has been simplified and represented by load zone *lz63*. Regarding the first one, "*N1-n9*" means that load zone *lz63* of 87MW can be supplied from either node *N1* or *n9*; regarding the last one, "*lz63(2)*" means that there are two load zones for a total of 109MW which can be supplied only from node *n17*.

The characteristics of the whole network are as follows:

- 198 nodes;
- 33 units: 17 hydraulic, 10 classical thermal, and 6 nuclear, with a total capacity of 8208MW (auxiliaries' consumption deduced from the rated power of the thermal units);
- 177 consumer loads for a total of 4254MW;
- 307 branches (lines, autotransformers, TCUL transformers and step-up transformers);
- 19 capacitors for a total of 220MVar and one reactor of 60MVar.

Different dynamic simulation parameters for this network can be found in Appendix D.

4.4.2.3 New England test network

The scheme of this network, its characteristics and different dynamic simulation parameters can be found in Appendix A.

4.4.3 Examples and results

This section will first give some examples of DAFFOR's functionality in search and validation mode, how the restoration strategy is applied (i.e., the flow and the control of the search process) and the backtracking facility. All these will be demonstrated using the part of the French national network (Figure 4.35), with the blackout scenario from Figure 4.37.

```
 Scenario_1
#
# Load level
#
  100.
#
# External support
#
  0
#
# UnitName    InSkel TS    Avail  Tmin[min]
#
gen7          1      ts1   1      240.0000
gen8          1      *     1      240.0000
gen9          1      *     1      270.0000
gen10         1      *     0      240.0000
gen3          1      ts2   1      330.0000
gen4          1      *     1      300.0000
gen5          1      *     1      240.0000
gen6          1      *     1      240.0000
gen1          0      *     1      240.0000
gen2          0      *     1      240.0000
#
# All branches available
#
```

Figure 4.37 Blackout scenario for a part of the French national network (from Figure 4.35)

4.4.3.1 General description

The blackout scenario from Figure 4.37 defines all nuclear production units to belong to the regional skeleton network (RSN), i.e., units G3 through G10. There are two units that are isolated (G3 and G7); all the others are shutdown, and can be coupled after the time defined by Tmin. Unit G7 is a black-start unit, and unit G3 needs voltage on its auxiliaries in order to be coupled. Unit G10 is supposed to be unavailable (e.g., under maintenance).

After the state had been initialised from the blackout scenario and as described in Section 4.4.1, the reasoning started with cycle r=1, in search mode, as shown in Figure 4.38. The depth of the search was set to N_{act}=3. Therefore, DAFFOR had to search for three consecutive actions which satisfy both the current power system state and the restoration strategy (energisation of the RSN, shown in Figure 4.22). The following three actions were found by DAFFOR:

- the first action (couple unit G7) was found according to Goal 3;
- the second and the third actions (connect branch n14n29_1, then n13n29_2) were found according to Goal 7.

The sub-sequence of 3 actions being found, DAFFOR showed the head of the sub-sequence through the dialogue (this is not shown in Figure 4.38) and suggested to execute the action on the power system (i.e., to couple unit G7). The action was accepted so that DAFFOR could start the next reasoning cycle, r=2.

```
# ################################# Cycle #1
# -------------- recherche
# --A1  enclencher_groupe_gen7            (EnclencherGroupeTensionMin bstart_oss en ossature)
#   --A2  enclencher_branche_n14n29_1     (RenvoiDeTension a groupe_ts2_oss en ossature)
#     --A3  enclencher_branche_n13n29_2   (RenvoiDeTension a groupe_ts2_oss en ossature)
#
# -------------- Vous avez  accepte  action A1  enclencher_groupe_gen7          (a1: gen on gen7)
#
# ################################# Cycle #2
# -------------- recherche
# --A1  enclencher_branche_n14n29_1       (R)
#   --A2  enclencher_branche_n13n29_2     (R)
#     --A3  enclencher_branche_n23n13_1   (RenvoiDeTension a groupe_ts2_oss en ossature)
#
# -------------- Vous avez  accepte  action A1  enclencher_branche_n14n29_1     (a2: bra on n14n29_1)
#
# ################################# Cycle #3
# -------------- recherche
# --A1  enclencher_branche_n13n29_2       (R)
#   --A2  ................
#     --A3  ..................
#
# -------------- Vous avez  accepte  action A1  enclencher_branche_n13n29_2     (a3: bra on n13n29_2)
#
```

Figure 4.38 Reasoning in search mode with a search of depth N_{act}=3

In the second reasoning cycle DAFFOR was again asked to reason in search mode. Therefore, it kept the second and third action from the sub-sequence in order to decrease the time of the search (they are both followed by (R)). These two actions became the partial solution of the current search, so that DAFFOR only needed to find the third action. When it was found (connect branch `n23n13_1`), according to Goal 7, the sub-sequence of three actions was completed so that DAFFOR showed the head of the sub-sequence, and suggested connecting branch `n14n29_1`. This suggestion was accepted and DAFFOR could continue its third reasoning cycle, *r*=3.

In the third reasoning cycle, the procedure was the same. DAFFOR was asked to continue in search mode so that it kept the partial solution from the previous cycle. When it found the third action, it suggested connecting branch `n13n29_2`, which was accepted.

This example shows how DAFFOR operates in search mode when its suggestion is accepted. It uses what was found in the previous search in order to decrease the time of the search (actions followed by (R)). It can be said that one reasoning cycle always corresponds to one control action which is supposed to be executed on the power system.

Figure 4.39 shows the fourth reasoning cycle for the same example. DAFFOR was asked to find an action (search mode). It found the sub-sequence with the head action "connect branch `n23n13_1`", which was rejected. DAFFOR was again asked to continue in search mode, and found another sub-sequence: its head action is different than before (connect branch `n31n14_2`), but the "old head action" found its place as the second action in the new sub-sequence. This shows that the rejected action is considered as invalid only for the power system state for which it had been found.

```
# ################################ Cycle #4
# ------------- recherche
# --A1  enclencher_branche_n23n13_1          (R)
#    --A2  ..........
#       --A3  ...........
#
# ------------- Vous avez refuse action A1  enclencher_branche_n23n13_1    ( a4: bra on n23n13_1 )
#
# ------------- DAFFOR recommence avec  recherche
# --A1  enclencher_branche_n31n14_2          (RenvoiDeTension a groupe_ts2_oss en ossature)
#    --A2  enclencher_branche_n23n13_1       (RenvoiDeTension a groupe_ts2_oss en ossature)
#       --A3  .............
#
# ------------- Vous avez  refuse action A1  enclencher_branche_n31n14_2    ( a4: bra on n31n14_2 )
#
# ------------- DAFFOR recommence avec  recherche
# --A1  prendre_niveaux_5_n14_2              (ReprendreClients en ossature)
#        ==>> a cause de l'incertitude sur la charge
# --A1  prendre_niveaux_5_n14_1              (ReprendreClients en ossature)
#        ==>> a cause de l'incertitude sur la charge
# --A1  prendre_niveaux_4_n14_2              (ReprendreClient s en ossature)
#        ==>> a cause de l'incertitude sur la charge
#       ........................
# --A1  prendre_niveaux_1_n14_2              (ReprendreClients en ossature)
#    --A2  enclencher_branche_n23n13_1       (RenvoiDeTension a groupe_ts2_oss en ossature)
#       --A3  ...........
#
# ------------- Vous avez  accepte  action A1  prendre_niveaux_1_n14_2      (a4: loa aset n14_2 )
#
```

Figure 4.39 Fourth reasoning cycle in search mode with a search of depth N_{act}=3

In a similar manner, the new suggested action (connect branch n31n14_2) was also rejected and DAFFOR was again asked to find another one (search mode). The search for the first action was performed according to Goal 10. Note that DAFFOR did not find the first action immediately: there were some violations of the operating limits in several trials (due to load uncertainty). However, after some search it suggested picking up the first unserved level of the consumer load n14_2 which was accepted.

Note also that the second action in the sub-sequence is the connection of the line n23n13_1 which is on the shortest path towards the TS2 unit G3 (i.e., the highest priority goal).

Figure 4.40 shows the continuation of the same example, when DAFFOR was asked to assess action "connect branch n13n24_2" in its fifth reasoning cycle. Note that there are no actions saved from the previous cycle (no sign (R) after any action), since the user suggested the action. It is considered by DAFFOR as the head of a sub-sequence of three actions, and it is necessary to search for the second and the third action. Again, DAFFOR did not find them on the first try, but it continued until the actions had been found which do not cause any violation of the operating limits. The head of the sub-sequence was suggested (i.e., DAFFOR approved the action suggested by the user), and the user accepted connecting branch n13n24_2. Therefore, DAFFOR proceeded to the next reasoning cycle, r=6.

```
# ################################ Cycle #5
# ------------- validation
# --A1  enclencher_branche_n13n24_2
#    --A2  enclencher_branche_n19n24_2       (RenvoiDeTension a groupe_ts2_oss en ossature)
#          ==>> Surtensions Vmax + [%]:
#    --A2  enclencher_branche_n23n13_1       (RenvoiDeTension a groupe_ts2_oss en ossature)
#       --A3  enclencher_branche_n19n24_2    (RenvoiDeTension a groupe_ts2_oss en ossature)
#             ==>> Surten sions Vmax + [%]:
#          ........................
#       --A3  prendre_niveaux_1_n23_2        (ReprendreClients en ossature)
#
# ------------- Vous avez  accepte  action A1  enclencher_branche_n13n24_2     (a5: bra on n13n24_2)
#
# ################################ Cycle #6
# ------------- recherche
# --A1  enclencher_branche_n23n13_1          (R)
#    --A2  prendre_niveaux_1_n23_2           (R)
#       --A3  ............
#
# ------------- Vous avez  refuse action A1  enclencher_branche_n23n13_1     (  a6: bra on n23n13_1)
#
# ------------- DAFFOR recommence  avec   validation
# --A1  enclencher_branche_n19n24_2
#       ==>> Surtensions Vmax + [%]:
#          ############ depth # 1 # - USER failed
#
# ------------- Vous avez  refuse action A1  enclencher_branche_n19n24_2     (  a6: bra on n19n24_2)
#
# ------------- DAFFOR recommence  avec   recherche
# --A1  enclencher_branche_n31n14_2          (RenvoiDeTension a groupe_ts2_oss en ossature)
#    --A2  ...........
#       --A3  ...........
```

Figure 4.40 Fifth and sixth reasoning cycles in validation and search mode with a search of depth N_{act}=3

The sixth cycle started in search mode, and DAFFOR suggested connecting branch n23n13_1. This action was rejected and DAFFOR was asked to assess the action "connect branch n19n24_2". Already at this level, DAFFOR found that this action would cause some over-voltages. Therefore, it disapproved this action suggested by the user. The latter agreed with DAFFOR's conclusion and decided to ask for the new action in search mode.

At present, DAFFOR has no rules which would enable the alleviation of the operating limits violations. So if the user had accepted the disapproved action "connect branch n19n24_2" (i.e., wanted to execute it despite the limit violations), DAFFOR would not have been able to continue the reasoning. The implementation of such rules will be the objective of further work.

Figure 4.41 shows reasoning cycle r=141, in which DAFFOR operated in search mode, and where the backtracking occurred. The first action (pick up one unserved level of load n28_2) was found according to Goal 23, and the second (increase power set-point of unit G4) according to Goal 15. In the search for the third action, DAFFOR found that the voltage should be increased (Goal 17), but there were no applicable actions for this goal, because of the violations of the operating limits.

```
# ################################ Cycle #141
# ------------- recherche
# --A1  prendre_niveaux_1_n28_2          (R)
#    --A2  augmenter_puissance_gen4      (R)
#      --A3 augmenter_tension_gen6       (TensionEstBasse: ChangerConsigneTension)
#           ==>> Surtensions Vmax + [%]:
#           ...............
#      --A3 prendre_niveaux_5_n21_2      (ReprendreClients)
#           ==>> a cause de l'incertitude sur la charge
#           ...............
#      --A3 enclencher_branche_n19n21_2  (EtendreReseau)
#           ==>> Surtensions Vmax + [%]:
#           ...............
#      --A3 enclencher_branche_n31n14_2  (EtendreReseau)
#            ==>> Surtensions Vmax + [%]:
#
#      ###### depth # 3 # --- remonter
#
#    --A2  augmenter_puissance_gen9      (FrequenceEstBasse)
#      --A3 augmenter_tension_gen6       (TensionEstBasse: ChangerConsigneTension)
#           ==>> Soutensions Vmin - [%]:
#           ...............
#      --A3 prendre_niveaux_5_n21_2      (ReprendreClients)
#           ==>> a cause de l'incertitude sur la charge
#           ...............
#      --A3 enclencher_branche_n19n21_2  (EtendreReseau)
#           ==>> Soutensions Vmin - [%]:
#      --A3 enclencher_branche_n19n24_2  (EtendreReseau)
#
# ------------- Vous avez accepte action A1   prendre_niveaux_1_n28_2     (a141: loa aset n28_2 )
```

Figure 4.41 Backtracking with a search of depth N_{act}=3

In a similar manner, DAFFOR failed to find the third action according to Goals 23 and 26. Since Goal 26 is the last one, DAFFOR concluded that it should backtrack through the second action (i.e., the last action that had been found). Therefore, it considered the second action as invalid, and started to search for a new one for the same (second) position in the sub-sequence. It found that the frequency should be increased (Goal 15), and chose the unit `G9` for increasing its power set-point. Then it continued the search for the third action, until it found that the branch `n19n24_2` could be connected (Goal 26). Since the sub-sequence of three actions was found, DAFFOR suggested the first action from the sub-sequence (to pick up one unserved level of load `n28_2`).

This example of backtracking is quite illustrative. If there had not been a backtracking mechanism, DAFFOR would have informed the user that it failed to find a solution. Backtracking made it examine another search path, i.e., searching for another action that is also valid for the current power system state.

* * *

The above discussion aimed at demonstrating DAFFOR's functionality in search and validation mode, the search for a sub-sequence of control actions, the backtracking mechanism and the manner in which DAFFOR focuses on a goal from the restoration strategy. In the following sections, the results of the restoration procedure, entirely suggested by DAFFOR, will be presented for three test networks (the mode is always search, and the suggested action is always accepted).

4.4.3.2 French national network

The blackout scenario from which the restoration has started is similar to the one from Figure 4.37. The only difference is the choice of the black-start (TS1) and the TS2 unit which belong to the RSN (units G8 and G5, respectively). The results of the dynamic simulation of the restoration procedure found by DAFFOR are shown in Figure 4.42:

- the frequency in Hz (the steady state limits are drawn with dotted lines);
- the percentage of the consumer load that is supplied; and
- the maximum and minimum voltage at a given moment; a dot represents the boundary for the node in question (the limits may be asymmetric).

In the beginning of the restoration, the RSN is created by energising a path from the black-start unit towards the TS2 unit. The first three transients in the frequency (between 20 and 40 minutes) are due to the supply of the auxiliaries of the shutdown thermal units in the RSN. When the frequency reaches the heuristic lower limit (see section 4.1.6.1.2), the set-point of a unit is increased. The RSN is created after about one hour. Then the network is extended and the consumer loads start to be picked up.

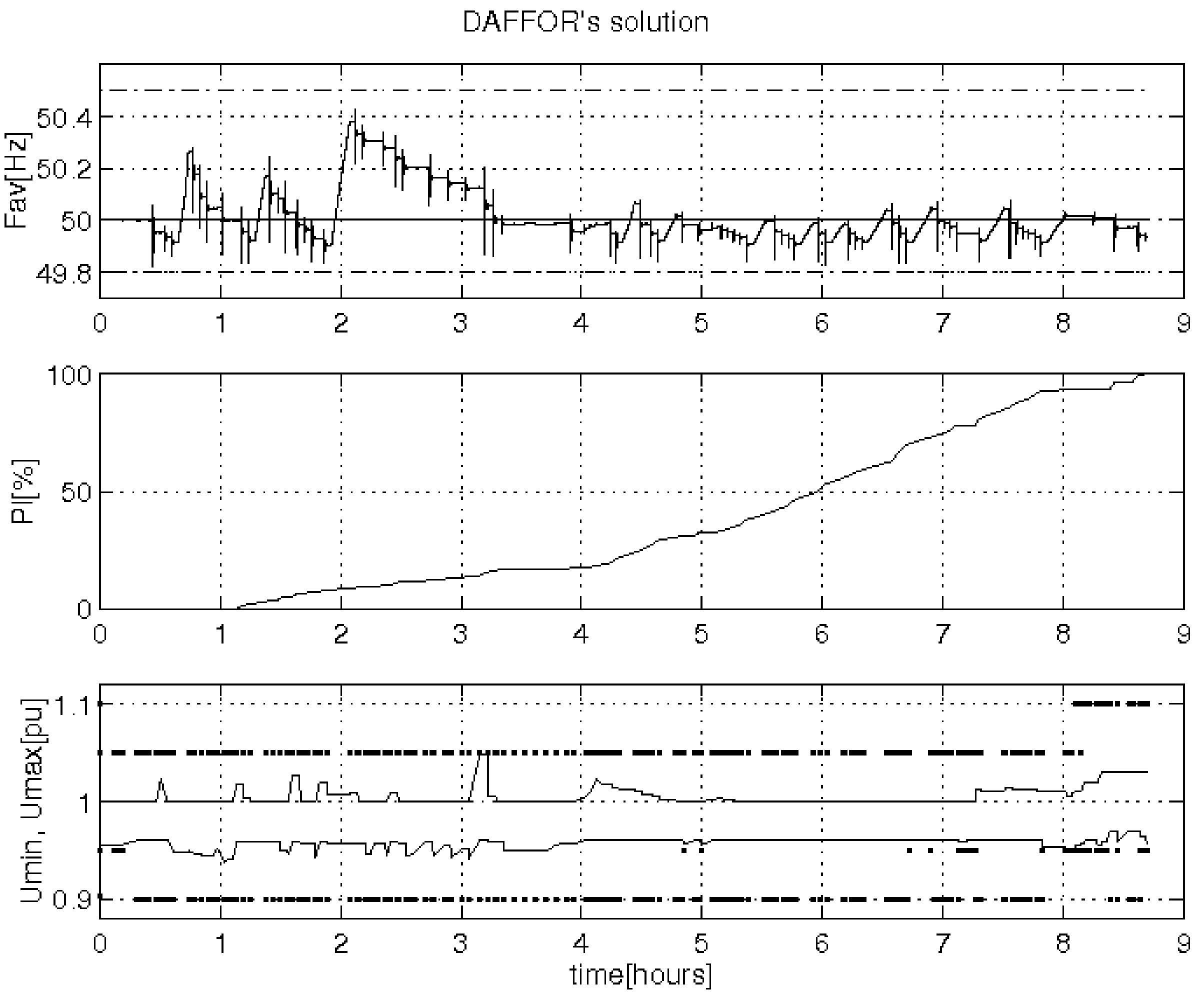


Figure 4.42 Frequency, load supply rate and minimum and maximum voltages during the restoration of the French national network (from Figure 4.35)

The rate of load pickup is rather small in the beginning, since there are only two coupled units. For the same reason, the response of the prime movers to cold load pickup is quite pronounced, even for small load increments. After about 2 hours of restoration, the power set-point of one of the coupled units is increased for the second time - this transient is zoomed in Figure 4.43. Let us recall that the unit's loading is performed here by default levels of 30%, 75% and 90% of the unit's rated power. As the unit has important rated power, and again there are only two coupled units, the increase in frequency is very important. However, it stays within the limits. As soon as the unit has been loaded, some load is picked up. This transient occurs around 127 minutes, as shown in Figure 4.43.

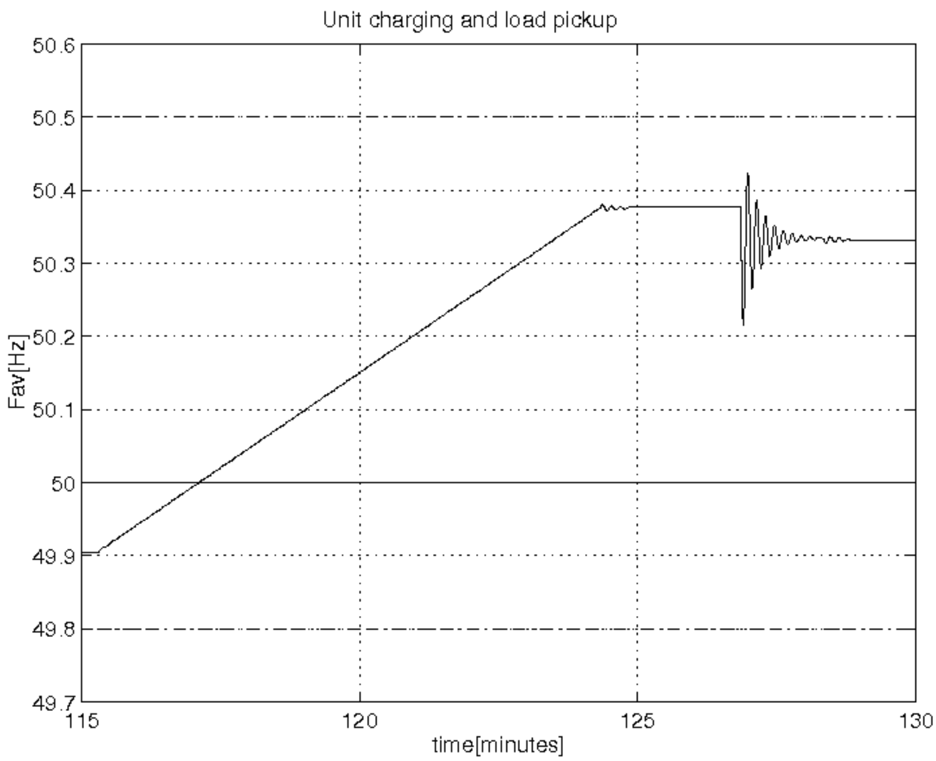


Figure 4.43 Zoom on the highest frequency transient from Figure 4.42

Focusing again on Figure 4.42, it can be seen that during the fourth hour little load has been picked up. On one hand, the voltage was rather low, and on the other, there were only big increments of load left to pick up in the energised part of the network. Therefore, the network was being extended during that period.

After 4 hours, the other thermal units were ready for coupling. It is evident that when the number of coupled units increased, the frequency excursions are much smaller (because of the higher inertia of masses), and that the load can be picked up much faster. Note that at the end of the eighth hour there were some nodes with the voltage quite close to the lower limit. For this reason, the voltage set-point of some units had to be increased, or capacitors had to be connected, before proceeding to the load pickup. Finally, the total load was supplied after about 8.5 hours.

4.4.3.3 French regional network

For this network (refer to Figure 4.36), all the units at the 400kV level belong to the RSN. Two units in node N4 were set as TS2 units, and one unit in node N2 was the black-start unit (TS1). In addition, one unit in node n15 (225kV) was also set as black-start unit (TS1). The minimum times of start-up for the shutdown units vary from 3.5 to 5 hours. The simulation results of the restoration procedure found by DAFFOR are shown in Figure 4.44.

The RSN was created after about 40 minutes. After that, the network has been extended towards the auxiliaries of the shutdown thermal units in the whole network. These energisations took about an hour. The consumer loads have started to be picked up after 1 hour and 40 minutes. Note again the transient of the frequency, at the beginning of the third hour, when a large nuclear unit is in ramping while there is a small number of coupled units. This permitted, however, a very large consumer at the 225kV level to be supplied (230MW with the single supply level, i.e., this consumer has no shedding levels). As soon as the other thermal units are coupled, the frequency excursions are much smaller. Also, the rate of the load supply is somewhat faster.

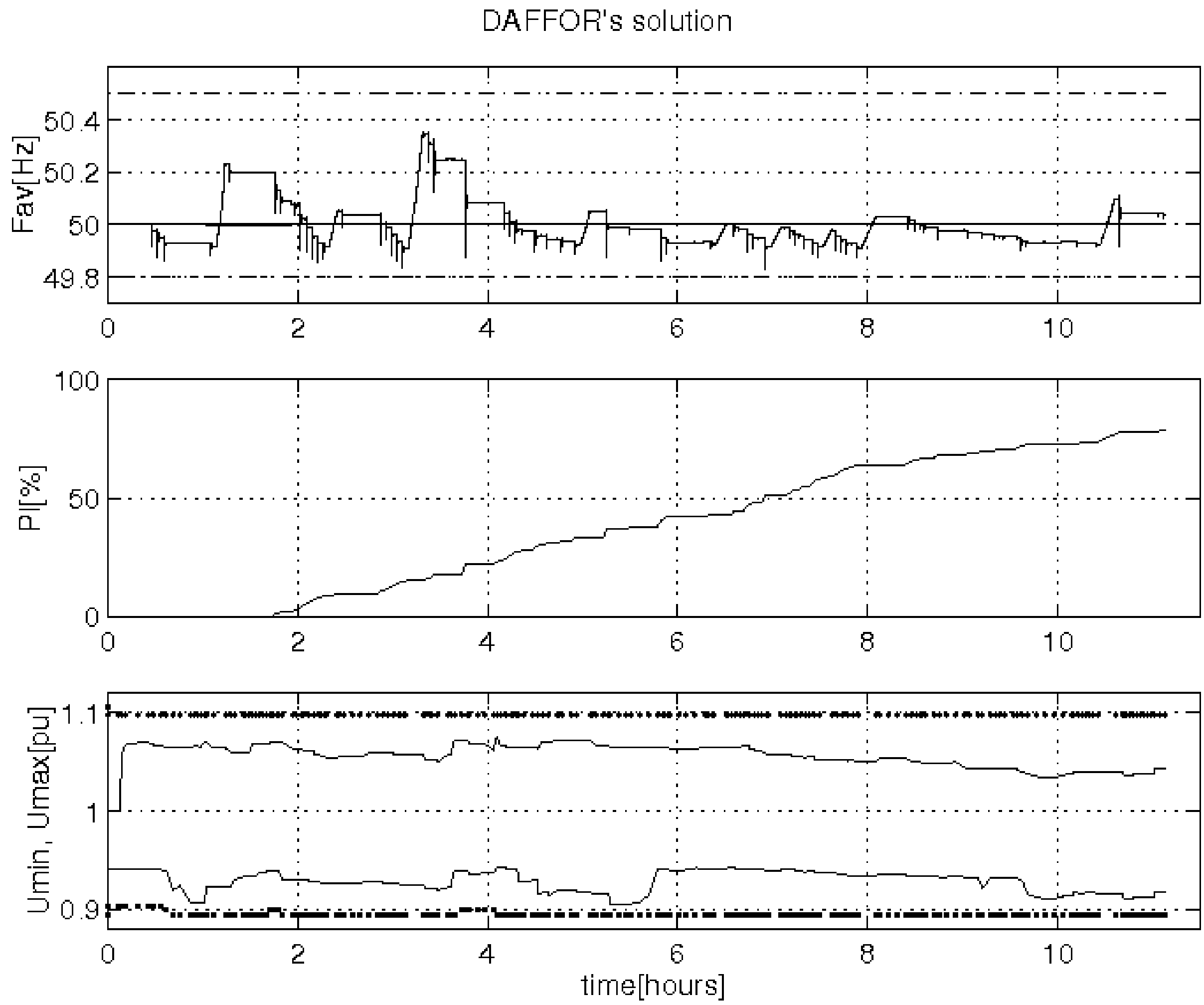


Figure 4.44 Frequency, load supply rate and minimum and maximum voltages during the restoration of the French regional network (from Figure 4.36)

From Figure 4.44 it can be seen that only 80% of the load was supplied after 11 hours. The 20% of load that was not picked up are the consumers in the load zones in the 63kV network. Let us recall that the tap positions of the TCUL transformers are locked at the beginning of the restoration, and the results from Figure 4.44 are obtained with the minimum secondary voltage (63kV side). We have estimated that the decision about the manual tap settings could be left to the operators: they can ask DAFFOR at any moment to assess such a change (or simply perform it independently of DAFFOR), but DAFFOR, at present, cannot suggest such an action. Therefore, the simulation has been stopped at this point.

4.4.3.4 New England network

This test network can be found in Appendix A. The blackout scenario is like the one shown in Figure 4.34 except for the external network support which is considered as available. Three units are defined to be in the RSN: `G2`, `G5` and `G8`. All three are black-start units, meaning TS1 (classical thermal units cannot emit the telemetry TS2). Since there is no shutdown thermal unit in the RSN, the strategy that has been applied in this case is the strategy proposed in this work (the load pickup strategy from Figure 4.24) with no need to apply the RSN-based strategy defined by EDF. It was mentioned that the proposed load pickup strategy might also apply if EDF's RSN-based strategy is not feasible. This is confirmed with the simulation results of the restoration procedure found by DAFFOR, which are shown in Figure 4.45.

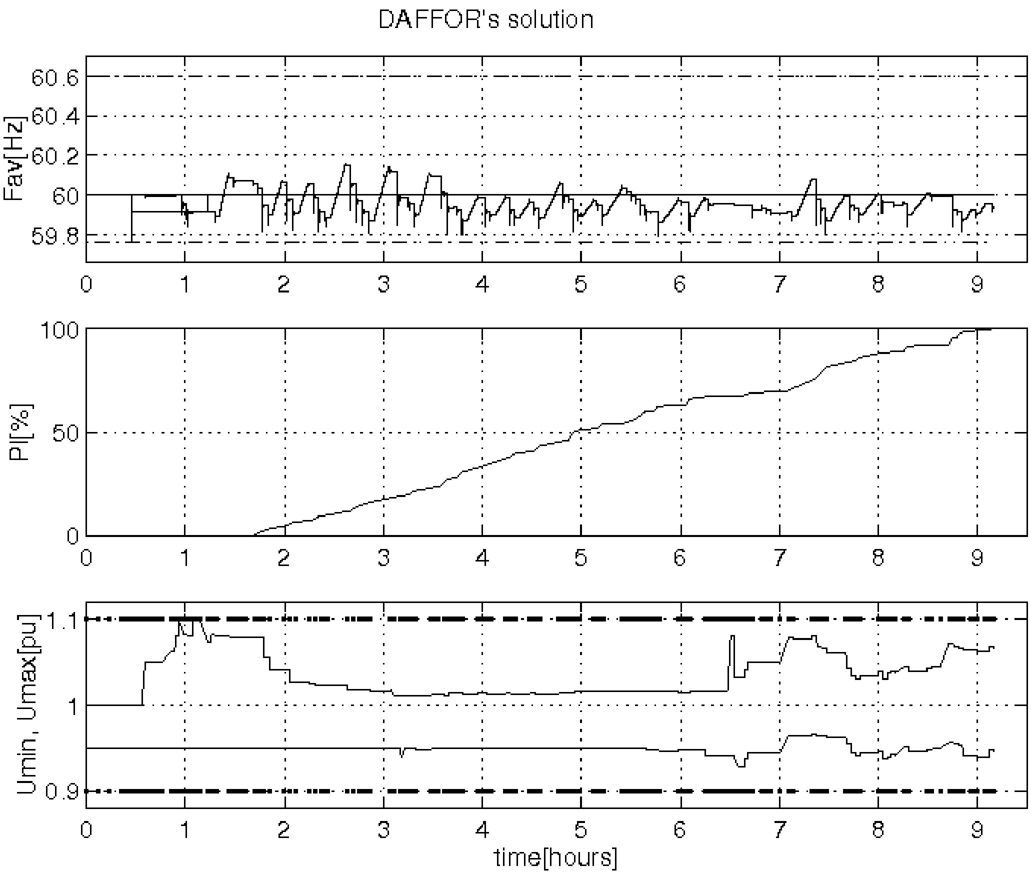


Figure 4.45 Frequency, load supply rate and minimum and maximum voltages during the restoration of the New England test network (from Appendix A)

After the black-start units are coupled, the paths are energised towards the shutdown thermal units in the network. As soon as the coupling nodes of the shutdown thermal units are energised, the auxiliaries can be supplied. These are shown by the first frequency transients.

The units are synchronised into a single electrical zone after about 1 hour and 15 minutes. During the energisation of the paths, while the units represented single electrical zones, the reactive power absorption limit of one unit was almost reached, so that the voltage had to be increased - this caused the maximum voltage peaks around the first hour of restoration.

The impact of the external support is evident over the whole restoration: the frequency excursions are much smaller than for the case of the previous two networks. In this particular network, there is a single equivalent unit, `G1`, but its inertia constant is 7-8 times higher than that of any of the real units. For this reason, the frequency is much more stable.

Since the priority in supply is given to the auxiliaries of the shutdown units, the consumer loads start to be picked up only after about 1 hour and 40 minutes. It can be seen that the load supply rate is quite regular, except during the seventh hour. During this time, the voltage in some nodes reached the lower heuristic limit (see Section 4.1.6.1.2), so that the voltage in the network had to be increased. This explains the maximum voltage peak in the middle of the seventh hour. In addition, the unserved load levels in the energised part of the network were too high, so the network was extended towards the other consumer loads. A similar situation applies to the load pickup plateau during the ninth hour. Finally, the total load was supplied after 9 hours and 7 minutes.

4.4.4 Discussion

The **time necessary for restoration** in the previous 3 examples may seem very long with respect to the amount of the load to pick up. It must be recalled that each control action not only results in the power system dynamics, which take a certain time, but that DAFFOR also considers the time of operators' manoeuvres $T_{man}(a_r)$ given in Table 4.7 per type of action a_r. In Figure 4.43 one can see clearly that from minute 125 to minute 127 nothing happens with the frequency: these two minutes represent the time of manoeuvres for a load pickup action. Only after this time elapses will the action physically be executed and the system responds by variations in frequency.

It is clear that for a network with many elements to operate on (i.e., many actions), the total restoration time is likely to be longer. For the French national network, 135 control actions have been found by DAFFOR (and simulated), while for the French regional network 169 actions were found. Additionally, in the latter the load has not been fully supplied. Comparing now the similar types of networks, that is to say the French national network and the New England one, the number of actions found (135 and 129, respectively) and the restoration time (8.5 and 9 hours, respectively) were similar. In conclusion, the restoration time depends on the complexity of the power system in question, i.e., on the number of control actions.

The restoration time could be decreased by more precisely defining the time of manoeuvres $T_{man}(a_r)$, i.e., by dividing it into two components, as shown in Figure 4.46.

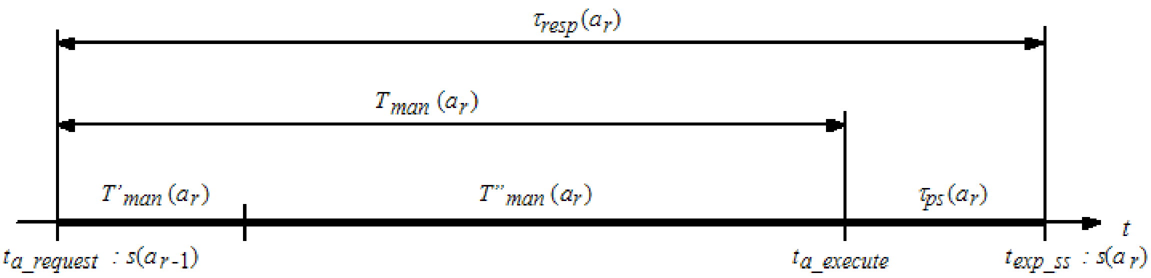


Figure 4.46 Time of manoeuvres decomposed on two intervals

The first component, $T'_{man}(a_r)$, could refer to the time needed by the operators in the control centre to perform a manoeuvre *themselves* (e.g., to press a button or to call to their colleagues in the field), and the second component, $T''_{man}(a_r)$, could refer to the time necessary for performing the manoeuvres remotely. The two parameters should be specified not only per type of action, but for all the elements in the database of the control centre in question. In this manner, if the operators in the control centre can execute action a_r directly, the parameter $T''_{man}(a_r)$ would be always set to 0. This would increase the amount of data to be handled in DAFFOR, but the restoration time would be significantly decreased.

At present, DAFFOR considers that the actions are executed sequentially, even if there are several electrical zones (islands). If there are, say, 2 islands, it is conceivable to make DAFFOR search for two actions for the current state, one for each island. In this case, the Dynamic simulator should have different time scales for each electrical zone. This enhancement to DAFFOR could decrease the restoration time in its first stage, when there are usually several electrical zones.

The **simulation time** depends on the complexity of the power system in question. To issue the restoration procedures for the French national and the New England test network, DAFFOR needed 7 and 9 minutes, respectively, while it took an hour for the French regional network. The reasons for such a difference are the following:

- the number of possible search paths, and
- the amount of data to handle.

It is, however, necessary to clarify that half an hour was spent for the first 8 hours of restoration when 65% of the load was supplied (see Figure 4.44). From the moment DAFFOR had to focus on the supply of the load zones, everything went much more slowly (see discussion in Section 4.4.3.3).

In our opinion, the computation time could be decreased at least 10 times if the Dialogue&control and the Knowledge base modules were implemented in C and/or C++ (instead of NEXPERT *OBJECT*™).

Chapter 5

Reasoning in real time

In the previous chapter, the functionality of the Reasoning kernel has been described in detail, and it was said that the Reasoning kernel represents the stand-alone DAFFOR. However, references have been made several times to this chapter when speaking of the real-time extensions which make the reasoning in real-time conditions possible. These extensions are realised thanks to the Real Time Update (RTUpd) kernel.

The Reasoning and the RTUpd kernel together make the real-time DAFFOR, shown in Figure 5.1 (this is somewhat modified Figure 3.1).

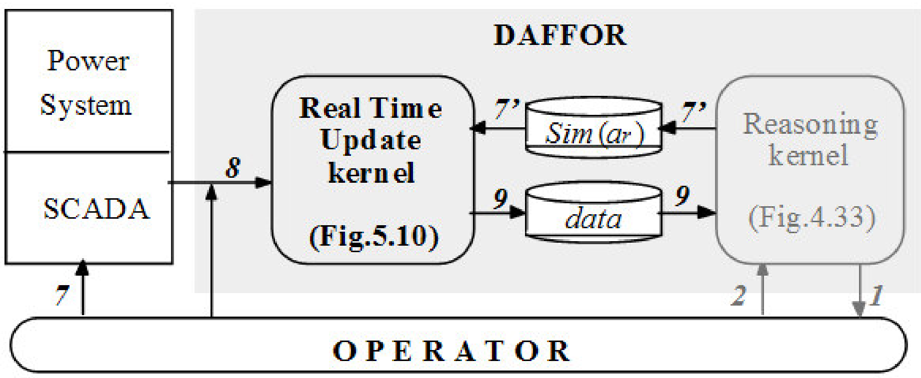


Figure 5.1 The real-time DAFFOR

The RTUpd kernel provides the Reasoning kernel with an image of the power system in real-time (step 9). It uses different kinds of information (steps 7' and 8) in order to generate the relevant data about the power system evolution and make it available to the Reasoning kernel (step 9). Thanks to this information, the Reasoning kernel can adapt its operation to the real-time evolution of the power system state. In other words, the Reasoning kernel can play its guidance role (steps 1 and 2) not only as a function of simulated data (which was the subject of Chapter 4), but also as a function of the real-time data.

Following a bottom-up structure of presentation, as in Chapter 4, the design and the function of the proposed RTUpd kernel will first be described in detail in Section 5.1. It will be shown how the RTUpd kernel uses the output of the Reasoning kernel (step 7') in order to generate the relevant real-time information, i.e., the interaction between the kernels will be focused on from the point of view of the RTUpd kernel.

Section 5.2 describes the interaction between the kernels from the point of view of the Reasoning kernel. In other words, it will be explained how DAFFOR performs its reasoning in real-time. This section will show to what extent the real-time requirements, stated in Section 3.1.2, have been met.

Finally, Section 5.3 presents and discusses validation architecture of the integral DAFFOR, with respect to requirements concerning the real environment use from Section 3.3.3.

Part of the contents of this chapter have already been published in [Kos97].

5.1 Real time update (RTUpd) kernel

The RTUpd kernel is DAFFOR's **real-time data interface**. It steadily gathers information from different data sources in order to generate the only real-time data available to the Reasoning kernel, as follows:

1) the coherent data set, representing the power system at a given moment, and
2) the message, which "indicates" to the Reasoning kernel how to adapt its operation to the current power system state.

The important information for the RTUpd kernel is the reasoning status, i.e., whether there is a terminated reasoning cycle (see Section 4.3.4.1). This internal feedback information to DAFFOR reflects what the operators do, and is used in the RTUpd kernel in order to identify the unforeseen events that may occur in real-time operation.

5.1.1 Update cycle

The RTUpd kernel works in terms of update cycles. An update cycle u is the time interval of constant duration T between two consecutive runs of the RTUpd kernel, as shown in Figure 5.2. At the beginning of update cycle u, the RTUpd kernel first gathers information from different sources. The sources of information will be discussed later; at this point, consider only that the RTUpd kernel processes that information in order to generate current update cycle data, denoted as *data*(u). The time it needs to get real-time information and generate *data*(u) is denoted by e_u. During the time interval (T–e_u), the RTUpd kernel is in stand-by, after which it is launched again in order to generate information *data*(u+1) in the next update cycle. The latter update process requires time e_{u+1}, which may be different from the previous update time e_u (e.g., due to a different quantity of information to process).

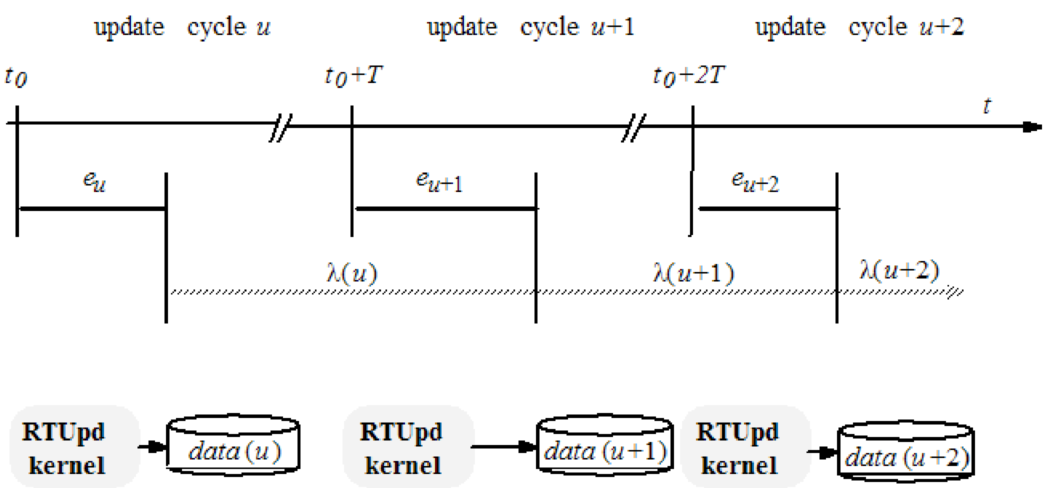


Figure 5.2 Illustration for update cycles on the real-time scale

Remember that the function of the RTUpd kernel is to generate *data*(u) for the Reasoning kernel; *data*(u) describes the power system state during update cycle u. The Reasoning kernel has this information available as soon as it has been generated, i.e., at the end of the interval e_u. Since the next update information *data*(u+1) is available at the end of the interval e_{u+1}, the **Reasoning kernel considers that data does not change between two updates**, i.e., during the interval $\lambda(u)=T+(e_{u+1}-e_u)$. In other words, the time $\lambda(u)$ is left to the Reasoning kernel to operate (i.e., search for the solution) without being interrupted.

5.1.1.1 Duration of update cycle

The duration of update cycles has been adopted as a constant T to ease the implementation[14]. It must be chosen in such a manner that the RTUpd kernel has enough time to process data for the most pessimistic case, i.e., it should be longer than or equal to the maximal estimated time interval e_u.

A variable duration of update cycles could be considered as well. For instance, the update u+1 could start as soon as the previous update u has terminated. In this case, the duration of update cycle u would equal the time interval e_u. However, if real-time information *data*(u) generated by the RTUpd kernel often changes from one update cycle to another, the Reasoning kernel might be forced to adapt its own operation too often if the update cycle (in this case e_u) is

[14] It seems quite logical to perform the update in a synchronous manner (most real-time applications do it that way). Indeed, it would be difficult to choose the factor that launches the update asynchronously such that it is neither too frequent nor too rare.

short. This might cause an inability of the Reasoning kernel to find a solution because of the lack of time.

5.1.2 Coherent data set

As mentioned in the introduction, the first task of the RTUpd kernel is to generate coherent information corresponding to the power system state in the current update cycle *u*. This information will be referred to as the **coherent data set** and is denoted as *Real*(*u*). In order to generate the coherent data set *Real*(*u*), the RTUpd kernel reads real-time information from two sources, as shown in Figure 5.3:

1) external world information (step 8): power system real-time data provided by SCADA (switch status and other telemetry) and information entered by the operators, and
2) DAFFOR's internal information (step 10'): RTUpd kernel's output generated in the previous update cycle, i.e. the last coherent data set, *Real*(*u*–1).

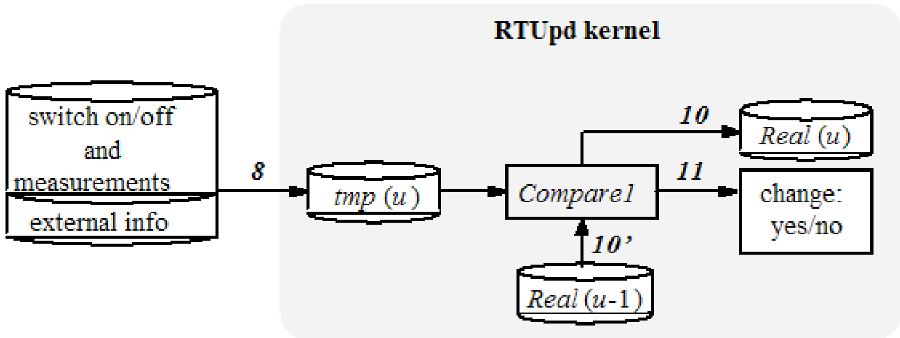


Figure 5.3 RTUpd kernel generates coherent data set thanks to real-time information from the power system and provided by the operators on one hand, and its own output from the previous update on the other.

The RTUpd kernel first gets real-time data from the power system and the information provided by the operators (step 8) and saves it in the temporary data structure, denoted by *tmp*(*u*). The information provided by the operators actually comes from the devices external to SCADA (telephone, radio, synoptic map, printer) and may be threefold:

1) switch status not available in SCADA,
2) corrections of invalid switch status, and
3) unavailability of a power system component (transformer, switch, etc.).

This information can be introduced by the operators at any time, but will only be processed at the same time as information coming from the power system, during time interval e_u.

The data just read, *tmp*(*u*–1), is then compared against the previous update output, *Real*(*u*–1) (step 10'), in the function *Compare1*. There are two intermediate results of this comparison:

1) If some real-time data is missing in *tmp*(*u*), the corresponding items from *Real*(*u*–1) will be assumed. The real-time data set obtained in this way is the coherent data set *Real*(*u*) (step 10); and

2) The current coherent data set, *Real*(u), is compared against the previous coherent data set, *Real*(u–1), in order to determine whether the power system state has changed since the last update (step 11).

The precise meaning of "whether the power system state has changed since last update" will be discussed in Section 5.1.5.2, as well as the function *Compare1*.

5.1.3 Message for reasoning kernel

The second task of the RTUpd kernel is to issue a "message" for the Reasoning kernel as a function of the power system evolution in real time. The Reasoning kernel receives the message and adapts its operation to be in step with the current state of the power system.

In order to generate the message for the Reasoning kernel, the RTUpd kernel should be capable of "deducing" the following:

- if the state has changed from the previous update, whether it was expected (due to a control action executed by the operators) or unexpected (due to an unforeseen event), and
- if the state did not change, but was expected to (due to a control action executed by the operators), there must have been an unforeseen event.

However, the intermediate results described in Section 5.1.2 (coherent data set *Real*(u) and state change status) are not sufficient for the RTUpd kernel to issue the meaningful message for the Reasoning kernel, since the RTUpd kernel itself has no information on what the operators do. Since the Reasoning kernel does have that information, **we propose that the RTUpd kernel use the following feedback information from the Reasoning kernel in order to assess the events in the power system**:

- reasoning **status**, i.e., whether there is an ongoing reasoning cycle, and
- reasoning **result** in case there is a terminated reasoning cycle, i.e., simulated time (4.13) and simulated steady state (4.22).

In the following discussion, it will be supposed that the RTUpd kernel has just generated the coherent data set *Real*(u) and the state change status for the current update cycle u, as described in Section 5.1.2. Depending on the reasoning status and the state change status, the RTUpd kernel will issue the appropriate message for the Reasoning kernel.

5.1.3.1 Ongoing reasoning

Let us suppose that there is an ongoing reasoning cycle r during the current update data processing e_u, as shown in Figure 5.4. The RTUpd kernel would interpret this fact as follows:

- the Reasoning kernel is still searching for control action a_r; therefore,
- no action has been suggested to the operators; therefore,
- the operators are not supposed to execute any action on the power system **known** to DAFFOR; consequently,
- there should be no changes in the power system state from the previous update.

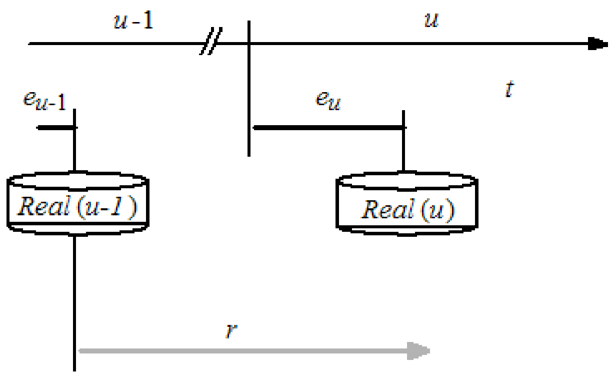


Figure 5.4 There is an ongoing reasoning cycle, *r*, in the current update cycle, *u*

If there have been no changes, no message need to be issued – the Reasoning kernel will simply continue its current cycle *r* without interruption.

However, if some change has been identified, this may be due to one of the following:

1) the operators have executed a control action which is **unknown** to DAFFOR, or
2) something unexpected has happened in the power system (e.g., the switch has been operated without a manoeuvring requirement, i.e., by a protection device; an industrial load connected without informing the control centre; etc.).

Any of the two cases above will be considered by the RTUpd kernel as an unforeseen event (Figure 5.5). Since the power system state changed in an unexpected manner, the Reasoning kernel must adapt its operation to the new situation, i.e., to the new power system state, *Real*(*u*). Therefore, the message which the RTUpd kernel issues in this case will be "*restart*". This case is represented by step A in Figure 5.5.

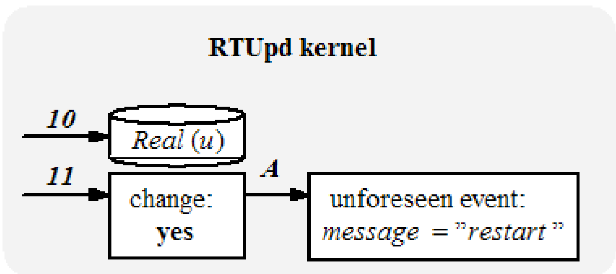


Figure 5.5 Message generation for the ongoing reasoning when an unexpected change has been identified

5.1.3.2 Terminated reasoning

Suppose now that there is a reasoning cycle *r* that had been completed before the current update data processing e_u, at the moment denoted as $t_{a\_request}(a_r)$, as shown in Figure 5.6. From the time $t_{a\_request}(a_r)$ on, the Reasoning kernel must wait for a message from the RTUpd kernel, in order to keep up with what will happen in the power system as a result of action a_r.

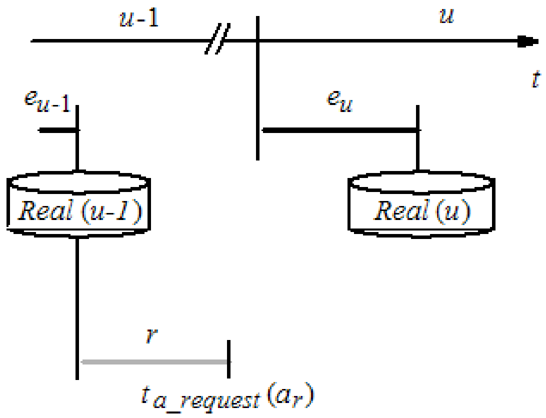


Figure 5.6 There is a terminated reasoning cycle, *r*, in the current update cycle, *u*

The fact that there is a terminated reasoning cycle is interpreted in the RTUpd kernel as follows:

- the Reasoning kernel has already suggested control action a_r to the operators, and they accepted it; therefore,
- the operators are supposed to execute on the power system action a_r, which is **known** to DAFFOR; consequently,
- some changes in the power system state should be identified soon, due to action a_r.

There are two questions the RTUpd kernel should be able to answer in this case: (1) how soon and (2) how much the state will change due to action a_r. To find answers to these questions, the RTUpd kernel will use the information resulting from the terminated reasoning cycle *r* as follows:

1) the simulated total response time $\tau_{resp}(a_r)$ to action a_r, given by equation (4.13), will be used as measure of expected time, and
2) the simulated steady state $Sim(a_r)$ after action a_r, given by equation (4.22), will be used as measure of expected change in the power system state.

However, it may happen that the state does not change soon, in which case a message should be issued to the Reasoning kernel as well. This requirement makes the determination of the meaning of "soon" important. In fact, some time limit must be imposed on the identification of the expected change in the RTUpd kernel, in order for it to not keep waiting (and making the Reasoning kernel wait too) for a change which does not occur at all.

5.1.3.2.1 Expected update cycle

The expected update cycle, denoted by *y*, represents the above mentioned time limit – it answers the question "How soon should the state change?" **The expected update cycle *y* is defined as the latest update cycle in which the RTUpd kernel should identify the changes in the power system state due to the executed control action a_r.** If no change is identified during the expected update cycle, the RTUpd kernel concludes that something unexpected happened.

The expected update cycle y is determined as a function of the simulated total response time $\tau_{resp}(a_r)$ (4.13) to action a_r by the following deductions:

- Suppose that reasoning cycle r has ended at moment $t_{a\_request}(a_r)$, which falls within update cycle u–1, as shown in Figure 5.7.a.
- The result of the terminated reasoning cycle is the steady state $Sim(a_r)$, which contains total response time $\tau_{resp}(a_r)$ expected after action a_r, already simulated during the reasoning cycle r itself. Since the result of reasoning is *simulated* data, it is represented on a separate *simulation* time axis t' in Figure 5.7.b.
- The translation of the simulated response time $\tau_{resp}(a_r)$ from simulation to real time axis (represented as dashed links between Figures 5.7.a and 5.7.b) enables the RTUpd kernel to determine the moment at which the expected steady state $Sim(a_r)$ should be reached in real time after action a_r. This moment is denoted as $t_{exp\_ss}(a_r)$. Figure 5.7 can be compared to Figure 4.10.
- Since the operators have accepted action a_r at moment $t_{a\_request}(a_r)$, the RTUpd kernel considers that they will need time $T_{man}(a_r)$ to perform the manoeuvres on the power system, and after that the power system should respond by some transients, during time interval $\tau_{ps}(a_r)$, until the steady state is reached, by moment $t_{exp\_ss}(a_r)$.
- The update cycle within which falls the expected steady state, is denoted as u'. For the given example $u'=u+1$, but in the general case u' may be any update cycle that starts after the current one or the current update cycle itself, i.e., $u'\geq u$. Since the expected steady state moment $t_{exp\_ss}(a_r)$ is during the time in which the RTUpd kernel is in stand-by, if the expected steady state had not been already identified (in updates u or $u'=u+1$), that should be done at the latest during the next update interval. The corresponding update cycle represents the expected update cycle $y=u'+1$.

The above deduction shows how the **simulated** total response time $\tau_{resp}(a_r)$ for action a_r is used in the RTUpd kernel to estimate the **real** time by which the steady state after action a_r should be reached.

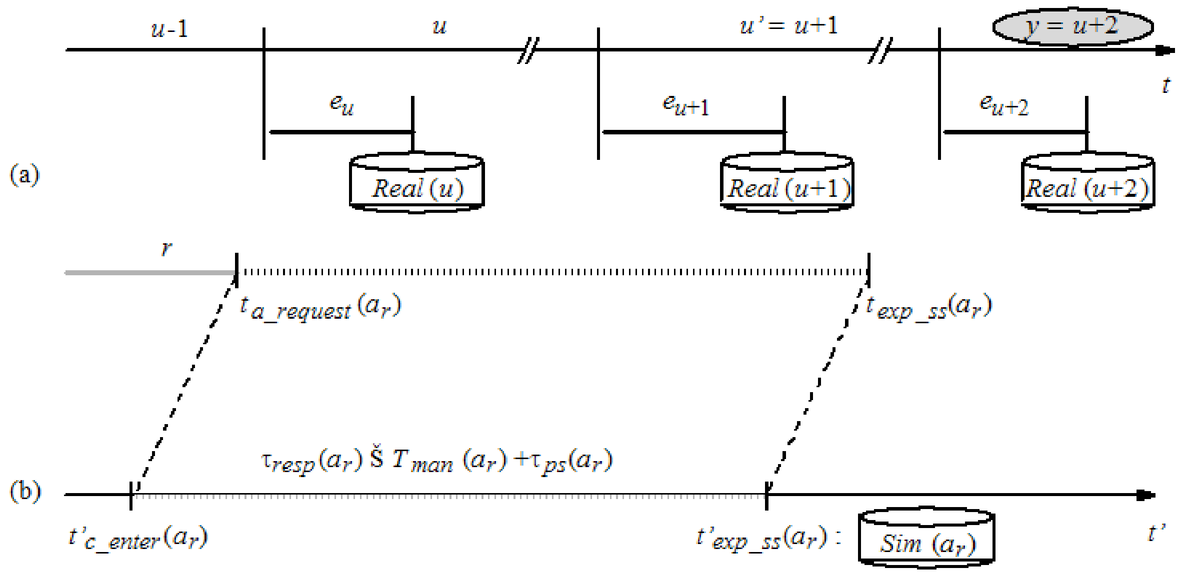


Figure 5.7 Determination of the expected update cycle y

The time of manoeuvres $T_{man}(a_r)$ (introduced in Section 4.1.4.1) has been supposed to be the upper limit time. For instance, if action a_r is the energisation of a line, the time of manoeuvres is at most 2 minutes (see Table 4.7). If the operators need only to press a button, the line energisation may take just a few seconds. For the example given in Figure 5.7, the steady state after the line is energised might be, therefore, identified already in the current update cycle u. However, if the operators have to ask their colleagues in a substation for the line energisation, the whole time of manoeuvres is likely to be longer than a few seconds, although less than 2 minutes. The parameter $T_{man}(a_r)$ for the given type of action has been chosen in such a manner that the worst case is represented. In other words, whatever the operators have to do in order to physically execute action a_r, that will take at most the time defined by the corresponding parameter $T_{man}(a_r)$. For that reason, and taking into account the definition of the expected update cycle y, the steady state $Sim(a_r)$ after action a_r should be identified either in the current update cycle or in some of the following ones, but at the latest in the expected update cycle y, i.e., while $u \leq y$.

However, if no change has been identified even during the expected update interval e_y, this may be due to one of the following:

1) for some reason, the operators have not executed action a_r on the power system although they told DAFFOR they would, or
2) something unexpected happened in the power system (e.g., the switch has been blocked so that action a_r produced no power system response although it has been executed).

Any of the two cases above will be considered by the RTUpd kernel as an unforeseen event (Figure 5.8). Since the power system state did not change at all, the Reasoning kernel must be informed that action a_r had no effect on the power system. Therefore, the message the RTUpd kernel issues in this case will be "*disable* a_r *and continue*". This case is represented by step B in Figure 5.8.

Note that the expected update cycle y may be defined only if DAFFOR has something to expect, i.e., when there is a terminated reasoning cycle r which gives the expected steady state $Sim(a_r)$ after action a_r. Therefore, for the case when the reasoning is still going on (Figure 5.5), the expected update cycle is set to 0, i.e., y=0.

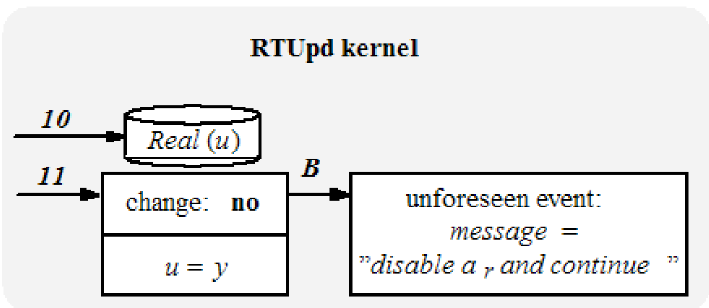


Figure 5.8 Message generation for the terminated reasoning when the expected change has not been identified

5.1.3.2.2 Expected steady state

In the previous section it was deduced that the RTUpd kernel should identify the steady state $Sim(a_r)$ after action a_r either in the current update cycle or in some of the following ones, but at the last in the expected update cycle, i.e., while $u \leq y$. Therefore, in each update interval e_u such that $u \leq y$, in which some change from the previous update has been identified, the RTUpd kernel compares both the current coherent data set $Real(u)$ and the expected steady state $Sim(a_r)$. This comparison enables the RTUpd kernel to answer the question "How much should the state change?", and generate an appropriate message for the Reasoning kernel, as shown in Figure 5.9.

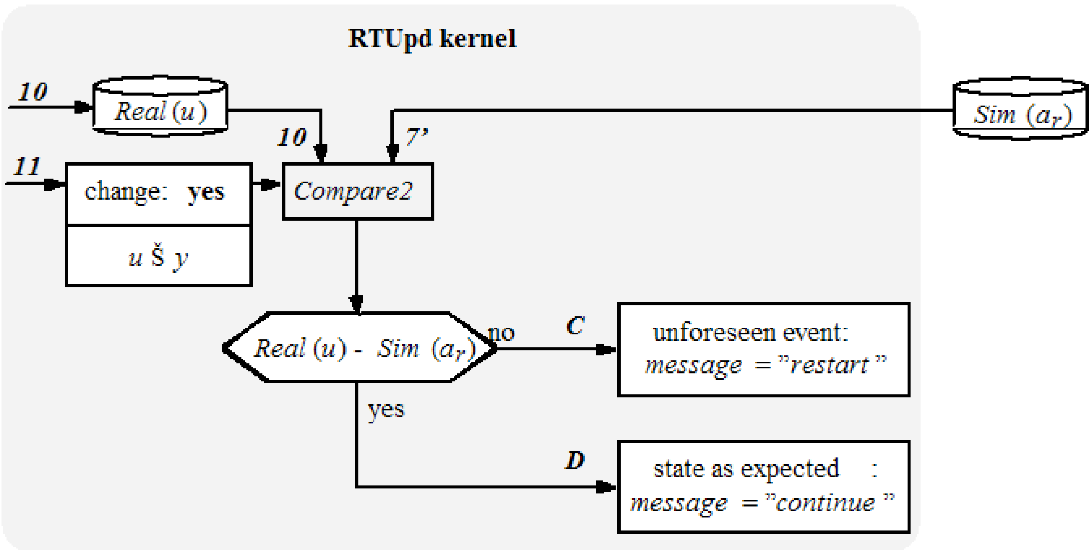


Figure 5.9 Message generation for the terminated reasoning when the power system state has changed differently than expected (step C) or as much as expected (step D)

The current coherent data set $Real(u)$ is compared against the expected steady state $Sim(a_r)$ in the function *Compare2* (steps 10 and 7'). The result of this comparison may be twofold:

1) The coherent data set $Real(u)$ **differs significantly** from the expected steady state $Sim(a_r)$. This difference may be due to some of the following:
   - for any reason, the operators did not execute action a_r on the power system (although they told DAFFOR they would) but instead executed some other action, **unknown** to DAFFOR, or
   - for any reason, the operators executed neither action a_r nor any other action, but something unexpected happened (e.g., the switch was operated without a manoeuvring requirement, i.e., by a protection device), or
   - the operators executed action a_r on the power system and in addition something unexpected happened.

   Any of the three cases above will be considered by the RTUpd kernel as an unforeseen event. Since the power system state changed in an unexpected manner, the Reasoning kernel must adapt its operation to the new situation, i.e., to the new power system state, $Real(u)$. Therefore, the message the RTUpd kernel issues in this case will be "*restart*". This case is represented by step C in Figure 5.9. Note that the message is the same as in case A (shown in Figure 5.5 and described in Section 5.1.3.1). However, the response of the Reasoning kernel will be somewhat different in these two cases since in one the reasoning is still going on, and in the other the reasoning has terminated. This will be discussed in Sections 5.2.1.1 and 5.2.2.2, respectively.

2) The coherent data set *Real*(u) is **close enough** to the expected steady state *Sim*(a_r). This means that the expected steady state has been identified and that the Reasoning kernel can continue its operation. Therefore, the message that the RTUpd kernel issues in this case will be "*continue*", which is represented by step D in Figure 5.9.

Function *Compare2*, as well as the meaning of expressions "differs significantly" and "close enough" will be discussed in Section 5.1.5.2.

5.1.4 Summary

In Sections 5.1.1 through 5.1.3, basic concepts concerning the operation of the RTUpd kernel and its dependence on the operation status of the Reasoning kernel have been explained. Figure 5.10 gives the flow chart of the RTUpd kernel that merges the above items in two passes: Pass 1 represents the operation of the RTUpd kernel described in Section 5.1.2, and Pass 2 its operation described in Section 5.1.3.

In **every** update cycle u, the RTUpd kernel generates the coherent data set *Real*(u) (step 10) and the system change status (step 11). In **certain** update cycles, it also generates a message (steps A, B, C and D) for the Reasoning kernel. The following list summarises the conditions under which it is necessary to generate a message for the Reasoning kernel:

A) if data has changed since the last update (*change*=true)
and there is an ongoing reasoning cycle ($y=0 \rightarrow u>y$)
==> message is "*restart*";

B) if data has not changed since the last update (*change*=false)
and there is a terminated reasoning cycle ($y \neq 0$)
and the current update cycle is the expected one ($u=y$)
==> message is "*disable a_r and continue*";

C) if data has changed since the last update (*change*=true)
and there is a terminated reasoning cycle ($y \neq 0$)
and the current update cycle is not later than the expected one ($u \leq y$)
and the real state is different than the expected one ($Real(u) \neq Sim(a_r)$)
==> message is "*restart*";

D) if data has changed since the last update (*change*=true)
and there is a terminated reasoning cycle ($y \neq 0$)
and the current update cycle is not later than the expected one ($u \leq y$)
and the real state is as expected ($Real(u) \approx Sim(a_r)$)
==> message is "*continue*".

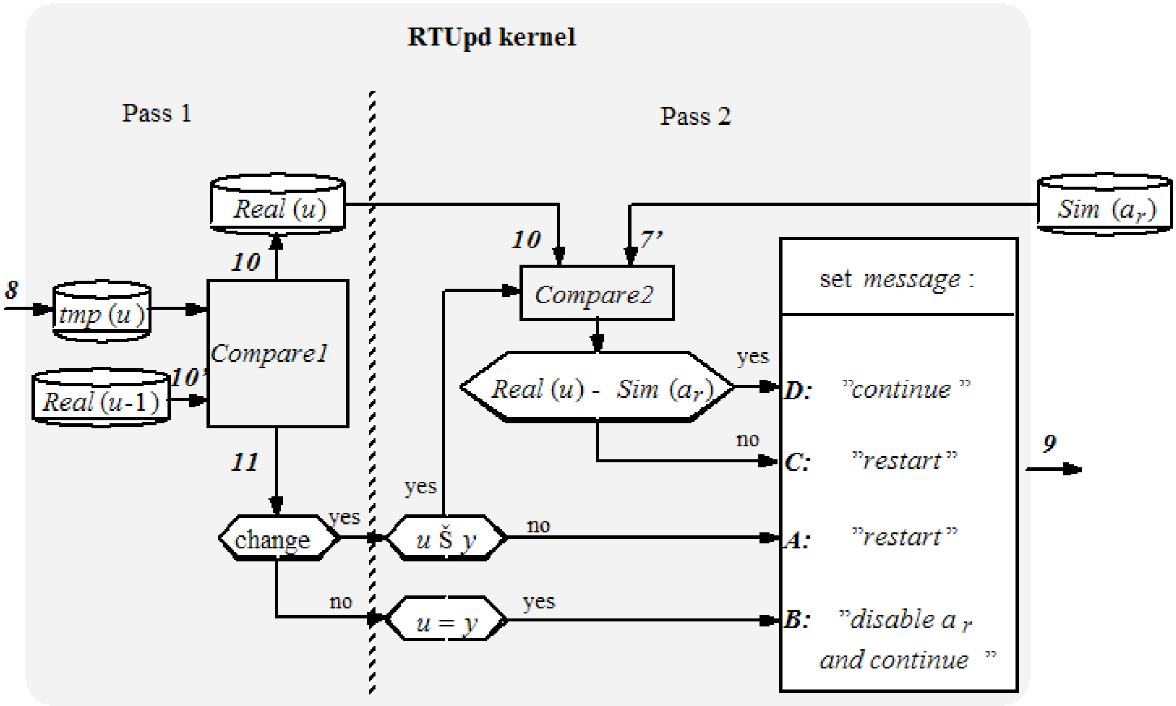


Figure 5.10 Flow chart of the RTUpd kernel

Note that in Pass 2 the RTUpd kernel identifies the unforeseen events in all the above-mentioned cases except D, in which it identifies the expected steady state resulting from action a_r executed by the operators on the power system.

Note also that when there is a terminated reasoning cycle (all above-mentioned cases except A), the Reasoning kernel is paused and is waiting for a message from the RTUpd kernel in order to resume its operation. Therefore, as soon as any of messages B, C or D are issued, the Reasoning kernel will continue its operation thanks to the reception of the message. In other words, there will be an ongoing reasoning, so that the expected update cycle y is reset (i.e., y=0). Consequently, the condition u>y is true only if y=0, i.e., when there is an ongoing reasoning.

Having this in mind, there are two other possible cases, which are not represented in Pass 2 in Figure 5.10 since no message need be issued:

E) if data has not changed since the last update (*change*=false)
and there is an ongoing reasoning cycle ($y=0 \rightarrow u>y$)
==> no message;

F) if data has not changed since the last update (*change*=false)
and there is a terminated reasoning cycle ($y \neq 0$)
and the expected update cycle has not been reached yet ($u<y$)
==> no message;

In fact, for case E it is not necessary to interrupt the ongoing reasoning since nothing has changed in the power system.

For case F some change is expected, but it still can be identified in the expected update cycle *y*. Regardless of changes in the power system, one of the messages B, C or D will be issued in any case at the latest during the expected update cycle *y*. Here can be seen the importance of the expected update cycle *y*: **once when the Reasoning kernel terminated its cycle *r*, it will not wait for a message from the RTUpd kernel longer than until the expected update cycle *y*; this message would indicate to the Reasoning kernel how to continue its operation**.

5.1.5 Discussion on data reliability

In the general case, *data*(*u*) is generated by the RTUpd kernel during the update cycle *u* on the basis of real-time data (from SCADA function), and through some comparisons between power system states. Since *data*(*u*) is the only "external world" data available to the Reasoning kernel, the accuracy of the reasoning is, consequently, completely dependent on the accuracy of *data*(*u*).

5.1.5.1 State estimation

One of the factors that strongly influences the reliability of *data*(*u*) (both *Real*(*u*) and message) is the information external to DAFFOR: the power system’s real-time data. The data on the power system is gathered by the SCADA function and treated inside the state estimator. During the restoration process, or at least in its first stage, the information available to the state estimator is likely to be incomplete (e.g., unobservable) and thus the state estimator might not converge. Even if the information is not incomplete, its quality might cause a convergence which is too slow for real-time purposes.

Both above-mentioned cases are sources of problems for the RTUpd kernel, since they result in a lack of the power system's real-time data as seen from the standpoint of the RTUpd kernel. Therefore a special, restoration-dedicated state estimator should be available in order for the RTUpd kernel to generate *data*(*u*) reliable enough for the correct operation of the Reasoning kernel.

5.1.5.2 Comparison functions

Another factor that influences the reliability of *data*(*u*) generated by the RTUpd kernel is the quality of DAFFOR's internal comparison functions *Compare1* and *Compare2*, introduced in Sections 5.1.2 and 5.1.3.2.2, respectively.

At present, the functions *Compare1* and *Compare2* take into account only the topology changes, computed on the basis of the switching status. As the operating status of the power system elements has a discrete nature (open or closed), it is easy to compare two power system states topologically. However, this is not enough for the creation of a sufficiently reliable *data*(*u*). The telemetry should also be considered in comparison functions, which is much more complicated to manage.

Function *Compare1* compares the current coherent data set *Real*(*u*) against the coherent data set from the previous update, *Real*(*u*–1), in order to deduce whether the power system has changed (see Section 5.1.2). At present:

- "change" is set to False if the coherent data sets *Real*(*u*) and *Real*(*u*–1) are topologically identical, and
- "change" is set to True otherwise.

The function *Compare2* compares the current coherent data set *Real*(*u*) against the simulated steady state, *Sim*(a_r), in order to deduce whether the power system state has changed as expected due to action a_r (see Section 5.1.3.2.2). At present:

- the coherent state *Real*(*u*) is said to be "close enough" to the expected state *Sim*(a_r) if they are topologically identical, and
- the coherent state *Real*(*u*) is said to be "significantly different" from the expected state *Sim*(a_r) otherwise.

Since the function *Compare1* compares data of the same type (*Real*(*u*) and *Real*(*u*–1)), in our opinion it should not be too difficult to determine threshold values associated to the telemetry data, which permit a conclusion as whether the state of the power system has changed. On the contrary, the comparison of power injections, frequency, voltages, etc., in the function *Compare2* is much more delicate, because the types of data to be compared are different: the coherent data set *Real*(*u*) contains telemetry, while the expected steady state data *Sim*(a_r) contains results of the simulation. In this case, it would be necessary to determine threshold values for the continuous power system variables (powers, frequency, etc.) in such a manner that the simplicity of DAFFOR's models compared to the real power system could be overcome.

The problem of determination of the threshold values to be used in the function *Compare1* and especially in the function *Compare2* is quite complex and demands further research.

5.1.6 Conclusion

The function of the RTUpd kernel is to steadily gather information from different data sources and to generate information which indicates to the Reasoning kernel what is happening in the power system in real time. In the update cycle *u*, the RTUpd kernel generates data *data*(*u*) which consists, in general, of two parts:

1) the coherent data set *Real*(*u*), and
2) the message for the Reasoning kernel.

Although the RTUpd kernel has no information about what the operators do, the Reasoning kernel does. Therefore, it has been proposed that the RTUpd kernel use reasoning status (ongoing or terminated reasoning) and reasoning results (simulated expected steady state and time), which are both functions of the interaction with the operators, in order to generate an appropriate message.

The message may be threefold (see Section 5.1.4):

- "*restart*", for cases A and C,
- "*disable* a_r *and continue*", for case B, and
- "*continue*", for case D.

For cases E and F, no message is issued and nothing changes in the operation of the Reasoning kernel.

Figure 5.11 indicates data generated by the RTUpd kernel (step 9) which influence the operation of the Reasoning kernel. Note that the coherent data set *Real*(*u*) needs to be used by the Reasoning kernel only when the power system has changed unexpectedly (cases 9.A and 9.C).

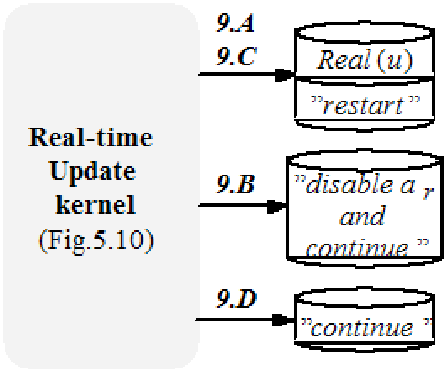


Figure 5.11 Data generated by the RTUpd kernel, during update cycle *u*, for the Reasoning kernel. Cases A, B, C and D are those enumerated in Section 5.1.4.

Finally, note that the RTUpd kernel is launched synchronously, every *T* seconds, whatever the current status, results or length of the reasoning cycle might be. However, the **operation of the RTUpd kernel in cycle** *u* depends on the reasoning **status** (the reasoning must either still be running or terminated) and the reasoning **results** (if the reasoning is terminated).

5.2 DAFFOR's real-time operation

This section concentrates on the operation of the Reasoning kernel as a function of the RTUpd kernel's outputs, shown by steps 9.A, 9.B, 9.C and 9.D in Figure 5.11. In other words, it will explain how the Reasoning kernel keeps up with the evolution of the power system in real time.

5.2.1 Ongoing reasoning

The Reasoning kernel in real-time operation depends on data generated by the RTUpd kernel. The very first reasoning cycle, r=1, can start only after the RTUpd kernel terminates data processing in its first cycle, u=1, i.e., after update interval e_1. In this case, the RTUpd kernel generates (step 9.C) the coherent data set *Real*(1) and the message "*restart*", as shown in Figure 5.12, which enable the Reasoning kernel to start its first cycle, r=1.

Referring to the transition expression (4.1), the initial state is denoted by s(0), since this is the first reasoning cycle (r=1). Therefore, the current power system state in the Reasoning kernel will be initialised with the coherent data set, i.e., s(0)=*Real*(1). Then the reasoning focused on action a_1 starts.

The reasoning cycle may extend over several update cycles if the power system state does not change, i.e., if no unforeseen event is identified. In this case (case E from Section 5.1.4), the RTUpd kernel does not issue any message for the Reasoning kernel. For the example in Figure 5.12, there is no output from the RTUpd kernel after the second and the third update intervals, e_2 and e_3, respectively. This fact enables the Reasoning kernel to complete its cycle r=1.

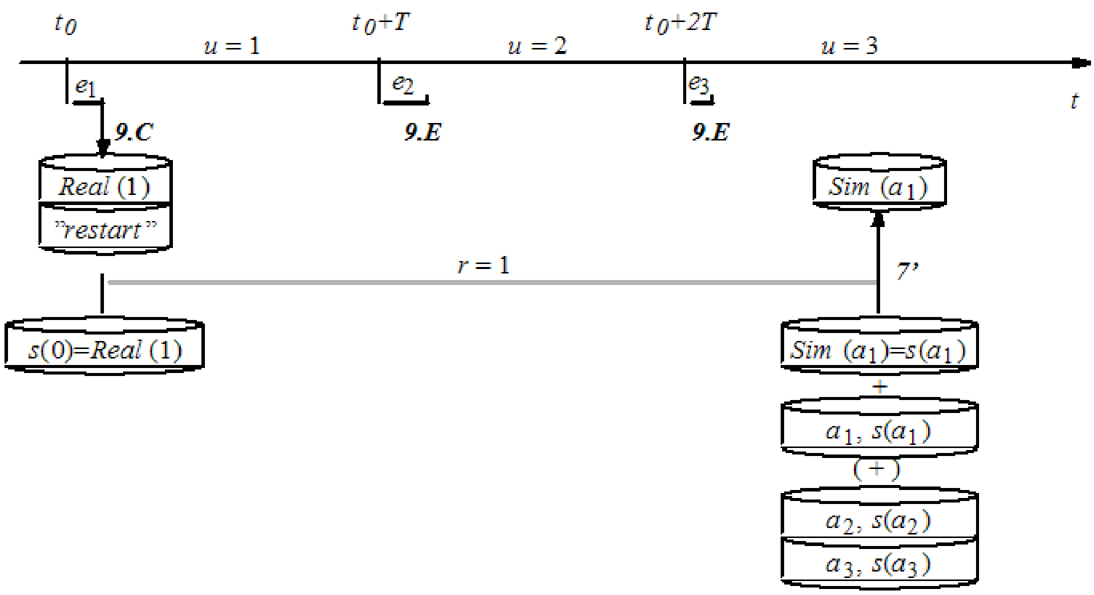


Figure 5.12 There were no messages during the reasoning cycle r=1, so it could terminate.

Remember that the reasoning cycle r=1 is terminated when the operators accept to execute action a_1 on the power system. As concluded in Section 4.3.4:

1) Action a_1 may have been suggested to the operators as the solution, and as such, it represents the head of sub-sequence (4.21) of N_{act} actions for r=1. In this case, at the end of reasoning cycle r=1, the Reasoning kernel contains not only action a_1, but the whole sub-sequence. Figure 5.12 shows this sub-sequence for N_{act}=3; or
2) Action a_1 may have been suggested by the operators and disapproved through the reasoning, in which case it is not the head of any sub-sequence (4.21), and actions a_2 and a_3 from Figure 5.12 would not exist.

For any of the two cases above, the expected simulated state will be initialised with the state resulting from action a_1, i.e., $Sim(a_1)=s(a_1)$.

The expected simulated state $Sim(a_1)$ (step 7') is the result of the terminated reasoning cycle r=1. The Reasoning kernel is waiting for a message from the RTUpd kernel; how it resumes its operation after the reception of the message will be explained in Section 5.2.2, which deals with the terminated reasoning.

5.2.1.1 RTUpd kernel's output 9.A

Let us consider now the example of the reasoning cycle r=1 which cannot terminate because the RTUpd kernel identifies an unforeseen event. This is shown in Figure 5.13, for the case in which the RTUpd kernel has generated some data in its cycle u=3, i.e., after update interval e_3 (step 9.A). Since the power system state has changed, the Reasoning kernel must keep up with this change: it must consider as its current power system state the one generated by the RTUpd kernel, i.e., $s(0)=Real(3)$. Since the action a_1 has been neither found nor accepted, the Reasoning kernel is said to restart the reasoning on action a_1, i.e., it restarts the same cycle, r=1.

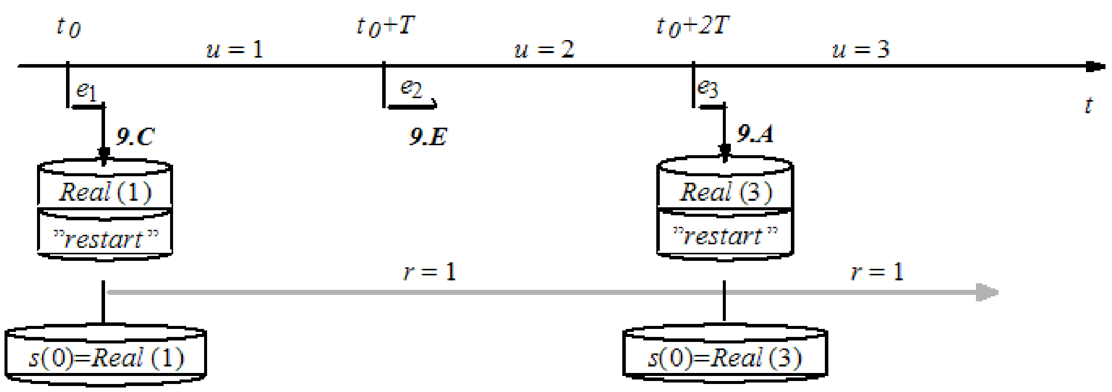


Figure 5.13 Message "*restart*" (step 9.A) has been received during reasoning cycle r=1: the cycle is restarted with new data.

5.2.2 Terminated reasoning

When the reasoning cycle r has terminated (because the operators accepted action a_r known to DAFFOR), the Reasoning kernel is waiting for a message from the RTUpd kernel (one of the outputs 9.B, 9.C or 9.10 from Figure 5.11). This message will indicate to the Reasoning kernel how and with which data to continue its operation, as a function of what has happened in the power system after the operators have executed action a_r.

The operation of the Reasoning kernel will be illustrated with the terminated reasoning cycle r=1 from Figure 5.12. The end of the reasoning falls in the update cycle u=3 within the interval in which the RTUpd kernel is in stand-by. Therefore, the RTUpd kernel will "see" that there is some simulated data ($Sim(a_1)$ from step 7') when it starts processing data during update interval e_4. It will conclude that there is a terminated reasoning cycle (r=1), and will determine the expected update cycle y.

5.2.2.1 RTUpd kernel's output 9.B

Let us suppose that the RTUpd kernel has determined that some change in the power system, due to action a_1, should occur at the latest in its sixth cycle, i.e., the expected update cycle is y=6. For the example shown in Figure 5.14, the RTUpd kernel generates no output after the forth and the fifth update interval (e_4 and e_5, respectively), since there has been no change identified (case 9.F from Section 5.1.4), and the Reasoning kernel keeps waiting. However, if the RTUpd kernel does not identify the change even in the expected update cycle y=6, it generates output in step 9.B.

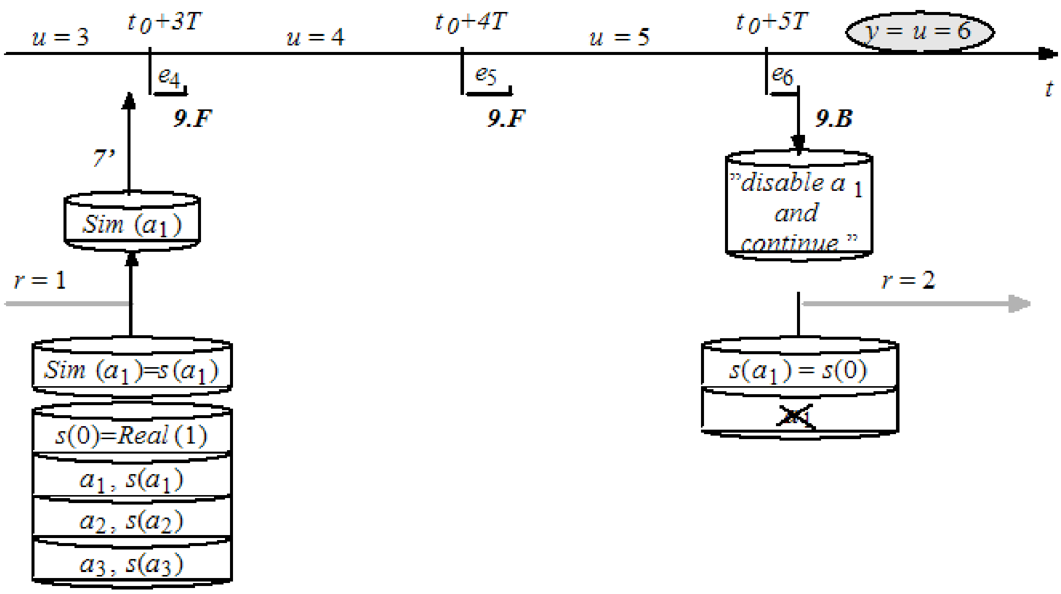


Figure 5.14 Message "*disable a_r and continue*" (step 9.B) has been received while waiting after reasoning cycle r=1: the next one (r=2) continues with old data.

The Reasoning kernel first disables action a_1, which has not caused any response from the power system[15]. Then it resets all the internal data it had except the old initial state $s(0)$, which remains the current state for the next reasoning cycle ($s(a_1)=s(0)$, $r=2$). Finally it starts its second cycle in order to concentrate on action a_2.

5.2.2.2 RTUpd kernel's output 9.C

Consider now that the RTUpd kernel has identified some change in the power system while processing data in the fifth update, e_5. In addition, it has concluded that the change was not as expected, i.e., it states that *Real*(5) and *Sim*(a_1) have been found significantly different. Therefore, it creates output in step 9.C.

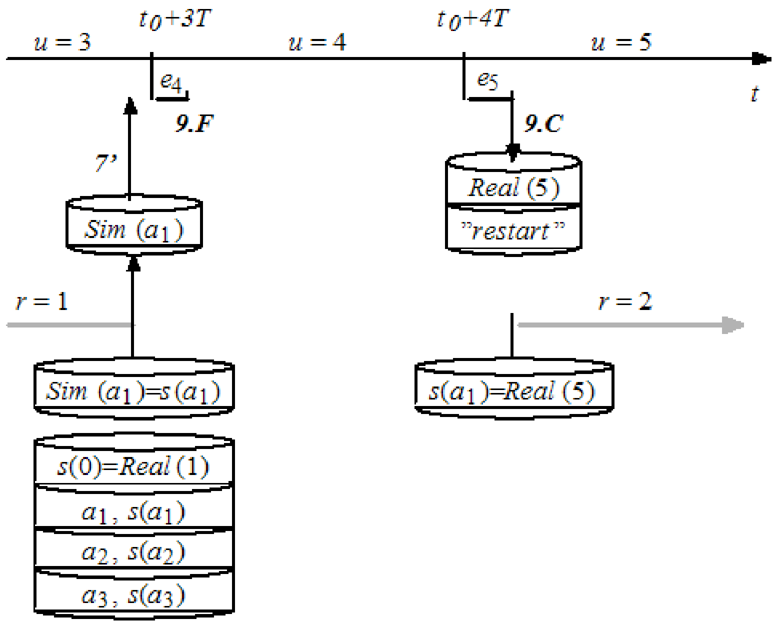


Figure 5.15 Message "*restart*" (step 9.C) has been received while waiting after reasoning cycle $r=1$: the next one ($r=2$) continues with new data.

The Reasoning kernel may continue its next cycle, but all its internal data is no longer useful. Therefore, it resets all the data it had generated in the first cycle, and starts the next cycle with data generated by the RTUpd kernel ($s(a_1)=Real(5)$, $r=2$).

5.2.2.3 RTUpd kernel's output 9.D

The last possible message is the one received in step 9.D. This is shown in Figure 5.16 for the case in which the RTUpd kernel has identified in its fifth update, e_5, that the change in the power system has been as expected, i.e., that states *Real*(5) and *Sim*(a_1) are close enough.

In this case, the Reasoning kernel may keep all the useful internal data it had generated in its first cycle, as follows:

---

[15] Note that the response on action a_1 may arrive later on, for instance in update cycle $u=7$, and if it happens, the RTUpd kernel will conclude that an unforeseen event occurred, and therefore interrupt the reasoning with output 9.A, as in Figure 5.12. Therefore, the action a_1 that has been disabled is not likely to be valid later.

- the current state for the next reasoning cycle, r=2, will be state $s(a_1)$ resulting from action a_1;
- if action a_1 has been found to be a solution, the other actions from the sub-sequence (a_2 and a_3) may be used as the partial solution in the next cycle if the operation of the Reasoning kernel is in search mode (see Section 4.3.2.4).

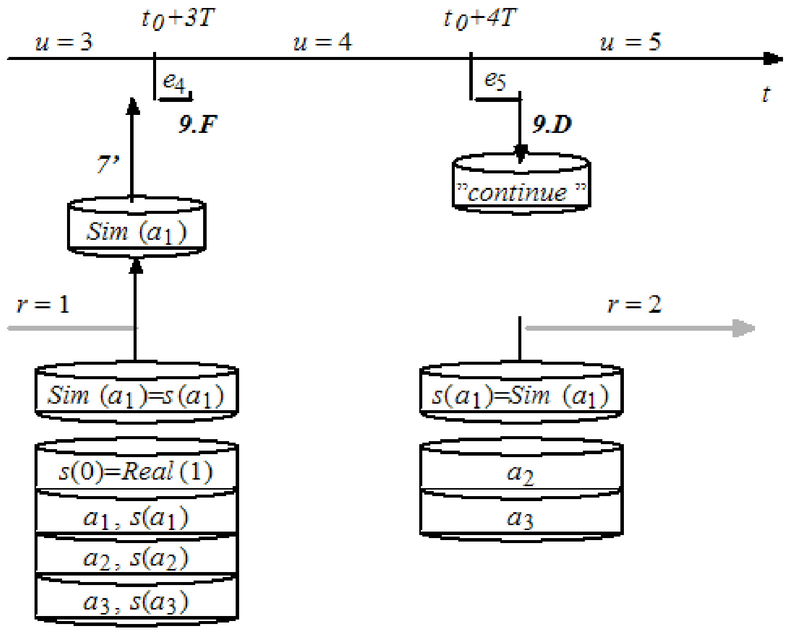


Figure 5.16 The message "*continue*" (step 9.D) has been received while waiting after reasoning cycle r=1: the next one (r=2) continues with data generated in cycle r=1.

5.2.3 Summary

In general case, the Reasoning kernel starts (or restarts) its cycle r after receiving one of the messages 9.A, 9.B, 9.C or 9.D (see Figure 5.11) generated by the RTUpd kernel in update cycle u, i.e., after update interval e_u. While the reasoning is going on and there is no message, reasoning cycle r can endure several update cycles. If the message is received during reasoning cycle r, the cycle must restart with new real-time data. When the reasoning cycle r has terminated, the Reasoning kernel is paused and is waiting for the message in order to know with which data to continue its operation, i.e., with which data to start its next cycle, r+1.

The attitude of the Reasoning kernel in real-time operation, which has been illustrated by examples in Figures 5.13 through 5.16, can be generalised as follows:

> *(9.A) As soon as the Reasoning kernel receives the message "restart"* ***during*** *its cycle* r, *it will immediately restart the same cycle with new data, i.e.,* $s(a_{r-1})$=Real(u).

> *(9.B) As soon as the Reasoning kernel receives the message "disable* a_r *and continue" after its cycle* r, *it will immediately disable action* a_r, *reset all the data it*

contains except the current state s(a_{r-1})*, and continue the next cycle,* r+*1, with the old data, i.e.,* s(a_r)=s(a_{r-1})*.*

(9.C) As soon as the Reasoning kernel receives the message "restart" ***while paused*** *after its cycle* r*, it will immediately reset all the data it contains, and continue the next cycle,* r+*1, with new data, i.e.,* s(a_r)=Real(u)*.*

(9.D) As soon as the Reasoning kernel receives the message "continue" after its cycle r*, it will continue the next cycle,* r+*1, while keeping all the useful internal data it had generated in cycle* r*: the state after action* a_r*,* s(a_r)*, and the sub-sequence without the head action, which is a partial solution for the current state* s(a_r)*.*

5.2.4 Current Implementation

Each of DAFFOR's kernels has its own real-time cycle:

- the RTUpd kernel's cycle *u* is of constant duration *T*, while
- the Reasoning kernel's cycle *r* is of variable duration, which depends on 3 factors: (1) the speed of finding the solution, (2) the kind and the speed of the response of the operators, and (3) the message issued by the RTUpd kernel.

The RTUpd kernel is executed in a synchronous manner, independently of the Reasoning kernel's operation and the possible data input provided by the operators: referring to Figure 5.1 from the introduction of this chapter, these are steps 7' and 8, respectively. It can also be seen that the RTUpd kernel is somewhat in the "background" from the point of view of the operators, i.e., it is situated more deeply inside DAFFOR. On the contrary, the Reasoning kernel is the interactive part of DAFFOR (steps 1 and 2), which must be active all the time. Its real-time operation does not depend only on the operators, but is controlled, in addition, by data generated by the RTUpd kernel (step 9).

Given the nature of the interactions between the kernels (inside DAFFOR) and towards the operators, DAFFOR has been implemented as a programme with two processes which share data, as shown in Figure 5.17.

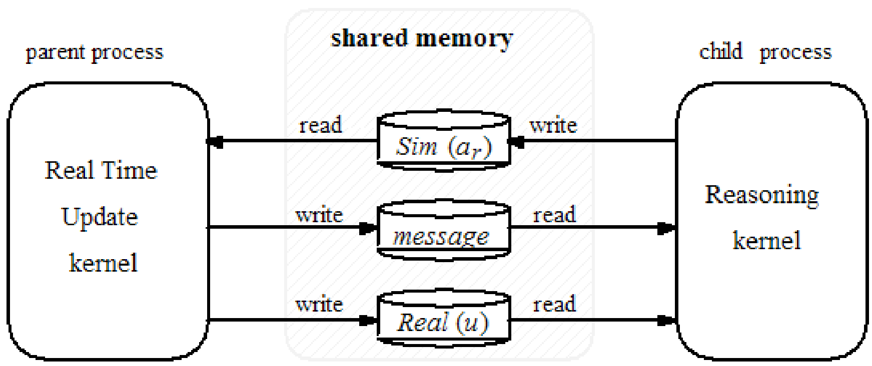


Figure 5.17 The DAFFOR's kernels are two processes that share data.

5.2.4.1 RTUpd kernel

The RTUpd kernel is a parent process which is executed synchronously, at the beginning of update cycle *u*, and which controls its child process (the Reasoning kernel) from the background, by means of the message it generates.

In update cycle *u*, the RTUpd kernel gathers data from the power system and applies topology conversion functions (discussed in Section 4.1.2.1) in order to create the coherent data set *Real*(*u*) which might be necessary for the Reasoning kernel. Therefore, the coherent data set *Real*(*u*) has the same format as data sets in the Reasoning kernel (see Section 4.1.6.1). The same data set is also used internally in the RTUpd kernel for the next update cycle (see Section 5.1.2).

In order to determine whether it is necessary to generate the message, the RTUpd kernel reads data generated by the Reasoning kernel (if it exists) from the shared memory segment. Both the message and the coherent data set are made available to the Reasoning kernel through the shared memory segment, in read mode.

5.2.4.2 Reasoning kernel

The Reasoning kernel is the child process which checks regularly whether there is a message from the RTUpd kernel, in order to keep up with the evolution of the power system in real time. In Section 4.3 a detailed description of the Dialogue&control (D&C) module has been given as the uppermost layer of the Reasoning kernel. It was also said that it handles data for the whole Reasoning kernel, in order to keep it consistent and updated. In addition, it represents the data link between DAFFOR's two kernels. The fact that data handling is centralised in one place (D&C module) reaches its full sense for the real-time operation of the Reasoning kernel.

In Figure 4.26, the stars around the data handling function represent the points in which the shared memory is accessed in order to see whether there is a message from the RTUpd kernel. The length of the reasoning cycle is defined by two factors which are not dependent on the RTUpd kernel's output:

1. the search process in the Knowledge base module may take some time, and in particular if backtracking occurs, and
2. the operators might not respond to a dialogue immediately after it appears.

Both cases must generally be considered as time consuming, even if this is not always true (i.e., even if the solution can be found quickly, and the operators always respond to dialogues quickly). Therefore, the existence of the message from the RTUpd kernel must be checked regularly. The operation which is performed the most often during the search is the assessment of an action, shown in Figure 4.17. The star in that figure represents the point at which the existence of a message from the RTUpd kernel is checked. If the message is encountered, the Knowledge base module returns to the D&C module, which will restart the reasoning. The existence of a message is also checked after each operator request. In this manner, the D&C module enables the Reasoning kernel be updated whenever necessary.

The update is performed in the "initialisation"-box of the D&C module (see Figure 4.33). All the operations on data that have been shown in examples in Figures 5.12 through 5.16 are the responsibility of the D&C module: setting the current power system state for the Reasoning kernel, resetting or keeping old data, restarting the current reasoning cycle or starting a new one, and providing the RTUpd kernel with the reasoning result.

5.2.4.3 Portability

DAFFOR has been developed and is running on an HP workstation, under the UNIX operating system. The RTUpd kernel has been implemented in C. Although most of the code is ANSI-compatible, there are calls to certain system function which depend on the operating system (e.g., time, signal and shared memory-related calls). Nevertheless, these system calls are concentrated in a small number of functions, so that it should not be difficult to modify them if DAFFOR is to be implemented on another hardware platform.

Portability issues concerning the Reasoning kernel and its modules have already been discussed in Sections 4.1.7, 4.2.4 and 4.3.3.

5.2.5 Conclusion

The objective of Sections 5.1 and 5.2 is to show how DAFFOR, with its two kernels, can answer the real-time requirements given in Section 3.1.2. The proposed Real Time Update (RTUpd) kernel enables the Reasoning kernel to keep up with the evolution of the power system in real time. This latter performs its guidance and interactivity functions with respect to both the operators' requests and the current power system state.

The RTUpd kernel is responsible for the creation of the image of the power system comprehensible to the Reasoning kernel. Its results are the coherent data set and the message if an unforeseen event has been identified. The Reasoning kernel, from its side, contributes to the identification of unforeseen events, since it simulates the expected power system state, corresponding to the action which is supposed to be executed by the operators. In fact, it provides the RTUpd kernel with information about the operators' feedback.

5.3 Validation of real-time DAFFOR

DAFFOR has been designed to run in a transmission power system control centre, just as any other programme in the Energy Management System (EMS). However, before the installation in a real EMS environment, it must pass through several development and validation stages. First of all, the stand-alone prototype has been developed and tested off-line (i.e., the Reasoning kernel described in Chapter 4). Then it has been extended with the Real Time Update (RTUpd) kernel which adds the real-time functionality to the prototype, as described in Sections 5.1 and 5.2. At present, the real-time prototype of DAFFOR is ready to be coupled with EDF's power system simulator in order to test DAFFOR's real-time functionality. This corresponds to the second development stage cited in Section 3.1.3.

EDF's power system simulator is itself a stand-alone version of the Operator Training Simulator (OTS) presented in [Jea88]. It plays the role of the power system, with data from a regional control centre. The validation architecture is shown in Figure 5.18.

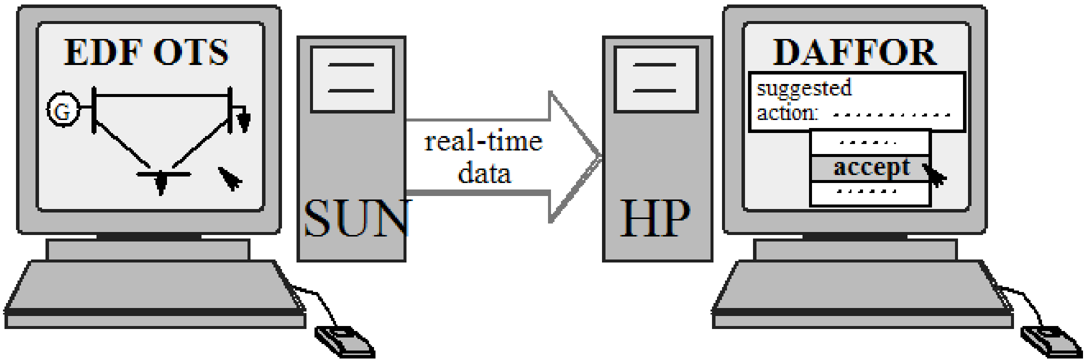


Figure 5.18 Configuration for validation of the real-time functionality of DAFFOR

The data supplied by EDF's OTS corresponds to the data that is available from SCADA, i.e., telemetry including switching status. In addition, the operators can introduce some unforeseen events either in the scenario file or on-line via the user interface of EDF's OTS. In general, the coupling with EDF's OTS will permit the following:

- to conclude what time the DAFFOR's RTUpd kernel needs to gather the real-time information, process it and generate the coherent data set and message, i.e., the length of the "active" update interval e_u in update cycle u;
- to determine the frequency of updates, i.e., the length T of the update cycle, with respect to the longest update interval e_u;
- to improve comparison functions in the RTUpd kernel by defining threshold values for continuous power system variables (frequency, powers, etc.), and
- to test the guidance and interactivity of DAFFOR as a function of the outputs of the RTUpd kernel.

It is important to note that DAFFOR and EDF's OTS are two computer applications which run on different machines. The inter-process communication is through UNIX sockets. The relevant information from EDF's OTS is made accessible to DAFFOR's RTUpd kernel in read mode only, which confirms the principle that only the operators can take action on the power system.

The third and the forth development stages from Section 3.1.3, i.e., the integration in a real environment (OTS and EMS, respectively) are out of the scope of this thesis. However, DAFFOR has been designed in such a manner that the integration does not require many modifications. The changes that are to be made concern on one hand the real-time data access functions of DAFFOR’s RTUpd kernel, and on the other hand, dialogue functions provided by the Dialogue&control module in the Reasoning kernel.

Chapter 6

A genetic algorithm approach for generation of load pickup strategy

Since EDF has no predefined restoration strategy which could apply to the second restoration stage (see Section 2.4.5), one of the objectives of this thesis was to propose one. This task seemed quite difficult at first because of the lack of human experience in this domain, which is why the first direction of research was towards a method that could automatically generate sequences of restoration control actions for the load pickup stage. The analysis of these sequences was expected to give some generic rules that could "compensate" the lack of human expertise.

The determination of an acceptable sequence of control actions to pick up load is a highly combinatorial optimisation problem with multiple constraints. Genetic algorithms have been proven to be quite a successful technique for solving that kind of problem. Therefore, a genetic algorithm (GA) has been used to generate *optimised sequences of restoration control actions* for different initial states of the power system after the skeleton network had been created. In order to follow the evolution of the power system variables (which result from the execution of the sequence of actions) and to avoid violations of operating limits, the GA has been coupled with the Dynamic simulator (DS), described in Section 4.1. The results of the DS are then used in the evaluation function of the GA.

After many runs of the GA, the obtained optimised sequences were analysed and compared. As a result, we have concluded that it is almost impossible to deduce generic rules that could describe the strategy for the second restoration stage (i.e., load pickup stage) from the sequences **as they are defined at present**. In addition, since rules cannot be extracted, further investigation is needed to give any explanation about the order of actions in the sequence. To overcome the encountered problem, Alba proposed in [Alb96] a method to classify the optimised sequences generated by the GA as functions of different power system variables in order to obtain a decision tree. The decision tree then contains rules, not about the actions in the sequence, but about the applicability of the whole sequence to the power system in a given initial state. The first results of this joint work have been published in [Kos96].

The principle of the genetic algorithms technique used for optimisation and its application to some power system problems are briefly discussed in Sections 6.1 and 6.2, respectively.

Section 6.3 first presents the GA-based approach proposed in this work for obtaining the optimised load pickup sequences. The parameters, the operators and the results of this method applied to a test network are discussed, and the present limitations are highlighted. After that, the combined approach proposed by Alba [Alb96] and the potential integration within DAFFOR will be addressed, before giving directions for the further research in this domain.

6.1 Genetic algorithm: Principle

Genetic algorithm (GA) was proposed by Holland [Hol75] in the early 70's as an attempt to model adaptive systems. It belongs to a wide class of methods, called evolutionary programming [Fog95], which are all inspired by natural evolution. GAs combine the random search and hill-climbing methods with the idea of competition. This technique can be used for different purposes, including optimisation.

As their name indicates, GAs are based on natural genetic mechanisms. Before proceeding to the introduction of GA principles, the correspondence between the terms in the real and the artificial world is shortly described below, and summarised in Table 6.1:

- One or more chromosomes combine in order to form the prescription for the forming and functioning of an organism. In this work, a single chromosome is considered and in GAs terminology it will be called an *individual*.
- The chromosome contains a number of elementary genetic structures - genes. Each of them has a certain function (e.g., a gene responsible for the colour of eyes). A gene is usually called a *character* in GAs.
- The gene holds encoded genetic information about the chromosome in its allele and locus (e.g., for the eye colour gene, allele may be "brown"). In GAs, the character is defined by its *value* and *position* inside the individual.
- Each organism in nature has certain degree of ability, or fitness, to adapt to the environment, to survive and to reproduce. The fitness represents, in fact, the decoded

genetic information contained in chromosomes. In GAs too, the decoded genetic information permits the degree of quality of an individual, or its *fitness* to be determined.

Table 6.1 Terminology in natural genetics and artificial genetics (GAs)

Natural Genetics / Biology	Genetic Algorithms
chromosome	individual
gene	character
allele	value
locus	position
fitness	fitness

GAs manipulate a population of individuals, denoted as parents, in order to produce a new population, denoted as children, through the steps stated in Figure 6.1. By applying different genetic operators to its parents, the child population is produced, whose individuals are expected to have higher fitness then their parents. In the next generation, the children become parents, and a new child population is created. The reproduction process continues during a given number of generations or until some other criterion is satisfied. This artificial reproduction of individuals simulates what happens with the organisms in the real world reproduction.

1. Initialize a population of individuals.
2. Evaluate each individual (i.e., determine its fitness).
3. Mate couples of individuals (parents) by applying genetic operators to them in order to obtain new individuals (children).
4. Evaluate children and replace some or all parents by inserting some or all children in the parent population.
5. If stop criterion is not reached, go to step 3.

Figure 6.1 Basic genetic algorithm operation

The idea of using GAs to solve difficult optimisation combinatorial problems resides in the fact that GAs perform search through the space on which the problem is defined in several directions. The population of individuals is randomly initialised with a set of feasible solutions, an individual being an encoded solution of the problem. The information contained in the individual must be decoded to evaluate its fitness, which is done by the evaluation (or fitness) function. It captures the objective function and the constraints of the problem; in fact, it plays the role of the environment. The search for the optimal solution starts from several points (i.e., initial population of individuals or solutions). Thanks to selective adaptation of individuals to the environment through a number of generations, the search is guided by the fittest among them. Although GAs explore a problem's space of solutions by random search, the exploration is done not across the whole solution space, but only through the regions that are chosen considering the value of the objective function. This makes the GAs capable of finding the global optimal solution, preventing them from "irreversibly climbing a wrong hill".

In the following sections, the basic principles of GA operation are briefly introduced. For details on any of these items, refer to the excellent books [Gol89] and [Dav91]. For the reader interested in genetics itself, we suggest one of the "scientific classics", [Gro91].

6.1.1 Reproduction strategy

Depending on the problem that is to be solved, there are different reproduction strategies:

- a couple of parents may produce one or more children (usually two);
- all the present parents may be deleted in order for all children to take their place, i.e., to become parents in the next generation – this is called *generational replacement*;
- a number of the best parents may be kept in the next generation – this is called *elitism*;
- only a small number of children may be created (one or two) and introduced into the parent population by replacing the same number of the worst parents – this is called *steady-state reproduction*, etc.

6.1.2 Encoding, decoding and evaluation

An individual in the GA method must be encoded in some manner in order to apply the genetic operators to it. Depending on the problem that is to be solved, it is necessary to select "...the smallest alphabet that permits a natural expression of the problem" [Gol89]. The individual must then be decoded and evaluated, in order to find its fitness. The following is a brief description of a few frequently used encoding techniques; the list is far from being exhaustive.

The most common encoding technique is *bit string* encoding, where the individual is represented by a string of "zeros" and "ones". The elementary mathematical basis of GAs has been established with this type of encoding. The individual represented with the binary alphabet can then be decoded in different manners. For instance, if the maximum of a function is to be found, the "binary" individual may simply be decoded in "decimal" one. In this case, the evaluation is performed directly, by decoding.

Other encoding techniques are *integer* and *real number* encoding. The first is usually used when the individual represents a set of discrete modifications that can affect variables (e.g., transformer tap change steps). The real number encoding may be used, for instance, when the individual represents a set of thresholds associated to variables (e.g., minimum and maximum admitted voltage). The individuals may then be evaluated thanks to the results of a power flow.

When a scheduling problem is to be solved, *permuted list* encoding is used. Each individual in the population contains as characters all the members of the list, but the order of characters differs from one individual to another. There are usually costs associated to characters as a function of their position inside the individual, so that decoding consists of computing the sum of costs. A typical problem whose solutions are encoded as permuted lists is the Travelling Salesman Problem.

6.1.3 Genetic operators

The basic GA operators are selection, crossover and mutation (the inversion operator will not be addressed here). The terms in italics refer to definitions in [Gol89].

The *selection* operator chooses the couple of parents that will produce one or more children. The better the fitness of a parent, the higher is its chance to be selected for reproduction. The most commonly used selection operator is the *biased roulette wheel*.

The *crossover* operator plays the role of the recombination mechanism. It mixes the genetic material of two parents in order to produce one or more children. The choice of crossover operator depends on both the problem that is to be solved and the encoding technique. The basic one is the *one-point* crossover, but there is also the *two-point* crossover, the *multiple-points* crossover, the *order-based* crossover, etc.

The *mutation* operator acts on a single individual, introducing a random modification in its genetic structure. If the individual is a parent, the mutation produces a child, leaving the parent unmodified; otherwise, the child is modified independently of the parents' genetic material. The function of the mutation operator is to maintain diversity in a population. The choice of the mutation operator depends mostly on the encoding technique. The basic one is a *single character* mutation, but there is also the *position-based* mutation, the *order-based* mutation, the *scramble* mutation, etc.

Both crossover and mutation operators are associated with a probability of application. The probability may be constant (i.e., given as a parameter), or variable (e.g., defined as a function of the number of generations). The mutation probability is usually much lower than the crossover one.

6.1.4 Conclusion

A GA is a powerful optimisation technique. Its main advantage compared to conventional optimisation methods is its capability of multiple directional searching in a problem's solution space. That is why a GA usually manages to find the global optimal solution. However, if one wants a GA to solve the given problem as expected and efficiently, there are many hills to walk up and down before reaching the right one.

In fact, the most important choices to be made are about encoding, decoding and evaluation. The solutions should be encoded so that: (a) they naturally express the problem, (b) the genetic operators applied to the coded individuals result as much as possible in new solutions that are feasible, and (c) solutions may easily be decoded and evaluated. The evaluation function should contain all the relevant information about the problem and its constraints, and should be "smart" enough to guide the search through good directions (by giving high fitness to good solutions).

The choice of reproduction strategy and genetic operators, as well as the settings for numerous related parameters, represent another challenge. Usually, several combinations of them must be tested before reaching the one that is good and efficient enough for a given problem.

Finally, for certain very difficult problems, the main disadvantage of GAs is the computation time they require to give a solution. Consequently, it is almost impossible to think of a real world application that is based on GA optimisation in real time. To overcome this problem (at least until more powerful computers become available), it is recommended to hybridise GA-based with conventional optimisation methods whenever it is possible. The GA search may be run first in order to generate a number of good solutions, which are used afterwards as initial points in the conventional optimisation method.

6.2 Application to power system problems

One of the most recent survey papers on the application of evolutionary techniques to power systems [Mir96] gives the list of 140 references. It may be seen that the interest in solving power system problems with GAs suddenly increased in the early 90's. The majority of the treated problems are in the domain of structural planning (generation-transmission-distribution system expansion, VAR planning, capacitor placement), operation planning (unit commitment and generator scheduling, maintenance scheduling, load dispatch, reactive power dispatch and voltage control) and power flow analysis.

As far as the service restoration problem is concerned (and in particular the load pickup stage), to our knowledge there is very little work. In references [Fuk96] and [Oya96], hybridised GAs for the restoration of **distribution** (radial) power systems have been proposed, and in [Kov96] a modified simple GA has been used to generate switching sequences in a large transmission substation.

6.3 Optimisation of load pickup sequences

As was mentioned in the introduction of this chapter, a genetic algorithm (GA) technique has been used in this thesis to generate optimised sequences of control actions for the second restoration stage – load pickup stage. The initial power system state for which an optimised load pickup (LP) sequence is generated by the GA is the state after the skeleton network had been created. The obtained LP sequence may be seen as a possible load pickup strategy for the given initial power system state. The power system state variables that may significantly influence the LP sequence are:

- the availability of production units and circuits,
- the availability of the external network support, and

- the loads that were supplied before the blackout; they are assumed to be the loads to restore thanks to control actions contained in the LP sequence.

Note that all these variables may be either defined in a blackout scenario file (see Section 4.4.1) or initialised randomly, which enables the creation of different initial states in order to obtain diversified LP sequences (e.g., very high pre-blackout load level combined with the unavailability of a production unit, very low pre-blackout load level combined with availability of all production units, etc.).

In the following sections, the formulation of the optimisation problem will be given first. Then the proposed GA-based solution method will be described in details by addressing all GA-related items (introduced in Section 6.1) as they have been implemented in this work. After the description of GA parameters, results of GA-based optimisation will be presented and discussed.

6.3.1 Problem formulation

The problem we are trying to solve is the restoration of a transmission power system after the skeleton network has been created. The solution of this problem is a load pickup (LP) sequence of control actions which, when applied to the power system in a given state, provide the supply of the pre-blackout load as soon as possible, and without violation of power system operating limits.

The restoration sequence defined by (4.2) may be divided into two parts as follows:

$$Seq(1,N_{all\_act}) = Seq(1,N_{skel}) + Seq(N_{skel}+1,N_{all\_act}-N_{skel})$$

$$Seq(1,N_{all\_act}) = [a_1 \ldots a_{Nskel} | a_{Nskel+1} \ldots a_{Nall\_act}]. \qquad (6.1)$$

where the first N_{skel} actions lead to the skeleton network creation, and the remaining actions represent the sequence that lead to total service restoration:

$$Seq(N_{skel}+1,N_{all\_act}-N_{skel}) = [a_{Nskel+1} \ldots a_{Nall\_act}]. \qquad (6.2)$$

In other words, (6.2) is the LP sequence. The states transition expression (4.1) that corresponds to the LP sequence (6.2) then may be written as:

$$a_r=f(s(a_{r-1})) \rightarrow s(a_r),\ r=(N_{skel}+1,N_{all\_act}-N_{skel}). \qquad (6.3)$$

The optimisation problem now may be defined:

> *Find a load pickup sequence (6.2) that, when applied to the power system in state* $s(a_{Nskel})$ *(6.3), leads to state* $s(a_{Nall\_act})$ *(6.3) in which the maximum of the pre-blackout load will be restored in minimum time, while respecting operating constraints (4.14-4.16).*

6.3.2 GA-based solution

The following sections describe how GA-based optimisation has been used to generate optimised LP sequences.

6.3.2.1 Encoding and decoding

Each individual in the population is an encoded representation of a possible solution, i.e., an LP sequence of restoration control actions (6.2). Since the Dynamic simulator (DS) had already been available when this research started, it seemed quite evident that we should encode solutions as sequences of DS commands, described in Section 4.1.3. Naturally, the DS is used as a decoder.

The DS commands that are used for encoding are those that correspond with control actions that are likely to be applied to the power system in the load pickup stage, as follows:

- `$ bra on <branch>` - to connect `branch`,
- `$ gen sby <unit>` - to supply auxiliaries of thermal `unit`,
- `$ gen on <unit>` - to couple `unit` to the network,
- `$ gen avol <unit>` - to increase voltage set-point of `unit`.
- `$ gen aset <unit>` - to increase power set-point of `unit`, and
- `$ loa aset <load> <n>` - to pick up `n` levels of consumer `load`,

The set of above DS commands for all transmission devices, loads and units in a network form the **list of LP actions**. Each element in the list of LP actions has an associated probability of application. In other words, the action that is likely to appear more frequently in the LP sequence will also appear more frequently in the list of LP actions. To illustrate this, refer to Figure 6.2 which shows a part of the list of LP actions for the New England test network, given in Appendix A. The number in parentheses indicates how many times the action appears in the list:

- For each `branch`, there are two actions `$ bra on <branch>`;
- For each `thermal unit`, there are two actions `$ gen sby <unit>` and five actions `$ gen aset <unit>`;
- For each `hydraulic unit`, there are two actions `$ gen aset <unit>` (this is not shown in Figure 6.2 because the test network has no hydraulic units);
- For each `hydraulic` and `thermal unit`, there are two actions `$ gen on <unit>` and one action `$ gen avol <unit>`;
- For each `consumer load`, there is at least one action `$ loa aset <load> 5`. The total number of actions depends on the initial load. The consumer load ln32 is of 9MW, so it is not worth considering lower undeserved levels (`n`<5). On the contrary, the consumer load ln27 of 280MW is likely to be picked up in several smaller increments.

```
bra on n11n12 (2)     gen on   gen2  (2)     loa aset ln27 1 (3)
.............         gen sby  gen2  (2)     loa aset ln27 2 (3)
bra on n30n39 (2)     gen aset gen2  (5)     loa aset ln27 3 (2)
                      gen avol gen2  (1)     loa aset ln27 4 (1)
                      .............          loa aset ln27 5 (1)
                      gen on   gen10 (2)     ...............
                      gen sby  gen10 (2)     loa aset ln32 5 (2)
                      gen aset gen10 (5)
                      gen avol gen10 (1)      ( ln27: 280 MW )
                                              ( ln32:   9 MW )
```

Figure 6.2 Sample list of LP actions for the New England test network

The list of LP actions is used to randomly create the individuals in the initial population (see Section 6.3.2.5) and for the mutation operator (see Section 6.3.2.3). By associating the probability of application to each action in the list of LP actions, random initialisation will have higher probability of generating feasible solutions.

An individual (LP sequence) contains a number of DS commands. Therefore, the **DS decodes the individual** by simulating the execution of actions (encoded by DS commands) in order to obtain the evolution of the power system when the LP sequence is executed. It also traces violations of operating limits.

6.3.2.2 Evaluation (fitness) function

Both number of violations and evolution of the power system variables are the results of the DS obtained during the simulation of a complete LP sequence. They are used by the evaluation function to estimate the "quality" of a sequence, i.e., to compute the fitness for each individual.

Here is a list of the DS results concerning operating limit violations (all of them are issued per electrical zone z; see the definition of electrical zone in Section 4.1.2.2):

- $unstab(z)$ - this flag is true if the unstable state has been encountered; the unstable state has been defined in section 4.1.6.2;
- $diverg(z)$ - this flag is true if a power flow did not converge;
- $N_{ldf}(z)$ - total number of power flows;
- $N_{ovl}(z)$ - number of overloaded lines and transformers; loading limits have been given by (4.16);
- $N_{vol}(z)$ - number of nodes with voltage limit violations; voltage limits have been given by (4.15);

  (note that both $N_{ovl}(z)$ and $N_{vol}(z)$ can be greater than $N_{ldf}(z)$, since several overloads and voltage violations can be found in a single loadflow)
- $I_{fre}(z)$ - total integral of absolute frequency deviation with respect to the nominal frequency f_{nom};
- $I_{fre+}(z)$ - integral of those parts of the frequency that are above its upper steady state limit F_{SSmax}, given by (4.14); and

- $I_{fre-}(z)$ integral of those parts of the frequency that are below its lower steady state limit F_{SSmin}, given by (4.14).

The global DS results (for the whole power system) are the following:

- F_{zon} - number of electrical zones after the sequence is executed;
- *load* - percentage of the pre-blackout load that has been picked up at the end of the sequence.

The objective of restoration is not only to supply the maximum of the pre-blackout load without operating limit violations, but also to have the entire power system reconnected (without islands, i.e., a single electrical zone). However, during the load pickup stage, there might be more than one electrical zone. Some of them might have contained operating limit violations, which disappear when the zones become connected. The task of the evaluation function is to assign a high fitness to the LP sequence which provides satisfactory transition from one power system state to another, i.e., the whole evolution and not only the final state. Since the maximal number of electrical zones corresponds to the number of production units, denoted as N_{gen}, the violation-related items apply to all electrical zones that might have existed, i.e., z=1,…, N_{gen}.

First of all, certain tests are performed in order to reject an LP sequence (i.e., assign it zero-fitness) if it causes significant operating limits violations:

$$fitness=0 \text{ if } unstab(z)=\text{true OR } diverg(z) \text{ OR } N_{ovl}(z)\geq ldf_l(z) \text{ OR } N_{vol}(z) \geq ldf_l(z). \quad (6.4)$$

If conditions (6.4) are found to be false for each z=1,…, N_{gen}, coefficients K_{fre}, K_{ovl} and K_{vol} relative to violations of frequency, power flow and voltage limits, respectively, are computed thus:

$$K_{fre} = (1/\mathrm{N}_{Ngen}) \times \sum(1 - (I_{fre+}(z)+ I_{fre-}(z)) / I_{fre}(z))$$

$$K_{ovl} = (1/\mathrm{N}_{Ngen}) \times \sum(1 - N_{ovl}(z) / N_{ldf}(z))$$

$$K_{vol} = (1/\mathrm{N}_{Ngen}) \times \sum(1 - N_{vol}(z) / N_{ldf}(z)). \quad (6.5)$$

Finally, the fitness of the LP sequence is a function of the percentage of the pre-blackout load that has been picked up (*load*), the number of electrical zones in the system restored by the LP sequence (N_{zon}) and the violations of operating limits expressed through coefficients (6.5):

$$fitness = 0.1 \times (load/ N_{zon})\times(K_{fre})^{efre}\times(K_{ovl})^{eovl}\times(K_{vol})^{evol}, \ (e_{fre},e_{ovl},e_{vol}\geq 1) \quad (6.6)$$

where exponents e_{fre}, e_{ovl} and e_{vol} penalise violations of frequency, power flow and voltage limits, respectively.

Coefficient K_{fre} is obtained from dynamic simulation (the long-term dynamics model), and coefficients K_{ovl} and K_{vol} are functions of operating limits violations obtained from static computation (power flow). All of them can vary from 0 to 1, and *load* from 0 to 100%. The factor 0.1 in fitness equation (6.6) has been introduced for the sake of readability during simulations. Since penalty exponents e_{fre}, e_{ovl} and e_{vol} are greater than or equal to 1, the fitness (6.6) can vary from 0 to 10.

It can be seen that time does not appear in fitness equation (6.6). In fact, it has been noticed that for similar power system states, all the optimised LP sequences have quite similar duration.

An optimised LP sequence which has non-zero fitness is said to be a feasible solution. The closer the fitness is to 0, the more operating limit violations the LP sequence causes. On the contrary, the LP sequence that has the maximal fitness (10) results in total pre-blackout load restoration with no voltage, load or frequency violations.

6.3.2.3 Genetic operators

The *selection* operator used here is the standard biased roulette wheel operator.

The *crossover* operator is a two-point crossover with variable length for the individual. As shown in Figure 6.3, two points are chosen for each parent, and elements between the two points (e_1 for p_1 and e_2 for p_2) are exchanged to obtain two children, c_1 and c_2. This kind of crossover keeps a certain level of redundancy in the elements of the individuals (LP actions), needed to increase the likelihood of obtaining a successful sequence. However, the maximal length of the individual is defined and limited by the decoder (DS).

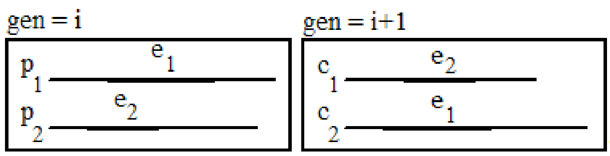


Figure 6.3 Two-point crossover with variable length for the individual

Finally, the *mutation* operator replaces a single element of a child with a randomly chosen element from the list of LP actions, introduced in Section 6.3.2.1. The actions that appear more frequently in the list of LP actions have a higher probability of being chosen.

6.3.2.4 Initial power system state

Since the problem focuses on the load pickup restoration stage, it is assumed that the skeleton network had been constructed by coupling the pre-defined production units, connecting them with single circuits, and supplying auxiliaries of the shutdown thermal units belonging to the skeleton network. It is assumed that no consumer load has been picked up.

Initial voltage set-points for all coupled units in the skeleton network are at their minimum values (i.e., 0.95pu as defined in Section 4.1.3.2), and their initial power set-points are as follows:

- 0MW for all hydraulic units,
- for all equivalent units, power set-points match the equivalent load at their nodes, and

- for all thermal units, power set-points match the load demand of their auxiliaries.

The above enumerated power system variables always have the same initial values. On the contrary, the variables which significantly influence the LP sequence (enumerated in Section 6.3) may have different initial values, as follows:

- some production units and circuits may be set as unavailable,
- the external network support may be set as available or not, and
- the level of the pre-blackout load may be set to a certain percentage which permits a lightly or heavily loaded system (e.g., 80% and 120% of nominal load, respectively).

6.3.2.5 Initial population

The initial population is composed of individuals (LP sequences), each of which is randomly filled with actions from the list of LP actions described in Section 6.3.2.1. An individual is evaluated to determine whether its fitness (6.6) is positive for the initial power system state. To speed up the later GA search, only individuals with positive fitness values are accepted in a population. This constraint can make the creation of the initial population from the list of LP actions a time consuming task, despite the probabilities of application associated to each action. This problem can be solved by randomly selecting the initial population from a previously created set of individuals with positive fitness; this set must contain acceptable sequences for any initial power system state.

6.3.2.6 Reproduction strategy

A pair of parents produces a pair of individuals by using the genetic operators. Only the individuals with positive fitness (6.6) are kept as children. The reproduction strategy applied is *generational replacement with elitism*. This means that a certain number of best ranked parents are copied in the next parent-generation, and the remaining parents are replaced with the children.

* * *

Now the flow chart of the GA optimisation described above can be represented as shown in Figure 6.4. *PopSize* and *EliteSize* stand for GA parameters which determine the number of individuals in the population and the number of elite members, respectively, and P_{cross} and P_{mut} are the probabilities of applying the crossover and the mutation operator, respectively.

First the initial generation of *PopSize* individuals, with positive fitness (6.6), must be created; these individuals become parents. Then *EliteSize* best parents are copied into the child generation, and the process of reproduction may start. For each generation, the objective is to create (*PopSize–EliteSize*) children with positive fitness (6.6) by applying genetic operators on the parents. A pair of parents is selected, and double-point crossover is performed on them with probability P_{cross}. Then the mutation operator is applied to both resulting individuals with probability P_{mut}, after which the individuals are evaluated. Any individual with positive fitness (6.6) becomes a child. When the population of children is completed, they become parents ready to create a new child generation. This artificial reproduction continues until a stop criterion is satisfied.

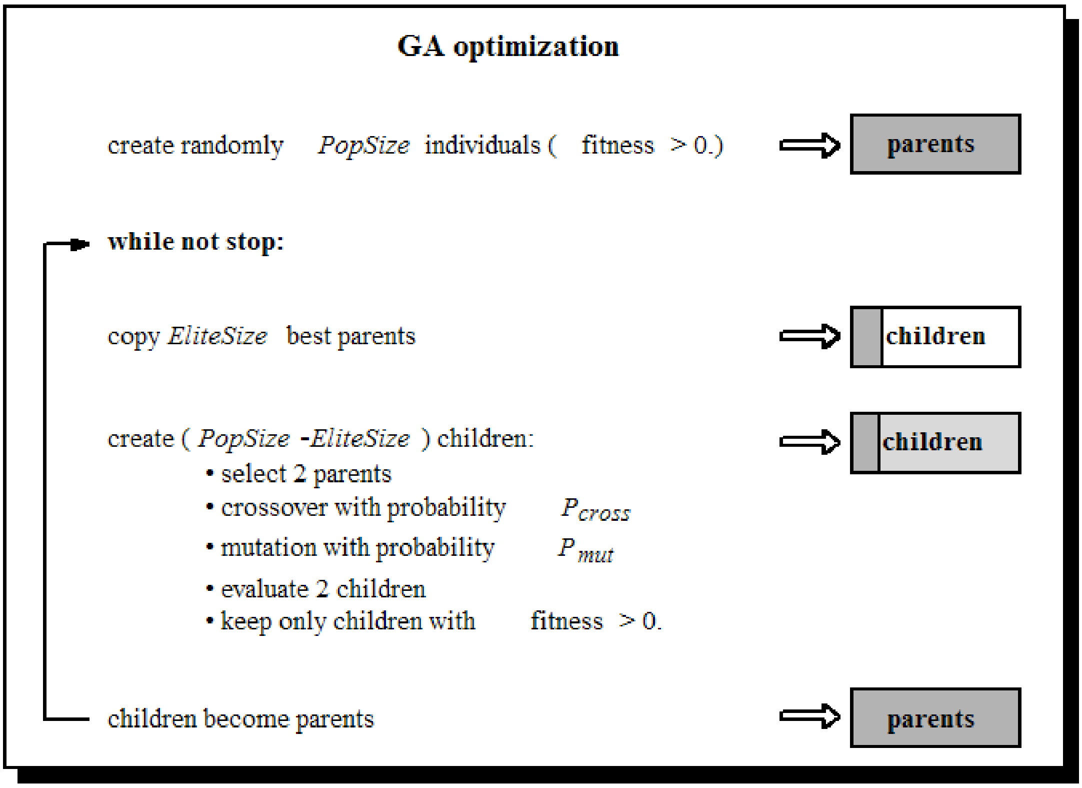


Figure 6.4 Flow chart of GA optimisation for generating optimised LP sequences

6.3.3 GA parameters

When the first results of this approach were reported in [Kos96], the initial power system state was the skeleton network created with **all** the units and **all** the branches in the network. New simulations were carried out with a more realistic initial state, i.e., the skeleton network comprising only **certain** units, connected with the **minimal number** of branches. Since there are more actions to execute in this latter case, most of the GA parameters had to be adjusted as follows:

- The maximal number of generations is *MaxGen*=100. Depending on the initial state (lightly or heavily loaded system, number of unavailable units and branches, presence of an external network), the convergence is usually between 20 and 50 generations, except for very "difficult" states.
- The number of individuals in a generation is *PopSize*=100. The size of the population had to be increased (from 30 to 100) because of the convergence problems. With only 30 individuals, the convergence proved to be either too slow (i.e., much more generations would be necessary) or too fast (i.e., convergence around a local optimum).
- The number of elite members is *EliteSize*=10 (i.e., 10% of parents survive).
- The maximal length of an individual is *SeqLen*=500. It is 2-3 times greater than the expected length of an acceptable sequence. Redundancy in the actions that compose an

individual is needed to reach 100% of restored load. The advantage of the redundancy in coding is the fact that an action which is useless in one sequence may be useful in another one, obtained through the crossover. In all cases, the DS simply ignores an invalid action.

- The probability of crossover is P_{cross}=1. This means that during the reproduction, each couple of selected parents will be double-crossed and thus produce a couple of new individuals.
- The mutation probability P_{mut} is a function of the current generation *gen*:

$$P_{mut}=0.4\times(1-e^{-0.1\times gen}). \qquad (6.7)$$

  In the previous tests, the mutation probability decreased with the generation number. With the adjusted GA parameters, decreasing mutation probability caused the small diversity in the later generations. Mutation probability (6.7) which increases with the number of generations may help the convergence for very "difficult" initial states, since it reaches the near-maximum value (0,4) around the 50th generation.
- There are 3 stopping criteria:

  (1) if an individual with very high fitness (*fitness*≥9) is obtained, and all the coefficients (6.5) related to violations of the operating limits equal 1, OR

  (2) if an individual with the maximal fitness (*fitness*=10) is obtained, OR

  (3) if the maximal number of generations *MaxGen* is reached.
- In our experiments, penalty exponents from the fitness function (6.6) have the following values: $e_{ovl}=e_{vol}=1.5$ and $e_{fre}=3$. The exponent e_{fre} which penalises violations of frequency K_{fre} (6.5) has been given a somewhat higher value because it is more reliable from the point of view of the dynamic evolution of the power system obtained by the DS.

6.3.4 Examples and results

This section presents the results of the proposed approach through four examples. The first two GA optimisations have as an objective to show how the optimised sequences may differ for the same initial state due to different seeds used in GA runs. The second pair of GA optimisation results is obtained for different initial states in order to show how the GA convergence depends on the initial state.

Each simulation has been carried out with the parameters defined in Section 6.3.3, in order to obtain an optimised LP sequence for the New England test network, shown and commented on in Appendix A. The skeleton network comprises 3 coupled thermal units (`G2`, `G5` and `G8`), connected by energised lines. For the initial state in which the external network is available, the equivalent unit is also coupled to the skeleton network (`G1`). In all the GA runs discussed below, it has been supposed that the external network is available. The only consumption in the skeleton network is that of the units' auxiliaries. The voltage and power set-points of the coupled units are set as described in Section 6.3.2.4.

6.3.4.1 The same initial state

Two GA optimisations, denoted as O1 and O2, have been run with different seeds. The initial power system state is the nominal one (with the external network support and all production units and transmission equipment available). The results of optimisations O1 and O2 are presented in three figures, each of them grouping information of the same type in order to facilitate the comparison.

6.3.4.1.1 GA convergence

Figure 6.5 shows the evolution of the maximum fitness value through generations in GA optimisations O1 and O2, respectively. For the nominal power system state and the adopted set of parameters, the convergence of the GA optimisations is reached between 30 and 40 generations. The maximum fitness in the initial population (generation 0) usually varies from 0.5 to 1.5, because the population is generated randomly. However, it may happen that there is an individual with relatively high fitness compared to other individuals in the initial population, which usually leads to premature convergence, as shown in Figure 6.5 for the optimisation O2 (the best individual in the initial population had a fitness of 2.5). The more representative case is that of the optimisation O1.

The LP sequences with a fitness over 5 are obtained quickly, i.e., not later than after 5 generations. The reason for this is the fitness function (6.6). If the network restored with the LP sequence has two electrical zones, its fitness cannot be greater than 5 because the fitness is divided by the number of zones. Closing a single branch, which connects two zones, may be sufficient to significantly increase the fitness.

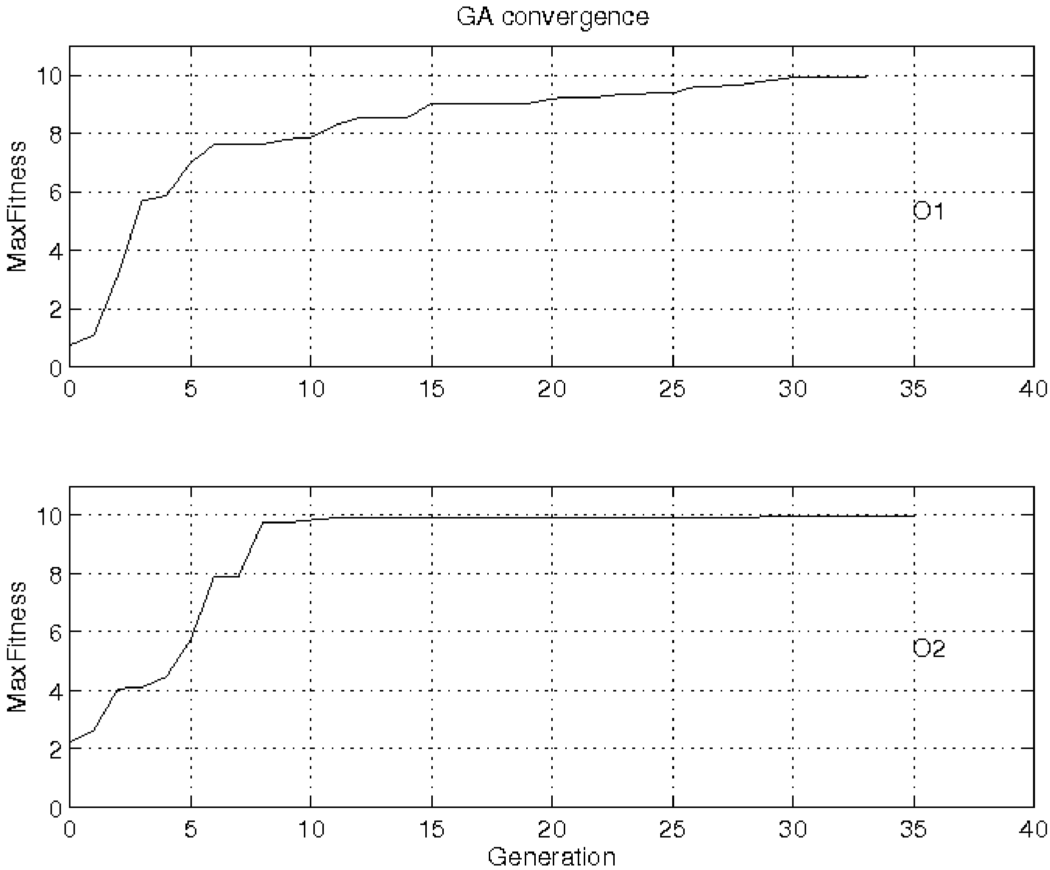


Figure 6.5 GA convergence for optimisations O1 and O2

The convergence to the maximum fitness (10) might be very slow. In fact, most limit violations caused by the LP sequence can be eliminated quickly, but there are some of them that persist from one generation to the other despite the recombination and mutation during reproduction (this will be discussed along with the simulation results in Section 6.3.4.1.3). Although very small, they cause the fitness to differ from 10. In order to avoid the reproduction until the last generation in such a case (which does not even guarantee that the maximum fitness will be reached), there are two additional stopping criteria which are tested after a generation has been created:

1) If there is an LP sequence which results in a single electrical zone, and which supplies at least 90% of consumer load without violations of the operating limits, the optimisation is to be stopped. Both optimisations O1 and O2 have been stopped due to this criterion: the resulting LP sequences are not those with the maximum fitness, but those with the minimum limit violations; and
2) If there is an LP sequence with the fitness "close" to 10 with a small threshold (e.g., 0.003), the optimisation is to be stopped.

6.3.4.1.2 Optimised LP sequences

Figure 6.6 shows the parts of the LP sequences obtained from optimisations O1 and O2 in generations 33 and 35, respectively, with the stopping criteria on the minimum violation of limits. Comparing the LP sequences, it can be seen that there is no evident similarity. For instance, among the first 10 actions in sequence O1, five transmission lines have been connected, and in O2 only 1; on the contrary, in sequence O2 more load has been picked up. The first two branches connected in the sequence O1 (actions 3 and 5) will also be connected in the sequence O2, but much later (actions 27 and 34, respectively). A similar remark applies to action 10 from sequence O2 which is on position 19 in sequence O1.

However, what is common to both LP sequences is the similar number of valid actions: 154 and 162 for sequences O1 and O2, respectively.

```
            O1                                        O2

  1   loa aset ln35_1 4                 1   gen aset gen2
  2   gen aset gen2                     2   loa aset ln18_1 2
  3   bra on n17n18                     3   gen aset gen8
  4   loa aset ln35_2 2                 4   loa aset ln2     5
  5   bra on n36n37                     5   gen aset gen8
  6   bra on n37n27                     6   gen aset gen8
  7   loa aset ln14_2 1                 7   loa aset ln13_2 3
  8   loa aset ln25   1                 8   loa aset ln35_2 2
  9   bra on n16n31                     9   gen aset gen5
 10   bra on n16n17                    10   bra on n1n1
 .....................                 .....................
 19   bra on n1n12                     27   bra on n17n18
 .....................                 .....................
154   gen on   gen9                    34   bra on n36n37
      end                              .....................
                                      162   gen avol gen5
                                            end
```

Figure 6.6 Parts of the optimised sequences obtained in GA runs O1 and O2. The number on the left of an action indicates its position inside the sequences.

6.3.4.1.3 Power system evolution

Figure 6.7 shows the evolution of the frequency and the rate of the load pickup when the optimised LP sequences O1 and O2 are applied to the power system defined by the initial state described in Section 6.3.4, i.e., after the skeleton network has been created. For this reason, the curves of the frequency and the supplied load start after about one hour. The fitness of both sequences O1 and O2 is around 9,6. Since they have been chosen by the first stopping criteria (stated in Section 6.3.4.1.1), they supply about 96% of the total pre-blackout load without violations of operating limits.

However, Figure 6.7 shows that in both cases small frequency limit violations, given by (4.14), persisted. For sequences O1 and O2, there are, respectively, three and two peaks under 59.76Hz. This indicates that the frequency-related coefficient K_{fre} from (6.5), which is used in the fitness function (6.6), should be defined in such a manner that the peaks be penalised more (e.g., instead of computing the ratio of frequency integrals in (6.5), the derivatives should be introduced).

The evolution of the frequency for the two LP sequences is quite different, which can be expected after a comparison of the sequences themselves: since the actions on power set-points of the production units and the load pickups are executed in two completely different manners, the aspect of the frequency curves differs from one sequence to the other. On the contrary, the rate of the load pickup is quite similar, as well as the time necessary to restore the same amount of load (about 11 hours).

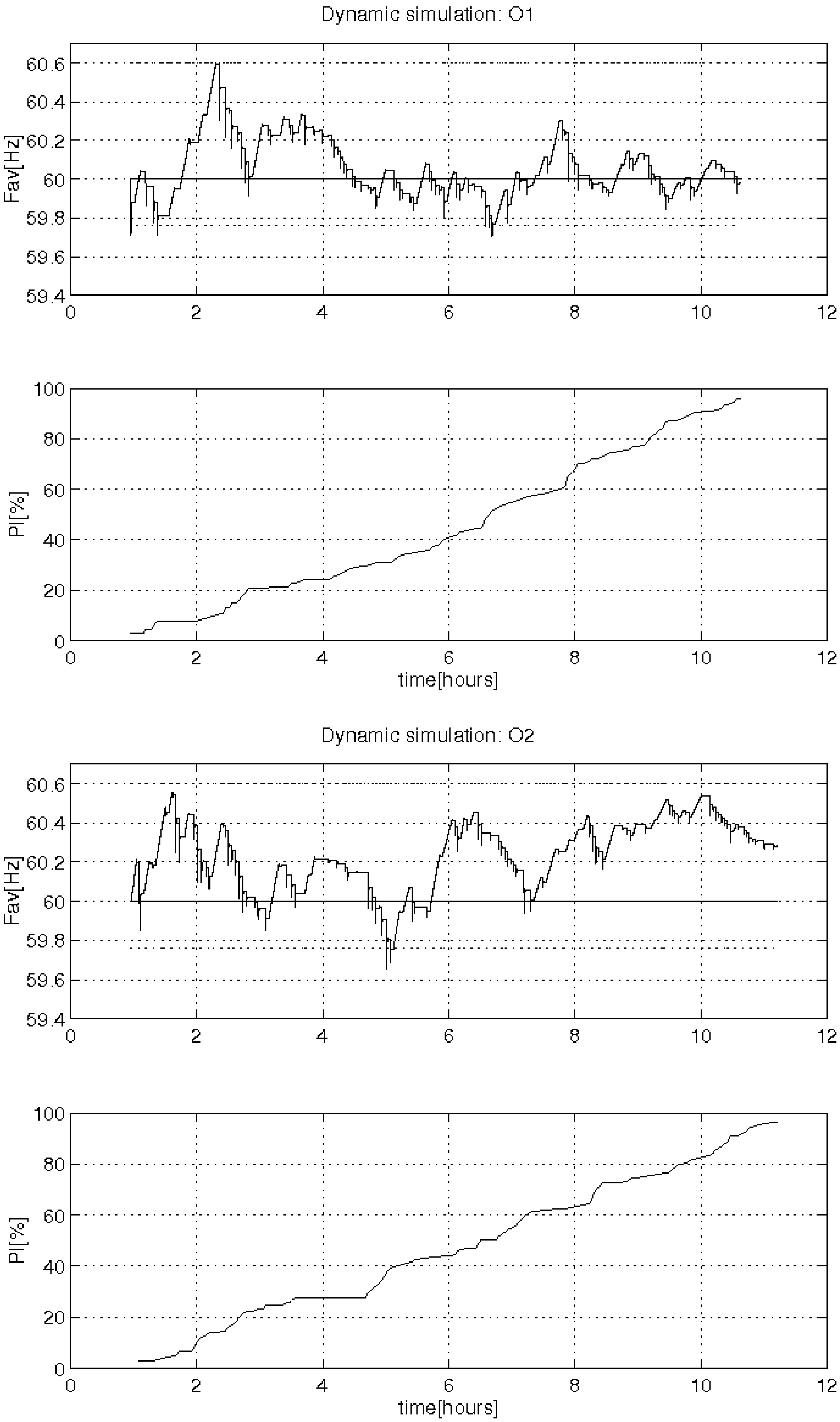


Figure 6.7 Dynamic simulation of the optimised sequences obtained in GA runs O1 and O2

6.3.4.2 Different initial states

Two more optimisations have been run, this time with the same seed (the one from the above optimisation O1), but with different initial power system states as follows:

- O3: lightly loaded network (80%); and
- O4: load in the network higher than nominal (110%), and line `n12n13` unavailable (this line is the heaviest loaded line in the nominal state, with a relative load of 84%).

6.3.4.2.1 GA convergence

Figure 6.8 shows the evolution of the maximum fitness value through generations in GA optimisations O3 and O4, respectively. The speed of the convergence depends strongly on the initial state.

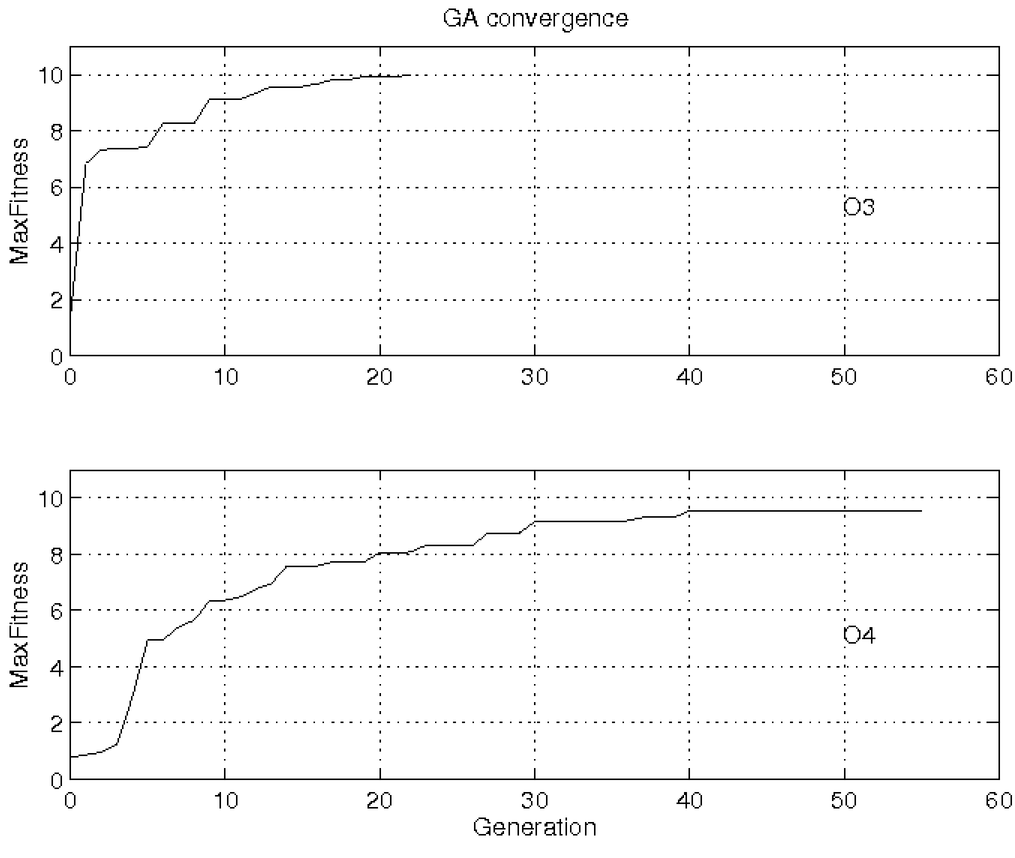


Figure 6.8 GA results for optimisations O3 and O4

The "easy" initial state, corresponding to optimisation O3, has resulted in a very fast convergence, i.e., after 22 generations. The initial population contained an LP sequence with fitness around 2, and already after the first generation, the best fitness increased to 7. When the power system conditions are favourable, premature convergence usually does not occur. The stop criteria that applied in this case was the second one from Section 6.3.4.1.1 (i.e., the obtained LP sequence had a fitness close to 10).

Optimisation O4 has been run for the state somewhat more difficult than the nominal one, so that the convergence has been slower. The LP sequence has been obtained at generation 55 due to the "minimum violations" criterion, as in the case of optimisations O1 and O2.

6.3.4.2.2 Power system evolution

Figure 6.9 shows the evolution of the frequency and the rate of load supply when the optimised sequences from optimisations O3 and O4, respectively, are applied to the power system.

Although the sequence O3 manages to pick up the total load, some violations of the operating limits could not be eliminated through the recombinations and mutations. These peaks appear during the first hour and a half of the load pickup stage of restoration, during which 25% of the load has been supplied. As in the case of sequences O1 and O2, the conclusion is that the fitness function (6.6) is not "smart" enough to capture the brief frequency violations.

On the contrary, sequence O4 does not supply the total pre-blackout load, but it does not cause any operating limit violation. Indeed, compared to all three of the other sequences, the frequency has the smallest excursions. Since the load level is higher than nominal, the restoration takes more time to be completed.

6.3.4.3 Computation time

The experiments have been carried out on an HP-UX 9000/735 workstation. The simulation by the Dynamic simulator of a single LP sequence for the New England test network takes between 5 and 6 seconds (i.e., the evaluation of a single individual). In the optimisations O1 and O2 (with the nominal power system state), the average time to create a generation is 11 minutes, while for "not-nominal" optimisations O3 and O4 this interval was 13 minutes. For the "difficult" initial state this is not surprising, but might seem strange for the "easy" state. In fact, when the load level is smaller than nominal, there should be fewer actions which increase the power set-point of production units when all of them are available. In addition, the spinning reserve is higher than in the nominal state, so the permitted load increments are also higher, which may lead to important excursions of the frequency.

In conclusion, the average time necessary to create a generation depends on whether the initial state is nominal (as defined in Section 6.3.4.1); if this is not the case, there is a computation time increase of about 20%. On the contrary, the duration of the whole optimisation process depends on the convergence speed, i.e., on the "difficulty" of the initial state.

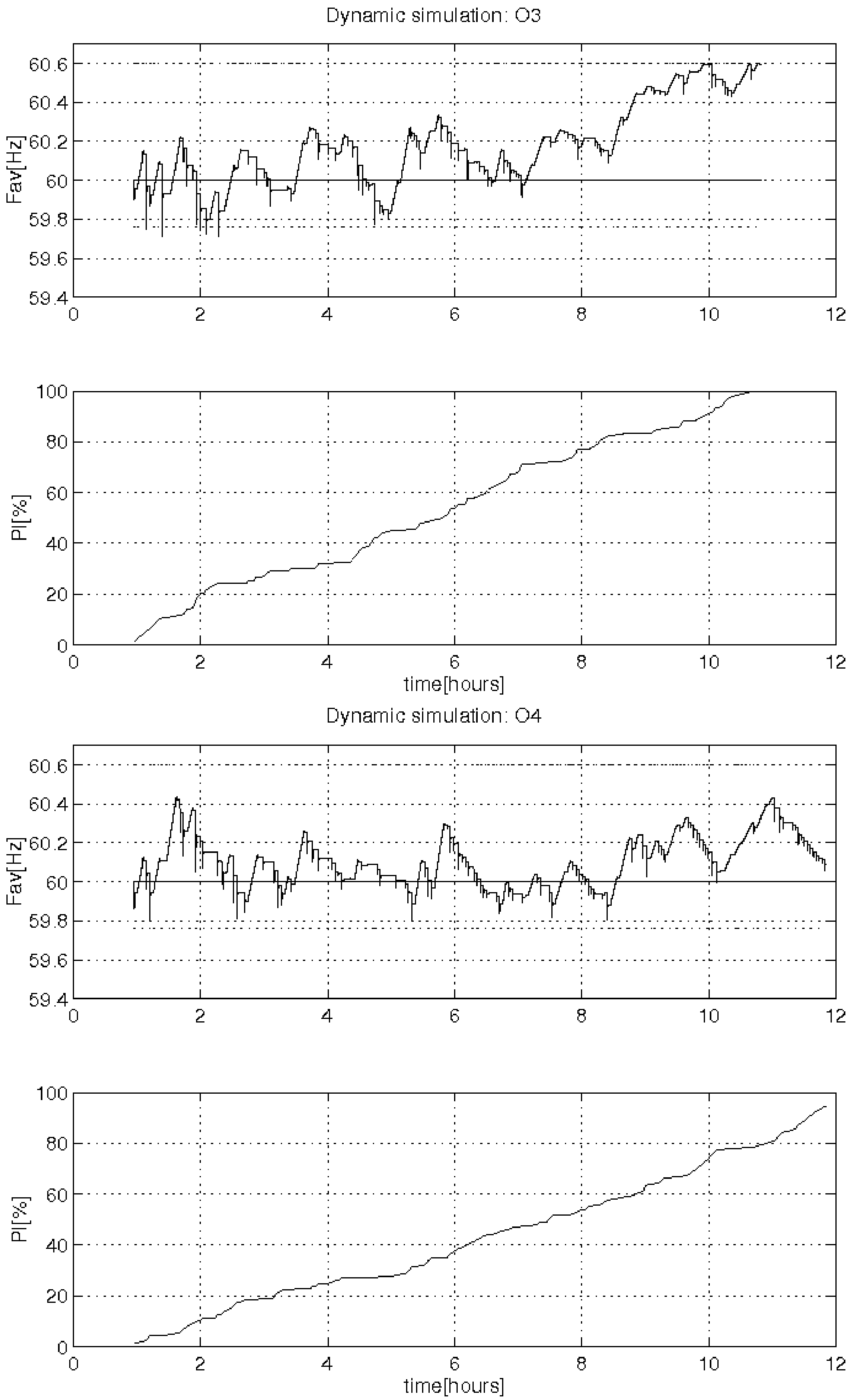


Figure 6.9 Dynamic simulation of the optimised sequences obtained in GA runs O3 and O4

6.3.5 Discussion

As shown in Section 6.3.4.1, the GA-based optimisation can generate many different optimised LP sequences for the given initial power system state, each of them leading to equally acceptable results[16], since they have high fitness values. Our hope was that the comparisons and analyses of different optimised LP sequences would permit the generic rules for the given state of the given power system to be deduced. Unfortunately, this was not the case. Due to the random nature of the GA technique and the coding method (single actions from the list of LP actions), it has been found that the obtained sequences have few common characteristics. What is common for all of them is the similar number of valid actions and the satisfactory result. On the contrary, it is impossible to explain why one action is, for instance, the first action in one LP sequence and not in the others. In other words, there is no regularity in the order of actions from one LP sequence to another, which made the deduction of rules based on the order of actions, i.e., the strategy itself, impossible.

However, we were curious to know whether the optimised LP sequences, as they are, could be exploited in some manner. Then Alba proposed an approach based on a machine learning technique which permits the LP sequences to be classified in a decision tree according to the characteristics of the power system in its initial state. The basics and the very first results of this approach can be found in [Kos96], and the detailed description with more exhaustive results in [Alb96]; in the following section only a brief overview of this methodology and the applicability of its results in the framework of Decision Aid Function FOr Restoration (DAFFOR) will be given.

6.3.6 Classifying with a decision tree

The GA-based optimisation permits an LP sequence to be generated for a given initial power system state. Regarding a single LP sequence, it has been concluded that it might be adequate for several power system states, thanks to the redundancy of the actions it contains. For instance, an action (e.g., connect line `n12n13`) which is invalid for one power system state (if line `n12n13` is unavailable) may be valid for another power system state (if line `n12n13` is available). Therefore, **one** LP sequence can be associated to **several** initial power system states. In the other sense too, **several** LP sequences can be equally adequate for a **single** initial power system state, as shown with examples in Section 6.3.4.1.

The GA-based optimisation can be used to generate different pairs (initial power system state, optimised LP sequence), and the relations among them can be "learned" and organised in a decision tree which deals with pairs (attributes and class). The core of a decision tree approach is the organisation of the **attributes** of a problem in a tree structure, in which the more relevant an attribute is, the closer to the root it is placed. The tree will be used to determine which attributes must be checked to **classify** a new situation.

---

[16] This fact only confirms the combinatorial nature of the restoration of transmission power systems.

The attributes are the variables which determine the initial power system state (**load level, availability of production units, transmission equipment and external network support**), and the classes are optimised LP **sequences**. Therefore, the obtained decision tree encodes the simple criteria (the values of the most relevant power system variables for the current initial state, or attributes) which permit the most adequate LP sequence (i.e., the class with the highest fitness) to quickly be chosen.

6.3.6.1 Possible integration in DAFFOR

The most time-consuming tasks (GA-optimisations and generation of the decision tree) are carried out off-line. These processes result in a "library" of LP sequences (6.2) which are "classified" in the decision tree. Given the current power system state, the decision tree chooses the most adequate one.

As soon as the regional skeleton network (RSN) has been created, DAFFOR could use the LP sequence chosen by the decision tree. Since the guidance functionality of DAFFOR is provided by its Reasoning kernel, the LP sequence should be exploited in the latter as explained below.

Referring to transition expression (6.3), the first reasoning cycle after the RSN has been created can be denoted as $r=N_{skel}+1$. Instead of controlling the search process for sub-sequence (4.21) of N_{act} control actions, the Dialogue&control (D&C) module could simply simulate the first N_{act} actions from the chosen LP sequence (6.2) thanks to the Dynamic simulator (DS). If the first action from the LP sequence is solution, $a_{Nskel+1}$, it can be suggested to the operators; if they accept action $a_{Nskel+1}$, it can be removed from the LP sequence, and the next one can be tested in the following reasoning cycle. Otherwise, the D&C module schedules the ordinary search for sub-sequence (4.21) which is performed in the Knowledge base module.

As long as the LP sequence (6.2) provides a solution, it can be used instead of launching the search process, which can speed up DAFFOR's response. Note, however, the following:

- it would not be possible to explain "why" or "how" the action has been found;
- if an unforeseen event occurs, the LP sequence is likely to become useless.

* * *

Since the decision tree is based on a given structure and the production capacity of the power system, it should be updated whenever a significant change occurs in the topology, installed capacity or load profile in it (e.g., a new transmission line has been built; an old production unit is no longer used, etc.). This means that additional GA-optimisations should be carried out (in order to add new LP sequences to the "library") and that the new decision tree should be created which will "re-classify" the LP sequences according to the new structure of the power system.

6.4 Conclusion

This chapter dealt with an original approach to the second stage of the transmission power system restoration problem. A genetic algorithm (GA) has been implemented which generates optimised load pickup (LP) sequences of restoration actions. The GA optimisation uses the Dynamic simulator for the evaluation of the LP sequences. The fitness function translates the objective of the optimisation, i.e., to find the LP sequence which provides the maximal supply of unserved load, while still respecting operating limits.

The approach was expected to help in the determination of the load pickup strategy. However, due to the combinatorial nature of the problem itself and the random nature of the optimisation technique, the obtained LP sequences did not permit us to *quickly* deduce strategic rules. Further investigations are needed in order to find the manner to interpret the order of actions in the LP sequences. Because of the lack of time, this research had to be postponed.

In order to improve the performance of the proposed GA-method, the reproduction should be performed without duplicates. This means that when an LP sequence with positive fitness is created, it should be compared against the other LP sequences that are already in the population. Because of the redundancy in coding, the LP sequence contains some invalid actions which are simply ignored by the Dynamic simulator. When comparing the LP sequences, only valid actions should be taken into account. Therefore, a sequence whose valid actions are exactly the same as the actions of another sequence should not be kept in the population.

The results presented in Figures 6.7 and 6.9 show that the fitness function should be improved in such a manner as to better captures the frequency limit violations. For instance, the ratio of the frequency integrals in (6.5) could be amplified if smaller than a given threshold so that the violations which are very small (compared to the total frequency integral) could have more impact than they have at present.

The performance of the obtained LP sequences could be improved by using the heuristic limits (4.17-4.20) in the Dynamic simulator when it is called from the evaluation function of the GA. For instance, if the frequency is low, any action that results in a load pickup should be declared as invalid. In this manner, the frequency excursions would certainly be much smaller.

Finally, other types of crossover and mutation operators could be considered. For instance, the restoration could be seen as a scheduling task, so that the order-based crossover and mutation operators (see Section 21 in [Dav91]) would replace the current ones.

Chapter 7

Conclusion

This thesis focuses on the problem of the restoration of transmission power systems. The increasing demand for electric power cannot be followed, at the same pace, by an increase of the production and transmission capacity of power systems. This fact makes the electric power utilities operate their systems closer and closer to the systems' limits. With such a tendency, it is likely that a small disturbance, occurring in a "bad" moment (e.g., peak load), can result in the cutoff of the power supply for many consumers, i.e., a blackout. The operators in control centres have the task of quickly and securely resupplying all the unserved consumers, which is not an easy task. This work proposes a computer tool which can play the role of the operators' adviser during restoration in real time: **D**ecision **A**id **F**unction **FO**r **R**estoration (DAFFOR).

The **principal objective** of this work was the conceptual design of DAFFOR in such a manner that it can "reason" in real time, which implies the capability of keeping in step with the evolution of the power system state, including unforeseen events. The proposed DAFFOR consists of two kernels:

- The Reasoning kernel (Chapter 4) has the task of assisting the operator during the restoration process, taking into account the real-time dynamic behaviour of the power system.
- This behaviour is analysed by the Real Time Update (RTUpd) kernel (Chapter 5) in order to generate a coherent representation of the current power system state, and the message

which indicates to the Reasoning kernel how to operate. One of the sources of information for the RTUpd kernel that enables it to identify unforeseen events is the simulated expected power system state, obtained thanks to the Dynamic simulator (Section 4.1), and which results from the execution of an action proposed by the Reasoning kernel.

After DAFFOR had been designed, we proceeded to the **second objective** defined in Section 1.2, i.e., the determination of the guidelines for the load pickup restoration stage:

- The first direction of research was towards a method which could automatically generate the load pickup (LP) sequences. A genetic algorithm (GA) has been used for obtaining the optimised LP sequences (Chapter 6).

After the results were analysed, the need for further investigations was identified in order to extract the strategic rules that could be implemented in DAFFOR. Although the results of the research have not yet been implemented in DAFFOR, the approach is, to our knowledge, an original one and as such it found its place in this thesis.

The **third objective** covered the development, implementation and validation of the stand-alone prototype of DAFFOR (Chapter 4). Since the work has been realised in cooperation with Électricité de France (EDF), their restoration strategy for the first restoration stage was first studied in detail. EDF's strategy, based on the creation of the regional skeleton networks (RSNs), has been implemented and tested with the data provided by EDF. Simultaneously, the abundant literature concerning restoration in general has been overviewed (Chapter 2), which permitted us first to determine and implement very simple guidelines for the load pickup strategy (**second objective**). The intensive simulations and the occurrence of backtracking made it possible to improve the performance of EDF's implemented strategy for the first restoration stage, and to refine the knowledge for the load pickup strategy. The stand-alone prototype of DAFFOR has been tested with two sets of data provided by EDF, as well as with the New England test network, and has shown a good efficiency.

As to the **fourth objective** of this thesis, the stand-alone prototype has been extended with the concept of the update in real time, as explained in Chapter 5. This resulted in the real-time prototype of DAFFOR, which is now ready for coupling with EDF's operator training simulator.

7.1 Contributions of this Work

The design of DAFFOR proposed in this work can be seen as a general model for any real-time control assistance application. The application should be provided with an appropriate internal dynamic simulator and knowledge base (in the Reasoning kernel) and comparison functions (in the RTUpd kernel). The interaction of such an application with the external world is twofold: (1) the RTUpd kernel is linked to the real-time database, and (2) the Reasoning kernel to the user interface. As far as the integration in a real environment (e.g., energy management system) is concerned, the first step is to couple the Reasoning kernel with the user interface (i.e., the stand-alone prototype), and then the RTUpd kernel to the real-time database: the modifications to be done in the kernels stay at the "surface", on the level of the data, but without affecting the interaction between the kernels.

Another contribution of this work is the use of the *Dynamic* simulator (as opposed to the *static* power flow), which is fast enough for application in real time. The Dynamic simulator enables the assessment of control actions taking into account the dynamic response of the power system, the time of that response as well as the time of manoeuvres, the units' ramping rates, the load shedding levels, etc.

Next, a restoration strategy has been proposed which focuses mainly on the second restoration period, and therefore is called the load pickup strategy. However, it has been conceived in such a manner that it can also apply to the first restoration stage.

Finally, a novel direction of research has been opened with the GA-based approach for generating the optimised load pickup sequences. The results presented in this work can be considered as a feasibility study of the approach, and more research must be carried out in order to be able to exploit its results in the desired manner.

7.2 Further work

This document has shown what has been done during a 3-year period. As the author was told several times, there is a moment when one must decide to stop the research and present what has been done up to that moment. Obviously, there is no work which is absolutely complete, which also applies to the presented work. Therefore, there are many things to be done in the future in order to improve the current performance and the results of both DAFFOR and the GA-based optimisation. Some of the directions for further work are enumerated below following the order of sections and chapters in this document.

(Section 4.1) The Dynamic simulator could be enhanced with: (1) the dynamic load model, and (2) the capability to take into account different time scales if there are several islands in the system.

(Section 4.2) The strategies and the heuristics implemented in the Knowledge base module must be intensively tested with the variety of initial states, which will permit the existing goals (and rules) to be improved, as well as more sophisticated rules to be determined and added. Once the knowledge is represented in a satisfactory manner, the whole module should be translated into a procedural language (e.g., C or C++) in order to: (1) guarantee the portability, (2) facilitate the integration in a real environment (EMS), and (3) significantly increase the speed of execution.

(Section 4.3) The Dialogue&control module is the first candidate for translation into a procedural language, independently of the state of the development of the Knowledge base module. Its search control and data handling functions should be somewhat modified for the case in which there are several islands in the power system to be considered simultaneously (and not sequentially, as at present).

(Section 4.4) The restoration time of the procedures found by DAFFOR could be decreased by dividing the time of manoeuvres into two time intervals. Instead of using a single parameter per type of action, two parameters should be identified and defined as a function of both (1) the type of action, and (2) the power system element.

(Chapter 5) At present, DAFFOR is ready to be coupled to EDF's operator training simulator. The coupling will permit us to: (1) set some parameters in the Real Time Update kernel, (2) improve the current states comparison functions, and (3) test the performance of the real-time prototype of DAFFOR. In addition, since DAFFOR is supposed to run during an abnormal power system state (restoration), a specialised state estimator needs to be available.

(Chapter 6) The GA-optimisation could certainly improve performance by: (1) improving the fitness function, (2) using the heuristic limits, (3) avoiding having duplicates in a population. Another type of genetic operators should be implemented and tested in order to see whether the optimised load pickup sequences contain some strategic rules, which are at present impossible to deduce.

Appendices

Appendix A contains the scheme, characteristics and dynamic simulation parameters for the New England test power system used for different simulations in this work.

Appendix B shows a part of the original C-include file with definitions for constants and types concerning the power system and its elements. There is a comment provided for each field. At the end of the comment, the letter "`c`" means that the field is **c**onstant, and the letter "`d`" that it changes (**d**ynamically).

Appendices C and D give different dynamic simulation parameters for the French test networks from Sections 4.4.2.1 and 4.4.2.2, respectively.

A. New England test network

Figure A.1 shows the New England test network.

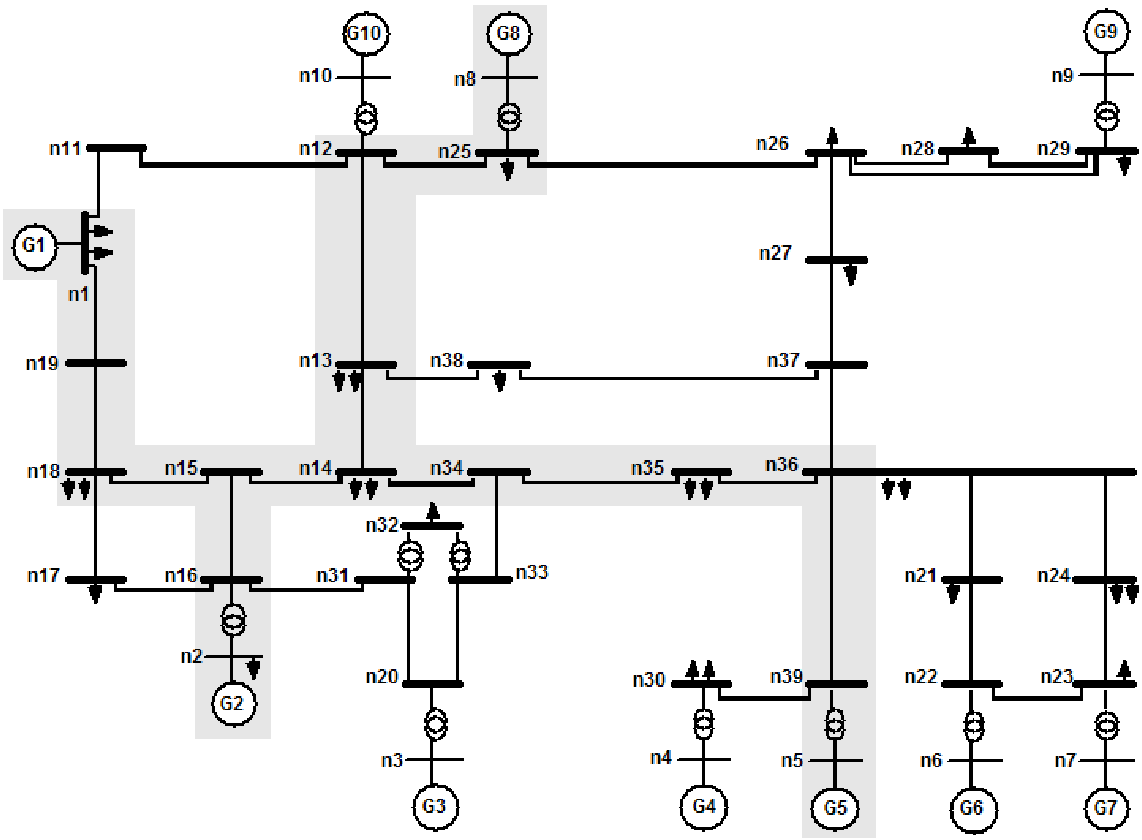


Figure A.1 New England test network, with highlighted skeleton network.

The characteristics of this network are as follows:

- 39 nodes;
- 10 units: 9 classical thermal and one equivalent (`G1`), with a total capacity of 7870MW (6370+1500);
- 19 consumer loads: 18 real and one equivalent for a total of 6150MW (5110+1040);
- 35 350kV lines, 2 350/200 kV/kV transformers and 9 step-up transformers;
- no compensation devices (condensers or reactors).

Some modifications have been made to the original data:

- real consumer loads of more than 300MW have been split in two loads on the same bus, and
- auxiliaries have been added for all thermal units (according to the nodal model shown in Figure 4.7.a), with a consumption of about 3% of the unit's rated power.

Following are the parameters for dynamic simulation:

```
Network type:         national

Base frequency:            60 [Hz]
Base power:                100.0 [MW]

Convergence criterion:                0.001 [pu]
Load level:                           1.000 [pu]
Max number of LDF iterations:         30
Acceleration coefficient     (delQ): 1.00
Acceleration sub-coefficient (delQ): 0.50
Initial voltage for PQ nodes:         1 [pu]
Start tap changing at iteration:      4

Integration step:               0.050 [sec]
Steady state criterion (Pm-Pe): 0.0010 [pu]
N of consecutive steady states: 30
delta Vmax   when put ON bra:   0.1000 [pu]
delta DEGmax when put ON bra:   30.0 [deg]
delta HZmax  when put ON bra:   0.1000 [Hz]
Transient frequency limits:     (58.8000-61.2000) [Hz]
Steady state frequency limits:  (59.7600-60.6000) [Hz]
Voltage set-point limits     :  (0.9500-1.0500) [pu]

Voltage is low    (for heuristic tests): 0.2000
Voltage is high   (for heuristic tests): 0.0300
Frequency is low  (for heuristic tests): 0.1800
Frequency is high (for heuristic tests): 0.1500
Spinning reserve  (for heuristic tests): 0.1000 [pu]
Voltage set-point increase step :        0.0200 [pu]
Default time of manoeuvres      :        120 [sec]
Time of manoeuvres for branch coupling : 300 [sec]
Time of manoeuvres for HUnit  coupling : 240 [sec]
Time of manoeuvres for TUnit  coupling : 420 [sec]
```

B. Type definitions for power system state

```
/*-------------------------------------------------------------------------------*/
/*                              User defined constants                            */
/*-------------------------------------------------------------------------------*/
#ifndef __ELEM_TYPES__
  #define __ELEM_TYPES__
  #define THERMO    1       /* classical thermal unit                             */
  #define HYDRO     2       /* hydraulic unit                                     */
  #define NUCL      3       /* nuclear unit                                       */
  #define EQUIV     4       /* equivalent unit                                    */
  #define TS1      "ts1"    /* telesignalisation TS1 for all thermal units        */
  #define TS2      "ts2"    /* telesignalisation TS2 for nuclear units            */
  #define LIN       1       /* transmission line                                  */
  #define TG        2       /* step-up transformer                                */
  #define AT        3       /* autotransformer                                    */
  #define TCUL      4       /* tap changer under load transformer                 */
  #define CLIENT    1       /* concumer load                                      */
  #define CONDENS   2       /* condensor                                          */
  #define REAC      3       /* reactor                                            */
  #define AUX       4       /* thermal unit's auxiliaries                         */
  #define AUT      'a'      /* automatic TCUL;       frequency control unit       */
  #define FIX      '*'      /* manuel TCUL;          constant output power unit   */
  #define MPR      'm'      /*     -----  ;          ramping unit                 */
#endif /*  __ELEM_TYPES__  */

#define PQ          1       /* PQ bus for loadflow                                */
#define PV          2       /* PV bus for loadflow                                */
#define SL          3       /* SL bus for loadflow                                */
#define PQV         5       /* PQV bus for loadflow                               */
#define T_PRI      'p'      /* regulated voltage on the primary side of TCUL      */
#define T_SEC      's'      /* regulated voltage on the secondary side of TCUL    */
#define TH_LOADING  5.      /* % of Pnom per minute for all thermal units loading */
#define SH_LEV      5       /* number of shedding levels                          */

#define BOTH_DIRECT 1       /* both sides without switch (branches)               */
#define I_DIRECT    2       /* side I without switch                              */
#define J_DIRECT    3       /* side J without switch (branches)                   */
#define NO_DIRECT   4       /* side J without switch (branches)                   */
```

```
/*---------------------------------------------------------------------------*/
/*                              Type definitions                             */
/*---------------------------------------------------------------------------*/
                    /*                  CONTROL PARAMETERS                   */
typedef struct {    /*_______________________________________________________*/
    double Vb;             /* base voltage (format edf only)                 */
    int    lre;            /* format lre if true, otherwise format edf       */
    int    applic;         /* sort of application             (402 for ldf)  */
    double sb;             /* base power                      (def:100[MVA]) */
    double epmva;          /* convergence criterion           (def:0.1[MVA]) */
    double cniv;           /* load level                      (def:1.0[pu])  */
    int    prnt;           /* print info to screen            (def:1)        */
    int    prfl;           /* print results in files          (def:1)        */
    int    itmax;          /* max iteration number            (def:30)       */
    double coef;           /* acceleration coefficient        (def:1.)       */
    int    initia;         /* initialising PQ-bus voltage:    (def:1)        */
                           /*    0 - equals to Vslack                        */
                           /*    1 - equals to 1 pu                          */
                           /*    2 - equals to Vset-point (code 11)          */
    double tol;            /* tolerance for V set-point       (def:0.1)      */
    double xgrd;           /* acceler. subcoeff. (for Q)      (def:0.5)      */
    int    itet;           /* start tap-changing iteration    (def: 4)       */
                           /*------------------------------------------------*/
    double freq;           /* base frequency                  (def:50.0[Hz]) */
    double tstep;          /* integration step                (def:0.1[s])   */
    double CRIT;           /* st_state criterion (Pm-Pe)      (def:1.e-3[pu])*/
    int    AFFICH;         /* frequency of print results      (def:20)       */
    int    CONS_SS;        /* number of consecutive steady states (def:20)   */
    double VCRIT;          /* delta Vmax   when put ON bra    (def:0.1[pu])  */
    double FICRIT;         /* delta DEGmax when put ON bra    (def:30.[deg]) */
    double OMECRIT;        /* delta HZmax  when put ON bra    (def:0.1[Hz])  */
                           /*------------------------------------------------*/
    int    REGIONAL;       /* with 63kV level (load zones)    (def:1)        */
    int    WITH_EXTERNAL;  /* with external network support   (def:0)        */
    double FRE_MIN;        /* min transient frequency limit   (def:0.98[pu]) */
    double FRE_MAX;        /* max transient frequency limit   (def:1.02[pu]) */
    double SS_FRE_MIN;     /* min steady state freq limit     (def:0.996[pu])*/
    double SS_FRE_MAX;     /* max steady state freq limit     (def:1.01[pu]) */
    double VSET_MIN;       /* min voltage set-point           (def:0.95[pu]) */
    double VSET_MAX;       /* max voltage set-point           (def:1.05[pu]) */
    double LOW_V;          /* for heuristic tests             (def:0.100)    */
    double HIGH_V;         /* for heuristic tests             (def:0.030)    */
    double LOW_F;          /* for heuristic tests             (def:0.15)     */
    double HIGH_F;         /* for heuristic tests             (def:0.15)     */
    double SPIN_RES;       /* for heuristic tests             (def:0.1)      */
    double V_STEP;         /* voltage set-point default step  (def:0.01[pu]) */
    int    MAN_TIME;       /* default time of manoeuvres      (def:120[s])   */
    int    MAN_COUPL_BRA;  /* time of man. for branch         (def:300[s])   */
    int    MAN_COUPL_HYD;  /* time of man. for hydro unit     (def:240[s])   */
    int    MAN_COUPL_THE;  /* time of man. for thermal unit   (def:420[s])   */
} Param;


                    /*                  NETWORK DIMENSIONS                   */
typedef struct {    /*_______________________________________________________*/
    int   Nnod;            /* number of nodes                                */
    int   Nbra;            /* number of branches                             */
    int   Ngen;            /* number of production units                     */
    int   Nloa;            /* number of loads                                */
    int   Nzon;            /* number of electrical zones                     */
    int   Nlzn;            /* number of load zones                           */
} NetDim;
```

```
                    /*                    ELECTRICAL NODE                          */
typedef struct {    /*_____________________________________________________________*/
    double Unom;        /* nominal (base) node voltage [kV]                      c */
    double Umax;        /* maximum permissible voltage [pu]                      c */
    double Umin;        /* minimum permissible voltage [pu]                      c */
    double lowVlim;     /* limit for low voltage [kV]                            c */
    double highVlim;    /* limit for high voltage [kV]                           c */
    double lowFlim;     /* limit for low frequency [pu]                          c */
    double highFlim;    /* limit for high frequency [pu]                         c */
    double Qmax;        /* maximum reactive production [pu]                      d */
    double Qmin;        /* minimum reactive production [pu]                      d */
    double Volt;        /* node voltage [pu]                                     d */
    double Usp;         /* specified (set-point) voltage [KV]                    d */
    double Vviol;       /* voltage violation [pu]                                d */
    double Teta;        /* node phase angle [rad]                                d */
    double Pg;          /* real generation [pu]                                  d */
    double Qg;          /* reactive generation [pu]                              d */
    double Pl;          /* active load [pu]                                      d */
    double Ql;          /* reactive load [pu]                                    d */
    double Gsh;         /* shunt conductance [pu]                                d */
    double Bsh;         /* shunt susceptance [pu]                                d */
    double Psh;         /* real power of shunt [pu]                              d */
    double Qsh;         /* reactive power of shunt [pu]                          d */
    double Perr;        /* real power loadflow error [kW]                        d */
    double Qerr;        /* reactive power loadflow error [kW]                    d */
    int    Equiv;       /* weather equivalent or not                             c */
    int    Id;          /* identifier                                            c */
    int    Ezon;        /* electrical zone id                                    d */
    int    Lzon;        /* load zone id                                          d */
    int    Type;        /* node type (PQ, PQV, PV, SL)                           d */
    int    lowV;        /* flag for (heuristic) low voltage (t|f)                d */
    int    highV;       /* flag for (heuristic) high voltage (t|f)               d */
    int    lowF;        /* flag for (heuristic) low frequency (t|f)              d */
    int    highF;       /* flag for (heuristic) high frequency (t|f)             d */
    char   Name[W_LEN]; /* name                                                  c */
} SmNode;
```

```
                    /*                              BRANCH                              */
typedef struct {    /*__________________________________________________________________*/
    double Zmod;        /* longitudinal impedance module [pu]                          c */
    double Gij;         /* longitudinal conductance [pu]                               c */
    double Bij;         /* longitudinal susceptance [pu]                               c */
    double Yij;         /* transversal semi-susceptance [pu]                           c */
    double Smax;        /* maximal power flow [pu]                                     c */
    double Vpri;        /* primary  voltage [pu]                                       c */
    double Vsec;        /* secondry voltage [pu]                                       c */
    double Vmin;        /* voltage/angle for min tap [kV|deg]                          c */
    double Vmax;        /* voltage/angle for max tap [kV|deg]                          c */
    double dtap;        /* tap step: (Vmax-Vmin)/Ttot [kV|deg]                         c */
    double teta;        /* phase difference (for cType=2) [deg]                        c */
    double Qsp;         /* transit set-point [MVar|MW] (for cType=3|4)                 d */
    double P_ij;        /* branch real power flow from I to J                          d */
    double P_ji;        /* branch real power flow from J to I                          d */
    double Q_ij;        /* branch reactive power flow from I to J                      d */
    double Q_ji;        /* branch reactive power flow from J to I                      d */
    double S_ij;        /* branch power flow from I to J [pu]                          d */
    double S_ji;        /* branch power flow from J to I [pu]                          d */
    double CH_ij;       /* relative power flow from I to J [%/100]                     d */
    double CH_ji;       /* relative power flow from J to I [%/100]                     d */
    double Ploss;       /* branch real power losses [pu]                               d */
    double Qloss;       /* branch reactive power losses [pu]                           d */
    double a;           /* branch turns ratio (1.0 for lines)                          d */
    double Vvar;        /* voltahe on tap side (secondary side)                        d */
    double dfre;        /* frequency difference between I and J [pu]                   d */
    double dvol;        /* voltage   difference between I and J [pu]                   d */
    double dang;        /* angle     difference between I and J [rad]                  d */
    double gaprob;      /* probability of availability (used in GA)                    d */
    int    IsTra;       /* transformer id (-1  for lines)                              c */
    int    cType;       /* control transformer type:                                   c */
                        /*        1 - tap change under load (TCUL)                       */
                        /*        2 - phase shifter                                      */
                        /*        3 - reactive power control                             */
                        /*        4 - real power control                                 */
    int    Ttot;        /* total number of taps                                        c */
    int    ficI;        /* id of the origine fict node                                 c */
    int    ficJ;        /* id of the extremity fict node                               c */
    int    Parall;      /* parallelism index (if 0, single circuit)                    d */
    int    defI;        /* id of the origine default node                              d */
    int    defJ;        /* id of the extremity default node                            d */
    int    I;           /* origine node id                                             d */
    int    J;           /* extremity node id                                           d */
    int    Ezon;        /* electrical zone                                             d */
    int    Lzon;        /* load zone                                                   d */
    int    On;          /* operating status (true if ON or SBY)                        d */
    int    StBy;        /* standby status (= cbreakI|cbreakJ id if I|J open)           d */
    int    availI;      /* whether available or not                                    d */
    int    availJ;      /* whether available or not                                    d */
    int    Tap;         /* transformer tap-position                                    d */
    int    Id;          /* identifier                                                  c */
    int    unitId;      /* unit id (for type TG only)                                  c */
    int    inBlock;     /* TRUE if TG is in block with its unit (NUCL)                 c */
    int    SSw;         /* how is connected to busbar (BOTH_DIRECT,...)                c */
    int    Type;        /* LIN, TG, AT, TCUL                                           c */
    int    Equiv;       /* weather equivalent or not                                   c */
    int    boucl;       /* zone(i)==zone(j) != NOVALUE                                 d */
    int    coupl;       /* zone(i)!=zone(j); zone(i),zone(j) != NOVALUE                d */
    int    exten;       /* zone(i) OR zone(j) == NOVALUE                               d */
    int    braON_ok;    /* test for action bra on                                      d */
    char   Name[W_LEN]; /* name                                                        c */
    char   PorS;        /* control transformer controlled side:                        c */
                        /*         p,P = primary side (default)                          */
                        /*         s,S = secondary side                                  */
    char   cntrl;       /* AUT, FIX                                                    d */
} SmBranch;
```

```
                    /*                            LOAD                              */
typedef struct {    /*______________________________________________________________*/
    double ShedLevel[SH_LEV]; /* shedding levels                                  c */
    double NotPicked[SH_LEV]; /* amount of unserved load per level [pu]           d */
    double Pl0;               /* initial real consumption [pu]                    d */
    double Ql0;               /* initial reactive consumption [pu]                d */
    double Ploa;              /* current real consumption [pu]                    d */
    double Qloa;              /* current reactive consumption [pu]                d */
    double setMin;            /* lower limit of argument for action loa set       d */
    double setMax;            /* upper limit of argument for action loa set       d */
    int    Id;                /* identifier                                       c */
    int    ficI;              /* id of the fict node                              c */
    int    SSw;               /* I_DIRECT, NO_DIRECT                              c */
    int    unitId;            /* unit id (for type AUX only)                      c */
    int    Type;              /* CLIENT, COND, REAC, AUX                          c */
    int    Equiv;             /* weather equivalent or not                        c */
    int    Ezon;              /* electrical zone id                               d */
    int    Lzon;              /* load zone id                                     d */
    int    defI;              /* default node id                                  d */
    int    I;                 /* current node id                                  d */
    int    On;                /* operating status (1|0)                           d */
    int    avail;             /* weather available or not                         d */
    int    NotPickedLevel;    /* number of levels left to be picked up            d */
    int    CanBePickedLevel;  /* number of levels possible to pick up             d */
    int    loaON_ok;          /* test for action loa on                           d */
    char   Name[W_LEN];       /* name                                             c */
} SmLoad;
```

```
                    /*                         LOAD ZONE                            */
typedef struct {    /*______________________________________________________________*/
    double Ploa;          /* total real load                                      d */
    double Qloa;          /* total reactive load                                  d */
    int    Id;            /* identifier                                           d */
} SmLzone;
```

```
                    /*                      ELECTRICAL ZONE                         */
typedef struct {    /*______________________________________________________________*/
    double Pgen;           /* real power production                               d */
    double Ploa;           /* real power consumption                              d */
    double Qloa;           /* reactive power consumption                          d */
    double Ploss;          /* real power losses                                   d */
    double Qloss;          /* reactive power losses                               d */
    double Pmec;           /* mechanical power output                             d */
    double Rspin;          /* spinning reserve                                    d */
    double Htot;           /* inertia                                             d */
    double ome;            /* current frequency                                   d */
    double dome_old;       /* previous integration frequency derivative           d */
    double Iofre;          /* integral of frequency  over SS upper limit          d */
    double Iufre;          /* integral of frequency under SS lower limit          d */
    double Imfre;          /* integral of frequency                               d */
    double Iofre1act;      /* same as Iofre, but for 1 action                     d */
    double Iufre1act;      /* same as Iufre, but for 1 action                     d */
    double Imfre1act;      /* same as Imfre, but for 1 action                     d */
    int    Id;             /* identifier                                          d */
    int    slackId;        /* slack node id                                       d */
    int    nnodOn;         /* number of nodes                                     d */
    int    nbraOn;         /* number of branches (status ON)                      d */
    int    ngenOn;         /* number of units (status ON)                         d */
    int    nTs1;           /* number of TS1 or Hydro_On                           d */
    int    num_of_ldf;     /* number of the current loadflow                      d */
    int    SIG_frequency;  /* number of frequency violations                      d */
    int    SIG_voltage;    /* number of overvoltages                              d */
    int    SgVolt1act;     /* same as SIG_voltage, but for 1 action               d */
    int    SIG_overload;   /* number of overloads                                 d */
    int    SgLoad1act;     /* same as SIG_overload, but for 1 action              d */
} SmZone;
```

```
                    /*                       PRODUCTION UNIT                      */
typedef struct {    /*____________________________________________________________*/
    double Pn;          /* rated turbine power                                c */
    double Sgb;         /* apparent power [MVA]                               c */
    double Ugb;         /* base voltage at its ends [kV]                      c */
    double Qgmin;       /* minimal reactive power [pu]                        c */
    double Qgmax;       /* maximal reactive power [pu]                        c */
    double r;           /* permanent speed droop coefficient                  c */
    double Ts;          /* servomotor time constant                           c */
    double H;           /* inertia time constant [sec]                        c */
    double dgmax;       /* maximum speed variation                            c */
    double dgmin;       /* minimum speed variation                            c */
    double Thp;         /* high pressure time constant                        c */
    double Tip;         /* intermediate pressure time constant                c */
    double Tlp;         /* low pressure time constant                         c */
    double fhp;         /* high pressure power fraction                       c */
    double fip;         /* intermediate pressure power fraction               c */
    double flp;         /* low pressure power fraction                        c */
    double Tr;          /* dashpot time constant                              c */
    double Tw;          /* water starting time constant                       c */
    double Delta;       /* transient speed droop coefficient                  c */
    double StartTime;   /* unit available after this time [min]               d */
    double gmax;        /* maximum gain                                       d */
    double gmin;        /* minimum gain                                       d */
    double Pmec;        /* mechanical power output                            d */
    double Pgen;        /* unit real production (electrical)                  d */
    double Qgen;        /* unit reactive production (electrical)              d */
    double Rspin;       /* spinning reserve                                   d */
    double Fmax;        /* maximum frequency                                  d */
    double Fmin;        /* minimum frequency                                  d */
    ....................../* some auxiliary variables                         d */
    double Pset;        /* mecanical power set-point                          d */
    double Vset;        /* voltage set-point                                  d */
    double Pgrad;       /* loading gradient per second                        d */
    double Pset_new;    /* new power set-point                                d */
    double time_now;    /* moment when loading starts                         d */
    double setMin;      /* lower limit of argument for action gen set         d */
    double setMax;      /* upper limit of argument for action gen set         d */
    double volMin;      /* lower limit of argument for action gen vol         d */
    double volMax;      /* upper limit of argument for action gen vol         d */
    double gaprob;      /* probability of availability (used in GA)           d */
    int    genSBY_ok;   /* test for action gen sby                            d */
    int    genON_ok;    /* test for action gen on                             d */
    int    Id;          /* identifier                                         c */
    int    Type;        /* THERMO, HYDRO, NUCL, EQUIV                         c */
    int    traId;       /* transformer id (except for type EQUIV)             c */
    int    auxId;       /* auxiliaries id (for types THERMO, NUCL)            c */
    int    inBlock;     /* TRUE if TG is in block with its unit (NUCL)        c */
    int    SSw;         /* how is connected to busbar (I_DIRECT, NO_DIRECT)   c */
    int    ficI;        /* id of the fict node                                c */
    int    defI;        /* id of the default node                             d */
    int    I;           /* id of the current node                             d */
    int    Ezon;        /* electrical zone                                    d */
    int    Lzon;        /* load zone                                          d */
    int    On;          /* operating status (true if coupled or isolated)     d */
    int    StBy;        /* true if in standby                                 d */
    int    Islnd;       /* true if isolated                                   d */
    int    InSkel;      /* is in skeleton (1|0)                               d */
    int    avail;       /* weather available or not                           d */
    char   Name[W_LEN]; /* name                                               c */
    char   TlSig[W_LEN]; /* TS1, TS2, *                                       d */
    char   cntrl;       /* AUT, FIX, MPR                                      d */
} SmUnit;
```

```
                    /*                       POWER SYSTEM STATE                      */
typedef struct {    /*____________________________________________________________*/
    double   s_time;      /* simulation time [sec]                                  */
    double   s_real_time; /* real time [sec]                                        */
    double   Ptot;        /* total CLIENT pre-blackout load [pu]                    */
    double   Psupplied;   /* currently supplied CLIENT load [pu]                    */
    NetDim   s_dim;
    SmNode   *s_nod;
    SmBranch *s_bra;
    SmLoad   *s_loa;
    SmUnit   *s_gen;
    SmZone   *s_zon;
    SmLzone  *s_lzn;
} SmState;
```

C. Parameters for the French national network

```
Network type:         national

Base frequency:            50 [Hz]
Base voltage (only *.edf): 225 [kV]
Base power:                100.0 [MW]

Convergence criterion:              0.001 [pu]
Load level:                         1.000 [pu]
Max number of LDF iterations:       30
Acceleration coefficient     (delQ): 1.00
Acceleration sub-coefficient (delQ): 0.50
Initial voltage for PQ nodes:       1 [pu]
Start tap changing at iteration:    4

Integration step:              0.040 [sec]
Steady state criterion (Pm-Pe): 0.0010 [pu]
N of consecutive steady states: 20
delta Vmax   when put ON bra:   0.1000 [pu]
delta DEGmax when put ON bra:   30.0 [deg]
delta HZmax  when put ON bra:   0.1000 [Hz]
Transient frequency limits:     (49.0000-51.0000) [Hz]
Steady state frequency limits:  (49.8000-50.5000) [Hz]
Voltage set-point limits     :  (0.9500-1.0500) [pu]

Voltage is low    (for heuristic tests): 0.1000
Voltage is high   (for heuristic tests): 0.0300
Frequency is low  (for heuristic tests): 0.1800
Frequency is high (for heuristic tests): 0.1400
Spinning reserve  (for heuristic tests): 0.1000 [pu]
Voltage set-point increase step :        0.0100 [pu]
Default time of manoeuvres      :        120 [sec]
Time of manoeuvres for branch coupling : 300 [sec]
Time of manoeuvres for HUnit  coupling : 240 [sec]
Time of manoeuvres for TUnit  coupling : 420 [sec]
```

D. Parameters for the French regional network

```
Network type:          regional

Base frequency:             50 [Hz]
Base voltage (only *.edf): 225 [kV]
Base power:                100.0 [MW]

Convergence criterion:               0.001 [pu]
Load level:                          1.000 [pu]
Max number of LDF iterations:        40
Acceleration coefficient     (delQ): 0.80
Acceleration sub-coefficient (delQ): 0.50
Initial voltage for PQ nodes:        1 [pu]
Start tap changing at iteration:     4

Integration step:                0.150 [sec]
Steady state criterion (Pm-Pe): 0.0010 [pu]
N of consecutive steady states: 20
delta Vmax   when put ON bra:   0.1000 [pu]
delta DEGmax when put ON bra:   30.0 [deg]
delta HZmax  when put ON bra:   0.1000 [Hz]
Transient frequency limits:     (49.0000-51.0000) [Hz]
Steady state frequency limits:  (49.8000-50.5000) [Hz]
Voltage set-point limits     :  (0.9500-1.0500) [pu]

Voltage is low    (for heuristic tests): 0.1000
Voltage is high   (for heuristic tests): 0.0300
Frequency is low  (for heuristic tests): 0.1800
Frequency is high (for heuristic tests): 0.1400
Spinning reserve  (for heuristic tests): 0.1000 [pu]
Voltage set-point increase step :        0.0200 [pu]
Default time of manoeuvres      :        120 [sec]
Time of manoeuvres for branch coupling : 300 [sec]
Time of manoeuvres for HUnit  coupling : 240 [sec]
Time of manoeuvres for TUnit  coupling : 420 [sec]
```

Bibliography

[Adi94] M. M. Adibi, L. H. Fink, "Power System Restoration Planning", *IEEE Transactions on Power Systems*, vol.9, no.1, 1994, pp.22-28

[Alb96] Juan-José Alba, "Switching Planning in Electric Power Systems: Application of Knowledge Engineering Techniques", in *Departamento de Electrotécnica y Sistemas*. Madrid: Universidad Pontifícia Comillas, 1996, pp. 174

[Anc95] J. J. Ancona, "A Framework For Power System Restoration Following a Major Power Failure", *IEEE Transactions on Power Systems*, vol.10, no.3, 1995, pp.1480-1485

[Ban95] J. Bann, et al., "Integration of Artificial Intelligence Applications in the EMS: Issues and Solutions", presented at Conference on Power Industry Computer Applications (PICA), Salt Lake City, Utah, May 7-12 1995, pp.456-463

[Che80] André Cheimanoff, Claude Corroyer, "The Power Failure of December 19, 1978", *RGE*, vol.89, no.4, 1980, pp.280-296

[Chu91a] R. F. Chu, et al., "Restoration Simulator Prepares Operators for Major Blackouts", *IEEE Computer Applications in Power*, vol.4, no.4, 1991, pp.46-51

[Chu91b] R. F. Chu, et al., "The Uses of an Operator Training Simulator for System Restoration", presented at Conference on Power Industry Computer Applications (PICA), Baltimore, USA, May 7-10 1991, pp.171-177

[CIG90] CIGRÉ Committee Report, "Modelling and Simulation of Black Start and Restoration of an Electric Power System - Results of a Questionnaire", Modelling of Abnormal Conditions Working Group (WG 38.02.02), *ÉLECTRA*, no.131, 1990, pp.155-169

[CIG93b] CIGRÉ Committee Report, "A Survey of Expert Systems for Power System Restoration", Task Force TF 38.06.04, *ÉLECTRA*, no.150, 1993, pp.87-105

[Cih69] T. C. Cihlar, J. H. Wear, D. N. Ewart, L. K. Kirchmayer, "Electric Utility System Security", *Proceedings of the American Power Conference*, vol.31, 1969, pp.97-109

[Cor93] J. M. Corera, et al., “On-Line Expert System for Service Restoration”, presented at Fourth Symposium on Expert Systems Application to Power Systems (ESAP), Melbourne, Australia, January 1993, pp.119-124

[Cou93] C. Counan, et al., “Major Incidents on the French Electric System: Potentiality and Curative Measures Studies”, *IEEE Transactions on Power Systems*, vol.8, no.3, 1993, pp.879-886

[Dar89] P.H. Darnault, et al., “An Expert System as a Guide for Information Synthesis and Decision Making in the Restoration of Power System after a Blackout”, presented at Second Symposium on Expert Systems Application to Power Systems (ESAP), Seattle, USA, July 17-20 1989, pp.399-404

[Dar91] Peter A. Darnel, Philip E. Margolis, *C: A Software Engineering Approach*, Springer-Verlag, Inc. ed. New York, 1991

[Dav75] D. R. Davidson, D. N. Ewart, L. K. Kirchmayer, “Long Term Dynamic Response of Power Systems: an Analysis of Major Disturbances”, *IEEE Transactions on Power Apparatus and Systems*, vol.PAS-94, no.3, 1975, pp.819-826

[Dav91] Lawrence Davis, *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991

[Del96] B. Delfino, et al., “Black-start and Restoration of a Part of the Italian HV Network: Modelling and Simulation of a Field Test”, *IEEE Transactions on Power Systems*, vol.11, no.3, 1996, pp.1371-1379

[Dij88] H. E. Dijk, van der A. W. Weegen, G. P. T. Roelofs, “An Expert System for Power System Restoration: Restoration Strategies”, presented at Symposium on Expert Systems Application to Power Systems (ESAP), Stockholm-Helsinki, August 22-26 1988, pp.8.9-8.16

[Dil88] T. S. Dillon, “Expert Systems - Potential and Limitations in the Application to Power Systems”, presented at Symposium on Expert Systems Application to Power Systems (ESAP), Stockholm - Helsinki, August 22-26 1988, pp.18.1-18.9

[Dil94] T. S. Dillon, E. Chang, “Solution of Power System Problems through the Use of the Object-Oriented Paradigm”, *Electrical Power & Energy Systems*, vol.16, no.3, 1994, pp.157-165

[DyL67] T. E. DyLiacco, “The Adaptive Reliability Control System”, *IEEE Transactions on Power Apparatus and Systems*, vol.PAS-86, no.5, 1967, pp.517-531

[EDF86] J. L. Boussin, “Plan de découpage du réseau et de reprise de service à l'aide des automatismes à manque de tension (AMU)”, EDF - Centre national des mouvements d'énergie, Paris, D.7061/TEP/T/85/JLB/CI/15, Feb 7, 1986

[EDF92] C. Kieny, A. Sbai, B. Blengino, “Ferrorésonance transitoire - Proposition d'un critère simple”, EDF - DER, Paris, HM-18/0205A, Dec 31, 1992

[EDF93a] Groupe de travail T34, “Note technique sur le calcul des temps de constitution des ossatures régionales”, EDF - Commission Interconnexion, Paris, D4002/56/DTE/93/337 PM, Sep 14, 1993

[EDF93b] P. Loevenbruck, P. Juston, “Étude de l'intérêt des selfs pour la constitution des ossatures régionales”, EDF - DER, Paris, HR-46/93/019, Sep 29, 1993

[EDF93c] Groupe de travail T34, “Études de nouvelles approches de la reconstitution du réseau après un incident généralisé - Note de synthèse sur travaux complémentaires”, EDF - Commission interconnexion, Paris, D4002-56/DTE/93/354 PM, Oct 22, 1993

[EDF94a] EDF, “Doctrine de Reconstitution du Réseau”, EDF - Exploitation du système Production Transport, Paris, D4002-56/DTE/94/625, Oct 18, 1994

[EDF94b] P. Michaux, “Rapport final - Maquettage de l'ossature régionale Loire”, EDF - Exploitation du système Production Transport, Paris, D4002-56/DTE/94/745, Oct 22, 1994

[EDF94c] Tatjana Kostić, “Évaluation des générateurs de système expert commerciaux”, EPFL-LRE/EDF-DER-FCR-CRR, Lausanne/Paris, III, Aug 31, 1994

[Fan91] J. Fandino, et al., “An Expert System as a Help for Power System Restoration after a Blackout”, presented at Third Symposium on Expert Systems Application to Power Systems (ESAP), Tokyo-Kobe, Japan, April 1-5 1991, pp.487-494

[Fin95] Lester H. Fink, Kan-Lee Liou, Chen-Ching Liu, “From Generic Restoration Actions to Specific Restoration Strategies”, *IEEE Transactions on Power Systems*, vol.10, no.2, 1995, pp.745-752

[Fog95] David B. Fogel, *Evolutionary Computation*. New York: Institute of Electrical and Electronics Engineers, Inc., 1995

[Fou97] N. A. Fountas, N. D. Hatziargyriou, C. Orfanogiannis, A. Tasoulis, “Interactive Long-Term Simulation for Power System Restoration Planning”, *IEEE Transactions on Power Systems*, vol.12, no.1, 1997, pp.61-68

[Fuk96] Yoshikazu Fukuyama, Hiroshi Endo, Yosuke Nakanishi, “A Hybrid System for Service Restoration using Expert System and Genetic Algorithm”, presented at International Conference on Intelligent Systems Applications to Power Systems (ISAP), Orlando, Florida, USA, January 28 - February 2 1996, pp.394-398

[Gai96] Z.L. Gaing, C.N. Lu, B.S. Chang, C.L. Cheng, “An Object-Oriented Approach for Implementing Power System Restoration Package”, *IEEE Transactions on Power Systems*, vol.11, no.1, 1996, pp.483-489

[Ger75] Alain Germond, “Computation of Ferroresonant Overvoltages in Actual Power Systems by Galerkin's Method”, presented at Conference on Power Industry Computer Applications (PICA), New Orleans, Louisiana, USA, June 2-4,1975 1975, pp.127-135

[Ger92] A. J. Germond, D. Niebur, “Survey of Knowledge-Based Systems in Power Systems: Europe”, *Proceedings of the IEEE*, vol.80, no.5, 1992, pp.732-744

[Gol89] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*: Addison-Wesley, 1989

[Gro91] François Gros, *Les secrets du gène*, vol. OJ14. Paris, 1991

[Gut87] J. Gutiérrez, M. Staropolsky, A. García, “Policies for Restoration of a Power System”, *IEEE Transactions on Power Systems*, vol.PWRS-2, no.2, 1987, pp.436-442

[Hig94] Shuji Higashi, et al., “Development of the Advanced Intelligent Restoration Guidance System for a Control Center”, presented at International Conference on Intelligent Systems Applications to Power Systems (ISAP), Montpellier, France, September 1994, pp.731-738

[Hol75] John H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975

[Hol92] A. T. Holen, Å. Botnen, P. Støa, J. J. Keronen, “Coupling Between Knowledge-Based and Algorithmic Methods”, *Proceedings of the IEEE*, vol.80, no.5, 1992, pp.745-757

[Hot90] K. Hotta, et al., “Implementation of a Real-Time Expert System for a Restoration Guide in a Dispatching Center”, *IEEE Transactions on Power Systems*, vol.5, no.3, 1990, pp.1032-1038

[Hua91] J. A. Huang, F. D. Galiana, G. T. Vuong, “Power System Restoration Incorporating Interactive Graphics and Optimization”, presented at Conference on Power Industry Computer Applications (PICA), Baltimore, May 7-10 1991, pp.216-222

[Hua95] J. A. Huang, L. Audette, S. Harrison, “A Systematic Method For Power System Restoration Planning”, *IEEE Transactions on Power Systems*, vol.10, no.2, 1995, pp.869-875

[IEE82] IEEE Committee Report, “System Restoration - Deploying the Plan”, Current Operational Problems Working Group, *IEEE Transactions on Power Apparatus and Systems*, vol.PAS-101, no.11, 1982, pp.4263-4271

[IEE87a] IEEE Committee Report, “Power System Restoration - A Task Force Report”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.PWRS-2, no.2, 1987, pp.271-277

[IEE87b] IEEE Committee Report, “Power System Restoration - The Second Task Force Report”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.PWRS-2, no.4, 1987, pp.927-933

[IEE88] IEEE Committee Report, “System Operations Challenges”, Current Operational Problems Working Group, *IEEE Transactions on Power Systems*, vol.3, no.1, 1988, pp.118-126

[IEE91] IEEE Committee Report, “Dispatcher Training Simulators - Lessons Learned”, a report by the Current Operational Problems Working Group, *IEEE Transactions on Power Systems*, vol.6, no.2, 1991, pp.594-604

[IEE92a] IEEE Committee Report, “Special Considerations in Power System Restoration”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.7, no.4, 1992, pp.1419-1427

[IEE92b] IEEE Committee Report, “New Approaches in Power System Restoration”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.7, no.4, 1992, pp.1428-1434

[IEE92c] IEEE Committee Report, “Overvoltage Control During Restoration”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.7, no.4, 1992, pp.1464-1470

[IEE93] IEEE Committee Report, “Bulk Power System Restoration Training Techniques”, a joint report by the Working Group on Operator Training and Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.8, no.1, 1993, pp.191-197

[IEE94a] IEEE Committee Report, “Special Consideration in Power System Restoration - The Second Working Group Report”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.9, no.1, 1994, pp.15-21

[IEE94b] IEEE Committee Report, “Analytical Tool Requirements for Power System Restoration”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.9, no.3, 1994, pp.1582-1591

[IEE94c] IEEE Committee Report, “Expert System Requirements for Power System Restoration”, System Operation Subcommittee, *IEEE Transactions on Power Systems*, vol.9, no.3, 1994, pp.1592-1600

[IEE95a] IEEE Committee Report, “Protective System Issues During Restoration”, System Operation Subcommittee, *IEEE Transactions on Power Systems*, vol.10, no.3, 1995, pp.1492-1497

[IEE95b] IEEE Committee Report, “Nuclear Plant Requirements During Power System Restoration”, Power System Restoration Working Group, *IEEE Transactions on Power Systems*, vol.10, no.3, 1995, pp.1486-1491

[IEE96a] IEEE Committee Report, “System Restoration Guidelines: How to Set-Up, Conduct, and Evaluate a Drill”, a report by the Operator Training Working Group, *IEEE Transactions on Power Systems*, vol.11, no.3, 1996, pp.1619-1629

[IEE96b] IEEE Committee Report, “Power System Restoration Training - Questionnaire Results”, a report by the Power System Restoration Training Task Force, *IEEE Transactions on Power Systems*, vol.11, no.3, 1996, pp.1630-1635

[Jea88] C. Jeanbart, Y. Logeay, M. Musart, “EDF's Dispatcher Training Simulator”, presented at CIGRÉ, Paris, France, Aug 28-Sep 3 1988, pp.Paper 39-12

[Kaf81] R. J. Kafka, D. R. Penders, S. H. Bouchey, M. M. Adibi, “System Restoration Plan Development for a Metropolitan Electric System”, *IEEE Transactions on Power Apparatus and Systems*, vol.PAS-100, no.8, 1981, pp.3703-3713

[Kak88] Naoto Kakimoto, Masayoshi Emoto, Muneaki Hayashi, “An Application of Artificial Intelligence to Restoration of Bulk Power Systems”, presented at Symposium on Expert Systems Application to Power Systems (ESAP), Stockholm-Helsinki, August 22-26 1988, pp.8.17-8.23

[Kak91] Naoto Kakimoto, Bunpin Lin, Muneaki Hayashi, "Restoration of Power System from Complete Outage by Expert System", presented at Third Symposium on Expert Systems Application to Power Systems (ESAP), Tokyo-Kobe, Japan, April 1-5 1991, pp.262-269

[Kea87] R. Kearsley, "Restoration in Sweden and Experience Gained from the Blackout of 1983", *IEEE Transactions on Power Systems*, vol.PWRS-2, no.2, 1987, pp.422-428

[Kir91] Daniel Kirschen, Terry Volkmann, "Guiding a Power System Restoration with an Expert System", *IEEE Transactions on Power Systems*, vol.6, no.2, 1991, pp.558-566

[Kob94] T. Kobayashi, D. Moridera, S. Fukui, K Komai, "Verification of an Advanced Power System Restoration Support System Using an Operator Training Simulator", *IEEE Transactions on Power Systems*, vol.9, no.2, 1994, pp.707-713

[Koj89a] Y Kojima, S. Warashina, S. Nakamura, K. Matsumoto, "Development of a Guidance Method for Power System Restoration", *IEEE Transactions on Power Systems*, vol.4, no.3, 1989, pp.1219-1227

[Koj89b] Yukio Kojima, Shigeru Warashina., Masakazu Kato, Hadime Watanabe, "The Development of Power System Restoration Method for a Bulk Power System by Applying Knowledge Engineering Techniques", *IEEE Transactions on Power Systems*, vol.4, no.3, 1989, pp.1228-1235

[Kön94] Siegfried König, et al., "An Expert System in Training Power System Restoration at the Stadtwerke Duisburg AG", presented at International Conference on Intelligent Systems Applications to Power Systems (ISAP), Montpellier, France, September 1994, pp.631-638

[Kos94] Tatjana Kostić, "Dynamic Simulator as a Part of an Expert System for Power System Restoration", in *Faculty of Electrotechnics*. Belgrade: Belgrade University, 1994, pp. 138

[Kos96] Tatjana Kostić, Juan-José Alba, Alain Germond, "Optimization and Learning of Load Restoration Strategies", presented at Power Systems Computation Conference (PSCC), Dresden, Germany, August 19-23 1996, pp.276-284

[Kos97] Tatjana Kostić, Rachid Cherkaoui, Partick Pruvot, Alain Germond, "Decision Aid Function for Restoration of Transmission Power Systems: Conceptual Design and Real Time Considerations", presented at Conference on Power Industry Computer Applications (PICA), Columbus, Ohio, USA, May 11-16 1997, pp.112-118

[Kov96] Attila Kovács, Attila K. Mergl, "Generating Switching Sequences - A Genetic Algorithm Approach", presented at International Conference on Intelligent Systems Applications to Power Systems (ISAP), Orlando, Florida, USA, January 28 - February 2 1996, pp.380-384

[Kro89a] G. Krost, D. Rumpel, E. N'Guessan, "Design of a Network Restoration Expert System with Training Facilities", presented at Second Symposium on Expert Systems Application to Power Systems (ESAP), Seattle, USA, July 17-20 1989, pp.392-398

[Kro89b] G. Krost, D. Rumpel, "Network Restoration Expert System", presented at IFAC - Simposium on Control of Power Plants and Power Systems, Seoul, Korea 1989, pp.479-482

[Kro91] Gerhard Krost, Dieter Rumpel, "Process Database Link and Reason Maintenance of a Power System Restoration Expert System", presented at Third Symposium on Expert Systems Application to Power Systems (ESAP), Tokyo-Kobe, Japan, April 1-5 1991, pp.242-249

[Kur93] K. Kuroda, et al., "Verification of a knowledge-based restoration guidance system in a local dispatching centre", *Electrical Power & Energy Systems*, vol.15, no.3, 1993, pp.185-192

[Lat77] James R. Latimer, Ralph D. Masiello, "Design of a Dispatcher Training System", presented at Conference on Power Industry Computer Applications (PICA), Toronto, Ontario, Canada, May 24-27 1977, pp.87-92

[Lio95] K. L. Liou, C. C. Liu, R. F. Chu, "Tie Line Utilization During Power System Restoration", *IEEE Transactions on Power Systems*, vol.10, no.1, 1995, pp.192-199

[Liu88] Chen-Ching Liu, Seung Jae Lee, S. S. Venkata, "An Expert System Operational Aid for Restoration and Loss Reduction of Distribution Systems", *IEEE Transactions on Power Systems*, vol.3, no.2, 1988, pp.619-626

[Liu92] C. C. Liu, T. Dillon, “State-of-the-Art fo Expert System Applications to Power Systems”, *Electrical Power & Energy Systems*, vol.14, no.2/3, 1992, pp.86-96

[Liu93] Chen-Ching Liu, Kan-Lee Liou, Ron F. Chu, Arne T. Holen, “Generation Capability Dispatch for Bulk Power System Restoration: A Knowledge-Based Approach”, *IEEE Transactions on Power Systems*, vol.8, no.1, 1993, pp.316-325

[Mar84] E. Mariani, F. Mastroianni, V. Romano, “Field Experiences in Reenergization of Electrical Networks from Thermal and Hydro Units”, *IEEE Transactions on Power Apparatus and Systems*, vol.PAS-103, no.7, 1984, pp.1707-1713

[Mar91] Hemant Marathe, et al., “An On-Line Operational Expert System with Data Validation Capabilities”, *IEEE Transactions on Power Systems*, vol.6, no.2, 1991, pp.882-889

[Mat92] K. Matsumoto, T. Sakaguchi, R. J. Kafka, M. M. Adibi, “Knowledge-Based Systems as Operational Aids in Power System Restoration”, *Proceedings of the IEEE*, vol.80, no.5, 1992, pp.689-697

[Mir96] Vladimiro Miranda, Dipti Srinivasan, L. M. Proença, “Evolutionary Computation in Power Systems”, presented at Power Systems Computation Conference (PSCC), Dresden, Germany, August 19-23 1996, pp.25-40

[Miy94] Minoru Miyazaki, et al., “The Development of a Restoration Support System for Application to Large-scale Bulk Power Networks”, presented at International Conference on Intelligent Systems Applications to Power Systems (ISAP), Montpellier, France, September 1994, pp.717-724

[Mon92] E. Mondon, et al., “MARS: an Aid for Network Restoration after a Local Disturbance”, *IEEE Transactions on Power Systems*, vol.7, no.2, 1992, pp.850-855

[Mor87] G. Morin, “Service Restoration Following a Major Failure on the Hydro-Québec Power System”, *IEEE Transactions on Power Delivery*, vol.PWRD-2, no.2, 1987, pp.454-463

[Mor89] S. Moriguchi, et al., “A Large-Scale SCADA System with Real-Time Knowledge-Based Functions”, presented at Second Symposium on Expert Systems Application to Power Systems (ESAP), Seattle, USA, July, 17-20 1989, pp.21-27

[Nag91] Takashi Nagasawa, et al., “Development of Restoration Guidance System for Control Center”, presented at Third Symposium on Expert Systems Application to Power Systems (ESAP), Tokyo-Kobe, Japan, April 1-5 1991, pp.479-486

[Omm94] P. van Ommeren, P. van Son, D. Rumpel, A. Litzinger, “Training the Restoration of Electricity Supply Systems in the Netherlands”, *European Transactions on Electrical Power Engineering*, vol.4, no.6, 1994, pp.547-550

[Oya96] Tsutomu Oyama, “Restorative Planning of Power System Using Genetic Algorithm with Branch Exchange Method”, presented at International Conference on Intelligent Systems Applications to Power Systems (ISAP), Orlando, Florida, USA, January 28 - February 2 1996, pp.175-179

[Par95] Y.-M. Park, K.-H. Lee, “Application of expert system to power system restoration in local control center”, *Electrical Power & Energy Systems*, vol.17, no.6, 1995, pp.407-415

[Par97] Y.-M. Park, K.-H. Lee, “Application of Expert System to Power System Restoration in Sub-control Center”, *IEEE Transactions on Power Systems*, vol.12, no.2, 1997, pp.629-635

[Raf96] Mehdi Rafian, et al., “Evaluating a Restoration Tool Using Consolidated Edison's Training Simulator”, *IEEE Transactions on Power Systems*, vol.11, no.3, 1996, pp.1636-1642

[Ros94] H. B. Ross, N. Zhu, J. Giri, B. Kindel, “An AGC Implementation for System Islanding and Restoration Conditions”, *IEEE Transactions on Power Systems*, vol.9, no.3, 1994, pp.1399-1410

[Rum92a] D. Rumpel, G. Krost, T. Ader, “A Training Simulator with an Advising Expert System for Power System Restoration”, presented at IFAC - Simposium on Control of Power Plants and Power Systems, Munich, FRG, March 9-11 1992, pp.209-214

[Rum92b] Dieter Rumpel, Gerhard Krost, “Natural Language Interface and Database Issues in Applying Expert Systems to Power Systems”, *Proceedings of the IEEE*, vol.80, no.5, 1992, pp.758-764

[Sak83] T. Sakaguchi, K. Matsumoto, “Development of a Knowledge Based System for Power System Restoration”, *IEEE Transactions on Power Apparatus and Systems*, vol.PAS-102, no.2, 1983, pp.320-329

[Sed83] Robert Sedgewick, *Algorithms*: Addison-Wesley Publishing Company, Inc., 1983

[Shi91] Y. Shimakura, et al., “A Knowledge-Based Approach to Power System Operational Aid”, presented at Third Symposium on Expert Systems Application to Power Systems (ESAP), Tokyo-Kobe, Japan, April 1-5 1991, pp.270-276

[Shi92] K. Shimakura, et al., “A Knowledge-Based Method for Making Restoration Plan of Bulk Power System”, *IEEE Transactions on Power Systems*, vol.7, no.2, 1992, pp.914-920

[Shi94] Y. Shimakura, J. Inagaki, S. Fukui, S. Hori, “Knowledge-based approach for the determination of restorative operation procedures for bulk power systems”, *Electrical Power & Energy Systems*, vol.16, no.3, 1994, pp.183-190

[Shi96] Jun-ichi Shinohara, et al., “Knowledge-Based Behavior Interface: Its Application to Power Network Restoration Support System”, *IEEE Transactions on Power Systems*, vol.11, no.1, 1996, pp.383-389

[Sus86a] I. Susumago, et al., “Development of a Large-Scale Dispatcher Training Simulator and Training Results”, *IEEE Transactions on Power Systems*, vol.PWRS-1, no.2, 1986, pp.67-75

[Sus86b] I. Susumago, M. Suzuki, T. Tsuji, K. Dan, “Mise au point d'une méthode de simulation en temps réel de la remise en service du système de production-transport après de grosses perturbations”, presented at CIGRÉ, Paris, France, Aug 27 - Sep 4 1986, pp.paper 38-03

[Tak88] I. Takeyasu, et al., “An Expert System for Fault Analysis and Restoration of Trunk Line Power Systems”, presented at Symposium on Expert Systems Application to Power Systems (ESAP), Stockholm-Helsinki, August 22-26 1988, pp.8.24-8.31

[Tan92] H. Tanaka, et al., “Research and Development on Expert Systems Applications to Power Systems in Japan”, *Electrical Power & Energy Systems*, vol.14, no.2/3, 1992, pp.71-85

[Tom87] K. Tomsovic, C.-C. Liu, P. Ackerman, S. Pope, “An Expert System as a Dispatchers' Aid for the Isolation of Line Section Faults”, *IEEE Transactions on Power Delivery*, vol.PWRD-2, no.3, 1987, pp.736-743

[Wad89] M. Wada, et al., “An Expert System Applied to the Distribution Automation System”, presented at Second Symposium on Expert Systems Application to Power Systems (ESAP), Seattle, USA, July 17-20 1989, pp.50-56

[Wal86] K. Walve, “Modélisation de composants de réseaux soumis à de fortes perturbations”, presented at CIGRÉ, Paris, France, Aug 27 - Sep 4 1986, pp.Paper 38-18

[Wel92] E. Welfonder, W. Sattinger, H. P. Asal, H. Weber, “Control Behaviour of Part Power Systems During Restoration After Blackouts”, presented at IFAC - Simposium on Control of Power Plants and Power Systems, Munich, FRG, March 9-11 1992, pp.263-269

[Wol87] B. F. Wollenberg, T. Sakaguchi, “Artificial Intelligence in Power System Operations”, *Proceedings of the IEEE*, vol.75, no.12, 1987, pp.1678-1685

[Wu97] Jaw-Shyang Wu, Chen-Ching Liu, Kan-Lee Liou, Ron F. Chu, “A Petri Net Algorithm for Scheduling of Generic Restoration Actions”, *IEEE Transactions on Power Systems*, vol.12, no.1, 1997, pp.69-76

CURRICULUM VITAE

Tatjana KOSTIĆ, born July 5, 1966, in Podgorica, Montenegro, Federal Republic of Yugoslavia

Education

October 1989
- Dipl.El.-Ing. degree in electrical engineering received from Belgrade University

June 1994
- M.Sc.Techn. degree received from Belgrade University

Professional Experience

October 91 -
- Staff Research Engineer at Electric Power Systems Laboratory (LRE), Swiss Federal Institute of Technology (EPFL)
- Responsible for the exercises of "Electrotechnics" for the 1st year Electrical Engineering and Computer Science students

October 91 - April 94
- Work on the project "Dynamic Simulator as a Part of an Expert System for Power System Restoration" with a Swiss Government grant until January 93, and financing of the Swiss National Foundation for Energy Research (NEFF) thereafter.

April 94 -
- Work on the project "Decision Aid Function for Restoration of Transmission Power Systems After a Blackout", co-financed by Swiss National Foundation for Energy Research (NEFF) and the R&D Division of Électricité de France (DER-EDF)

February / March 95
- Staff exchange in the framework of the European Programme COMETT at the R&D Division of Électricité de France (DER-EDF)

Publications

[1] T. Kostić, J.-J. Alba, A. J. Germond, "Optimization and Learning of Load Restoration Strategies", presented at the 12th Power Systems Computation Conference (PSCC), August 1996, Dresden, Germany, pp.276-284

[2] T. Kostić, R.Cherkaoui, P.Pruvot, A. J. Germond, “Decision Aid Function for Restoration of Transmission Power Systems: Conceptual Design and Real Time Considerations", presented at the 20th IEEE Conference on Power Industry Computer Applications (PICA), May 11-16, 1997, Columbus, Ohio, USA, pp.112-118

Languages

Serbian (mother tongue); French, English and Spanish (oral and written); Japanese (learning oral and written)