

# Chapter 6

## SPS Design Procedure and Reliability Evaluation

SPS is often perceived to be highly reliable because significant redundancy is typically considered in the design of an SPS. Diagnostic and self-check features to detect and alarm when essential components fail or critical functions are not operational are often used. In the U.S., standards exist to require that each SPS owner has an SPS maintenance and testing program and that the design of SPS both in terms of circuitry and physical arrangement should facilitate periodic testing and maintenance [6.1].

More specific requirements for SPS reliability can be found in [6.2-6.9]. For example, *WSCC Reliability Criteria for Transmission System Planning* [6.3] requires that systems which rely on SPS to meet the performance levels specified by these criteria must ensure that the SPS are highly reliable. System studies shall be conducted to assess the consequence of SPS failure unless it has been satisfactorily demonstrated the SPS failure is not credible. Therefore, failure of planned or existing SPS will only be studied if the SPS failure is considered credible or if the credibility of SPS failure has not been established. The credibility of failure of an existing SPS may be demonstrated by actual operating performance.

Most of today's requirements are qualitative, rather than quantitative. This is as it should be, since reliability criteria must be broad enough to capture a wide range of situations. However, there is presently little guidance in the industry to develop, study and assess, and maintain SPS reliability. For example, there are formal methods for identifying failure modes in SPS, and for quantifying their reliability, yet very few utilities use them as indicated by industry survey results summarised in [6.10]. In fact, in [6.10], over 70 percent of utilities indicated that they had no reliability model and made no reliability computations. A majority of the responses indicated that verification of reliability was done via system monitoring with subsequent adjustments. Yet, in this approach, dependability assessment requires an operation, and since SPS are normally dormant systems, this approach is not very effective in ensuring dependability. Therefore, it is important to develop and use formal methods to model, assess and study SPS reliability. Our objective in this chapter is to further motivate the need for SPS reliability analysis and to provide a structured framework for doing it. A secondary objective is to identify some specific analysis techniques that can be used within this framework.

In the next section, the classification of SPS failures is provided and some specific instances of SPS failure are described. Section 6.2 describes a number of technical standards useful in developing SPS design procedures. In Section 6.3, a framework for SPS design, adapted from these standards, is provided and some techniques useful in SPS reliability assessment are summarised. An illustration is provided in the Appendix.

### 6.1 Examples of SPS Failure

An SPS event can be classified into one of the following categories:

- desirable operation,
- undesirable operation, or
- failure to operate.

An SPS operation may be desirable or undesirable, depending on the consequence of the operation relative to the consequence had the SPS not operated. If the consequence of the operation is less severe than the consequence had the SPS not operated, the operation is desirable. This is case, for example, when the action of a generation rejection scheme trips one out of three units following a disturbance when otherwise, all three units would have lost synchronism.

If the consequence of the operation is more severe than the consequence had the SPS not operated, the operation is undesirable. Undesirable operation may either be unintended, due to a hardware, software, or human error, or it can be intended (according to the design), but still undesirable due to a fault in the design logic. A nuisance operation, when an SPS takes unnecessary action when there is no disturbance in the system, is an example of an undesirable, unintended operation. An example of an undesirable, intended operation is when a generator rejection scheme operates and trips a unit following a disturbance for which it was designed to operate, but had the SPS not operated, the plant would have been stable. This situation can occur if the disturbance is single phase to ground fault and the design criteria is based only on three phase faults.

An SPS failure to operate occurs when the SPS fails to respond as designed to conditions for which the SPS is supposed to operate. An SPS may fail to operate as expected for several reasons, among which are:

- hardware failure,
- faulty design logic,
- software failure, or
- human error.

Hardware failure occurs when some physical stress exceeds the capability of one or more installed components. Faulty design logic may occur as a result of inappropriate or incomplete study procedure during the design. Software failure results from errors in vendor written and user written embedded, application, and utility software. The vendor software typically includes the operating system, I/O routines, diagnostics, application oriented functions and programming languages. User written software failure results from errors in the application program, diagnostics, and user interface routines. Human errors can be classified according to whether they are associated with construction, operation, or maintenance.

When correctly operating, SPS significantly improve system response following a contingency. However, the failure of SPS to accurately detect the defined conditions, or the failure to carry out the required pre-planned remedial action, can lead to serious and costly consequences. The survey by IEEE-CIGRE [6.10] in 1992 suggests that the cost of SPS failure can be very high as most of the respondents selected the highest cost category when asked to estimate the cost of an operational failure of SPS.

Here, the U.S. NERC *System Disturbance Reports* from 1986-1998 [6.11] have been reviewed. Of the 30 cases that involved the operation of SPS, 21 were reported as successful operation of SPS, while 9 involved operational failures<sup>1</sup>. The reasons for these failure cases include flaw in logic design, software failure, hardware failure, incorrect setting, and inadvertent failure to arm. The following are brief descriptions of these failure cases:

---

<sup>1</sup> In the U.S., all disturbances which result in significant loss of load must be reported to the U.S. Department of Energy's (DOE) Emergency Operation Center (EOC) (See Appendix A of the 1995 NERC Report on System Disturbances). However, only a selected few of these disturbances are described in the reports. Therefore, 9 failures out of 30 occurrences should not be interpreted as a statistic representing SPS reliability in the U.S., as it is likely that there were more occurrences and more failures during the time period of interest.

WSCC - Northeast/Southeast Separation Scheme - April 4, 1988:

Scheme: System separation.  
Reason: Flaw in design (the scheme was susceptible to misoperation due to short bursts of communications circuit noise).  
Consequence: 1902 MW of generation was lost and 253 MW of load was interrupted.  
Lessons learned: Faulty design logic.

NPCC - Hydro-Québec - April 18-19, 1988:

Scheme: Load rejection.  
Reason: Hardware failure.  
Consequence: Systemwide blackout.  
Lessons learned: Hardware failure.

NPCC - Hydro-Québec - November 15, 1988:

Scheme: Load rejection.  
Reason: Hardware failure.  
Consequence: 3950 MW of load was interrupted.  
Lessons learned: Hardware failure.

WSCC-British Columbia Hydro/TransAlta Separation - January 7, 1990:

Scheme: Controlled opening of lines.  
Reason: Not armed (inadvertently).  
Consequence: It caused 230 kV Cranbrook-Nelway circuit to trip on the subsequent swing and resulted in separation (islanding) of the eastern part of the BCHA/TAUC system from the Interconnection.  
Lessons learned: Human error.

WSCC-Garrison - Taft 500 kV No.1 and 2 outages - January 8, 1990:

Scheme: Var Compensation (trip two 500 kV bus reactors).  
Reason: Flaw in the logic design.  
Consequence: It caused the unnecessary dropping of generation at Hauser, Morony, and Ryan (119 MW) as well as the loss of customer load (25 MW) in Helena.  
Lessons learned: Faulty design logic.

WSCC-SE Idaho/SW Wyoming Outage - September 12, 1991:

Scheme: Generator rejection.  
Reason: Hardware failure (telemetry that automatically arms this scheme was out of calibration).  
Consequence: It caused the loss of a second 345 kV line which led to further loss of transmission by overload and out of step conditions.  
Lessons learned: Hardware failure.

WSCC-Pacific AC Intertie Separation - November 17, 1991:

Scheme: System separation.  
Reason: Software failure in PG&E SPS programmable logic controller caused the delay in initiating remedial actions (also maybe hardware failure).

Consequence: Fail to separate WSCC system into two islands, but did not produce any severe problems (it was expected that there would be load lost and out-of-step conditions).

Lessons learned: Software failure and/or hardware failure.

WSCC-Minnesota - Wisconsin Interface 69 kV conductor burn down - October 13, 1992:

Scheme: Controlled opening of lines.

Reason: Incorrect setting.

Consequence: Two 69 kV lines in the Northern States Power and Dairyland Power Cooperative service burned open causing the lines to fall to ground and trip out.

Lessons learned: Human error.

MAPP & MAIN - Eastern MAPP-Western MAIN Interface Separation -November 6, 1997:

Scheme: Controlled opening of lines.

Reason: Flaw in design (opened the circuit at an ampere level below its setting, possibly due to an unbalanced load.).

Consequence: Resulted in low voltages in the south-western Wisconsin, eastern Iowa and western Illinois (Cordova), heavy loading of parallel, lower voltage transmission systems, and a large phase angle across the open tie at Arpin.

Lessons learned: Faulty design logic.

## **6.2 Existing ISA and IEC Standards**

It is likely that individual companies have documented procedures for performing SPS design, installation, and start-up. However, the effort fails to identify documentation of these procedures in the literature or in publicly available documentation, available on the Internet and elsewhere, with just a few exceptions, including [6.4-6.8]. Yet, most of these are quite general and tend more towards «criteria» rather than «procedures». Indeed, the 1993 IEEE/CIGRE survey conducted by Anderson and LeReverend [6.10] found that most often utility criteria for SPS contained at most general requirements for equipment redundancy.

It is found, however, that the other industries have confronted quite similar problems. One of them in particular is the process control industry. This industry is comprised of companies in the petroleum, pharmaceutical, power, chemical, pulp and paper, and textile, and supporting areas. Often, the failure consequence of the various processes implemented can be very high, and so a great deal of attention is paid to standardising procedures for designing, installing, and maintaining safety instrumented systems (SIS). These are systems that are comprised of sensors, relays, breakers, communication equipment, and logic solvers that «take the process to a safe state when predetermined conditions are violated» [6.12]. The SIS equipment and function are quite similar to the equipment and function of SPS in power systems.

The procedures described in this document to be utilised in the design stage of SPS are borrowed heavily from three SIS standards. These standards include ISA SA84.01-1996 (including ISA dTR84.02), IEC 61508, and IEC 61511. ISA SA84.01-1996 [6.12] addresses the application of SIS for the process industries, including integrity levels for electrical (E), electronic (E), and programmable electronic (PE) systems. These systems include electromechanical relays, solid state logic, programmable electronic systems, motor-driven timers, solid state relays and timers, hard-wired logic, and combinations of the above. ISA-

dTR84.02 [6.13] is a supporting document for ISA SA84.01 that provides evaluation approaches for SIS reliability. The focus of this document is on modelling and calculation. IEC 61508 [6.14] provides definitions, requirements, and methods of assessing functional safety integrity levels of E/E/PE safety-related systems. It is generally based and applicable to all E/E/PE safety-related systems irrespective of the application. IEC draft 61511 [6.15] is a process industry sector implementation of IEC61508. It is primarily concerned with safety instrumented systems for the process industry sector. It provides general framework, definitions and requirements, and guidelines on the application of hazard and risk analysis.

There are three basic ideas on which these ISA and IEC materials depend. One is that the potential harm or danger can be measured by *risk*, which is the combination (usually the product of) the probability of occurrence of the harm and the severity of the harm. A second basic idea is that the safety instrumented functions, which mitigate or prevent the harm and are therefore much like SPS, can be characterised by their *safety integrity*. This is the probability of a safety instrumented function satisfactorily performing the required functions under all the stated conditions within a stated period of time. In the cited standards, safety integrity is quantified by a *safety integrity level* (SIL). The SIL is a discrete number, 1, 2, 3, or 4, which specifies the requirements of the safety instrumented functions to be allocated to the safety instrumented systems. SIL 4 has the highest level of safety integrity, and SIL 1 has the lowest level. Each SIL has associated target failure measures, according to whether the mode of operation is low demand operation where frequency of demand for operation is not more than once per year or high demand operation where this frequency is greater than once per year. For low demand operation, the average probability of failure to perform the design function on demand should lie in the range:  $10^{-4}$  to  $10^{-5}$  (SIL 4),  $10^{-3}$  to  $10^{-4}$  (SIL 3),  $10^{-2}$  to  $10^{-3}$  (SIL 2), and  $10^{-1}$  to  $10^{-2}$  (SIL 1). For high demand operation, the probability of a dangerous failure per hour should lie in the range:  $10^{-8}$  to  $10^{-9}$  (SIL 4),  $10^{-7}$  to  $10^{-8}$  (SIL 3),  $10^{-6}$  to  $10^{-7}$  (SIL 2), and  $10^{-5}$  to  $10^{-6}$  (SIL 1). The third basic idea embedded in these documents is that risk and SIL are keys in showing how the establishment and maintenance of safety-instrumented system integrity involves many activities over the lifetime of the equipment. This idea is captured via use of the term *safety life cycle*, the necessary activities involved in the implementation of safety instrumented function(s) occurring during a period of time that starts at the concept phase of a project and finishes when all of the safety instrumented functions are no longer available for use.

### 6.3 SPS Step-by-Step Development

Some procedures recommended in the documents described above for use in guiding SPS development are adapted. The adapted procedure consists of six main steps:

- 1) Identify specifications and logic design.
- 2) Perform SPS hardware and software design
- 3) Perform SPS installation, commissioning, and pre-start-up tests.
- 4) Establish operating and maintenance procedures.
- 5) Perform pre-start-up safety review.
- 6) Perform SPS review and modification.

Some guidelines for each of these steps are provided in the following subsections.

This procedure and the guidelines provided for the various steps should not be used as a self-contained guide for SPS development nor should it be viewed as comprehensive and complete. Rather, it serves to outline in broad and general terms some of the basic issues associated with SPS development with the emphasis on reliability. Individual designers might

use this as a basis for which to write their own SPS development guidelines. In doing this, it is strongly recommended reviewing the above described standards.

### 6.3.1 SPS specifications and logic design

The objective of this step is to design the logic and identify the specifications for the SPS. This step should result in the following information:

- SPS specification:
  - Type of SPS to be employed and its intended function.
  - Location within the system where the SPS will be installed.
  - System reliability criteria necessitating employment of the SPS.
  - SPS performance requirements for which the SPS is required to meet.
- Logic design:
  - Events and conditions for which the SPS will operate.
  - Events and conditions for which the SPS will not operate.
  - The consequence resulting from a failure of the SPS.

#### SPS specification:

For most SPS operating today, SPS specification was accomplished via studies carried out by an experienced engineer familiar with the network. Typically, the engineer becomes aware of a particular disturbance for which the power system does not meet the reliability criteria, either for some expected conditions, or for conditions that would be economically attractive if the disturbance-performance criteria could be satisfied. Most often, this awareness comes about as a result of simulation studies. One approach, suggested in [6.16], is to automate contingency analysis in a way that randomly simulates a large number of credible disturbances under various operating conditions and flags contingencies that do not satisfy disturbance-performance criteria. These contingencies are then candidates for further study by the engineer.

Subsequent to identification of a contingency in violation of disturbance-performance criteria, the engineer begins to study the problem, probing different solution strategies until one or more feasible ones become apparent. It is assumed that at least one of the identified feasible strategies include adoption of an SPS. At this stage, the engineer normally performs some rough economic analysis to determine which of the solution strategies are most economic. If this analysis identifies the strategy including SPS adoption as most economic, then a document describing all of the information prescribed in the SPS specification should be prepared. The information pertaining to the first three items in the SPS specification is normally available at this point from the studies already completed. In addition, the engineer should identify any specific performance requirements associated with the type of SPS under consideration.

#### Initial logic design:

The logic design may be interpreted as a search for a rule. A high-level statement of this rule is:

- IF{armed  $\cap$  activated} THEN{actions}, or IF{A  $\cap$  B} THEN {C }.

There are three information requirements regarding the rule:

- determination of the arming condition, A;
- determination of the activation conditions, B; and
- identification of the actions to take, C.

The *arming* conditions are typically specified based on one or more operational parameters in the network, e.g., load, generation, flow, or voltage level. The *activation* conditions are

typically specified based on detection of one or more events or responses in the network. Detection of a circuit outage is the most common event-based activation. Response-based detection may include underfrequency, generator over-speed, and others. *Actions* are typically specified in terms of network changes such as generator tripping, load rejection, capacitor insertion, etc., but they may as well take the form of controller setting changes as in the case of excitation boosting.

The rule may be quite simple, e.g., IF{ $A \cap B$ } THEN {C}, or it may be more complex, e.g., IF{ $A \cap B1$ } THEN {C1} OR IF { $A \cap B2$ } THEN {C2}. More complex rules are clearly possible. The designer should understand that increased rule complexity usually comes at the expense of an increased number of SPS failure modes, not only in the logic itself, but also in the interaction of the logic with the logic of other SPS logic, and in the hardware and software necessary to implement the rule. Increased rule complexity almost always results in decreased reliability performance level and therefore, if necessary, provides incentive to consider other solution strategies.

The study work associated with SPS specification, described above, typically provides enough information so that the engineer can provide the initial logic design. However, the integrity of the rule is paramount. Past efforts at validating SPS rules have relied on engineering experience and judgement, coupled with tedious trial and error testing involving manual computer simulations. This is still the case, and the essential contribution of human judgement in this process should be fully embraced. Yet, today, there is both motive and method to improve upon this approach. The *motive* is that there exist greater risk in the use of SPS due to: their proliferation; the increased variability of network conditions and associated uncertainty in SPS logic integrity; and the increased number of network participants resulting in increased financial exposure to an SPS owner should an SPS fail. The *method* lies in the power of the computer and its ability to perform large numbers of simulations of system conditions so as to test and refine the integrity of the rule. Additional insight into this method is provided next.

#### **Logic validation and refinement:**

It is desired to probe the integrity of a specified SPS rule using computer simulation and refine the rule when problems are observed. The following is a procedure for doing this:

- 1) *Simulation tool and network model*: Identify the appropriate simulation tool and the corresponding model. The simulation tool could be a power flow program, or, more likely, it could be a short-term time domain simulation program, or a mid- to long-term time domain simulation package, or some other analysis tool. The model should include:
    - the network of interest;
    - the ability to simulate the activating event;
    - the model of the SPS: its proposed arming, activating, and action logic; and
    - models of all components necessary to the study of the SPS logic integrity: this might include other protection and control equipment and especially other SPS that could be activated as a result of the identified initiating event.
  - 2) *Simulation*: Use computer automation to test the integrity of the rule over a large number of operating conditions, and for a defined number of events. This step should create a large database, with each record providing the operating conditions, the events, and the performance measures characterising the acceptability of the system response.
  - 3) *Analysis*: Analyse the resulting database; modify and refine the rule as needed.
  - 4) *Stopping*: If the rule has been modified, repeat steps 2 and 3. Otherwise, stop.
- Steps 2 and 3 require some further elaboration, provided in what follows. Additional information about SPS simulation and analysis can be found in Chapter 5.

**Simulation:**

The simulation procedure should provide that the SPS logic is tested for a wide range of conditions and events, including those that result in SPS activation and those that do not. This procedure should include intelligence for updating or modifying the operating conditions and the events for each successive simulation. This can be done in a variety of ways, but some randomisation in the selection is typically appropriate. The simulation procedure should also include the ability to store various parameters characterising the operating conditions and various performance measures characterising the acceptability of each simulation. One approach to security assessment automation, described in [6.17], is based on Latin Hypercube sampling, also called *structured Monte Carlo sampling*. Here, the operating range of interest, as defined by n-operational parameters, forms an n-dimensional hyperspace. This hyperspace is segregated into hypercubes. The simulation procedure then steps through each hypercube and performs a simulation for each one. The specific operating condition to be used in the simulation is chosen at random within the designated hypercube. The granularity (i.e. the number) of the hypercubes determines how many simulations are done for each automated security assessment run. The hypercube step-through procedure results in a uniform sampling of operating conditions throughout the operating range. The random selection within each hypercube maximises the number of different values chosen for each operating parameter over the entire simulation run.

**Analysis:**

The analysis of the database resulting from the automatic security assessment run is done with appropriate software. A minimal analysis would pick out the simulations that result in unacceptable system response. More advanced software would provide the ability to learn from the database. A statistical approach has been proposed in [6.18, 6.19]. Machine learning techniques using decision trees have been suggested for similar purposes [6.20, 6.21]; these techniques are attractive as they are also capable of automatically deriving and refining the rule.

### 6.3.2 Hardware and software design

The objective of this step is to perform the SPS hardware and software design based on the rule obtained from the procedures described in the introduction to this chapter. Some guidelines useful in performing this design are provided in what follows. These guidelines are focused on enhancing reliability and are intended to complement those provided in Chapter 4. First, some good practices for general use in hardware and software design are presented. Then, a design refinement that can be used to enhance SPS reliability is described. Finally, the issues about software and human reliability assessment are addressed.

**SPS components - good practices:**

The following «good practices» are oriented towards enhancing SPS reliability and should be used together with the design criteria provided in Chapter 4. These were extracted from industry references [6.4-6.6].

*Logic solver:* The logic solver is that portion of an SPS that performs one or more logic functions used to execute the SPS application logic and initiate protective actions. Although it may be electrical or electronic, it is assumed in what follows that it is a programmable electronic (PE) system such as a microprocessor [6.22], micro-controller, programmable logic controller (PLC), or application-specific integrated circuit (ASIC). If the logic solver is purchased externally, the supplier should provide an integrated design including input module(s), output module(s), maintenance interface device(s), communication(s), and utility software. The logic solver should have a published mean time to failure (MTTF), unsafe

failure mode listing, and frequency of unsafe failure mode. It should have a method (internal and/or external) to protect against covert faults (such as a “watchdog” timer). The logic solver should be designed to ensure the process will not restart automatically when power is restored, unless it is required to do so. Detected failure of the logic solver should not result in an unsafe system condition, if the appropriate, documented, response action is undertaken.

*Logic solver software:* In developing software necessary for the logic solver, good software development practices should be followed. For example, a software requirements specification and a software design document should be developed. These documents should specify the functionality of the design using functional blocks so that the programmer does not need to make any assumptions about the functionality of each software module. Software architecture should be clearly identified including specification of the operating system, databases, input/output subsystems, communication subsystems, programming and diagnostic tools, and programming languages used. Coding standards should specify good programming practices (e.g., readability, traceability, checkability, and analysability), proscribe unsafe language features, and specify procedures for source code documentation. Testing plans should be able to show that each individual software module, each software subsystem, and the entire software system performs their intended functions and does not perform any unintended functions.

*Sensors:* The sensors are devices that measure the power system condition. Generally, they include relays and breaker/switch status detectors. Relays may be current, voltage, power, frequency, rate of change of each of these, out-of-step, generator power output level, line loading power level, etc. Neither loss of current nor loss of power can be used alone to determine that a line is open, because they both go through zero as power flow reverses direction on the line. Caution must be taken in determining settings required to distinguish between local faults or system problems by using rate of change of current, voltage, power, or frequency. Out-of-step relays may be used in some cases for detecting pending instabilities. However, these are usually applied only where it is acceptable to wait until the swing is “coming out of” the swing setting of the relay before taking corrective action. Studies must be performed to determine the proper setting to prevent out-of-step tripping on recoverable swings.

*Communication equipment:* This equipment communicates the power system conditions as measured by the sensors to the logic solver and the logic solver output to the final or actuating elements. When using communication channels to provide remote logic indication, the design logic should be such that loss of a signal, channel noise, or failure of the communication channel does not give wrong logic information at the receiving end. It should be designed to ensure correct and adequate signal transmission during large system disturbances. Channel equipment used in SPS should be dual channel, shift up/shift down type. All communication equipment and channel equipment should be monitored and alarmed to the appropriate dispatch or maintenance centre so that appropriate action may take place upon failure. Communication channels should be well labelled or identified in some manner such that personnel working on a channel can readily identify the proper circuit.

*Power supplies:* No single battery or DC power supply failure should prevent the SPS from performing its intended function. Each battery should be provided with its own charger. The regulation of the dc voltage should be such that, under all possible charging and loading conditions, voltage within acceptable limits is supplied to all devices. DC systems should be monitored to detect abnormal voltage levels, DC grounds, and loss of AC supply to the battery chargers.

*Monitoring devices:* Each SPS should have instrumentation to monitor and analyse the SPS operations. For example, data logging with sequence of events timing can be designed into the SPS at many points. Such data logging assists in the accurate analysis of any operation in the overall scheme, either correct or incorrect.

**Design refinement for enhancing reliability:**

This section draws on the material provided in the ISA and IEC standards [6.12-6.15], particularly IEC 61508 [6.14]. It is assumed that, based on identification of the rule, an initial SPS design has been completed. The purpose of the design refinement procedure is to enhance the SPS reliability. This procedure consists of repeating the following basic steps until the desired performance level is reached: 1) evaluate the design, 2) assess the evaluation, and 3) modify the design.

1) *Evaluate the design:* This is a two-step procedure. The first step is to perform a failure analysis of the design in order to identify the possible failure modes, their initiating events, and the possible consequences of these failures if they should occur. This step is generally qualitative but may also include quantitative analysis if desired. Different methods are available for performing it, and a few of them are summarised below [6.14]. Additional information regarding these and other methods can be found in [6.14] and the references contained therein. Often, the best results are achieved by applying more than one method.

- Hazard and Operability (HAZOP): Here, a team of engineers performs a structured examination of a design. A leader drives the procedure by presenting each part of the system in connection with several guide words. Every applied condition or failure mode is considered for its feasibility, how it could arise, the possible consequences, and how it could be avoided.
- Failure Modes and Effects Analysis (FMEA): This is a «bottom-up» method that starts with a detailed list of all components in the system. An entire system can be analysed one component at a time. Alternatively, the system can be hierarchically divided into subsystems and modules as required. The FMEA technique is generally poor at identifying combinations of failures that cause critical problems. Since each component is reviewed individually, failures due to combination of components are not addressed. Common cause failures are rarely identified since they require more than one component failure. An extension of this method, called Failure Modes, Effects, and Criticality Analysis (FMECA), results in a criticality ranking of all components.
- Common Cause Failure Analysis: This method determines the potential failures in multiple systems/sub-systems that would undermine the benefits of redundancy, because of the appearance of the same failures in the multiple parts at the same time. For example, if a system is installed in a single room, an air-conditioner failure might reduce the benefits of redundancy.
- Event Tree Analysis: This method is a «bottom-up» approach. It begins with the determination of a «bottom event», which is a basic or initiating event and aims to determine the possible consequences of the event. Possible consequences of this event are described in a tree fashion using logical operators. Intermediate consequences are similarly analysed, and so on, until a terminal event is reached. A terminal event is one that has no other immediate consequences.
- Fault Tree Analysis: Fault tree analysis is a «top-down» approach. It begins with the determination of a «top event», which is the immediate cause of a serious consequence. Combinations of causes for this top event are described in a tree fashion using logical operators. Intermediate causes are similarly analysed, and so on, back to «basic» or initiating events, where the analysis is then repeated for other «top events». The method is good at finding combinations of failures that may cause problems.

- Cause-Consequence Analysis: This technique can be regarded as a combination of fault tree and event tree analysis. Starting from a critical event, a cause consequence graph is traced backwards using fault tree construction techniques and forwards using event tree construction techniques.

Recognising that HAZOP and FMEA are inductive methods, and common cause, event tree, fault tree, and cause-consequence are deductive methods, it is often effective to use at least one of each to gain complementary benefits. A simple illustration of hardware reliability assessment using FMEA and corresponding Markov modelling for a Generation Rejection Scheme (GRS) is presented in the Appendix.

The second step in the design evaluation is to perform quantitative failure analysis. This step may not be required for low complexity SPS, but it should always at least be considered. The deductive methods of failure mode identification are amenable to quantitative analysis, but these methods are not capable of modelling time varying effects of failures throughout the system life cycle. Therefore, Markov models are usually considered more rigorous. In Markov modelling [6.23], the status of the system with regard to its identified failure states, modelled as nodes, and the failure or repair events, modelled as node interconnects weighted by failure or repair rates, is represented as a graph. With time modelled in discrete increments (for example, once per hour), calculations can be made showing the probability of being in each state for each time interval. Markov modelling is applicable for systems for which at any given time the subsequent system state only depends on the state at the given time and is not affected by the state at any preceding time. It can be assumed that an SPS has this characteristic. Markov modelling is well suited for use in SPS reliability modelling because its flexibility provides that it can account for the variety of features that are common in SPS. Specifically, Markov modelling can incorporate independent and common cause failures, partial and full repairs, maintenance, and diagnostic coverage. Most importantly, it provides that all of these features can be modelled as a function of time. This is in contrast to probability methods that provide steady state results and are accurate only for short repair times and low failure rates. When there is a large number of states, methods of state reduction can be applied to obtain state probabilities. In very complex models, the graph can be computer simulated to obtain state probabilities. This technique is suitable for modelling systems in which the level of redundancy varies with time due to component failure and repair.

2) *Assess the evaluation:* The result from the design evaluation should be assessed to determine whether the design is satisfactory or not. In this step, the result of the design evaluation is compared to a standard. As indicated at the beginning of Section 6.2, existing SPS design standards are too general to make this step very meaningful. More specific standards existing within individual companies could be identified and used for this purpose, if available. Alternatively, recommended approaches published in the ISA/IEC documents [6.12-6.15] could be adopted. One such approach is based on identification of four risk classes: I (intolerable), II (undesirable, and tolerable only if the cost of risk reduction grossly exceeds the improvement gained), III (tolerable if the cost of risk reduction exceeds the improvement gained), and IV (negligible). These risk classes are identified according to the combination of frequency (or probability) and consequence, as illustrated in Table 6.1.

Frequency	Consequence			
	Catastrophic	Critical	Marginal	Negligible
Frequent	I	I	I	II
Probable	I	I	II	III
Occasional	I	II	III	III
Remote	II	III	III	IV
Improbable	III	III	IV	IV
Incredible	IV	IV	IV	IV

**Table 6.1.** Identification of risk classes.

Numerical designations for the various levels of frequency and consequence would normally be used. If the result of the design evaluation indicates the SPS design falls into classes I, II, or III, then risk reduction is necessary, as described in below. Another attractive approach utilises a so-called risk graph [6.15-part 3]. A risk graph is a method of making decisions on a complex issue by breaking down the overall problem into a number of relevant issues which can then be considered on an individual basis. This approach results in a specified safety integrity level (SIL) appropriate to the situation under study, and then the design evaluation results can be compared against the performance measures corresponding to the specified SIL (see Section 6.2) to determine if further risk reduction is necessary.

3) *Modify the design:* Design modification is necessary if design evaluation described above results in an indication that risk reduction is required. Here, the first step is to identify the critical elements. These are components, subsystems, and/or parameters within the design that are responsible for unacceptable risk. The second step is to identify improvements corresponding to each identified critical element. These are typically measures to control failures and could include:

- Using more reliable components.
- Implementing additional defences against common mode failures.
- Increasing diagnostic coverage using, for example, comparison checks between parallel signals, designs which fail to a detectable state for specific failure modes (using normally energised modes), and test and/or cut-out switches at the initiating points, logic receiver or monitoring locations, and at sites where devices are to be operated or tripped.
- Increasing redundancy.
- Reducing proof of test interval.

Following selection of one or more design improvements, the design evaluation should be repeated.

**Software and human reliability assessment:**

The SPS design refinement approach described in the previous section is broad and inclusive. In particular, it is applicable to reliability problems introduced by software problems and also human errors. However, these are not familiar areas to many engineers, therefore additional comments about them are provided here.

*Software Reliability:* Most of the software typically used in SPS is not very complicated. But as shown in Section 6.1, software is one of the factors that cause SPS to fail. Software reliability concerns itself with how well the software functions meet the design requirements. Defined precisely, software reliability is the probability of failure-free operation of a computer program for a specified period of time in a specified environment. Software reliability can be improved by flawless design and error-free coding [6.24]. Studies have

shown that most software errors are introduced during specification development. Adopting formal methods to develop specifications may reduce this sort of errors.

A software reliability model (SRM) specifies the general form of the dependence of the failure process on the principle factors that affect it, namely fault introduction, fault removal, and the environment. A good SRM is simple, gives good predictions of future failure behaviour, and computes useful quantities. SRMs are mathematical models that represent failures as a random process that is characterised by either times of failures or number of failures at fixed times, which can be classified into four categories: *times between failures models*, *failure count models*, *fault seeding models*, and *input domain based models* [6.24, 6.25], described further in what follows:

- *Times between failures models* include models that provide an estimate of the times between failures in a software. The key assumptions of these models are
  - a) independent time between successive failures,
  - b) equal probability of exposure of each fault,
  - c) embedded faults are independent of each other, and
  - d) no new faults introduced during corrective actions.

The most common approach is to assume that the time between, say, the  $(i-1)^{th}$  and the  $i^{th}$  failures, follows a distribution whose parameters depend on the number of faults remaining in the program during the interval. Estimation of the parameters are obtained from observed values of times between failures and estimates of software reliability, mean time to next failure, etc., are then obtained from the fitted model.
- *Failure count models* include models that estimate the number of faults or failures experienced in a specific time. The key assumptions are
  - a) test intervals are independent of each other,
  - b) testing intervals are homogeneous distributed, and
  - c) number of faults detected during different intervals are independent of each other.

Parameters of failure rate can be estimated from observed values of failure counts or from failure times. Estimates of the software reliability, mean time to next failure, etc., can again be obtained from the relevant equations.
- *Fault seeding models* include models that assess the number of faults in the software at time zero via seeding extraneous faults. The key assumptions are
  - a) seed faults are randomly distributed in the software, and
  - b) indigenous and seeded faults have equal probabilities of being detected.

The basic approach taken here is to “seed” a known number of faults in a program which is assumed to have an unknown number of indigenous faults. The program is tested and the observed number of seeded and indigenous faults are counted. From these, an estimate of the fault content of the program prior to seeding is obtained and used to assess software reliability and other relevant measures.
- *Input domain based models* include models that assess the reliability of software when the test cases are sampled randomly from a well known operational distribution of software inputs. The key assumptions are
  - a) input profile distribution is known,
  - b) random testing is used (inputs are selected randomly), and
  - c) input domain can be partitioned into equivalence classes.

The basic approach taken here is to generate a set of test cases from an input distribution that is assumed to be representative of the operational usage of the program. Because of the difficulty in obtaining this distribution, the input domain is partitioned into a set of equivalence classes, each of which is usually associated with a program path. An estimate

of program reliability is obtained from failures observed during physical or symbolic execution of the test cases sampled from the input domain.

*Human Reliability:* Human errors are an important factor to be considered in SPS reliability. Of the 8 SPS failures described in Section 6.1, at least 3 of them were partly due to human errors (No.1, 4, 8). Human error must be taken into account in order to obtain a precise and accurate measure of SPS reliability. Human errors can be categorised into several types as operating, design, construction, and maintenance [6.24, 6.26]. Operating error results from humans operating the equipment incorrectly such as motivational error, use of wrong procedures, failure to follow procedures, etc. Design error results from inadequate design. Construction error results from poor workmanship such as the use of an incorrect component or failure to follow the design. Maintenance errors result generally from wrong repair or installation. Sometimes design, construction, and maintenance errors are also the cause of operating error. Normally, quality assurance programs are designed and implemented to minimise the occurrence of these types of human error [6.24]. Several human reliability models have been developed [6.24, 6.26, 6.27], including simulation methods, expert judgement methods, and analytical methods.

### **6.3.3 Installation, commissioning, and pre-start-up test**

The objective of this step is to install the SPS and to ensure it is installed in accordance with the design and performs in accordance with the SPS reliability requirements. The commissioning step should include confirmation of proper equipment and wiring, operational energy sources, calibrated instruments, and operational logic solver. The pre-startup test should include confirmation that the SPS communication system (where required) is operational, that relays, logic and final control elements perform correctly in accordance with the design, that the SPS provides the proper operation display, and that manual shutdown systems operate properly.

### **6.3.4 Operating and maintenance procedures**

The objective of this step is to ensure that the SPS functions are in accordance with SPS reliability requirements developed throughout the SPS operational life. The SPS operating and maintenance procedures should be available to the technicians and engineers responsible for the operation and maintenance of the SPS facilities. Improving the familiarity of these well-written procedures for these individuals is an effective means to avoid human error and prevent subsequent degradation of reliability.

The SPS operating procedures should be written to explain the operating criteria of the SPS. These procedures are typically part of the bulk system operating procedures and should include:

- when and how the SPS is to be armed or disarmed,
- when the SPS will take action, and
- how the system operator is to respond to an SPS operation.

A maintenance program should be established which includes written procedures for maintaining, testing, and repairing the SPS. SPS maintenance should include:

- periodic functional testing,
- periodic preventive maintenance, and
- repair procedures for detected faults, with appropriate testing after repair.

The following items can be considered when developing a maintenance program:

- How often can the scheme be taken out of service without degrading the system?

- What schedule or power flow changes must be made in order to test the scheme?
- What is the length of time to test the scheme?
- What number of times is the scheme expected to operate during a given time interval?
- What is the potential exposure to a false or inadvertent operation caused by personnel during testing?

### 6.3.5 Pre-start-up safety review

The objective of this step is to provide the last safety review before the start-up of SPS. It should verify that:

- SPS was constructed and installed in accordance with the SPS reliability specification.
- Operating, maintenance and emergency procedures pertaining to the SPS are in place and are adequate.
- Adequate employee training has been completed.

### 6.3.6 Review and modification

The objective of this step is to ensure adequate management of SPS verification and modification. A review process should be conducted periodically or whenever there are major changes in operating practices or physical changes to the power system. The review should include review of:

- the suitability of the scheme by comparing the system conditions that motivated the original need for the scheme with the system conditions expected to be encountered in the near future; and
- the consequence resulting from a failure of the scheme.

If it is found during the review process that the SPS is no longer needed, the scheme should be disabled or removed. If it is found that some modifications of the existing scheme are necessary due to changes in expected operating conditions, then a full design evaluation should be considered.

## References of Chapter 6

- [6.1] North American Electric Reliability Council, *NERC Planning Standards* (draft), June, 1997.
- [6.2] Western Systems Coordinating Council, *WSCC Operating Procedures*, December, 1993.
- [6.3] Western Systems Coordinating Council, *Reliability Criteria*, March, 1997.
- [6.4] Western Systems Coordinating Council, *Guide for Remedial Action Schemes*, April, 1991.
- [6.5] Northeast Power Coordinating Council, *Special Protection System Guideline*, February, 1992.
- [6.6] Northeast Power Coordinating Council, *Bulk Power System Protection Criteria*, August, 1995.
- [6.7] Northeast Power Coordinating Council, *Maintenance Criteria for Bulk Power System Protection*, November, 1996.
- [6.8] Northeast Power Coordinating Council, *Procedure for Reporting and Reviewing Proposed Bulk Power System Protection*, September, 1996.
- [6.9] North American Electric Reliability Council, *Reliability Assessment 1997-2006*.
- [6.10] CIGRE WG 39.05, P. M. Anderson and B. LeReverend, et al., *Industry Experience with Special Protection Schemes*, *Électra*, No. 155, pp. 103-127, August, 1994.
- [6.11] North American Electric Reliability Council, *System Disturbances, 1986-1998*.
- [6.12] The Instrument Society of America (ISA), *Application of Safety Instrumented Systems for the Process Industries*, ISA S84.01, February, 1996.

## *System Protection Schemes in Power Networks*

- [6.13] The Instrument Society of America (ISA), *Electrical/Electronic/Programmable Electronic Systems for Use in Safety Applications – Safety Integrity Evaluation Techniques*, ISA-dTR84.02 (draft), September, 1997.
- [6.14] The International Electrotechnical Commission, *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, IEC-61508, Parts 1-7, April, 1998.
- [6.15] The International Electrotechnical Commission, *Functional Safety Instrumented Systems for the Process Industry Sector*, IEC-61511, Parts 1-3, June, 1999.
- [6.16] L. Wehenkel, C. Lebrevelec, M. Trotignon, J. Batut, *Probabilistic Design of Power System Special Stability Controls*, CPSPP'97, IFAC-CIGRE Symp. On Control of Power Systems and Power Plants, Beijing, August, 1997.
- [6.17] V. Van Acker, S. Wang, J. McCalley, G. Zhou, M. Mitchell, *Data Generation using Automated Security Assessment for Neural Network Training*, Proc. of the 1997 North American Power Symposium, pp. 142-148, Laramie, Wyoming, October, 1997.
- [6.18] P. Cholley, C. Lebrevelec, S. Vitet, M. de Pasquale, *A Statistical Approach to Assess Voltage Stability Limits*, Symposium Proceedings of the Bulk Power Systems Dynamics and Control IV: Restructuring, editors L. Fink and C. Vournas, pp. 219-224, Santorini, Greece, August, 1998..
- [6.19] C. Lebrevelec, Y. Schlumberger, M. de Pasquale, *An application of a risk based methodology for defining security rules against voltage collapse*, Proc. of the IEEE/PES SM, pp. 185-190, Edmonton, Canada, July, 1999.
- [6.20] L. Wehenkel, *Automatic Learning in Power Systems*, Kluwer Academic Publishers, Boston, 1997.
- [6.21] C. Olaru, P. Geurts, and L. Wehenkel, *Data Mining Tools and Applications in Power System Engineering*, Proc. of the 13<sup>th</sup> Power Systems Computation Conference, pp. 324-330, Trondheim, Norway, June, 1999.

- [6.22] P. C. K. Lau, M. Grover, and W. Tanaka, *Reliability Assessment of Special Protection Systems*, CIGRE paper AA-11, presented at the CIGRE Symposium on Electric Power System Reliability, Montreal, September, 1991.
- [6.23] J. McCalley and W. Fu, *Reliability of Special Protection Systems*, IEEE Trans. on Power Systems, Vol. 14, No. 4, pp.1400-1406, November, 1999.
- [6.24] M. Modarres, M. Kaminskiy, and V. Krivtsov, *Reliability Engineering and Risk Assessment: A Practical Guide*, Marcel Dekker, 1999.
- [6.25] Y. K. Malaiya and P. K. Srimani (Editors), *Software Reliability Models: Theoretical Developments, Evaluation and Applications*, IEEE Computer Society Press, 1990.
- [6.26] K. B. Misra, *New Trends in System Reliability Evaluation*, Elsevier, 1993.
- [6.27] B. S. Dhillon, *Mechanical Reliability: Theory, Models and Applications*, American Institute of Aeronautics and Astronautics, Washington DC, 1988.

