A COMPREHENSIVE ANALYSIS OF DISTRIBUTION AUTOMATION SYSTEMS

by

Liam Murphy and Felix Wu

Memorandum No. UCB/ERL M90/72

13 August 1990

COVER PAGE

A COMPREHENSIVE ANALYSIS OF DISTRIBUTION AUTOMATION SYSTEMS

by

Liam Murphy and Felix Wu

Memorandum No. UCB/ERL M90/72

13 August 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

Abstract

The Generation and Transmission systems of modern electric utilities are operated and controlled in real-time by EMS computers. By contrast, the Distribution system is usually not even monitored in real-time. A Distribution Automation System (DAS) is one suggestion to allow a utility to coordinate the operation of its Distribution system in real-time. In this report we consider the Distribution system as a network with microprocessor capability at the nodes. The functional components which could be part of a DAS are described. Proposed solutions to the functions on which research has been done are discussed, and the issues involved in implementing the remainder are explored. The various subsystems essential to the operation of a DAS - the communication, computation and control systems - are discussed and their interactions highlighted. In particular, recent advances in these areas which promise to be suitable for DAS applications, such as spread spectrum signalling, parallel processing, and expert systems, are outlined. The report concludes with some comments on the design decisions facing a utility considering Distribution Automation, and some performance measures which enable the utility to compare alternative designs are defined.

[1] Research supported by National Science Foundation Grant ECS-8715132

Contents

Chapter 1

Introduction

The objective of a power system is to supply load demand reliably and economically. In attempting to meet this objective, the power system operator must consider two main types of constraint : *load constraints* and *operating constraints*. Load constraints arise due to the requirement for the system to supply its customers with electric energy, while maintaining a balance between supply and demand. Operating constraints impose limits on voltage, current and other system quantities in order to operate the equipment safely and efficiently. The operation of a power system can be divided into three distinct states : **normal, emergency** and **restorative**. The system is in its normal state when both the load and operating constraints are met ; it is in an emergency state when the operating constraints are violated, and in a restorative state when only the load constraints are violated. Therefore there exists a need to improve economy and reliability of supply in the normal state, to remove violations of the operating constraints in the emergency state, and to remove violations of the load constraints in the restorative state. Note that these constraints are coupled, eg. if a line is overloaded then to remove the violation of its operating constraint we might consider tripping out the line, however this will lead to loss of load thus violating the load constraint.

In the generation and transmission systems of modern electric utilities, operators rely on Energy Management Systems (EMS) to coordinate the actions called for to meet these needs. Usually built around supercomputers or mainframe computers, EMS's monitor various quantities such as aggregate demand, voltage, generation, breaker status and line flows in real-time. Based on this data, an EMS performs many computational and control functions, among them :

- solving the economic dispatch and unit commitment problems
- minimising transmission losses eg. by volt/var control
- scheduling energy interchanges between utilities
- contingency analysis ie. predicting the effects of outages
- state estimation of non-telemetered quantities
- co-ordinating emergency responses eg. by alarm processing
- forecasting system and individual bus loads
- determining the network topology

The benefits of an EMS, both economically and in terms of better control of the system, are well-established : for instance, unit commitment programs achieve consistent fuel-cost reduction of 0.5 to 2 percent - up to $20 million on an annual billion-dollar fuel bill. And by considering the predictions of the effects of various outages, operators can readjust unit generation or voltage levels to prevent problems from arising or escalating.

The distribution system, on the other hand, is usually not monitored in real-time ; information about the system is obtained through such means as customer complaints and monthly meter readings. Planning and operation of the system relies heavily on historical data, and no attempt is made to match control actions to the system dynamics. Such approaches as have been tried to improve the operation of the distribution system have, almost exclusively, been limited in their scope and failed to take the operating needs of the system as a whole into account. An integrated 'systems' approach is needed in order to maximise the performance benefits of possible monitoring and control schemes and to balance the sometimes conflicting requirements imposed by such schemes.

A Distribution Automation System (DAS) is a system that enables a utility to monitor, co-ordinate and operate its distribution system in a real-time mode from remote locations. In this report we examine some of the issues involved in designing a DAS ; however, since the characteristics of the distribution system vary from one utility to another, resulting in unique requirements and priorities for a given utility, we phrase the discussion in the most general terms possible. For example, most distribution systems are operated radially - a tree structure with a unique path from each node to the source - but we will explore the suitability of our suggestions for loop systems, where multiple paths of power flow from the source to each node provide a higher level of service reliability. We consider the distribution system and its backbone communications system as a network with microprocessor capability at the nodes, and examine the factors affecting decisions about control and communications hierarchies and centralised vs. distributed processing.

Several projects have been, or are currently, implemented in an effort to determine the feasibility of automating distribution system functions. Published results from load control, real-time pricing, load modelling and forecasting, reactive power control via capacitor-switching, network reconfiguration and remote meter-reading experiments have increased awareness of the potential and the difficulties associated with such schemes. Methods for the economic evaluation of these innovations to determine their benefits to the utility and its customers have been proposed. A critical component of any automation scheme is the communications network : among the possibilities are the physical media (eg. coaxial and fibre-optic cable), power-line carrier schemes, radio-based schemes, and the use of telephone lines. Recent advances in parallel and distributed processing, and suggestions for applying expert systems to power system operations, have implications for the design of a DAS.

Once a list of possible Distribution Automation functions has been identified, the next step in the design of a DAS is to determine what parameters characterise such a system - in other words, the decisions which have to be made at the planning stage. In general, these decisions are not independent ; for instance, the communications system cannot be determined separately from the degree of local intelligence each node is required to possess.

For any particular utility there are many possible DAS's and some method is needed to compare different automation scenarios on the basis of the DAS parameters each possesses. Each utility will have its own priorities when comparing possible automation systems and so the weighting of the various DAS parameters will not be constant. For example, a utility with a low load factor might view load management as a high-priority function because of the relatively large relief to be gained by peak-load shifting, whereas another utility with a comparatively high load factor might view the reduction in losses due to capacitor-switching as more important. Consequently, we restrict ourselves in this report to identifying the basis for comparing different choices of DAS parameters and the effects the different choices have on the operation of the DAS.

Chapter 2

Distribution Automation Functions

The operation of the distribution system can be divided into three distinct states : normal, emergency and restorative. In the normal state, the load and operating constraints are met, and the aim is to operate the system to maximise efficiency, reliability and economy of operation. In addition we require the system to be robust with respect to the most likely emergency conditions, which means we take account of the future consequences of current control actions. If the system is predicted to remain in the normal state under prespecified emergency conditions then it is said to be secure. If one or more of the operating constraints are violated, the system is in the emergency state, while violation of only the load constraints puts the system into the restorative state. In either case the aim is to return the system to a normal state, but the priorities are usually different : in an emergency state we try to remove the constraint violations as quickly as possible, whereas in a restorative state we try to supply as much of the prefault load as possible, usually under the requirement that the operating constraints continue to be met.

We examine in this Chapter the functions of a Distribution Automation System (DAS) that enable us to carry out the above objectives. We consider first normal operation, where the distribution system is in its normal state. Then we look at DAS emergency response, where the distribution system is in an emergency or restorative state. Current practice in distribution systems is to base operating decisions on crude models of customer behaviour. However, it has been proposed that significant benefits to both a utility and its customers can be realised by better information about customers' loads and a greater involvement of the customer in the decisions affecting their electricity supply. We discuss separately those DAS functions which directly involve the customer or customer equipment. All DAS functions rely on the acquisition, transfer and processing of data for their operation. We mention some of the issues in data acquisition and storage and outline how data may be used to assist in the operation of selected DAS functions, while deferring a discussion of data transfer and processing to Chapters 3 and 4.

2.1 Normal Operation

We restrict our attention in this section to the case where the load and operating constraints on the system are met, and so we look for ways to improve system efficiency, economy of operation, reliability, and security. Reliability is a measure of the availability and quality of service to meet customers' load demands, while security is a measure of how robust the system is to possible emergency conditions (called contingencies).

Among the functions concerned with normal operation which we consider as candidates for automation are

- Feeder reconfiguration for loss minimisation and overload prevention
- Integrated volt/Var control
  - bus voltage control
  - feeder remote point voltage control
  - substation reactive power control
  - feeder reactive power control
- Cold load pickup

2.1.1 Feeder reconfiguration for loss minimisation and overload prevention

The system configuration refers to the network topology as determined by the status of switches (ie. open or closed), which we assume are under some form of remote control. Then we associate a state of the system configuration with a list of the status of all the *tie-line*, or Normally-Open (NO), and *in-line*, or Normally-Closed (NC), switches in the system. Thus if there are N switches under our control, there are 2^N possible states of the system configuration. **Feeder reconfiguration** is the operation, in real-time, of these switches in order to change the state of the system configuration.

The main objective in changing the feeder section connectivity through reconfiguration is to minimise the real power losses in the system. Given the present state of the system configuration, we adjust the status of the switches under our control so that losses are at a minimum. If we could continuously control the switches, then we could - in theory at least - achieve this objective, assuming we always knew the present state of the system. However, in practice, the opening or closing of switches is not instantaneous, there are operational constraints on how often switches can be opened or closed, and calculating the state of the system and the resulting configuration with minimum losses takes some finite amount of time, so we are limited to periodically changing the system configuration in such a way that losses would be minimised if the state remained the same until the next change. Another reason for reconfiguring the network is to improve the utilisation of existing equipment. This is done by load balancing, ie. by dividing the power flows required to meet load demands among the available lines so that no line is 'unnecessarily' overloaded. By unnecessary overload we mean that a line is overloaded but, by reassigning part of its load to another available line or lines, none of them would be loaded beyond their respective capacities. Thus if by changing the state of the system configuration we can remove overloads and still service the load, we have in some sense improved the efficiency of supply. Of course, it may happen that a line is overloaded and we cannot redistribute the load without overloading another line or lines; in this case, feeder reconfiguration can only minimise the overload and the decision on whether to continue to meet the load demand or not depends on other factors.

One of the benefits of a reconfiguration capability is that real power losses in the distribution system are reduced (whether or not they are minimised). Reduced losses lead to operational savings through decreased fuel costs; avoided or deferred generation capacity expansion; and avoided or deferred addition or replacement of feeders and feeder equipment. The prevention of overloads by redistributing the load in a more balanced fashion increases system reliability since the number of outages is reduced, and improves security of operation since possible emergency conditions resulting from line overloads are avoided. The ability to remotely switch feeders or feeder sections in and out of service means that crews do not

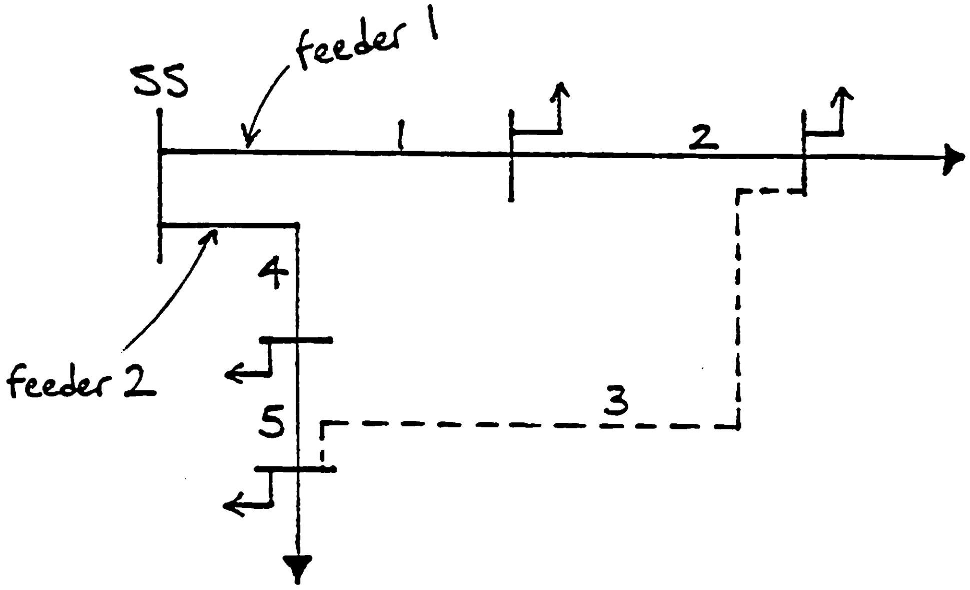


Figure 2.1: Model for branch exchange

have to be dispatched for these kinds of routine switching operations.

Problem Formulation :

A review of the various approaches to the feeder reconfiguration problem is given in [1], and we follow the problem formulation presented there. We assume a radial network, ie. each node of the network has a unique path back to the source node. We assume that the loads can be represented as constant-power loads, and that every switch in the system is associated with some line-section. We consider make-before-break switching, in which open switches are closed (possibly forming a loop) before closed switches are opened.

Consider Figure 2.1, where solid lines represent branches currently in service and dotted lines represent branches with open switches. A branch is a conceptual entity corresponding to all physical elements between the pair of switches at the branch ends. Suppose branch 3 is closed; since this forms a loop consisting of branches 1 to 5, one of the branches in the loop must be opened to restore the radial structure. Suppose branch 4 is opened;

then the loads on branches 4 and 5, which were on feeder 2, have been transferred to feeder 1 by this switching sequence. We call this operation a *branch exchange* between branches 3 and 4. More complicated switching strategies are composed of many such exchanges applied successively.

Let the distribution system connectivity be described by a graph which represents the system with all switches closed; this graph will change if outages occur or additions are made to the system. A particular state of the system configuration corresponds to a spanning tree of this graph, and so reconfiguration may be formulated as a minimal spanning tree problem : we seek the spanning tree which minimises the objective function while satisfying constraints on voltages, line capacities, and reliability. Note that the 'cost' associated with a particular branch may change with time as well as with changes in the system configuration, so the problem is more complex than the standard minimal spanning-tree case in which the branch cost functions are fixed.

Note that reconfiguration for loss minimisation or load balancing involves the same type of operation (branch exchange) and the same data (system parameters and load demands), and in each case a load flow calculation must be used to solve the minimisation. The essential difference between these problems is in the specification of the objective function. We make use of the *DistFlow branch equations* presented in [2] that describe real and reactive power flows in radial distribution systems. Defining

P_i,Q_i = real and reactive power flows into the sending-end of branch i+1 connecting nodes i and i+1
V_i = voltage magnitude at node i
r_i = resistance of branch i+1

we can denote the real power losses in the system by

$$c_p = \sum_{i=0}^{N-1} r_i \frac{P_i^2 + Q_i^2}{V_i^2} \tag{2.1}$$

This is used as the objective function in feeder reconfiguration for loss minimisation.

In order to distribute the required load among the available branches we need a measure of how much a branch is loaded. This is provided by the square of the ratio of the complex power at the sending-end of the branch, S_i, to the rated kVA capacity of the branch, S_{imax}, which we denote as

$$c_i = \frac{S_i^2}{S_{imax}^2} \equiv \frac{P_i^2 + Q_i^2}{S_{imax}^2} \tag{2.2}$$

Then as an objective function we could define (as in [1])

$$c_b = \sum_{i=0}^{N-1} c_i \tag{2.3}$$

One problem with this c_b is that the resulting loading pattern may be 'uneven', by which we mean that, in the corresponding optimal state of the system configuration, some branches might be loaded close to capacity while others might be lightly loaded. If we wish to ensure as equitable a loading pattern as possible, a better choice for c_b is

$$c_b = \max_i c_i \tag{2.4}$$

With this choice of objective function, some lines will of course be loaded more heavily than others (unless the line capacities are all the same), but the 'proportional' loadings of the lines (ie. with respect to their different capacities) will be as even as possible.

The solution to the reconfiguration problem is the spanning tree which minimises the objective function while satisfying all the constraints. A search over the set of states of the system configuration (ie. the set of all possible spanning trees) is computationally infeasible, since this set will be very large for practical networks and since to examine each state we must run a load flow to evaluate the objective. The approach taken in [1] is based on the observation that branch exchanges can be used to generate spanning trees from the present one and then this subset of the set of all spanning trees examined in the hope of decreasing the objective function. Calling this subset S, at each step we choose the branch exchange corresponding to the element of S which decreases the objective function the most without any constraint violation. The search is not over all possible spanning trees and so the solution can only be guaranteed to be locally optimal. In addition, performance of the algorithm based on this procedure depends on the choice of the branch to be opened after a branch has been closed to create a loop, and on the calculation of the objective function for each of the candidate trees. In [1] two approximate load flow methods are presented to reduce the number of times DistFlow must be run to once per search level : *simplified DistFlow* and *forward/backward DistFlow updates* . The approximate power flows are used in ranking candidate trees, so errors in the estimated figures may result in a different search than that based on an exact load flow calculation.

Implementation issues :

The above formulation is general and could be applied to non-radial networks. However a solution method based on branch exchanges would require more intensive computation due to the increase in the number of possible spanning trees compared to the radial case, and could not take advantage of the structure of the system to reduce the number of candidate trees to be examined since no assumptions about the system structure are made.

The use of feeder reconfiguration has implications for the protection scheme used to shield distribution system equipment from the effects of emergencies. If load is transferred from one feeder to another, the protection co-ordination established prior to the transfer may no longer be effective in coping with possible fault currents. Thus reconfiguration should interface with the setting of protection characteristics so that system security is not compromised. Another consequence of a feeder reconfiguration capability is the possibility that the direction of power flow in a branch reverses after a sequence of switching operations; for instance, the loads on branch 4 in 2.1 are upstream of those on branch 5 prior to the branch exchange between 3 and 4, and downstream afterwards. Since distribution system protection schemes typically exploit the unidirectional flow of power obtained with a radial topology, some of the attendant simplifications will be lost when power must be considered as capable of flowing in either direction through a branch.

It should be noted that loss minimisation and load balancing could be in conflict, for example if the configuration with minimum losses indicated that a certain line should be loaded to its capacity while other lines remain relatively lightly loaded. In such a case, the operating protocol (ie. the priorities assigned to the various automation functions) decides whether loss minimisation or load balancing dictates the operating configuration. Such a decision depends on the sensitivity of the respective solutions to changing configurations; for instance, if the solution to the loss minimisation problem was robust to slight changes in the state of the system configuration, which in turn had a large effect on the solution to the load balancing problem, then we would choose the configuration 'close' to the minimum-loss state that resulted in the most even line loading pattern, since we expect that the losses in such a configuration will not be much higher than the minimum.

Simulation results presented in [1] for the loss minimisation problem compare the reduction in losses achieved by the approximate load flow methods with that obtained by exact solution. The results show that the proposed approximations are computationally efficient and give conservative results close to that obtained using an exact load flow. The simulation also bears out the observations previously reported that the 'best' branch exchanges occur on the lower-voltage side of the loop, and that the system voltage profile increases as losses are reduced. It is suggested that the best approximation method to solve the loss minimisation problem is a hybrid of the two methods examined, trading off the less intensive computational load of the simplified DistFlow technique against the more accurate forward/backward DistFlow update solution.

An alternative formulation of the feeder reconfiguration problem is contained in DISTOP [3], PG&E's program for reconfiguring the distribution network in order to minimise real power losses. DISTOP has as 'hard' constraints that the reconfigured network must remain radial and that the loads prior to reconfiguration must still be served. Constraints on line capacities and voltage limits are treated as 'soft', meaning they may be ignored if a solution which only satisfies the hard constraints offers a significant advantage over the best solution that can be obtained taking all constraints into account. The optimum pattern of line flows (corresponding to the configuration with minimum losses) is defined as

$$\arg\min_{J} \; L \; = \; \sum_{i=1}^{m} R_i \mid J_i \mid^2 \tag{2.5}$$

subject to

$$A\,J \; = \; I$$

where

J = ($J_1,...,J_m$) are the line flows

I = ($I_1,...,I_n$) are the nodal injections

A = incidence matrix, representing the network connectivity

R_i = i^{th} line resistance

The algorithm used to approximate this minimisation is based on closing all NO switches, which creates a meshed network; solving for the optimal flow pattern; and using the resulting loadings of the switches to determine the sequence in which switches should be opened to return to a radial topology while not violating the hard constraints. The algorithm is computationally efficient and robust, and simulation results show savings of from 4 to 35 %.

2.1.2 Integrated volt/Var control

Voltage control involves maintaining the supply voltage within a certain range about a specified value. The necessity for voltage control arises from the fact that customers' equipment is designed for a constant voltage called the nominal or operating voltage. However, the losses in the system cause the voltage to drop as we move away from the substation down the feeder. Due to economic considerations, a utility cannot offset these losses and provide each customer on the feeder with a constant voltage matching the operating voltage of the customer's equipment, so instead utilities supply electricity within a framework of preferred voltage levels and ranges of allowable variation at these levels.

The generation of reactive power, or Vars, at a point remote from the load is uneconomic from the utility's point of view since the average value of this component of the complex power delivered is zero - hence the term 'useful' or 'active' power as an alternative for real power - yet reactive power flow gives rise to line losses and reduced utilisation of on-line capacity. Capacitors are a source of reactive power and supply leading Vars to a load across which they are connected, and their use as local generators of reactive power is well-established. Var control involves managing the flow of reactive power in the system, and would be required even if the system were lossless because the loads are generally not resistive but present (complex) impedances and so cause complex power to flow.

It is well-known that reactive power couples strongly with voltage magnitude and only weakly with voltage angle. A capacitor used for reactive power control affects the voltage of loads to which it is connected, typically supplying a voltage boost relative to the (fixed) substation voltage. An in-phase transformer controls the magnitude of its secondary voltage and so, if the load seen on the secondary side is voltage-sensitive, the reactive power delivered to the load will be affected. Thus we consider co-ordinating voltage and Var control to exploit this interdependence.

One of the principal objectives of integrated volt/Var control is to maintain all voltages within given tolerances of their nominal levels. This regulation of the system voltage profile leads to improved reliability of service; for example, the life of equipment is decreased if it is subjected to voltages very much below or above its operating voltage, and so keeping its supply voltage away from these harmful operating levels increases its expected time to failure. The quality of service customers experience is also improved by keeping voltages close to their nominal values, since excessive voltage variation interferes with the operation of customer equipment and could compromise safety; for instance, nuisance tripping of earth-leakage circuit breakers (ELCB's) due to the voltage varying out of the bandwidth about the set-point of the breaker inconveniences the customer and might convince them to trip out the ELCB permanently, removing one layer of protection against higher fault currents. Another aim of integrated volt/Var control is to minimise real power losses in the system by managing the flow of reactive power, leading to operational and investment-related savings for the utility through reduced fuel and maintenance costs and avoided or deferred capacity expansion costs. By reducing the component of power flow necessary to supply the losses in the system, we reduce line loadings while still meeting the load demand and so more efficient use is made of present equipment without having to sacrifice performance.

Voltage regulation :

The load demand and conditions of operation of a distribution system vary with time, and consequently so does the voltage at a load point. The range over which the voltage varies is divided into three zones, namely

(1) the favourable (or preferred) zone
(2) the tolerable zone
(3) the extreme zone

The *favourable zone* includes the nominal voltage level and is characterised by satisfactory operation of customer equipment. The *tolerable zone* lies directly above and below the favourable zone; supply voltages in the tolerable zone result in adequate equipment operation although perhaps less than warranted by the equipment manufacturer. Voltages outside the tolerable zone are in the *extreme zone* and are usually associated with emergency conditions; unless such voltage levels are temporary, equipment damage and/or failure may occur. The aim of voltage regulation is to contain system voltages to the favourable zone, and many methods have been developed to adjust voltage levels to meet this criterion. Among these are :

- feeder load balancing (among the phases)
- increasing the size of feeder conductors
- changing feeder sections from single-phase to multi-phase
- increasing primary voltage levels
- use of voltage regulators, both in the substation and on the feeder
- use of shunt and series capacitors, both in the substation and on the feeder

Imbalance among the phases of a 3-phase feeder means each phase must be regulated separately, so the motivation for ensuring approximately equal loading on each phase should be clear. Some discussion of the next three techniques may be found in [4]; however, in practice utilities have found that the most economical way of regulating voltages is to apply step-type voltage regulators or capacitors at the regulation points, and so we concentrate our treatment on these latter two techniques.

Voltage regulators are applied both in the distribution substation and at remote regulation points on the feeder. If two or more feeders connected to a substation bus have similar load profiles, then one is selected as the reference and all these feeders are regulated according to the requirements of the reference feeder : this is called **bus regulation** . If the load profile of a feeder indicates that the voltage drop from maximum favourable voltage at the substation would cause customers at remote points on the feeder to experience non-favourable voltage levels, voltage regulators are placed at those points on the feeder where voltage falls below the minimum permissible level : this is sometimes called **supplementary regulation** , or feeder remote point voltage control.

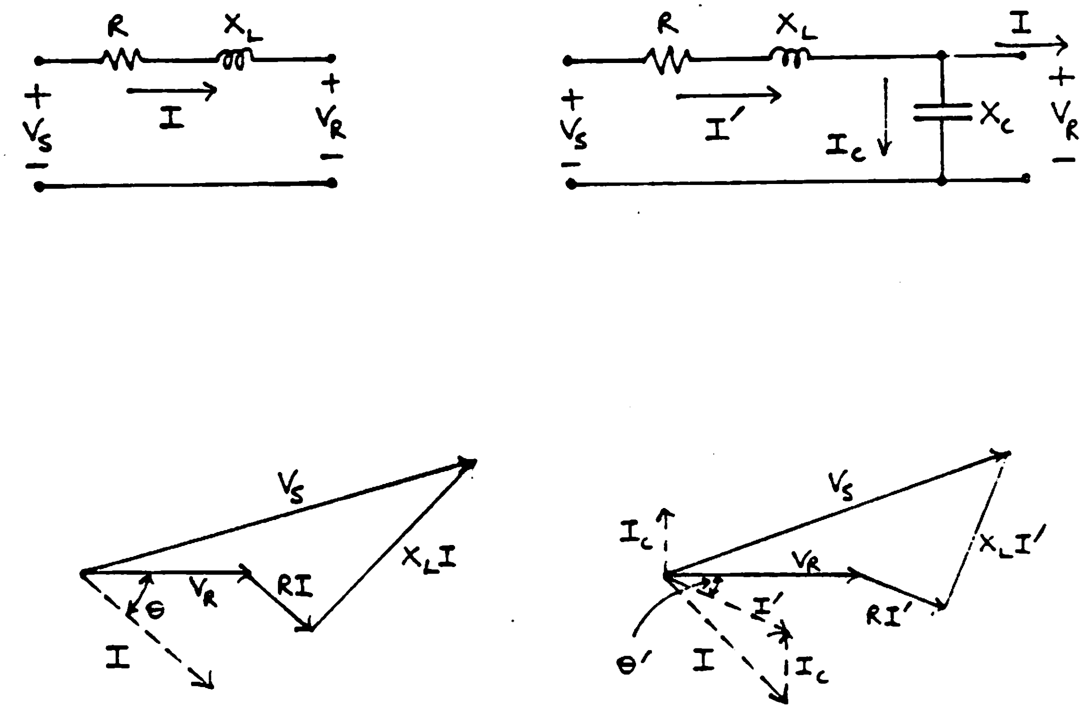
A step-type voltage regulator is usually an autotransformer designed to adjust the line voltage by ±10% through the use of different tap settings; 32 tap points yield $\frac{5}{8}$% voltage change per step. To maintain the voltage at a remote regulating point in the favourable zone without regard to the magnitude or power factor of the load, step-type regulators use *line-drop compensation* [5] which simulates line losses between the regulator and the regulation point; thus the voltage at the regulator relay is proportional to the voltage at the regulation point. An alternative is to monitor the remote-point voltage directly and send the measured value to the regulator relay which adjusts the regulator tap setting as appropriate.

Use of capacitors :

As mentioned above, capacitors regulate the voltage and reactive power flow at their point of installation. A shunt capacitor does these by changing the power factor of the load to which it is connected, while a series capacitor directly offsets the inductive reactance of the line in which it is applied.

Series capacitors are used to minimise or suppress the voltage drop caused by inductive reactances. The reactance presented by an inductor is positive and so a capacitor - which presents negative reactance - can be used in series to compensate and give a net reactance of zero (at least approximately). Used in this way, a series capacitor provides a *voltage boost* proportional to the magnitude and power factor of the through current, and can thus be considered as a voltage regulator providing a voltage rise which increases as the load grows. However, series capacitors do not give as much power factor improvement as shunt capacitors, and the line current is not significantly affected by their application which implies negligible changes in the real power losses. Furthermore, series capacitors are subject to the transient voltages and currents accompanying line faults and thus require elaborate protection and bypassing schemes to avoid damage or failure. Other implementation problems associated with the use of series capacitors are discussed in [4], and as a result shunt capacitors are usually preferred for voltage and Var flow control. The use of series capacitors in distribution systems is limited to reducing voltage flicker, making use of their faster response to load fluctuations.

The basis for using shunt capacitors with lagging power factor loads is illustrated in Figure 2.2. Comparing circuit quantities before and after the switching-in of the capacitor, we note that the magnitude of the source current is decreased, which implies *reduced line losses* ; the voltage drop between source and load is also decreased, hence the load experiences a *voltage boost* relative to the fixed substation voltage; and the power factor is increased (towards unity), which means the load demands *less* reactive power for the same real power as before. We consider next how these properties of shunt capacitors may be used in an automated system.



voltage drop $= RI_R + X_L I_I$,

$I_R = \mathrm{Re}\{I\}$

$I_I = \mathrm{Im}\{I\}$

voltage drop $\doteq RI_R + X_L I_I$
$- X_L I_c$

$\Rightarrow$ voltage rise $\doteq X_L I_c$

also $\theta' < \theta$

$\Rightarrow \cos\theta' > \cos\theta$

& $|I'| < |I|$

Figure 2.2: Use of shunt capacitors

Problem formulation :

The capacitor placement problem involves the determination of the location, type (fixed or switched) and size (ie. capacitance) of capacitors which, if installed in the distribution system, would minimise the real power losses while maintaining the voltages in the system within allowable limits. In its general form this is a very complex problem, and many solutions to it have been proposed (see [2] for a summary and list of references). Our discussion here is based on its formulation as a mixed integer programming problem, as presented in [6],[2].

Assume a radial distribution network topology with N nodes in the system. Assume that the load at time t takes on one of a finite number of discrete values, and that all loads in the system vary in a conforming way so that a common load variation curve, S(t), exists. S(t) might be the output of a load-forecasting program. Then any load in the system, L(t), can be written

$$L(t) = L_0 \, S(t)$$

where L_0 represents the peak value of L(t) over the period considered. The assumption about conforming loads is made to simplify the presentation and is not restrictive, since in practice we would group together loads which have similar variation curves and solve a more complicated but otherwise identical problem. With the above assumptions, the time period under examination may be divided up into intervals, in each of which the load demand on the system is constant. Let the number of different load levels be n. For each of the n load levels we have power flow equations, voltage constraints and capacitor control constraints. We solve the power flow equations using the *DistFlow branch equations* derived in [2]. Defining P_i, Q_i and V_i as in Section 2.1.1 and

Q_{ci} = reactive power injection from the capacitor at the i^{th} node,

the i^{th} branch flow equation can be written

$$x_{i+1} = f_{i+1}(x_i \, , \, u_{i+1}) \qquad (2.6)$$

where

$x_i = (P_i,\ Q_i,\ V_i^2)^T$ and $u_i = Q_{ci}$.

We can define

$u = (u_1, ..., u_{n_c})^T$

as an n_c-dimensional control vector, where $n_c \neq$ N unless a capacitor is installed at every node. Collecting the equations for a three- phase feeder with n_f branches and $l_f + 1$ laterals yields the $3(n_f + l_f + 1)$ DistFlow equations written in vector form as

$$G(x\,,\,u) = 0 \tag{2.7}$$

Since we have n different load levels, the DistFlow equations which must be solved to determine the state, x, given the load profile and the capacitor settings (as dictated by u) are

$$G^i(x^i\,,\,u^i) = 0\,,\,i = 1,\ldots,n \tag{2.8}$$

where x^i and u^i are the state and control vectors corresponding to the i^{th} load level, respectively.

The voltage constraints specify upper and lower bounds on the magnitudes of the node voltages and so can be transcribed into the form

$$H^i(x^i) \leq 0\,,\,i = 1,\ldots,n \tag{2.9}$$

A capacitor can be one of two types : *fixed* or *switched*, where by switched-type we mean that the capacitance is continuously variable and under our control. Fixed capacitors supply constant reactive power independent of load level, and so we have

$$u_j^1 = u_j^2 = \ldots = u_j^n \tag{2.10}$$

for the j^{th} fixed capacitor. We assume that the settings of the k^{th} switched capacitor can be controlled at all load levels, and that the setting at peak load (namely level 1) is greater than those at lower load levels. This gives

$$0 \leq u_k^i \leq u_k^1 \quad \forall\ i \tag{2.11}$$

for the k^{th} switched capacitor.

Let the real power losses in the system at the i^{th} load level be denoted by $p_i(x^i)$. Then the objective function to be minimised can be written

$$C = \sum_{i=1}^{n} T_i \, p_i(x^i) \tag{2.12}$$

If T_i is the duration of the i^{th} load level during the period under consideration, then C is the energy loss during the period : this is the objective function in [6]. We can also look on the T_i's as weighting factors which determine the relative importance of power loss reduction at the various load levels; for example, if we are only interested in reducing losses under peak load then $T_1 = 1$ and $T_i = 0$, i≠1. Since the real power losses in the system depend on the load demand, possibly resulting in negligible benefit to be gained by loss reduction when the system is lightly loaded but significant reduction potential under heavy load conditions, it is an advantage to be able to choose the T_i's to reflect this disparity.

There is also a cost associated with placing a capacitor. We approximate the typical 'staircase' cost function by a linear function of the form

$$f\,(u_k^1) \;=\; c_k \cdot e_k \;+\; r_k \cdot u_k^1 \,,\; for\, 0 \;\leq\; u_k^1 \;\leq\; u_k^{max} \cdot e_k \tag{2.13}$$

where u_k^1 and r_k represent the size and marginal cost of the k^{th} capacitor, and the k^{th} *placement decision variable* e_k is 1 or 0 depending on whether the k^{th} capacitor is placed or not.

Suppose now that the types and locations of the n_c capacitors at our disposal are known. Let the set of fixed capacitors be denoted by C_f and the set of switched capacitors by C_s. Then the problem is

$$min\, C \;=\; \sum_{i=1}^{n} T_i \, p_i(x^i) \;+\; \sum_{j=1}^{n_c} (c_k \cdot e_k \;+\; r_k \cdot u_k^1) \tag{2.14}$$

subject to

$$G^i\,(x^i \,,\, u^i \,) \;=\; 0 \,,\, i \;=\; 1,\ldots,n \tag{2.15}$$

$$H^i\,(x^i \,) \;\leq\; 0 \,,\, i \;=\; 1,\ldots,n \tag{2.16}$$

$$0 \;\leq\; u_k^1 \;\leq\; u_k^{max} \cdot e_k \;\forall\, k \tag{2.17}$$

$$0 \leq u_k^i \leq u_k^1 \, , \quad k \in C_s \tag{2.18}$$

$$u_k^i = u_k^1 \; \forall \, i \, , \quad k \in C_f \tag{2.19}$$

$$e_k = 0 \, or \, 1 \; \forall \, k \tag{2.20}$$

which is a (nonlinear) mixed integer programming problem.

We rewrite the problem more compactly as follows :

$$\min_{e,u} (C \mid e \in E \, , \, u \in U) \tag{2.21}$$

where E and U are the constraint sets for the placement decisions e and settings u, respectively. But this is equivalent to

$$\min_{e \in E} (\min_{u \in U} C) \tag{2.22}$$

which suggests decomposing the problem into a 'slave' problem

$$C_s (e) = \min_{u \in U} C \tag{2.23}$$

and its corresponding 'master' problem

$$\min_{e \in E} C_s (e) \tag{2.24}$$

The slave problem is : for a particular system load variation curve, find the optimal sizes of the capacitors in the system assuming their number and locations are known. The large number of equality constraints can be eliminated by using DistFlow to solve them for x in terms of u. It is shown in [2] that the Jacobian matrix $\frac{\delta G}{\delta x}$ is well-defined and thus the Implicit Function Theorem assures us that such a solution, x(u), can be found. A technique based on the *Phase I - Phase II Feasible Directions* method is used to solve the resulting inequality-constrained problem. Use is made of the geometry of the feasible region to minimise the number of currently-violated constraints that need to be considered at each iteration. In practice, switched-type capacitors are more expensive than fixed capacitors, so to determine the type of each capacitor once its size has been decided we first assume all capacitors are of switched-type; solve the problem considering only the switched capacitor settings and treating the fixed capacitors as reactive loads; take the capacitance found at the lightest load level (ie. level n) as fixed; and iterate until none of the switched capacitors are in at light load. Details of the solution algorithm may be found in [2].

The master problem is : for a particular system load variation curve, decide the status (on or off) of capacitors of given types and sizes at given locations in the system. Since there are n_c capacitors under our control, the decision graph has 2^{n_c} nodes each representing a particular set of placement decisions. If we assume that one decision is made at a time, then the decision graph can be arranged in a hierarchy with, say, the $\{e_k \equiv 1\}$ node at the top and the $\{e_k \equiv 0\}$ at the bottom, and - traversing the graph from top to bottom - each branch can be labelled with the capacitor to be removed to effect the transition between the nodes connected by that branch. The sorting procedure proposed in [6] considers all capacitors currently placed and evaluates the reduction in the objective function obtained by removing each of them in turn, and takes out the capacitor offering the smallest reduction. The solution methods to the slave and master problems are iterated until the reduction in the objective function offered by taking out the 'best' capacitor is below a preset tolerance.

Implementation issues :

The formulation of the capacitor placement problem outlined above is sufficiently general to allow for non-radial distribution system topologies. In general, the insertion of a shunt capacitor does not affect the current or power factor downstream from the point of application, and this fact may be used in a radial system to allow the specification of the required capacitances at each of the possible sites on the feeder in turn starting with the one furthest away from the substation. However, in a general network no such simplification may be made since power may flow in either direction through a branch and thus the possible sites for the application of capacitors cannot be ordered according to distance from the supply.

In practice we expect that only discrete amounts of capacitance may be switched, ie. that each u_k^i, k ϵ C_s, can only take on discrete values. We might wonder if the solution to the capacitor placement problem assuming continuously-varying capacitor settings (the 'continuous' problem) could be a guide to the solution of the problem assuming discrete-valued u_k^i (the 'discrete' problem). For example, suppose in the solution of the continuous problem that we calculate $u_1^1 = 0.75(u_1^{max})$. Then, forced to choose between the two nearest allowable values of u_1^1 - say $0.6(u_1^{max})$ and u_1^{max} - we choose the closest value as the control in the discrete problem - in this case, $0.6(u_1^{max})$. This can be thought of as a 'best-guess' heuristic for assigning the controls in the discrete problem based on the controls in the continuous problem. It can be shown that, under a constant-P,Q load assumption, if the real power injected into the network at the source node is a *convex* function of the node capacitive susceptances then the solution to the continuous problem, when used with the above best-guess heuristic assignment scheme, will be a good guide to the optimal solution of the discrete problem [7]. Conditions under which the real power injection is of this form are stated in [8] and the convexity proven.

We assumed throughout the above discussion that the load demands on the system could be modelled as constant-P,Q loads. The complex power injected into the system is the sum of the load powers and the losses in the system, so if we have constant-power loads then reducing the injected power is equivalent to reducing losses. The accuracy of a constant-P,Q model of the load demand has often been questioned [9],[10]; however, as noted in [8], if we only require capacitors to be switched at intervals then it is possible that the real power, at least, drawn by the loads will be independent of voltage when averaged over one such interval. If this assumption is not true, or if the interval between capacitor switchings is short enough to defeat a load-cycling strategy intended to ensure constant real power consumption, then it is possible that the reduction in real power supplied due to the reduction in power loss will be outweighed by the increase in power consumed by the loads as a result of the voltage boost accompanying the switching-in of a capacitor. One solution to this problem is to incorporate constraints into the problem formulation on the frequency with which capacitor settings may be adjusted, so that the real power averaged over one adjustment period can be made constant.

There are many issues to be decided in the area of regulation co-ordination, communication and control schemes to implement the solution to the capacitor placement problem, the possibility of applying expert system techniques to volt/Var control, and so on. We defer discussion of these and related areas until Chapters 3 and 4, when they can be framed in a system-wide context.

2.1.3 Cold Load Pickup

This refers to the transient surges that accompany the reconnection of loads to the system, or the switching actions used to implement feeder reconfiguration. These transients are also a consequence of faults on the system, and the ability to distinguish between fault-induced and 'normal' transients is thus a desirable feature of a DAS. Since the issues in

Cold Load Pickup are related to the protection system, we defer the discussion to Section 2.2.3.

2.2 Emergency Response

The *emergency response* of the system refers to system operation when not in the normal state. In this section we consider that either operating constraints are violated, putting the system into an emergency state, or that some of the load constraints are not met, which means the system is in a restorative state. The aim in each case is to return the system to a normal state, but the priorities are usually different; in an emergency state the emphasis is typically on removing the constraint violations in minimum time, while in a restorative state the objective is to reinstate supply to as much of the prefault load as possible, usually with the requirement that the operating constraints continue to be satisfied. The basic principle is that the protection system detects a fault condition and operates to clear it; then the switching and sectionalising operations determine whether the fault has affected any part of the system, find and isolate faulted system components, and take the appropriate steps to restore service to as much of the unfaulted portion of the system as possible.

The functions related to system emergency response considered as part of a Distribution Automation System are

- Protection
    - automatic breaker reclosing
    - overcurrent protection
    - harmonic restraint
    - underfrequency protection
    - adaptive relaying
- Bus/Feeder switching and sectionalising
- Cold load pickup
- Load-shed commands

2.2.1 Protection

There are many different faults which can occur in power systems. A lightning strike on a line can give rise to a conducting path between the line and earth, or between two phases of a three-phase feeder. Equipment can wear out or its insulation can be degraded such that conducting paths are short-circuited. Switching operations cause transients which can exceed the rated capacities of the lines. The consequences of a fault may be serious; for instance, arcing and high temperatures in an oil-filled transformer can lead to fire and explosion. Even for less severe faults, excessive voltages and currents can stress insulation beyond its breakdown point and overheat equipment, reducing its expected time to failure. When some component of the system is unavailable for use we say an *outage* has occurred. The consequences (whether immediate or cumulative) of faults can be described in terms of equipment outages. The **protection system** attempts - among other things - to decrease the frequency of outages by removing faults from the system, and hence minimise the effects of fault conditions on power system operation.

Protection System Devices :

Before going on to discuss the role of microprocessor-based relays in an automated protection system, we review briefly the present practice in overcurrent protection.

We impose the following criteria on the protection scheme :

- the protective devices should not operate in the normal state;
- when a fault occurs, the device immediately *upstream* of the fault should operate in order to clear the fault while removing no more of the system than necessary;
- if this protective device fails, the next upstream device should operate to clear the fault, and so on. This is referred to as *backup protection* .

We note that faults which are initially temporary in nature may become more serious if allowed to persist, possibly initiating a cascade of faults at points removed from the original fault location. Thus we want the protection system to continue to function under the unbalanced or asymmetrical conditions that often accompany faults. We might also like the protection scheme to have the capability for single as well as three-phase tripping.

The simplest protective device used in distribution systems is a **fuse** . Fuses detect overcurrent, interrupt it, and withstand the transient and steady- state voltages which arise

from opening the circuit. There are two main types of fuse :
current-zero awaiting , in which the arc due to the overcurrent is extinguished when this current goes to zero (which happens twice per cycle); in expulsion fuses the gases produced from the material in the fuse by the excessive current are de-ionised at a current zero and prevent the arc from re-forming, while in vacuum fuses the contacts are far enough apart that once the current goes to zero, it cannot re-ionise the vacuum between them;
current-limiting , in which the fuse action is characterised by the insertion of resistance into the current path, thus limiting the current through the fuse and hence the circuit.
Fuses suffer from a number of problems which limit their applicability; for instance, the difficulty in matching their protection characteristics with the operating characteristics of the system components they are supposed to protect over the entire range of operation, and the necessity of sending a line crew to replace a fuse which has operated before it can be used again. Thus we turn our attention to protective devices which can overcome these drawbacks.

A **circuit breaker** is a device with contacts which are closed in normal operation and open under fault conditions to interrupt the fault current. Again, the arc which forms between the contacts when they separate is extinguished at a current zero and the gases or vacuum enclosing the contacts prevents the arc from re-forming. The status of the breaker (ie. whether the contacts are open or closed) is controlled by a *relay* , which monitors the current and decides when a fault is present. After closing the breaker contacts - referred to as 'tripping the breaker' - the relay waits a fixed number of cycles and recloses them, so that if the fault was temporary service can resume immediately. Refinements to this basic technique include relays which reclose several times after differing time delays to improve the chances that the fault can be cleared if it is not a permanent condition; after a specified number of reclosings the relay trips out the breaker permanently. The advantage of preventing outages due to temporary faults must be weighed against the increased total fault energy to which the protected equipment is subjected in several reclosures. The trip characteristic of the relay can be matched to the characteristic of the device being protected; however, in a 'hard-wired' relay, once the trip logic has been installed it cannot be changed except by line crew.

Consider first the protection of a radial system. The principle of backup protection is straightforward : suppose we wish to protect against faults in between a particular breaker, B_i, and the next downstream breaker, B_{i+1}. We set a threshold value of current required to trip B_i, called the *pickup current* for B_i. Then we set the threshold current for the breaker immediately upstream, B_{i-1}, at the same value but introduce a time delay called the *co-ordination delay*, T_c, so that B_i will operate before B_{i-1} for any fault located downstream of B_i. This method for setting the value of B_{i-1}'s *backup current* does not affect its primary function of protecting against faults located between it and B_i, since we set the 'instantaneous' threshold for B_{i-1} (ie. B_{i-1}'s pickup current) at a higher level to cope with the higher fault current magnitudes associated with closer-in faults. After the co-ordination time has expired, the threshold decreases to the backup level mentioned above. The section of feeder between B_i and the next downstream breaker, B_{i+1}, is referred to as B_i's *zone of protection*. The discussion about B_i and B_{i-1} extends pair-wise to all breakers in the system. However, in non-radial system topologies a fault can be fed from either end of the feeder section in which it occurs, and relay co-ordination cannot be implemented as above under the constraint that we only remove as much of the system as is necessary to isolate the fault. The solution in this case is to use relays with a 'directional' feature, meaning the relay responds to faults in one direction only. We say the relay protects against faults in its *forward* direction to indicate which way a relay looks along a feeder. By considering pairs of forward-looking and reverse-looking relays we can establish a similar relay co-ordination to the radial case. Such a directional feature is based on the fact that the impedance of a typical feeder section is relatively inductive, and so the fault current *lags* the voltage by an angle close to 90^o for faults in the forward direction and *leads* the voltage by an angle slightly greater than 90^o for faults in the reverse direction.

The problem with relay co-ordination based on a deliberately-introduced time delay T_c is that, with multiple breaker failure, the time taken to clear a fault may be too long to avoid damage to equipment and/or cascaded faults along the feeder. The solution to this problem is to design the relay on a different principle. Under fault conditions the current increases since the impedance seen by the source is reduced, assuming small fault impedance, while the voltage at the fault location drops; thus the impedance seen by a relay looking toward a fault changes by a greater percentage than does the current monitored by the relay. A relay which operates on the basis of voltage-to-current ratio is called an *impedance relay* ; since the pickup value of impedance for an impedance relay is directly related to length of line (assuming constant line impedance), we also refer to this as a *distance relay*. The length of line corresponding to the pickup impedance is referred to as the *reach* of the relay, and by ensuring that the reach of B_{i-1} does not extend into B_i's zone of protection we can co-ordinate relay operation as before. Note that a fault beyond the reach of B_{i-1} but upstream of B_i still causes delayed tripping of B_{i-1}, so we have a compromise between the wish to maintain service to as much of the system as possible and the speed with which faults are to be cleared. To give impedance relays a directional feature to allow their use in a non-radial configuration, we can use them in conjunction with the directional relays described before, or modify their operating characteristic to only respond to faults in one direction.

The last type of relay we mention is the *differential relay* , which works by monitoring the currents immediately upstream and downstream of the device to be protected and examining their difference; if this difference is greater than a given threshold then we conclude that current is leaking to earth through a fault path and the relay trips the breaker. One problem with this kind of relay is the difficulty in protecting a transformer where an exact match with the turns ratio would be needed to ensure that the breaker would not operate under high-current but normal conditions. We can either raise the pickup current of the differential relay - possibly degrading protection under light-current conditions - or use a 'proportional'-type relay where the pickup current itself is proportional to the average of the monitored currents and thus adjusts automatically to the loading of the device under protection.

A more detailed discussion of the various protective devices mentioned above and their operation may be found in [11] or [12].

Microprocessor-based relays :

A recent development in the design of protection systems has been the introduction of digital relays using microprocessor technology to control the operation of one or more circuit breakers [13]. Digital relay design ranges from dedicated computers which replace the various types of relay mentioned above on a one-to-one basis, to integrated systems which implement relay co-ordination using communications links between the relays so that the relay settings can be adjusted in a coherent manner to reflect changing system conditions. We will not discuss here the hardware structure or signal processing techniques used in microprocessor relays; details of this component of the design of digital relays may be found for example in [13]. Instead, we concentrate on the relaying algorithms which have been proposed for digitally-based protection systems.

There are essentially two types of relaying algorithms : one is based on modelling the fault signal waveform (voltage or current) and using estimates of the waveform parameters to decide whether or not to trip the breaker, while the other models the faulted system by a differential equation and uses estimates of the system parameters to control its operation. We refer to these two distinct protection philosophies as the *waveform model* and the *system model* , respectively.

(1) Waveform model : the idea behind this model is that the parameter(s) of interest for relay operation is contained in the fault signal waveform(s), eg.
the *peak value* of sinusoidal current is needed for overcurrent protection;
the fundamental frequency *voltage and current phasors* are needed for an impedance relay;
the *magnitudes of the harmonics* are needed for harmonic restraint;
the *fundamental frequency* is needed for underfrequency protection.

As an example of a so-called 'window' algorithm, consider the problem of estimating the amplitude and/or phase of a sinusoidal signal y(t), where in general

$$y(t) = A\cos(w_o t + \phi)$$

where we assume that the fundamental frequency w_o is known. We can rewrite this as follows :

$$y(t) = Y_s \sin(w_o t) + Y_c \cos(w_o t) \tag{2.25}$$

where the representations of y(t) are related by

$$A = \sqrt{Y_s^2 + Y_c^2} \tag{2.26}$$

$$\phi = -\tan^{-1}\left(\frac{Y_s}{Y_c}\right) \tag{2.27}$$

Suppose we sample y(t) every δt seconds, and define y_{-1} = y($-\delta$t), y_o = y(0), y_1 = y(δt). Then we have an over-determined system whose solution may be written

$$Y_s = \frac{y_1 - y_{-1}}{2\sin\theta} + c_1(y_1 - 2y_o\cos\theta + y_{-1}) \tag{2.28}$$

$$Y_c = \frac{y_1\cos\theta + y_o + y_{-1}\cos\theta}{1 + 2\cos^2\theta} + c_2(y_1 - 2y_o\cos\theta + y_{-1}) \tag{2.29}$$

where c_1 and c_2 are arbitrary. In this case the window, or number of samples of the waveform used in estimating the parameters of interest, is of length 3. The *Mann-Morrison* algorithm corresponds to choosing $c_1 = 0$ and $c_2 = \frac{\cos\theta}{1 + 2\cos\theta}$, and acts as a nonlinear filter which uses the waveform and its time-derivative to estimate the waveform magnitude. Unfortunately the greatest amplification of this filter is in the neighbourhood of the third harmonic (ie. 180 hz.) and so significant nonlinear distortion can occur. The *Prodar-70* algorithm corresponds to $c_1 = 0$ and $c_2 = \frac{1}{\sin^2\theta} - \frac{\cos\theta}{1 + 2\cos^2\theta}$, and uses the first and second derivatives to estimate the waveform magnitude. The frequency response of this algorithm is designed to reject signals below the fundamental, but again the harmonic content of the waveform under examination is amplified relative to the fundamental. One way of reducing the high-frequency gain is to set both c_1 and c_2 equal to 0, but this degrades the low-frequency performance. Another problem with window-type algorithms is the fact that the window will contain both prefault and postfault values until one full window-length after a fault occurs, and so there is a built-in delay before the algorithm responds as designed to the fault condition; shortening the window lessens the ability to filter out the harmonic components. Because of these shortcomings, an alternative approach to estimating the relaying parameters is to treat the problem as stochastic parameter identification [13].

We have the following representation for the k^{th} sample of y(t) :

$$y_k \ = \ Y_s \sin(\, k w_o \delta t \,) \ + \ Y_c \cos(\, k w_o \delta t \,) \ + \ \epsilon_k \tag{2.30}$$

where ϵ_k is a random variable which models all components of the waveform of interest not otherwise represented. This can be written more generally as

$$y_k \ = \ \sum_{n=1}^{N} Y_n \, s_n(\, k \delta t \,) \ + \ \epsilon_k \tag{2.31}$$

where the $s_n(t)$ are known signals and the Y_n are the unknown coefficients to be estimated. Suppose we use m measurements to estimate the N unknown parameters, then clearly we need $m \geq N$ and we can write the resulting m equations in matrix form as

$$y \ = \ S\,Y \ + \ \epsilon \tag{2.32}$$

where y is an m-dimensional measurement vector, Y is an N-dimensional vector of unknowns and S is an m-by-N matrix with known entries. We assume that the error vector has zero mean, ie. E(ϵ) = 0, and covariance matrix W = E($\epsilon\,\epsilon^T$), where the superscript T denotes transpose. Let the estimate of Y be denoted Y^*, then the mean-square error between the estimated values Y_n^* and the true values Y_n is

$$J = \sum_{n=1}^{N} E((Y_n^* - Y_n)^2) \tag{2.33}$$

and it can be shown that the minimum-least-squares unbiased estimator for Y is

$$Y^* = (S^T W^{-1} S)^{-1} S^T W^{-1} y \tag{2.34}$$

If we assume the errors are independent and identically-distributed then the error covariance matrix W is just a scalar multiple of the m-by-m identity matrix I_m, and the previous equation for Y^* simplifies to

$$Y^* = (S^T S)^{-1} S^T y \tag{2.35}$$

which is the basis for *least-squares fitting algorithms*. Note that the matrix $(S^T S)^{-1}$ can be computed off-line to save computation time; however, a bad choice of the basis signals $s_n(t)$ could result in this matrix being full, so that even if we only save those rows of $(S^T S)^{-1}$ corresponding to the components of Y necessary for relaying purposes, the computational burden is considerable. Two popular algorithms based on the above simplification are Fourier-type algorithms, where the $s_n(t)$ are sines and cosines of integer multiples of the fundamental frequency and thus $S^T S$ is a diagonal matrix, and algorithms which use Walsh functions as the basis signals so that again the $s_n(t)$ constitute an orthogonal set.

(2) System model : in this case the line in the forward direction of the relay is modelled as a series RL circuit, giving rise to a differential equation relating the relay voltage and current of the form

$$v(t) = R\,i(t) + L\frac{di(t)}{dt} \tag{2.36}$$

Extra terms are sometimes added to the right-hand side to account for mutual inductances between the lines. The problem here is to estimate the values of R and L and compare them to their expected values under fault conditions. Integration of the differential equation using the trapezoidal rule allows the formulation of a number of different algorithms depending on the number of samples of voltage and current used [13]. An advantage of this type of algorithm is that even if the frequency drifts away from the fundamental, the calculation of R and L is still correct provided the voltage and current signals continue to satisfy the differential equation. The methods used to estimate the unknown parameters are similar to those outlined above.

Implementation issues :

Among the advantages offered by an automated protection system based on microprocessor relays, perhaps the most important is the capability to shape the trip characteristic of any particular relay through software changes only. Thus the relay trip characteristic can be complex and can be tailored to suit the operating characteristic of the device under protection. This makes for more efficient relay operation since the relay's trip logic is not limited by hardware considerations, and allows for modular relay design in which the equipment itself is standardised throughout the system since the decision-making is controlled by software. If we can adjust the trip characteristic of a relay in real-time then the protection scheme can track changing system conditions, avoiding the compromise faced with conventional relays where the requirement to trip under fault conditions must be balanced against the desire not to trip under normal but heavy-load conditions. The possibility of adapting relay protection characteristics to operational needs also suggests a way of accounting for the desire to maintain continuity of service to loads, by raising the pickup value of the relay parameter enough to avoid tripping the breaker while reconfiguring the system to relieve the overload through load balancing. The consequent reduced outage time due to the decreased frequency of false tripping (ie. trips when the system is not in fault condition) improves the reliability of operation as measured by the reliability indices defined in Section 5.2.

Relay co-ordination, which allows protective devices further up the feeder to offer backup protection to those nearer the fault location, is easier to implement in a system involving digital relays with communications links between them than in conventional relaying systems. We first note that a general principle underlying co-ordinated relay operation is that the decision on whether to trip or not should be local to the relay and as independent as possible of the trip decisions of other relays. This maintains reliability in the event of relay or breaker failure; consider for example the problems associated with the failure of the communication link between a relay and its breaker, where the relay sends a trip signal to the breaker and also informs the next upstream relay that the fault has been dealt with - if the backup relay resets its co-ordination time on the basis of this incorrect information, the fault may have further undesirable consequences. The data rates required between the data acquisition transducers and the relay computer are too high in practice to consider the remote control of one relay's operation by another relay or a central computer [13], which is another reason for taking trip decisions at the relays they affect. However, the provision of communication paths between relays can be used to enhance effectiveness, for example in the concept of *pilot protection* . This scheme is one realisation of the integrated protection system mentioned before, and involves 'fail-safe' transmission in the reverse direction of relay status. A relay transmits a message in the reverse direction when it detects a fault in the forward direction, and so the absence of a message under fault conditions is interpreted by the backup relay as indicating that the fault is in its zone of protection or that the downstream relay is defective. In this way we do not have to introduce fixed co-ordination times to ensure backup protection, and so the efficiency of the protection system is improved. The increased likelihood of clearing a temporary fault before it progresses in severity means reduced equipment overloads and hence lower maintenance and repair costs and increased reliability.

The introduction of microprocessor capability to a relay allows a wide range of background tasks not usually possible in a conventional relay to be carried out. The relay computer can monitor, process and transmit operational and fault data to a database so that a better understanding of the events leading up to the onset of a fault can be obtained. Digital relays incorporate self-testing diagnostics and can monitor their peripherals for evidence of failure. Because of their enhanced data-processing ability, different types of relay can make use of the inputs from a single transducer, and can carry out basic bad data detection through consistency checks. We can also expect easier interfaces with the other Distribution Automation functions since the information does not have to be converted from analog to digital form once it resides in the relay.

There are many other issues related to the operation of microprocessor-based relays, such as to what extent the protection functions can be carried out locally, how the inter-relay communications required in an integrated protection system can be implemented, or the specification of the weights attached to the various load and operating constraints. In addition, the precision and dynamic range required for Distribution Automation applications may differ from those found in present-day digital relays. We will return to some of these issues in later chapters, where their interdependence with the other functions of a Distribution Automation System will be clearer.

2.2.2 Bus/Feeder switching and sectionalising

When the protection system has operated to clear a fault, the power system moves into a restorative state in which some of the prefault load demand may not be satisfied. This is referred to as an outage, and the unavailability of the affected system components is due to the steps taken to remove the fault. The problem the system operator is then faced with is twofold; first, equipment damaged during the faulted state must be located (and often removed) from the system for repair or replacement; second, that part of the system unaffected by the fault must be identified and brought into service to try and meet the load constraints. We note that, if we do not impose the constraint that the system must not go into another emergency state, taking the actions necessary to fulfill these aims could result in overloads and decreased system security; however there may be situations in which these are acceptable operating risks, at least in the short term. For the remainder of this discussion, we will assume that the switching and sectionalising operations described are subject to the operating constraints imposed on the system. The objective of bus and feeder switching and sectionalising is thus to return the system to the prefault normal state (or as close to it as possible), and is concerned with decreasing the *duration* of outages and, in some sense, the 'severity' of supply disruption.

Since outages occur after the removal of a fault from the system, reliability of operation is improved by minimising their duration. This can be seen from the effects a reduction in outage time has on the indices used to quantify system reliability (5.2). Faster location and isolation of faulted equipment helps to prevent overloads and thus also contributes to the improvement in operational reliability. Decreasing the time during which service is not available to customers means increased revenues and an improvement in customer satisfaction which is harder to determine but, especially in a competitive electricity market, perhaps ultimately more important to the utility. The capability to automatically identify and remove equipment affected by a fault and to restore service to the unfaulted portion of the system means savings in crew time required to perform these tasks.

Problem formulation :

To determine whether a fault has had any effect on the system, we compare the status of the protective relays to their prefault values. If none of them has changed, either the fault was temporary and has been satisfactorily cleared or the protection system is defective. Even if the fault has indeed been cleared, it may have degraded the insulation of some component so that its safety margin has been violated. Thus we should co-ordinate with the monitoring functions at this stage to check for overloads. On the other hand, the status of one or more relays may have changed from its prefault value. In this case, any equipment located downstream of such a relay has been disconnected from the source. Again, we should also check to see if there are overloads registering on any of the system components still in service.

Suppose now that we know certain relays have tripped. It is possible that equipment downstream of these relays is serviceable, and so the next problem we face is to find equipment which has suffered damage or failure due to the fault. This problem is considerably simplified if the protection system has the feature described in Section 2.2.1 that the relay closest to the fault trips, since ideally we would know that the fault was in that relay's zone of protection. Of course we have to account for the possibility that some relays or breakers have failed, tripping a relay further away from the fault.

One approach to this problem of *fault location* is based on the use of an adaptive Kalman filter for the monitored voltage and current [14]. It is assumed that the transients accompanying a fault have parameters that are random, and so the problem becomes one of estimating the state in the presence of fault-induced noise. The state equation is of the form

$$x_{k+1} = A_k x_k + w_k \tag{2.37}$$

where x_k denotes the voltage (or current) at time k and w_k is a random variable with known distribution. The measurements of the state obey the equation

$$z_k = H_k x_k + v_k \tag{2.38}$$

where z_k is the measurement at time k and v_k is the noise in this measurement. We assume that the distribution of the measurement noise is known. Use is made in [14] of the difference in the statistical properties of w_k in faulted and unfaulted conditions to derive two Kalman filter models of the system, one which assumes the phase is faulted and the other which assumes the features of an unfaulted phase. To decide which filter is modelling the system correctly, weight factors calculated by the two filters are compared; as the measurement sequence progresses, one converges to 1 and the other to 0, indicating the correct and incorrect models respectively. Once the correct model is determined, the other filter is shut down to reduce the computational burden. The correct filter continues to process the inputs until x_k converges to its steady-state value and, assuming that a fault occurred, the ratio of the voltage and current phasors is used by a distance protection scheme to determine the fault location. Tests reported in [14] on simulated data using a sampling rate of 16 samples/sec. showed convergence of x_k in a half-cycle, and consequently in the same time the location of the fault was found.

The isolation of faulted equipment is straightforward once its location has been determined either by a fault location algorithm or through an overload check. The relay on the upstream side closest to the affected component is tripped out and the relay tagged so that it cannot be opened automatically but only when the operator has verified that it is serviceable again.

The problem of restoring service to the unfaulted system while observing the operational constraints can be formulated as a special case of the load balancing problem discussed in Section 2.1.1. The issue here is what to use as an objective function c_b. One possibility is to minimise the *unserved demand* . Let the load demand of the i^{th} branch available for service be denoted L_i; then we can define an indicator variable for the i^{th} branch as I(i), where I(i) = 1 if the i^{th} branch is NOT included in the spanning tree being considered, and 0 if it is. Let M denote the number of branches available for service. Then the objective function can be written

$$c_b = \sum_{i=1}^{M} I(i)\,L_i \tag{2.39}$$

We might consider minimising the *number of unserved branches* , regardless of the individual branch demands; in this case, the objective function is simply

$$c_b = \sum_{i=1}^{M} I(i) \tag{2.40}$$

Denote the number of customers on the i^{th} branch by C_i, then we could minimise the *number of customers affected* by using

$$c_b = \sum_{i=1}^{M} I(i)\,C_i \tag{2.41}$$

A fourth possibility is used as the basis for PG&E's PICKUP algorithm [3], where the objective is to reconfigure the distribution network by switching operations to minimise the total length of unserved branches, subject to the hard constraint that the reconfigured network remains radial and the soft constraints that all the prefault loads must be served and the operational constraints observed. Let l_i denote the length of the i^{th} branch available for service. Instead of making the line capacities hard constraints which cannot be exceeded by a feasible solution, l_i is multiplied by the square of the ratio of the current through the i^{th} branch to its rated capacity in the objective function, so that overloaded lines are less likely to be part of the optimal configuration than branches loaded below their rated value. Let m denote the number of branches in the system. The problem is then

$$\min_{J} \ c_b = \sum_{i=1}^{m} l_i \mid \frac{J_i}{J_i^R} \mid^2 \tag{2.42}$$

subject to

$$A\,J = I \tag{2.43}$$

where

J = ($J_1, \ldots, J_m$)^T is the vector of line flows
J^R = ($J_1^R, \ldots, J_m^R$)^T is the vector of line ratings
I = ($I_1, \ldots, I_n$)^T is the vector of nodal injections
and A = network incidence matrix.

The effects on the solution of the different choices for c_b are discussed next.

Implementation issues :

Minimising the unserved demand is probably the most natural choice for c_b since this will result in a normal state close or, if the optimal c_b^* is zero, equal to the prefault state. Here close refers to the fact that as much of the prefault demand as possible is served. A problem with this c_b arises if the optimal loading pattern is distributed geographically in such a way that 'pockets' of demand isolated from the bulk of the system must be served. In such a case we might wish to disconnect these remote loads until the intermediate loads are available again or maintain supply to those loads closer to the source to simplify the voltage regulation problem. Minimising the number of unserved branches could produce a normal state far from the prefault state since the branches omitted from the optimal solution could be those with the highest load demands. However, such a scheme will produce a reasonable loading with the minimum number of switching operations if the demands are approximately the same. Minimising the number of customers disconnected will only produce an acceptable solution if all the customers are of the same type, ie. we do not consider residential customers in the same objective function as industrial customers due to the significant difference in the sizes of their demands. Under this restriction, we minimise the disruption to customers and so improve the intangible customer satisfaction that may come to play an increasing role in such decisions. Using the PICKUP algorithm for service restoration can give a solution which violates some of the operating conditions, but incorporation of weighting factors into the terms of c_b could achieve the required balance between removing faulted equipment and maintaining supply to the loads.

The extension of the above to non-radial network topologies suffers from the same problems mentioned in Section 2.1.1, namely the difficulty in reducing the number of candidates for branch exchanges from that of a complete search and the consequent increase in the computational burden. In both the radial and non-radial cases, we expect some reduction in the number of candidates to be examined since certain branch exchanges will not be available if one of the branches contains faulted equipment. In a large system, however, this simplification will be negligible for all but the most serious faults.

2.2.3 Cold load pickup

The problem here is that high transient currents accompany the reconnection of a load or transformer to the system, or the starting of a motor. The system is in a normal state except for these transients, so we do not wish to have the protection system operate and incur the problem of service restoration. One solution we will see in Section 2.3.1 is to stagger the return of load demands so that the magnitudes of the associated transients are reduced. This type of solution applies to the case where there are enough 'small' load demands that, by cycling their return, we can significantly reduce the transient magnitudes. However, this approach cannot be used when a transformer or a single load is being reconnected if we do not want to cycle the component in normal operation. We could increase the pickup value of the protecting relay, but this compromises the light-load protection performance. A third possibility is to note that such transients have a distinctive strong harmonic content and so a relay whose operation is based on examining the harmonics of the through current can discriminate between normal and transient currents. Let the harmonic components of the waveform of interest be denoted H_i, i = 2,3,... ; then we can define the *harmonic content* of the waveform by

$$hc = \sqrt{\sum_{i \geq 2} H_i^2} \tag{2.44}$$

and trip the relay if hc exceeds a threshold value. This type of relay operation is called *harmonic restraint* . In practice we might add constraints on the maximum even and odd harmonic components, and also explicit constraints on the magnitudes of the lower-order harmonics since they usually dominate.

The protection characteristic of a relay used to solve the cold load pickup problem should be adaptive to system conditions, since unless we expect a high inrush current in normal operation, we cannot assume that the excessive currents are not due to faults.

2.2.4 Load-shed commands

These are commands issued by the EMS computer which controls the operation of the bulk power system, calling for immediate disconnection of a specified percentage of the load demand and usually with no indication of which loads to switch off. Assuming the load-shedding may be indiscriminate, one solution is to maintain a priority list of all loads in the system for just such an emergency and use this prioritisation to drop load. The determination of the priority assigned to a particular load may depend on its size, location, type and perhaps the current loading of the branch to which it is connected. We defer further discussion on designating loads as interruptible or non- interruptible until Section 2.3.1; and since the issues in implementing load-shed commands are related to the communications used to pass through the commands and the degree of local control over load reduction, we defer consideration of this function until Chapter 3.

2.3 Customer Functions

In this section we consider the functions in a Distribution Automation System which directly involve the customer or customer equipment. At present the role of the customer in the power industry is passive : customers receive and pay for a product - power - which has various attributes such as level of reliability, supply voltage, and cost, while having relatively little input into the design of this product or choice of its attributes. Customer response is limited to crude signals such as whether or not aggregate energy consumption is reduced when the price of electricity is raised. Because of the long time-delays between utility decisions being made and the responses to these decisions being observed, and the 'smoothing' effect of lumping customer loads at the substation level for the purposes of operating the system, it has not been necessary for utilities to know in detail how individual customers or groups of customers behave, and so the distribution system is essentially not monitored from the feeder level down. Utilities are usually only concerned with this part of the system when a fault occurs which interrupts service to some of its customers, and distribution planning relies heavily on historical load data from which industry-recognised load indices are estimated. However, it has recently been pointed out that substantial benefits to both utilities and customers can be achieved by various strategies which call for knowledge of the composition of customer loads, the differences in customer load patterns and real-time information on customer responses to utility control signals. Such strategies often require the customer to make more complex decisions about their demand characteristic, and this more careful evaluation of electricity usage is expected to lead to more efficient system operation, assuming that customers are not overwhelmed with data which confuses rather than assists their decisions. These new strategies will require educating the customer about their operation as well as a marketing drive to persuade customers to participate, and so involve a 'hidden' implementation cost which should be considered in the decision on whether to make these functions available or not.

Some of the functions in a Distribution Automation System which directly involve the customer are

- Direct Load Control
- Real-Time Pricing
  - spot pricing
  - priority service pricing
  - wheeling

- Remote Metering
  - remote meter reading
  - Var billing
  - remote programming of meter
- Customer Services

2.3.1 Direct Load Control

Direct Load Control refers to the ability of a Utility to control (ie. disconnect) or cycle part or all of a particular component of its load. This component may be one customer's load, or the customers may be divided into groups according to their load characteristics, needs, etc., and each group controlled separately.

The primary objective of direct load control is to alter the aggregate energy consumption pattern so that peak demand is reduced, avoiding the need for the more costly peaking equipment used by Utilities to supplement baseline capacity. This objective can be met by load denial or load deferral; and since the aim is to shift demand in time, the total energy consumption may increase or decrease. If the peak load can be reduced enough so that higher-cost peaking units are not needed, the utility benefits through decreased fuel and transmission costs. The existence of a significant number of customers accepting direct load control can substitute for reserve margin additions, and the deferred expansion of generation, transmission and distribution capacity due to the more efficient use of present capacity leads to avoided or deferred costs. For instance, a utility planning to add incremental peaking capacity could instead use direct load control to maintain the desired reserve margin, and wait for load growth to justify the addition of lower-cost baseline generators. System security is enhanced through the use of direct load control since the utility has an increased effective operating reserve margin and a wider range of options in dealing with an insecure state. A secondary objective is to reshape the load profile to satisfy optimal operating conditions ; in this case, direct load control may be used in conjunction with network reconfiguration eg. to reduce losses or to co-ordinate with optimal generation schedules. The staggered return of load required under cold load pickup conditions (2.2.3) can be implemented if the utility can control or cycle customer loads, and the capability for removing loads that would otherwise disallow certain switching actions by causing short-term overloads can speed service restoration after a disturbance (2.2.2). Remotely connecting and disconnecting service when there is a change of tenant saves crew-time in visiting the premises. Since customers are directly affected by direct load control , there will inevitably be some inconvenience caused by control actions; however, if savings obtained from the use of direct load control are passed on to the customer in the form of reduced electricity rates, then a balance may be reached where enough customers are sufficiently compensated for the inconvenience of load interruptions that the global social welfare of both utility and customers is increased. The use of direct load control to shed load during emergency conditions comes under the heading of Pass-Through Commands.

Problem Formulation :

There exists a wide variety of load types in the service area of a typical Utility. In addition, customers having similar types of load may have very different expectations of the ability of the Utility to meet their load demands, and each customer typically distinguishes between the various components of their load rather than regarding it as an aggregate demand for electricity. In order to allow for different load behaviour, we divide the load available for direct control into J groups in such a way that the j^{th} group, j = 1,..., J, may be controlled as a unit. For example, classify each customer as industrial, commercial or residential, and within each class, group loads which have similar characteristics and are located close to each other - one such group might be the air conditioners in the houses fed by a particular distribution transformer. Note that we make provision for customers to assign part of their load as controllable and the remainder as uninterruptible (at least nominally), eg. an integrated-circuit chip manufacturer would presumably not designate that portion of their load associated with production as a candidate for direct load control because of the prospect of losing an entire batch of silicon wafers. We assume that all the load groups can be described by the same load model, whose parameters take on different values depending on the particular load types being modelled. Since we envisage direct load control as an ongoing process, the load model must reflect the time-varying nature of the load and its response to the control strategy, and so we derive a dynamic model of the controllable load. Our analysis draws heavily on the results presented in [15].

As with most control problems, the search for the 'best' control strategy is limited to strategies which meet certain conditions. In this case, the constraints arise from a desire to ensure that the effects of direct load control are distributed somewhat equitably among the controllable groups, ie. that groups are not disconnected arbitrarily often or for arbitrarily long periods. We also impose a limit to the 'impact' of the control strategy on any given group (in a sense to be defined below).

Early attempts at load management relied on enumerating possible control schedules and selecting the one most closely approximating the desired optimality conditions. This approach thus required the candidate schedules to be prespecified, and rescheduling control on the basis of new information in the middle of a control period was impossible. Of course, if all the possibilities are considered then the global optimum is obtained, and if this optimum is non-unique all the optimal schedules are found; however, this technique is not used in practice since it imposes too great a computational burden. We can state the problem as follows : select a control strategy for each of the J load groups such that some system-wide objective function is minimised, while satisfying the constraints mentioned above and allowing the possibility of resheduling the controls at any time.

We start with some definitions. The **controllable load** is the load that is subject to direct load control . The **system load** is the actual load presented by the system, while the **diversified load** is the load that occurs, or would occur, when the load is not controlled. The **delta load** is the actual load less the diversified load, either for the system as a whole or for the j^{th} group. The **energy demand** in a period T is related to the delta load during T by

$$energy\ demand\ in\ T\ = -\int_T (delta\ load)\ dt$$

When the delta load is *positive* it is called the **payback** , because it represents the amount of energy required by the controlled device to bring it back to equilibrium (zero energy demand). Thus we think of each load group as an energy-storage device. See Figure 2.3.

We consider the interval [0,T] where t $\doteq$ 0 denotes the current time. Since the load is randomly varying we cannot know the load profile on [0,T] ahead of time, but we assume we have a forecast generated by the Load Forecasting function. If we regard the diversified load for the j^{th} group as a random variable, then we can deduce its probability density function from its forecasted profile as shown in Figure 2.4. Defining

$PL_j(t)$ = diversified load of j^{th} load group at time t

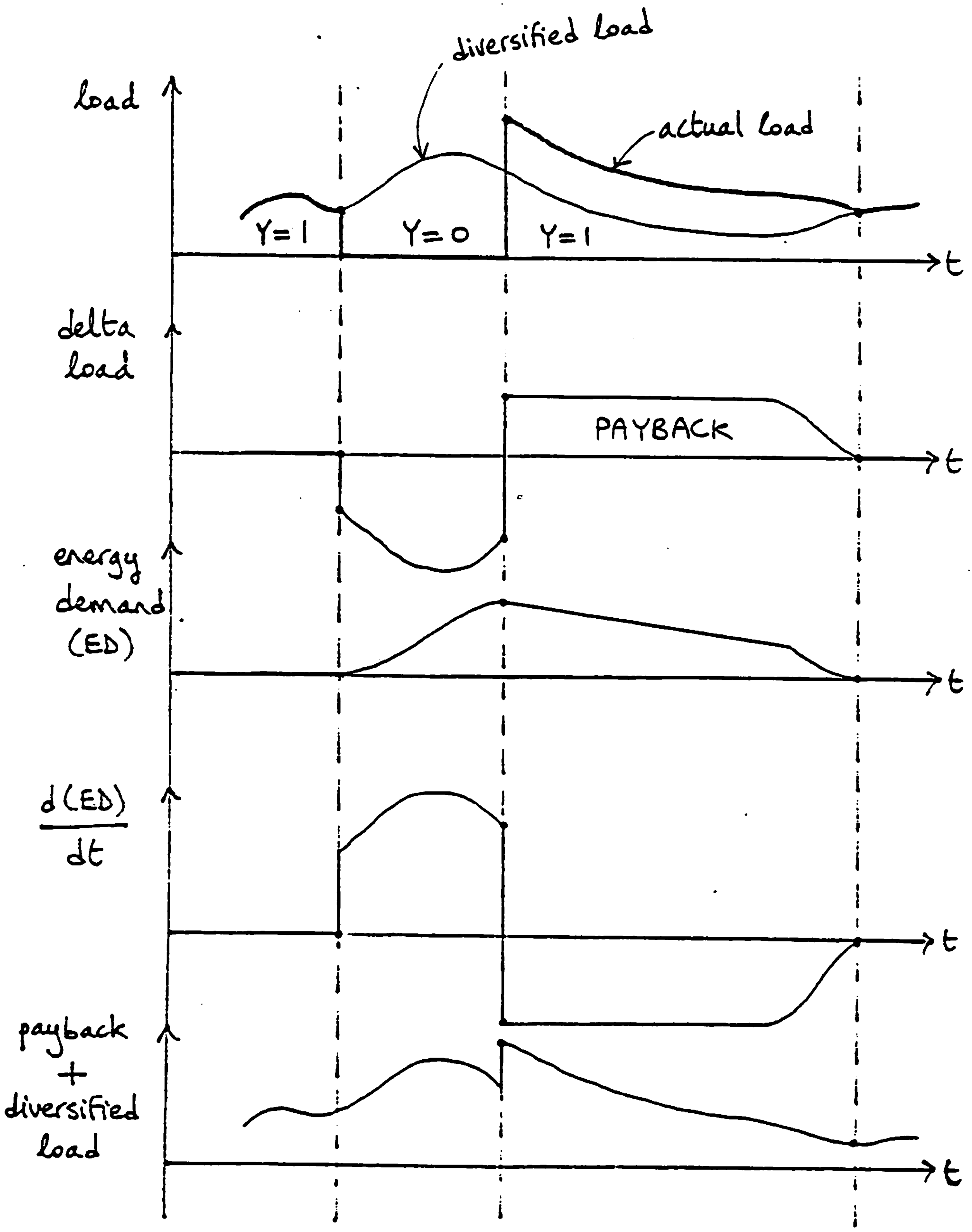


Figure 2.3: Controllable load definitions

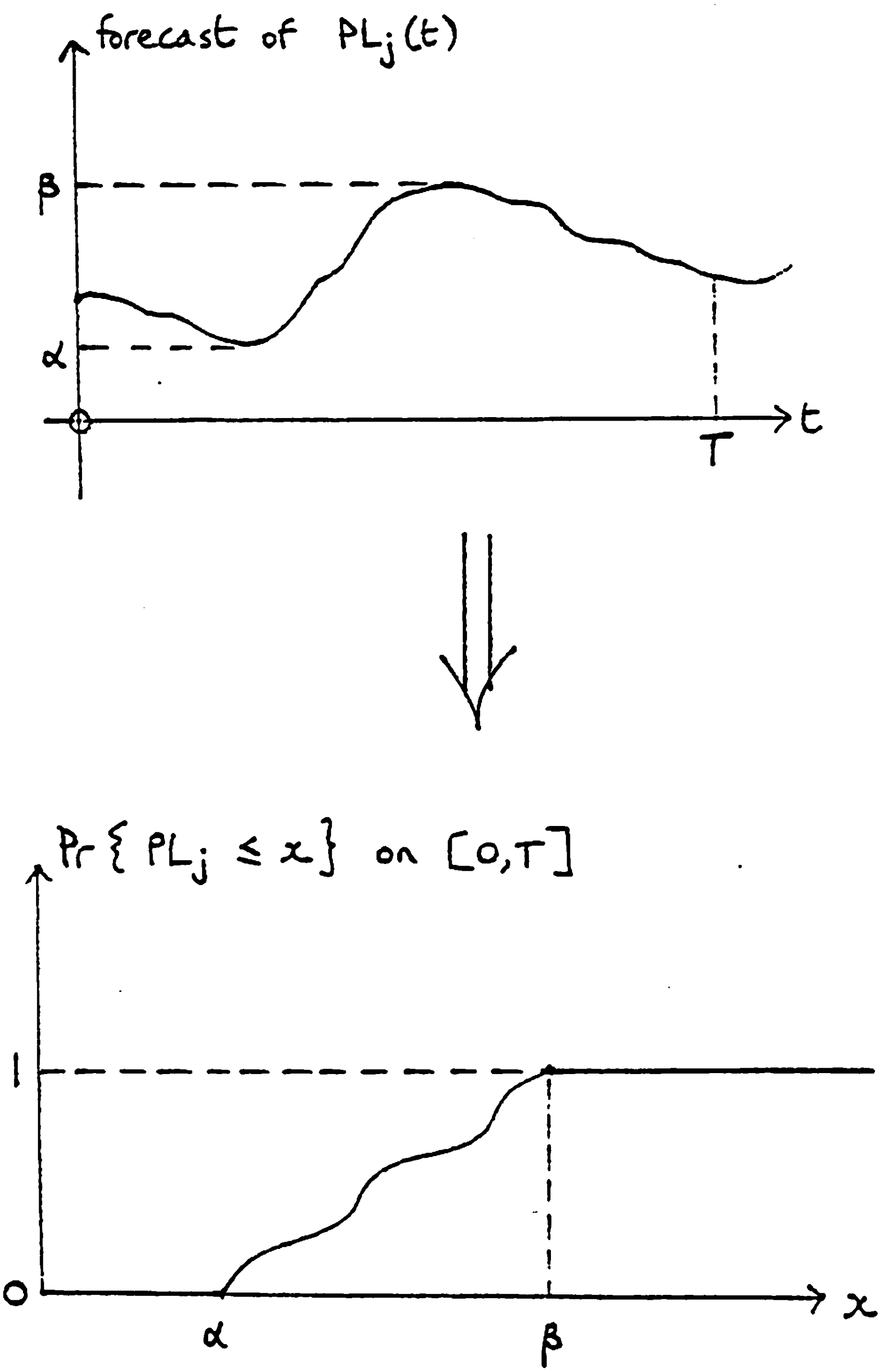


Figure 2.4: Load as a random variable

we can consider $PL_j(t)$ as a random variable with a known distribution.

The energy demand of the j^{th} load group at time t, $ED_j(t)$, is assumed to satisfy a differential equation of the form

$$\frac{d(ED_j(t))}{dt} = -Y_j(t) \cdot PB_j(t) + (1 - Y_j(t)) \cdot \alpha_j \cdot PL_j(t) \tag{2.45}$$

where

$Y_j(t)$ is the j^{th} control variable : $Y_j(t) = 0$ means the j^{th} load group is disconnected at time t, $Y_j(t) = 1$ means it is connected at time t, and intermediate values of $Y_j(t)$ indicate a cycling or partial disconnection;
$PB_j(t)$ = payback of the j^{th} group at time t;
α_j = efficiency factor : if $\alpha_j = 1$ then all the deferred energy during a control period would have to be 'paid back'.

As in [15], we model $PB_j(t)$ as a function of $ED_j(t)$:

$$PB_j(t) = min(\beta_j \cdot ED_j(t) , PMAX_j - PL_j(t)) \tag{2.46}$$

where

β_j = device-dependent constant;
$PMAX_j$ = nameplate load of the j^{th} group.
See Figure 2.5.

Because of this relation between payback and energy demand, we can rewrite equation 2.45 more compactly as

$$\frac{d(ED_j(t))}{dt} = f_{1,j} (ED_j(t) , Y_j(t) , PL_j(t)) \tag{2.47}$$

We assume for the moment that $PL_j(t)$ is given over the interval [0,T], ie. we take the forecasted load as being the diversified load. Define J-dimensional vectors ED(t), Y(t), and PL(t), whose j^{th} components are $ED_j(t)$, $Y_j(t)$, and $PL_j(t)$ respectively. Then we have the following vector differential equation for the energy demands in the system :

$$\frac{d(ED(t))}{dt} = [f_{1,1}(ED_1(t) , Y_1(t) , PL_1(t)), \ldots , f_{1,J}(ED_J(t) , Y_J(t) , PL_J(t))]^T \tag{2.48}$$

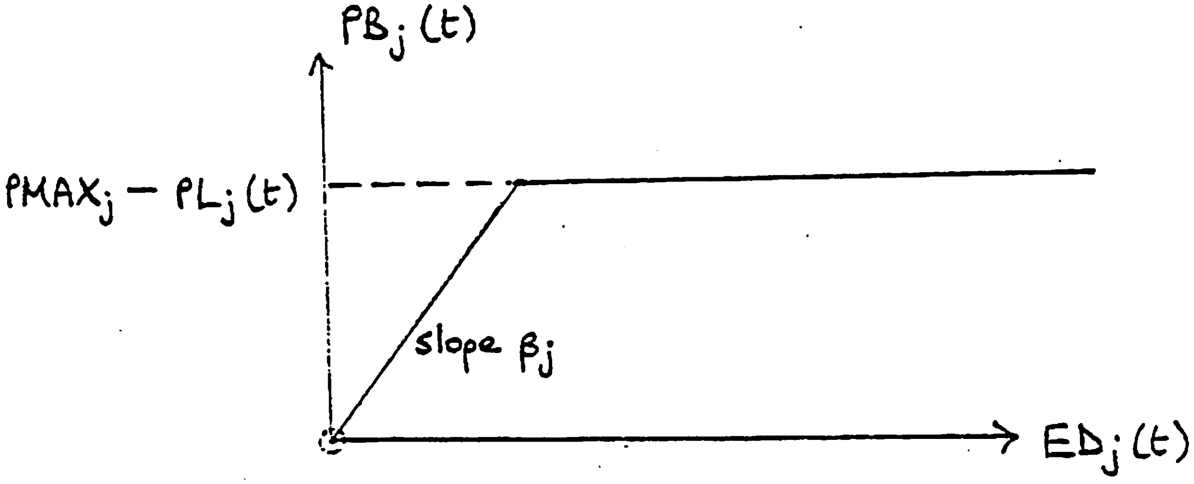


Figure 2.5: Payback as a function of energy demand

or, equivalently,

$$\frac{d(ED(t))}{dt} = f_1(\, ED(t)\, , Y(t)\, , \, PL(t)\,) \tag{2.49}$$

Let Y^j denote the set of control strategies for the j^{th} group which satisfy the constraints that the minimum time between control periods is T_j and the maximum time under control is $T^{j,max}$, for some constants T_j and $T^{j,max}$; then we choose $Y_j(t) \epsilon Y^j$, $j = 1, \ldots, J$. In order to limit the inconvenience suffered by customers subjected to direct load control in [0,T], we also impose the constraint $ED_j(t) \le ED_j^{max}$, $j = 1, \ldots, J$ for some device-dependent ED_j^{max}.

We can define the system diversified load at time t, L(t), as

$$L(t) = \sum_{j=1}^{J} PL_j(t) \tag{2.50}$$

and the system delta load at time t, DEL(t), as

$$DEL(t) = \sum_{j=1}^{J} (\, Y_j(t) \cdot PB_j(t) \; + \; (\, Y_j(t) - 1\,) \cdot PL_j(t)\,) \tag{2.51}$$

The objective function we wish to minimise is then

$$C = \max_{t \epsilon [0,T]} (\, L(t) + DEL(t)\,) \tag{2.52}$$

To approximate this minimax problem, we replace C with C_a, where

$$C_a = [\int_{t=0}^{T} (\, L(t) + DEL(t)\,)^p \, dt\,]^{1/p} \tag{2.53}$$

for p large; indeed, in the limit as p tends to infinity, C_a tends to the infinity-norm of $(\, L(t) + DEL(t)\,)$ on [0,T], which is C. Combining the expressions for L(t) and DEL(t) yields

$$C_a = [\int_{t=0}^{T} (\sum_{j=1}^{J} Y_j(t) \cdot (\, PB_j(t) + PL_j(t)\,)\,)^p \, dt\,]^{1/p} \tag{2.54}$$

Recall that we assume $PB_j(t)$ depends on $ED_j(t)$ and $PL_j(t)$ through equation 2.46, and so we can define

$$PB_j(t) + PL_j(t) = f_{2,j}(\, ED_j(t)\, , \, PL_j(t)\,), \; j = 1, \ldots, J \tag{2.55}$$

or in vector notation,

$$PB(t) + PL(t) = f_2(\, ED(t)\, ,\, PL(t)\,) \tag{2.56}$$

and so we can write

$$C_a = [\int_{t=0}^{T} < Y, f_2 >^p \; dt\,]^{1/p} \tag{2.57}$$

where we have dropped the time-dependence for clarity and the inner-product is defined as

$$< x, y > = x^T \cdot y$$

Let the optimal control vector be Y^* with corresponding energy demand vector ED^*, and define $f_2^* = f_2(\, ED^*\, ,\, PL\,)$. For any control vector we have $Y = Y^* + \delta Y$, for example if we had already computed Y and we somehow knew Y^* then δY is the correction we would make to Y to achieve optimality. Corresponding to Y is energy demand $ED = ED^* + \delta ED$, and linearising $f_2(\, ED\, ,\, PL\,)$ about the optimal point yields

$$f_2(\, ED^* + \delta ED\, , PL\,) = f_2^* + Jf_2 \cdot \delta ED \tag{2.58}$$

where the *Jacobian* $Jf_2 = (\, \frac{df_2^i}{dED_j}\,)_{i,j=1}^{J}$ is time-varying because it is evaluated at $ED^*(t)$.

Note that we have $C_a = [\, \int_{t=0}^{T} < Y^* + \delta Y\, ,\, f_2^* + Jf_2 \cdot \delta ED >^p \; dt\,]^{1/p}$, so expanding the inner-product gives the following optimality conditions :

if $< \delta Y\, ,\, f_2^* > = 0$ *and* $< Y^*\, ,\, Jf_2 \cdot \delta ED > = 0$ *and* $< \delta Y\, ,\, Jf_2 \cdot \delta ED > = 0$ then Y is optimal.

The optimality conditions give us three equations in the four unknowns Y^*, ED^* (which defines f_2^*), δY and δED. However, by equation 2.49 with Y and ED as above, linearising about the optimal values gives

$$\frac{d(\delta ED)}{dt} = Jf_1 \cdot \delta ED + Df_1^Y \cdot \delta Y \tag{2.59}$$

where again the Jacobian Jf_1 and J-by-J matrix of partial derivatives Df_1^Y are time-varying because they are evaluated at Y = $Y^*(t)$, ED = $ED^*(t)$. The solution of this differential equation relates δY, Y^* and ED^* on [0,t] to $\delta ED(t)$, and so provides the fourth equation needed to (hopefully) solve for the unknowns.

We assumed above that control variable for the j^{th} group, $Y_j(t)$, could be varied continuously between 0 and 1; in practice this is achieved by cycling the load group on and off during the period in question so that on average over the period the desired fraction of the load is presented to the system. We also assumed continuous control, in which case the control vector traces out a trajectory in the space of control variables over the interval [0,T]. We examine now the discrete-control case, ie. the control settings are considered for adjustment every δt seconds. In this discrete-time case we must replace equation 2.45 with a difference equation for the energy demand of the j^{th} controllable group :

$$\frac{1}{T} ED_j(k+1) = \frac{1}{T} ED_j(k) - Y_j(k) \cdot PB_j(k) + (1 - Y_j(k)) \cdot \alpha_j \cdot PL_j(k) \qquad (2.60)$$

The payback function of the j^{th} group is still modelled as in equation 2.46 with the continuous-time variable t replaced by the discrete-time variable k. Analagously to equation 2.49, by defining J-dimensional vectors ED(k), Y(k) and PL(k), we can define a function $f_1(\cdot, \cdot, \cdot)$ such that

$$ED(k+1) = f_1(ED(k), Y(k), PL(k)) \qquad (2.61)$$

This discrete-time problem is solved by Dynamic Programming (DP) techniques, where we regard the energy demand as the 'state' of the DP problem and so equation 2.61 describes the evolution of the state in discrete-time. However, conventional DP solves for the optimum controls given the final state, and changing the final state means the problem must be re-solved. In our case we require the solution starting from the current (known) state, and so the method of solution is *forward DP* [16]. Forward DP allows us to examine various final states so that we can estimate the sensitivity of the optimal controls to variations in the operating conditions at time T, and has the practical advantage that, when the optimum is found up to time k, the optimum at time k+1 uses the calculated optimum at time k as an initial state and so the problem does not have to be re-solved at each new time instant. Thus if we learn the state of the system at any time during the control period, we can reschedule the controls using this new information as the initial condition, without affecting the optimality of the solution before this time. The drawback with forward DP is that the optimal controls determine what the previous state *should* have been, not the next state, and so if a deviation from the optimal state sequence occurs (as would happen if we learn the state at time k and it differs from the calculated optimal state at time k), a new set of optimal controls must be computed. However, since it is desired to select the best control strategy, starting from a known initial state, under various contingencies, forward DP is the better choice.

Up to now we have assumed that the diversified load for the j^{th} load group is deterministic and given on [0,T]. Of course in practice the load is randomly varying and we treat $PL_j(t)$ as a random variable with a known distribution on [0,T], as explained before. In the stochastic case we solve the forward DP problem to find the minimum expected cost-from-start, but apart from this modification the solution method is as outlined above.

Implementation issues :

As mentioned in the Introduction, load management techniques - including, but not limited to, direct load control - are more attractive to utilities with relatively low load factors. A low load factor indicates that the average load over the period in question is considerably lower than the peak, suggesting that large savings can be obtained if the use of high-cost peaking units can be avoided by peak-shaving since the peaking units are presumably not in service for most of the period. A relatively high load factor, on the other hand, means that these peaking units are on for a greater fraction of the period under study, and so more 'intrusive' load control may be called for to avoid their use. We can use the load factor over [0,T] to help decide whether or not to run the direct load control function : from the probability distribution of PL(t) on [0,T], we calculate the load factor and if it is below some specified threshold we assume the use of peaking units will be such that significant benefits will result from direct load control .

The approach taken above produces control schedules for the J controllable groups so that the *system* peak load is minimised, and thus we cannot guarantee that $(Y_j^*(t))_{t=0}^T$ will be the optimal control strategy for the j^{th} group considered on its own. This set of controls could give rise to locally sub-optimal operation if the behaviour of the j^{th} group affected other groups, and so only if the j^{th} group is in some sense independent of the other ($J-1$) groups will the globally optimal controls also be locally optimal. In practice this condition will be approximated if the load demand of the j^{th} group is small compared to the system load demand, and identification of such groups leads to reduced communications requirements since $Y_j^*(t)$ can then be calculated using only local information.

Among the criticisms made of direct load control , perhaps the most important one for a utility increasingly concerned with maintaining customer satisfaction in the more competitive environment of the future is that of *customer inconvenience* . Two arguments can be advanced in favour of direct load control : first, since the times of greatest likelihood of interruption for those customers agreeing to load control could be made available to customers, participating customers would have the option of rescheduling their demand patterns to offset their lower reliability of service at times of system peak. Second, even if their loads are interrupted, customers will presumably be forced to examine their electricity usage and decide which components of it are essential, next in importance, and so on; this increased awareness of their demand characteristic should stimulate more efficient use of power and improved energy conservation. On a more philosophical note, one might argue that the disadvantages of direct load control related to customer inconvenience are best left to the customer to consider, and that customer quality of service is enhanced when the choice between agreeing to direct load control or not is available regardless of the customer's decision.

The presence of the payback phenomenon can cause problems when direct load control is used since it increases the load presented to the system after loads that have been under control are reconnected. It seems reasonable to suppose that those loads which are most likely to be candidates for direct load control, such as water heaters and air conditioners, are precisely those which best fit our model of the load as an energy-storage device and thus can be expected to exhibit some degree of 'memory' with respect to deferred energy once they are brought back into service. A recent survey of utilities which practice some form of direct load control [17] reported that the restoration of load which had been under control can cause increased peaks, high inrush currents and/or a temporary loss of stability. This problem, referred to as cold-load pickup, calls for further actions to maintain satisfactory operation as discussed in Section 2.2.3, and complicates the decision about whether to run direct load control or not.

In the worst case, direct load control could induce wasteful energy consumption on the part of participating customers who anticipate interruptions and adjust their demand to avoid the consequences, and then are not interrupted. One problem often raised in connection with the use of direct load control as a system resource is its availability [18]. Clearly, load control is only an option when the controllable loads are connected; however, it has been claimed that the controllable loads are those which drive the system into peaking conditions in the first place and thus if they are not in service the peaking generators will not be in use either. This argument suggests that one criterion (from the utility's point of view) for choosing loads to be interruptible is that peaking units can always be avoided if enough of the controllable load is disconnected. The controllability and verifiability of direct load control depends only on the communications system used to carry the control signals and gather load data and is not dependent on customer response. We have seen above that, in theory at least, direct load control can be rescheduled in real-time in response to changing system conditions or as more information becomes known about the system. In general, load control possesses the qualities necessary to be considered as a system resource, and the magnitude of the relief obtainable through its implementation should determine whether it is the appropriate control in a particular operating condition.

2.3.2 Real-time pricing

Real-time pricing refers to any scheme in which the price set by the utility for its electricity reflects the time-varying costs of supply, and is updated as system conditions change over time so that the utility may regard price as a control signal. We can think of real-time pricing as a form of *indirect load control* ; in subsection 2.3.1 we looked at direct load control , where the utility selects and implements control strategies affecting utility-owned equipment (such as switches, cycling devices, and so on), whereas in indirect load control the customer selects and implements consumption strategies affecting customer-owned equipment.

Interactions between a utility and its customers related to the use of price as a control signal must satisfy four basic requirements [19] :

- **economic efficiency** : customers should be encouraged to adjust their usage of electricity to match utility marginal costs, perhaps subject to revenue reconciliation and transaction costs;
- **equity** : a particular customer's charges should reflect the utility's costs to serve that customer (ie. eliminate or reduce cross-subsidies between customers);
- **customer choice** : customers should be provided the opportunity to specify their reliability of supply and usage pattern from a menu of options;
- **utility control** : the engineering constraints associated with planning and operating the system should continue to be met.

Examination of existing utility-customer transactions reveals that they do not fulfill the above requirements. For example, flat or time-of-use rates are related to a utility's marginal costs only on the average over the billing period, and certainly do not account for the detailed variations in the incremental cost of supply that give rise to the fluctuations in the utility's short-term marginal costs. Cross-subsidisation between customer classes is common, where customers who reduce demand at times of system peak pay as much as those whose demands cause the peak. As mentioned before, the role of the customer in present-day power systems is a passive one, with little choice offered to the customer concerning the conditions of delivery of their electricity except the option to demand or not.

PG&E has since 1985 conducted a real-time pricing experiment to investigate whether customers would respond to hourly price signals broadcast 24 hours in advance by adjusting their demand characteristics [20]. As of 1987 it had been established that the price signals could be accurately and reliably transmitted to participating customers, who were drawn from the large commercial and industrial load classes. However data presented in [20] indicates that customers do not respond in any appreciable way to the hourly price variations. This is because customers do not review the real-time prices often enough to adjust their demands. However, load-shifts from weekdays to weekends and from summer to winter suggest that customers are willing to alter their demand patterns once there is evidence of possible savings. We note here that a dilemma exists in trying to decide which type of customer should receive price signals in order to maximise benefits to both utility and customers; industrial and large commercial customers offer larger potential reduction but are usually constrained in rescheduling their demands by production schedules and labour availability, while smaller commercial and residential customers offer comparatively small reductions individually but have greater freedom to re-shape their demand characteristic and allow the possibility of a smoother controlled load profile. Another problem which makes it more difficult to start a real-time pricing project is the necessity - as borne out by the PG&E experience - for customers to devote more time than usual to deciding their demand patterns, and possibly investing in automatic load controllers or cycling devices, in order to maximise their benefits. Other factors specific to the PG&E experiment which make it difficult to draw general conclusions are :

the increased on-peak prices imposed by PG&E to ensure that there would be no loss of revenue - if a significant number of customers were involved in the scheme, as opposed to

the 9 reported in [20], we would expect these on-peak prices to be lower and consequently have less effect on customer responses;
the communication system used (Western Union's EasyLink notification system) has only been tested under the artificial conditions of the demonstration project - since our interest in this report is the integration of the various functions into a Distributed Automation System, where many different communications signals will be in transit at any one time, the reliability of the price broadcast must be regarded as unknown in the general case;
no attempt was made in the PG&E experiment to account for the diversity in customer demands and preferences, and thus the element of customer choice was still essentially lacking.

The four requirements for satisfactory utility-customer transactions mentioned above can be met by treating power as a product traded on an open 'energy market' [19]. Of course since this product cannot be stored cheaply we impose a balance constraint between supply and demand which must be satisfied at all times. The necessary conditions for such an open market are

- the supply side should have varying supply costs
- the demand side should have varying demands which are capable of responding to changes in price
- some mechanism for buying and selling the product
- no monopolies on either supply side or demand side

It can readily be verified that electrical energy meets these conditions. Indeed, marginal fuel costs vary as different generators are brought in and out of service; customers demands vary continually and can be adjusted as the price is varied, at least to some degree; the introduction of communication and control equipment allows a two-way flow of information between utility and customer in real-time; and neither side permits monopolistic behaviour - the supply side is regulated to restrain market manipulation, while on the demand side there are too many customers in most utility service areas to allow even the largest to dominate the others. Thus, in theory at least, electrical energy may be viewed as a commodity which can be bought and sold in real-time, subject to certain operating restrictions.

Spot pricing

The principle underlying spot pricing of electricity is that an appropriate charge for a unit of energy is the *marginal cost* to the utility of generating that energy and supplying it to the customer. Since this marginal cost depends on the load demand currently being met, and since this demand is varying randomly, the utility must continuously adjust the price it charges its customers if this principle is to be followed. Thus the price a customer sees reflects the actual cost the utility incurs in supplying them with electricity given the current load demand. Note that the utility may include a revenue reconciliation term in the spot price to recover its embedded capital costs, in which case the price seen by customers includes a factor representing the investment in plant and equipment that the utility made in order to supply them with electricity. The essential difference between flat or time-of-use prices and spot prices is in the frequency with which each is calculated : at present, electricity rates are adjusted once a year whereas we will consider re-computing the spot price every hour.

There are two fundamentally different approaches to the use of price as a control signal in a power system. One approach, which we will denote the 'market-clearing' approach, attempts to ensure the desired balance between supply and demand by raising the price at on-peak times to drive enough customers out of the market that the demand falls off, and lowering the price at off-peak times to market otherwise idle capacity. The second approach, which we refer to as the 'cost-covering' approach, is not concerned with attempting to influence the load demand but rather adjusts the supply as needed and passes the costs on to the customers; in this approach the price is treated more like an information signal since the utility is not trying to cause a particular customer response. There are also two different ways of calculating marginal costs; in short-run marginal costs no account is taken of future capital expenditures resulting from present operating decisions, while long-run marginal costs include the effects of current operation on future outlay. We will see that a natural way of classifying customers is based on whether they affect future marginal costs or not, and customers whose loads are large enough that they exhibit significant inter-temporal linking see an adjusted price (compared to smaller customers) which can be thought of as arising from consideration of long-run marginal costs.

Problem formulation :

Our discussion is based on results presented in [21]. Since future states of the system are affected by current operating decisions, we need a dynamic theory of pricing and we solve the spot pricing problem on an interval [0,T]. Denote the decision times by k, where k takes on discrete values in [0,T]. Let the operational decision for the n^{th} participant at time k be $d_n(k)$ and let $x_n(k)$ denote the state of the n^{th} participant's plant at time k. At time k some of the factors affecting future decisions are unknown; for example, if the n^{th} participant is the utility then random generator outages will affect its ability to supply the load, while if the n^{th} participant is a customer then temperature variations make the usage pattern of their air-conditioner impossible to predict exactly in advance. We model this uncertainty by a random process $\eta_n(k)$ whose statistics we assume are known. Define $y_n(k)$ as the net consumption of the n^{th} participant at time k, ie. electrical energy consumed less energy generated. Finally we have the notion of a benefit to the n^{th} participant at time k, $b_n(k)$. With these definitions we can define a *participant model* as follows :

$$x_n(k+1) = X_n(\, x_n(k)\,,\, d_n(k)\,,\, \eta_n(k)\,,\, k\,) \tag{2.62}$$

$$y_n(k) = Y_n(\, x_n(k)\,,\, d_n(k)\,,\, \eta_n(k)\,,\, k\,) \tag{2.63}$$

$$b_n(k) = B_n(\, x_n(k)\,,\, d_n(k)\,,\, \eta_n(k)\,,\, k\,) \tag{2.64}$$

$$d_n(k) \in D_n(\, x_n(k)\,,\, \eta_n(k)\,,\, k\,) \tag{2.65}$$

Let the price announced at time k be $\pi(k)$, then the net profit seen by the n^{th} participant at time k is

$$NB_n(k) = b_n(k) - \pi(k) \cdot y_n(k) \tag{2.66}$$

and profit maximisation for the n^{th} participant requires choosing $d_n(k)$ to satisfy

$$\max_{d_n(k)} [\, NB_n(k) + E(\sum_{j>k} NB_n(j) \mid x_n(k)\,,\, \eta_n(k)\,)\,] \tag{2.67}$$

For the purposes of deriving the conditions for socially optimal behaviour, suppose that there exists a 'central decision maker' which chooses all the $\{d_n(k)\}_{n=1}^{N}$ where N is the number of participants. Define

$$Y^b = \{ \{ y_n(k) \}_{n=1}^N \in \{ Y_n \}_{n=1}^N \mid \sum_{n=1}^N y_n(k) = 0 \} \quad (2.68)$$

Then the central decision maker ensures socially optimal behaviour by choosing $\{d_n(k)\}_1^N$ to satisfy

$$\max_{\{d_n(k)\}_1^N} \sum_{n=1}^N (b_n(k) + E(\sum_{j>k} b_n(j) \mid \{ x_n(k), \eta_n(k) \}_{n=1}^N)) \quad (2.69)$$

under the constraint that the corresponding $\{y_n(k)\}_1^N \in Y^b$, which is the balance constraint between supply and demand.

Each participant receives the current price $\pi(k)$ and a forecast of future parameters, and is assumed to choose $d_n(k)$ to satisfy equation 2.67. The problem can then be stated as follows :

find the price $\pi(k)$ such that, for each participant, $d_n(k)$ chosen to satisfy equation 2.67 is also the decision satisfying equation 2.69 .

In [21] these decision problems are formulated as Dynamic Programming problems which lead to Kuhn-Tucker optimality conditions, and the relation between these optimality conditions which results in individual profit-maximising coinciding with socially optimal behaviour is derived. As in subsection 2.3.1, we note that forward Dynamic Programming is needed since we start at a given initial condition and determine the optimal solution as time evolves.

The customers can be divided into two classes : those whose behaviour influences future marginal costs and those which display no such *inter-temporal linking* . It is shown in [21] that for small participants, short-run marginal pricing will induce socially optimal behaviour; and since we assume that this marginal cost cannot be predetermined, this means that spot pricing must be used to attain the optimum. Large participants - ie. those which display significant inter-temporal linking - are made aware of the effects of their current decisions on the future benefits of all participants through an additional term in the spot price they see :

$$SP_L(k) = SP(k) + c_L(d_L(k) , \{ d_n^*(k) \}_{n=1}^N , E(SP(k+1) \mid x_L(k) , d_L(k) , \eta_L(k))) \quad (2.70)$$

where

$SP_L(k)$ = spot price seen by large participant L at time k;
SP(k) = spot price seen by all small participants at time k = short-run marginal cost at time k;
$\{d_n^*\}_1^N$ = optimal decisions for all participants at time k;
and $c_L(\cdot)$ is such that

$$c_L(k) \;=\; 0\,,\; d_L(k) \;=\; d_L^*(k) \qquad (2.71)$$

$$> \;0\,,\; otherwise \qquad (2.72)$$

Note that we have not assumed any revenue reconciliation components in the calculation of the spot price. Suppose we wish to include a provision to recover embedded capital costs. In [19] one scheme for recovering investment costs is outlined, in which the marginal costs used in the above to calculate the spot prices are multiplied by a factor of $(1 + m)$; m is chosen to satisfy the utility's revenue requirements with regard to a reasonable rate of return as well as recovering capital costs. For example, if the utility's capital costs are low enough that annual revenue from marginal cost-based spot prices exceeds revenue requirements, $m < 0$: such a case might occur if the utility had limited generation capacity, high fuel costs and low capital costs. On the other hand, a utility with high installed capacity and low fuel costs would suffer a revenue shortfall if its spot prices were based solely on marginal costs and so $m > 0$.

The above formulation is sufficiently general to include the case where some customers generate electric energy independently of the utility and sell any power in excess of their own needs to the utility. The spot price which the utility pays the customer is called the *buy-back spot price* , and is calculated on the basis of marginal costs of production as before. In [19] the spot price is split into three components : operating cost terms, quality of supply terms, and revenue reconciliation terms. The first two components are independent of whether the customer is buying from or selling to the utility and so, if revenue reconciliation is ignored, the spot price and the buy-back spot price are the same. However, revenue reconciliation is not symmetrical and we need different values of m - say m_{sell} and m_{buy} - to calculate the two spot prices. We have also neglected consideration of transaction costs in the above discussion; the need to install and operate the communications and control devices which allow the energy marketplace to function imposes an additional term in the calculation of the spot prices. Of course in the context of Distribution Automation these communication costs may be incremental given the range of functions implemented and so can be removed from the spot price computation.

Priority service pricing

In Subsection 2.3.1 we assumed that, under direct load control, a customer can designate part of their load as controllable and the remainder as uninterruptible. Due to the randomly-varying nature of *both* the system load demand *and* the supply-side ability to meet that demand, however, it is more meaningful to quantify reliability of service in terms of the probability of meeting the demand. Thus instead of just two types of load - controllable and uninterruptible - we consider various levels of reliability, where a higher reliability is equivalent to a lower likelihood of interruption. It should be noted that we do not assume that every component of the system load demand is controllable; instead we simply acknowledge that the supply of electricity to any load is uncertain, and that the utility can control (at least to some extent) the likelihood of supplying any particular load. In order to obtain the benefits accompanying the use of direct load control, a utility will have to defer or deny some loads at on-peak times to avoid bringing more expensive peaking units into service, and so it must have a schedule of interruptions giving the order in which loads should be controlled to maximise the benefits. Clearly, a load for which the customer has accepted a lower reliability of service should be interrupted before one with higher reliability, and it follows that there is a direct relationship between the probability of meeting a load and the cost of meeting that load. This leads to the concept of **priority service pricing** .

Priority service pricing is a form of product differentiation that increases the range of choices available to customers, where the product in this case is the *reliability of service* to a load as measured by the probability of that load being met [22]. The idea is to unbundle this reliability into a number of *priority classes* , each priced to reflect the cost to the utility of providing that quality of service. The priority classes make up a menu of options offered to participating customers, and every customer chooses a level of priority service for each component of their load demand. We note that electricity supply to customers has traditionally been highly reliable - the low probability of interruption usually due to the redundancy built into the system at the planning stage. However, the wide variation in the values different customers place on service reliability, and the variation in these values for the different components of a typical customer's load, indicate that customers require a range of reliability levels rather than one uniform level across their demand characteristic. In order to be able to offer such a range a utility must design its priority service menu to ensure that enough customers choose lower-priority service to enable the provision of the higher reliability expected by those choosing higher-priority service.

There are some load curtailment schemes in use at present, in which typically a few large customers are offered lower rates and interruption compensation in return for allowing their loads to be controlled by the utility for the express purpose of load-shedding in low reserve margin operating conditions. Priority service differs from this approach in its underlying principles : all customers are offered essentially the same menu of priority service options, and in times of reduced reserve capacity their load increments are interrupted in reverse order of their *interruption costs* as indicated by their selection of priority service levels. Of course, quantifying their interruption costs is likely to be difficult for most customers, and the utility cannot expect to have such detailed knowledge about each customer, but as experience is gained with the scheme the relative priority levels desired for the load increments can be interpreted as the relative magnitudes of these costs. In practice the time over which customer choices of priority levels are binding will be the shortest period over which such an arrangement is economically feasible, and we will not consider how this period could be determined since it depends largely on technological factors. However, it is useful to regard spot pricing as a special case of priority service pricing in the limiting case where the forward contract period goes to zero, since it brings out the essential difference between the two schemes. Spot prices are revised continuously, in theory at any rate, and are simply energy charges based on energy consumed between price updates. Priority service contracts have a longer time horizon and the price should be the expectation of the spot prices in the events of receiving power under the chosen priority levels [22]. In effect the prices for priority service are demand charges reflecting the capacity needed to sustain each priority level.

The existence of a number of customers requiring low-priority service allows a utility to increase the reliability of service to those customers willing to pay more for higher priority supply within a given capacity allocation, and so there is a better matching of customer needs to service received than under present-day conditions. This improvement in customer satisfaction is hard to calculate but will play an increasing role in utility decisions as the competition to supply power intensifies. As with direct load control , the use of priority service can substitute for incremental peaking capacity additions until the load has grown sufficiently to justify additional baseline capacity, with resulting decreases in fuel and transmission costs. System security is enhanced by the additional flexibility a utility has in dealing with insecure states, since an orderly and equitable way is available to drop load to co-ordinate with the necessary switching actions. Revenues are increased when a utility sells lower-reliability power at reduced prices from otherwise idle capacity. From a planning point of view, the price customers are willing to pay for the highest priority level provides a direct indication of the value customers place on incremental capacity additions.

Customers benefit from priority service pricing because it reduces the sum of their appliance investments, energy and service charges, and expected interruption costs [22] . Aggregate interruption costs are lower and those customers choosing lower priorities are compensated appropriately. The introduction of priority service will benefit most those whose interruption costs are much higher or much lower than the average. Customers with very high interruption costs may find that the charge for high priority is small compared to the savings obtained by less frequent interruptions; for example, manufacturers whose production processes rely on uninterrupted electricity supply. On the other hand, loads with very low associated interruption costs can offer rate reductions which exceed the effects of more frequent disconnections; for example, rural customers whose agricultural pumping can be deferred at virtually no cost.

Implementation issues :

A priority service menu should have the following features [22] :

- customer preference diversity should be exploited in order to group customers according to their willingness-to-pay to avoid interruption
- product differentiation with respect to power delivery attributes that affect the cost of electricity (such as voltage level of service or service reliability) OR for which customers have diverse preferences (such as warning time before an interruption)
- differential pricing of the various service conditions in such a way that those customers with the highest interruption costs are induced to choose the highest priority level and other customers are induced to choose less reliable service

Under these conditions, customer preferences are revealed by their choice of priority levels; and while customer interruption costs are assumed to be unknown to the utility, these preferences are interpreted as signals of the relative magnitudes of the associated interruption costs since these costs presumably influenced the choice of priority levels. The design of a priority service menu involves setting the rates in such a way that moving to the next higher bracket of interruption costs induces the selection of the next higher priority level. In this way lower priority choices are implicitly compensated by higher priority choices, as required if the scheme is to be equitable. For example, consider the highest priority level : power at this priority level will continue to be supplied until all lower priority power has been interrupted. Thus the charge for the highest priority power should equal the marginal expected interruption costs incurred by the lower priority power in order to exempt the highest priority level from being interrupted. This is an example of the principle of *economic exchange* and forms the basis of a fair and efficient rate schedule. Such a pricing structure is necessary to encourage customer self-selection in a way that indirectly reveals their interruption costs [22] . As a further consequence, note that the utility needs only the *preference distribution* over all its customers, and not individual customer interruption cost data, to design a menu that induces a distribution of service contract selections that meets its supply constraints.

There are many different possible implementations of a priority service-based market, the fundamental differences being who takes responsibility for the predictions used to set priority levels and their associated prices. In the type of scheme considered up to now, the utility predicts interruption frequencies or guarantees a maximum number of interruptions for each priority level during the period of the contract. Customers then self-select from the resulting menu in such a way that interruption costs correspond to priority levels chosen. It is essential that the distribution of customers' selections should lead to system operation that justifies the probability assessments on which they were based - this is referred to as the *balance constraint*. Responsibility for satisfying the balance constraint rests entirely with the utility, and it must bear the risk of customer selections not being realisable given the supply constraints. The other extreme involves the utility making available to its customers a menu of priority levels and associated charges with no guarantee on the reliability of service at any of the levels. The utility also publishes supply and demand projections for the period of the contract, and it is the customer's responsibility to interpret this information and estimate the frequencies of interruption for the various levels.

Of course (since this is impractical for the majority of customers) we might expect the utility to give non-binding interpretations of its predictions, but the risk of violating the balance constraint is shifted to the customers. Another suggestion [22] involves the utility selling 'priority points' to customers which are designated by each customer as belonging to components of their load; the utility interrupts first those load increments with the fewest priority points, then those with next fewest, and so on. Again customers have to estimate the likelihood of interruption as a function of the number of priority points, based on past data and utility-provided predictions. This type of approach exposes the customer to risk with respect to their electricity supply, and if customers are risk-averse then either the utility or independent parties could offer interruption insurance : if this insurance is offered independent of the priority level subscribed to then the premium should reflect the level chosen, while if the insurance is tied to a particular priority level it can be shown that it is optimal for the utility to interrupt in order of increasing compensation owed as a result of the interruption.

One question that arises in connection with priority service is, when a customer's load is interrupted how does the customer know that all lower priority loads have been interrupted, thus necessitating the interruption of their higher priority load ? Clearly, in the case that customers are compensated for having their load interrupted, the utility will interrupt in increasing order of the resulting compensation. The utility may offer higher compensation to lower priority levels if other factors make this desirable, for example from continuity-of-supply considerations. If compensation is not being offered, it is also possible that the customer with higher priority is being interrupted while lower priority load demands continue to be met; unless the utility is behaving irrationally this would presumably be because of other reasons such as load shedding to prevent overloads from arising. These considerations show that customers can sometimes be 'unnecessarily' interrupted to benefit other customers, when such interruptions are called for taking the system as a whole into account.

In practice we expect to implement both spot pricing and priority service pricing in a Distribution Automation System. Yet the concept of a varying price for electricity seems to defeat the purpose of a forward contract between utility and customer. How can the utility (or anyone else for that matter) predict probabilities of meeting the various levels of service when the price varies with time over the period of the contract ? Shortening the contract period improves the accuracy of the predictions but we assume that the shortest feasible contract period is still several multiples of the update period for the spot prices. The solution is, for each priority level there is a cost to the utility to maintain that level of reliability; thus for each level we can calculate the marginal cost of supply at that level, which defines the spot price for that level of service. Customers choosing a particular priority level are in fact choosing the associated spot price to be the price of the electricity consumed at that level. Thus the notion of priority service has to be modified to account for the time-varying price of electricity, and with this modification the two pricing schemes may be combined.

Wheeling

Wheeling refers to the transfer of electrical energy from a supplier to a consumer, where this transfer is over the transmission and/or distribution network of a third party referred to as the *wheeling utility* , denoted W. The transfer of energy between remote suppliers and consumers is conceptual, and in practice W uses the energy from the supplier to meet part of its load demand and generates the energy specified by the wheeling transaction at the interconnection with the consumer. Wheeling rates are used to determine the payments to be made to W by the supplier or the consumer, or both, to compensate for the generation and network costs incurred by W as a result of the wheeling transaction. The optimal wheeling rate for a given transaction is the marginal impact of wheeling on W's costs, assuming that an incremental change in W's net interchange with the consumer is matched by an equal and opposite change in its interchange with the supplier and that the scheduled net interchange of all other interconnected utilities is held constant. As with the other forms of real-time pricing we have considered, the wheeling rate structure should be such that suppliers and consumers are induced to make 'efficient' decisions based on operating costs, projected demands, and so on. For example, a supplier should only wheel power to a remote consumer when the expected benefits outweigh the fraction of the wheeling rate it must bear; a consumer should only buy power wheeled by W from a remote supplier when the cost (wheeling transaction costs borne by the consumer + rates for wheeled power) is lower than that for power from any other source.

The supplier may be a utility or an independent power producer (IPP), that is, a customer with some installed capacity who can supply power to the grid. The consumer may be another utility or a customer, or perhaps a group of customers pooling their demands in order to avail of reduced rates for bulk loads. Wheeling between utilities is usually referred to as *wholesale wheeling* , while wheeling from a utility to a customer is referred to as *retail wheeling* . Since wholesale wheeling involves only the bulk power system, we focus our attention on retail wheeling which must by definition impact the distribution system. In order to allow for the possibility that the power being wheeled is generated by an IPP, we extend the definition of retail wheeling to include all wheeling transactions in which customers are involved. In terms of a utility implementing Distribution Automation, the case where the supplier and consumer have no interface through which to wheel power is irrelevant since the utility's Distribution Automation System in itself is not enough to permit the transaction; the customer's utility must also have some means of monitoring the detailed behaviour of the customer so that the specified net interchanges can be verified.

These observations imply that wheeling in the context of Distribution Automation refers to the transfer of power from an IPP to another customer or customers in the utility's service area using the utility's distribution network. The wheeling rate in this case may include a component enabling the utility to recover its embedded capital costs and some rate of return, although it is still possible that the utility's revenues are decreased compared to those it would obtain if it supplied the consumers. There are two main reasons why a utility might find it desirable to allow retail wheeling even if it involved a loss of revenue. First, it increases the range of options available to the utility's customers and thus increases customer satisfaction. Since the utility may be forced by regulatory restrictions to accept IPP-generated power, it makes sense to include this source of power as an option to customers to determine the demand for wheeled power. Second, the designation of IPP's power as a system resource implies that the utility can apply the same operating constraints to this source of power as to its own generation, and thus issues of availability and reliability of IPP power - which are among the objections put forward by utilities - can be quantified and compared to the utility's operation on an equal footing.

Further discussion on some of the issues involved in the wheeling of electric power is provided by [23].

2.3.3 Remote metering

Remote metering refers to the capability for a utility to collect customer data by transmitting this data over a communications link between the customer and a data-processing node. In its simplest form this function requires only one-way communication (from the customer to the utility), but we will consider the more general case where two-way communications are possible. A basic assumption is that every customer meter has some microprocessor capability : specifically, we assume that the meter can store data in digital form and can initiate communication with at least one designated processing node.

Remote meter reading involves the storage of customer kwh data in the meter's memory, from which it as accessed when required by the utility for billing purposes. In a one-way communication environment, the reporting schedule of each meter is set in the meter's hardware and any changes to this schedule require a visit from utility personnel. Apart from this manual intervention, which we expect will be infrequent, no labour costs are incurred in the collection of customer usage data, thus saving the approx. $10 per meter per year this collection costs a typical utility at present. This saving by itself would not justify the implementation of remote meter reading; however, a digitally-based meter can also provide voltage readings and switch status data which would be required for other Distribution Automation functions such as voltage regulation, direct load control and capacitor switching. In particular, if both load voltage and load current are monitored, then both the real and reactive power consumed by the customer's load can be computed and stored. Knowledge of reactive power consumption is the basis of *Var billing* , and could be used to determine the actual cost to the utility of meeting the customer's load demand in cases where the load power factor is significantly less than unity and capacitor switching is infeasible or undesirable. Applications like these represent potential 'opportunistic' benefits to the utility once the decision has been made to incorporate this function into a Distribution Automation System, since they can be added at negligible incremental cost once the equipment is in place.

Remote meter reading is a *background task* for the Distribution Automation System, in the sense that it does not involve critical real-time operations and so can defer to other functions in times of restricted access to system resources. In the type of operation we have been considering, this suggests the need for each meter to have a retry facility when denied access to its designated receiver, as well as some means of recognising that data cannot be transmitted. In a two-way communications setup, on the other hand, meter reporting schedules can be downloaded from a control processor and thus no retry facility is required : the data transmission is simply deferred and this fact noted so that the data can be requested once normal system operation has resumed.

The existence of a two-way communications path between a processing node or nodes and a customer's meter allows the utility to send control signals to the customer by *remote programming of the meter* . As well as the data deferral and subsequent collection described above, the utility could control how often the meter transmitted its stored data and what readings it sent, so that in normal operation only customer usage data is transmitted unless the load survey function discussed in Section 2.4 requested additional readings. The direct load control function could scan those meters associated with controllable loads (or some representative fraction of them) to determine the extent of the available load relief. When the processing node assigned to a meter or meters is known to be unavailable, the utility can re-route the metered information to another data processor rather than defer its collection. Of course, implementation of such additional functions gives rise to increased communication traffic, and thus the operational enhancements due to this remote control capability must be traded off against longer delays in data transmission and/or an increased error rate.

2.3.4 Customer services

In this subsection we suggest other functions a utility might offer its customers as part of its Distribution Automation System. Initially at least the emphasis will be on 'opportunistic' functions which can be offered at small incremental cost once the investment has been made in an underlying communications and control system. As more experience is gained with the system by both utility and customers, other functions which are presently infeasible may be added; the system should be flexible enough to permit such modular expansion.

Monitoring of customer load data (cf Section 2.4) allows a local microprocessor-based device such as that used in the customer's meter to compare measured voltages or currents against preset thresholds and issue alarms if the measured quantities are found to be unacceptable. For example, if a refrigerator is running for extended periods when the air temperature is not abnormally high, the door seals or the coolant may be faulty and a warning from the control device would avoid the higher repair or replacement costs incurred if the problem is not dealt with [7]. The status of electronic alarms could be monitored and the tripping of any alarm could automatically trigger a request for help to security personnel or the police, as well as logging the sequence of events for later analysis. This is easily adapted to devices used to monitor disabled, ill or elderly customers, where significant changes in the measured parameters could trigger help signals to the appropriate emergency services. The use of such 'intelligent' monitoring devices is not limited to responding to events; if the customer has access to computing power as well as a two-way interface with the utility, they could control the various electrical devices in their home simply by software commands acting on the relevant addresses in memory. For instance, a customer subscribing to spot pricing may wish to set a series of thresholds to which the price signal is compared - depending on the outcome of this comparison, the customer's load demand is automatically adjusted to improve their usage efficiency, as well as setting a maximum price the customer is willing to pay and above which their load demand is automatically curtailed.

The installation of a communications and control system between the utility and customers, and the investment by customers in some form of computing capability such as a personal computer, allows the possibility of a range of services being offered to a customer in their home. Applications such as home shopping and banking, which have already received initial testing, are among the likely candidates for in-home services. Tele- and video-conferencing could provide for the real-time exchange of information, and remote education and training facilities with interactive features could become feasible. The entertainment industry can be expected to generate both new applications and the customer demand to justify their inclusion in the system. As technology advances, other functions not possible under present conditions can be expected to arise, and a system designed to integrate the operational requirements of a utility and its customers is ideally placed to take advantage of such trends.

2.4 Data Monitoring And Processing

The acquisition and transfer of data is of critical importance to the correct operation of a Distribution Automation System. A typical function requires input data, often in real-time or from real-time measurements, which must be accessed from a storage location and processed, and the results passed in real-time to appropriate locations. The storage and processing may be centralised or distributed, and the required data rates vary depending on the function's priority, but the basic problems of data acquisition and processing within a given allowed error rate are the same. We note that the notion of 'real-time' depends on the function being considered : for example, load-shedding at times of peak load or under emergency conditions might require decisions on which loads to disconnect to be made within a few seconds, while in a spot pricing scheme the spot price might be updated once every hour. We consider the problem of data transmission in Chapter 3, and so in this section we concentrate on issues such as what should be measured and how these measurements should be used to generate the necessary control actions.

Each function in a Distribution Automation System imposes certain data requirements on the system. For example, capacitor switching for loss minimisation (Section 2.1.2) needs the system load variation curve (such as might be produced by a load forecasting program), the real and reactive power injected into the line at the substation, and the bus voltage magnitude, as well as capacitor cost functions and the values of the line parameters. Each function also produces data, which may be used to actuate controllable devices, or as input to other Distribution Automation functions, or stored for later reference. Continuing the capacitor switching example, the function produces decision vectors containing the optimal size and status of each switchable capacitor installed in the system, as well as calculating the corresponding state of the system. The capacitor switching function should be integrated with the voltage regulation, feeder reconfiguration, relay co-ordination, load control, real-time pricing and Var billing functions, which involves ensuring that the data produced by any of these functions is accessible to any of the other functions when needed. Note that identifying a group of functions which share data can lead to savings, since for example the average data requirement for each of the functions can be reduced relative to the case where each function is executed independently of the others.

The data used by the various Distribution Automation functions is stored in a database, access to which is controlled by an applications program called a *database manager*. We observe that data obtained from the monitoring devices placed on the system is typically not in a useful form, and so some processing or data concentration is required before putting the result into the database. In the capacitor switching example, we compute the real power consumed by each branch using the Distflow equations but we are only interested in the system real power loss associated with each load level in order to find the optimal solution (equation 2.12). Thus the database manager should have temporary storage available in which intermediate quantities can be stored and accessed without the delays inherent in the use of a system database. The operation of the distribution system imposes a *priority structure* on access to data stored in the database. For example, in emergency conditions the service restoration function should take precedence over functions concerned with normal operation (such as loss minimisation) and background tasks (such as remote meter reading), while in normal operation remote meter reading should defer to loss minimisation if they compete for system resources. Functions should be prioritised based on their requirements for real-time data as well as their impact on system operation, so that functions which need data in a matter of seconds (such as load shedding) are serviced before those whose data requirements are less strict (such as voltage regulation). The database manager should be able to recognise higher-priority requests and defer lower-priority requests until the higher-priority ones are serviced without losing any requests for data.

We discuss briefly three possible applications for a Distribution Automation database.

1. **Power quality diagnostics** : as well as monitoring voltage magnitudes for loadflow purposes, we monitor voltage harmonic magnitudes, durations of transients, occurrence of spikes in the voltage waveform, system frequency, and any other quantities which provide an indication of the integrity of the electricity supplied to customers with respect to its nominal characteristics. The issues of how many measurements should be taken, where they should be taken, and how they should be interpreted, will be resolved when experience is gained with their values under various operating conditions and their usefulness as indicators of the quality of supply. Such experience will lead to improvements in the quality of power used by customers - resulting in longer lifetimes for customer appliances and increased customer satisfaction - and the better understanding obtained about the dynamics of distribution system operation can be taken into account in future planning and operating practices, presumably leading to 'better' decisions.

2. **'Expert Monitoring'** : refers to the application of contingency analysis techniques to the distribution system. For example, on the basis of monitored quantities we decide to invoke the network reconfiguration function since some lines are loaded much more than others, leading to an insecure state where slight changes in the conditions of operation will cause overloads. One suggestion in this area is the *intelligent alarm processor* used to co-ordinate incoming alarms and screen them for possible false alarms. An incoming alarm is logged and its effect on the state of the system and other quantities of interest simulated; then as these quantities are measured, those alarms which produce a match between simulated and observed values are accepted. The incoming alarms can also be

ranked as to their importance according to the simulated effects they have on the various quantities. Another possibility is to use the difference between a measured parameter and its predicted value (based on other measurements, historical data, and so on) to trigger an alarm if this difference is greater than a tolerance chosen on the basis of previous conditions known to lead to an emergency.

3. **Load survey and forecast** : We proposed in Section 2.3.3 that customer load data be collected automatically using a communications link between a processing node and a microprocessor-based meter located on the customer's premises. To monitor the effects of operating strategies which affect the customer's load demand, such as direct load control or capacitor switching, the utility needs to sample this telemetered data and estimate the relevant parameters based on this subset of the systemwide data. The real-time data obtained in this way can then be used to develop more effective operating strategies and to identify where the predicted results do not match the observed outcomes. Such data can be used as the basis for estimating the indices used in planning system expansions, such as participation and load factors. The data samples can also be used as the input to a load forecasting model which exploits the assumed correlations between demand and variables such as time of day, temperature, and demand immediately prior to the sampling instant to predict the load demand a certain time in advance. The load model on which this approach is founded can also be updated and refined as more recent data is obtained about changes in the operating conditions, making this model *adaptive* to the state of the system.

Chapter 3

Communications

A typical power system is an interconnected collection of densely-connected subsystems, characterised by voltage level of service, geographical location, and so on. Each subsystem is operated on the basis of assumptions about the conditions of the other subsystems, and since these assumptions are usually not founded on real-time knowledge of the system the overall strategy is sub-optimal. This suggests the need for a communications system that, by providing each subsystem with data about the rest of the system relevant to the subsystem's operation, will allow the power system as a whole to be operated more effectively. In the context of a Distribution Automation System (DAS), the communications network is a critical part of the design, since the operation of each function imposes its own data requirements (both input and output) which vary widely among the DAS functions, as noted in Section 2.4. Without data transfer between the various system components, the potential for co-ordinating the DAS functions to meet system-wide objectives is lost.

In this Chapter we discuss some of the issues involved in the design of the communications system in a DAS. We first outline the common types of digital communications networks, concentrating on packet networks due to their growing importance. The layered model of a packet network is introduced and the issues at each level mentioned. Performance metrics which permit alternative system configurations to be compared are defined. The different media which can be applied to power systems are listed and two of the more common ones - broadcast radio and distribution line carrier - are discussed. A recent alternative to conventional narrow-band signalling is spread spectrum, and we indicate the advantages it has over narrow-band signalling schemes. Finally we mention some of the choices available when deciding on the topology of the communications system.

3.1 Network Issues

The information to be transferred between the microprocessor-based devices located in the distribution system (and in the bulk power system) is inherently digital, so there is no practical alternative to digital transmission of this data. Digital communications networks usually allow *switching*, where a user can initiate connections to specific other users. In particular, when the network must be able to handle simultaneous transmission of many bit-streams - as in Distribution Automation - switching capability must be provided to reconfigure the point-to-point connections in the network. There are two basic types of switching : **circuit-switching** and **packet-switching**.

Circuit-switching is a method of digital communication in which the sender transmits data to the receiver over a dedicated end-to-end path through the network which is maintained for the duration of the transmission. For example, most telephone connections are circuit-switched for the duration of the call. The network transports a constant-rate bit-stream between the transmitter and receiver for a relatively long period of time (ie. minutes or longer), using time-division multiplexing (TDM) to combine several lower-rate bit-streams from the users into a high-speed bit-stream suitable for the inter-node links of the communication network. The bit-interval of this high-speed link transmission is called a *frame*. The user-generated input bits are interleaved by the TDM protocol such that each input is only connected to the high-speed link for a fraction of the frame called a *time-slot*, as shown in Figure 3.1. Since the receiver needs to know the slot boundaries and the order in which the input bit-streams were interleaved, the TDM protocol must insert control bits (called framing bits) into the output bit-stream. Due to this overhead, if each of N users transmits data at the same bit-rate B_i, the high-speed link works at bit-rate $B_o < N \cdot B_i$. The use of circuit-switching avoids having to provide enough transmission capacity for every possible connection, since only those connections likely to be required in the next period are assigned a time-slot in the frame by the TDM protocol.

The circuit-switching approach provides a fixed bit-rate (corresponding to a reserved time-slot in the frame) to each user in the network. A circuit-switched network accepts a fixed maximum number of users, above which further requests to use the network are blocked. It cannot give time-varying bandwidths to users whose requirements change with time, and so such users must be provided with a circuit based on their maximum bit-rate requirement. Thus for applications requiring different bandwidths at different times -

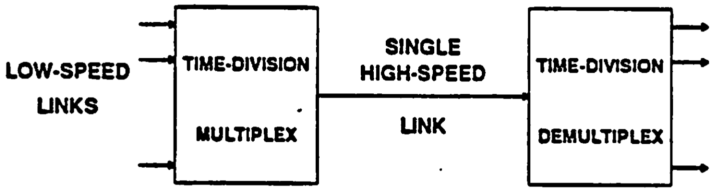


Figure 3.1: Time-division multiplexing

such as feeder load balancing, which during service restoration is a high-priority task but during normal operation might defer to capacitor switching or load control - the circuit-switched approach exhibits inherent inefficiency. Packet-switching is a method of digital communication in which messages are divided into groups of bits, called *packets*, and transferred to the receiver over 'virtual' circuits which are dedicated to the message transfer only for the duration of the packet's transmission. The bandwidth allocated to a user is proportional to the average bit-rate required rather than the peak, yielding an improvement in network efficiency which depends on the ratio of the peak to average bit-rate demanded by the service.

Packet-switching dynamically allocates the available bandwidth among the users, so that any given user 'sees' a variable bit-rate provided by the network. A packet is analagous to a time-slot in TDM, although typically a packet contains hundreds of bits compared to the eight per slot common in TDM. Packets are then interleaved on the high-speed inter-node link; the difference between this method and TDM is that, because the order of the packets is not pre-determined, the packet-switching protocol can accept more packets from higher-priority services than from lower-priority ones and so the higher- priority user is provided with a higher bit-rate, at the expense of longer queueing delays for the lower-priority packets. A packet-switching approach allows a user to transmit simultaneously to several other users without incurring the overheads associated with setting up

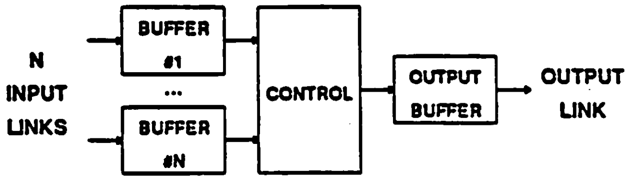


Figure 3.2: Block diagram of a statistical multiplexer

and taking down circuit connections to each one.

Since the beginning and end of a packet do not occur at prespecified times, packet-switching is *asynchronous* and control bits (called synchronisation fields) have to be added by the packet-switching protocol at the start and end of each packet. The packet-switching protocol is carried out by a device called a *statistical multiplexer*, which takes a number of 'packetised' input bit-streams and combines them into a high-speed output suitable for the network links, as shown in Figure 3.2 [24]. A possibility exists that the sum of the incoming bit-rates is higher than the link bandwidth - indeed, if the output bit-rate of the statistical multiplexer is always greater than the total incoming bit-rate, we might consider using a simpler TDM scheme. The statistical multiplexer addresses this problem by using internal buffers to hold input data when the link cannot support the total incoming bit-rate. It takes advantage of the statistics of the variable-rate inputs to make more efficient use of the link bandwidth, at the cost of increased device complexity and the queueing delays inherent in the temporary storage of input data.

Present digital communication systems provide constant bit-rate channels. Packet networks are therefore constructed by connecting packet-switching nodes together using constant rate bit-streams, ie. circuits. These circuits connect pairs of nodes, in contrast to the circuit-switching approach where the users are connected by an end-to-end circuit. The trend in digital communication is towards an integrated network access arrangement, in

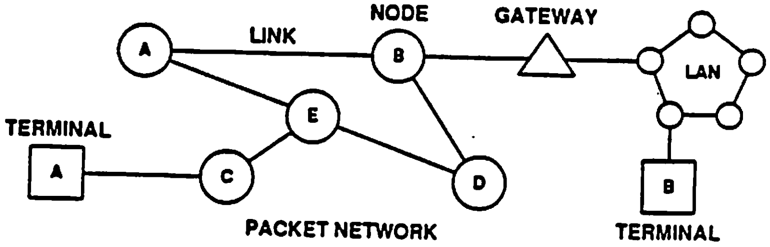


Figure 3.3: Part of a packet network

which the user would have access to a menu of different services (such as data transmission, voice, or video) and the capability to access more than one such service simultaneously through a single interface. The *Integrated Services Digital Network*, or ISDN, is an integrated access arrangement that has been standardised and undergone extensive field trials [25]. Integrated access would make the network appear 'integrated' to the user, although the services could still be supported internally by logically distinct subnetworks as at present. The integration of these internal communications subnetworks is the objective of *broadband ISDN* (BISDN), using some form of packet-switching as the basic protocol since it adapts easily to the bit-rate corresponding to the desired service and to time variations in this bit-rate. Due to its importance in the digital communications networks of the future, we pause to consider the general features of a packet network, before going on to discuss the various network layers at which design decisions must be made.

3.1.1 Packet Networks

A simple packet network is shown in Figure 3.3. The local area network (LAN), which connects users less than about one km apart, is itself usually a packet network, and the gateway is used to convert between the possibly different protocols used in the two networks. There are two basic methods of routing packets through a packet network : *virtual circuit* and *datagram*. Virtual circuit techniques are more common, and involve the establishment of a route through the network from one user to another along which all packets between this pair of users travel. The setting-up and dismantling of a virtual circuit is similar to those for a circuit in a circuit-switched network; however, there are two essential differences. First, in the present case, a number of virtual circuits may share a common physical circuit, unlike the circuit-switching technique. Second, the utilised bandwidth varies dynamically during transmission according to the demand, as opposed to the fixed bandwidth of a dedicated circuit. In virtual circuit routing, the address field in each packet identifies the virtual circuit associated with the packet rather than the destination address, and this information is stored in a lookup table when the virtual circuit is established.

In datagram routing, a packet is individually routed through the network without regard to its relation to other packets in the network. This form of packet routing is more difficult to implement, since the packet-switching nodes must decide where to route the incoming packets based on their knowledge of the state of the rest of the system; however, if communications bottlenecks develop and the rest of the system learns this, the overloaded links can be avoided. Packets contain the addresses of their source and destination, and routing is done using the destination address and whatever information the routing node has about the other nodes. Thus when this information is out-of-date or wrong, bad routing decisions can be made which will slow down the progress of a packet to its intended receiver, and so the data about the status of the packet-switching nodes should be regularly updated through high-priority messages. Two factors would reduce the resulting overhead : it may be that the increased delays suffered by packets due to inappropriate routing decisions do not degrade the performance of the application sufficiently to justify the extra computational burden associated with increasingly-frequent status updates. For example, if the service is a background task such as remote meter reading, the data is not needed in real-time anyway so even considerable slowing-down might be acceptable. Secondly, in a large network such as that needed to cover the distribution system of a typical utility, we expect that packets will pass through many intermediate nodes in going from sender to receiver. Thus most routing decisions will be *local* in nature rather than global, which means only the status of its neighbouring nodes is needed for a routing node to decide where to send the incoming packet.

A major difference between circuit-switching and packet-switching is in the nature of *blocking* and *delay*. Recall that, with circuit-switching, the network may block a request to set up a circuit due to a lack of available bandwidth on an intermediate link. Once an end-to-end connection has been established, a fixed bandwidth is allocated to the circuit and the delay between transmission and reception is fixed. In a packet-switched approach, the network may block a request for a virtual circuit due to excessive traffic at intermediate nodes. Once a virtual circuit is established, the available bandwidth is allocated dynamically and packets suffer time-varying queueing delays. This delay is statistical in nature, so we refer to the average delay, the delay distribution, and so on.

3.1.2 Network Layers

Among the issues addressed by the designer of a packet network are

- What access protocols will ensure efficient sharing of the communications channel among multiple users ?
- How can connectivity between the network nodes be determined ?
- How can this connectivity information be used to route data through the network ?
- How can reliable inter-node communication be achieved ?
- Should the network architecture be centralised or distributed ?
- What features are needed at the interface between the network and a user ?

The design of a packet network is complicated by the interactions between these issues, and so there is usually no 'optimal' design. The usual approach is to conceptually divide the network into *layers* and make design decisions by focusing on one layer at a time. However, these layers are in fact highly interdependent, and so the design process includes resolving tradeoffs between the requirements resulting from the decisions at each layer. The overall decision on the packet network architecture depends on the environment the network must operate in, the desired performance, the degree of flexibility required, the cost of the proposed design, and so on.

Three levels of design decisions are identified in [26] for the case of packet radio networks, although they apply also to the more general packet network considered here :

1. Physical
2. Network Management

3. User/Network Interface

1. Physical Level

Decisions at this level concentrate on issues affecting the transfer of data from one node to another. The Physical Level thus corresponds to the physical and data-link layers of the Open System Interconnection (OSI) model of a packet data network [24]. The *physical layer* establishes digital link connectivity between the nodes, while the *data-link layer* is concerned with the transfer of data between a pair of connected nodes.

Link connectivity from node A to another node, B, refers to B's ability to correctly receive digital information transmitted by A at a specified minimum rate. The neighbourhood of A is then the set of nodes that can exchange data with A. In multiple-access applications, the link connectivity of any pair of nodes is determined not only by the status of the pair but also by the status of the neighbourhood of either node. For example, in a packet radio network, node A may not be able to establish a connection to node B if B is already receiving a packet from a neighbouring node. The physical layer appears to the data-link layer as a bit-stream with associated bit-rate and operational requirements such as signal power. The advantage of the layer model is that different media or modulation methods can be substituted at the physical layer without affecting the operation of the higher layers : as long as the same bit-rate can be supported by the link, the same applications will be run as before, though of course the operational requirements may change. We will return to this flexibility in the choice of communications medium and signalling method in Section 3.3.

The function of the data-link layer is to ensure that a packet required by the layer above to traverse a route through the network is transmitted error-free over each individual link of the route. Thus the fundamental problem in this layer is the establishment of reliable communications between the nodes at the ends of a link. Among the issues affecting the successful transmission of a packet between a pair of nodes are the acknowledgement mechanisms (ARQ) used to notify the transmitting node of the reception of the packet by its intended receiver. Since the quality of a digital link may vary, ARQ is usually augmented by Forward Error-Correction (FEC) coding. For example, if the probability of error in detecting the bits in a packet is high enough, an ARQ-only scheme would result in low throughput since most of the packets would be rejected; the use of FEC would allow more packets to be accepted by their receivers, thus increasing throughput, at the expense of increased node computation time. The balance between the use of FEC and the resulting increase in computational burden should be capable of being varied from link to link, and also in time to track changing network conditions.

2. Network Management Level

Decisions at the Network Management level concentrate on issues affecting the movement of data through the network. In the OSI model these issues are addressed by the network and transport layers. The basic problem is the provision of an end-to-end communications path between two users using the capabilities provided by the functions on the Physical level. The services at the Network Management level must include packet routing strategies and system-wide flow control, as well as techniques for recovering the original message which was divided into several packets prior to transmission.

A fundamental question posed at this level is how two nodes should determine the existence of a logical link between them, and where this information should be stored. Direct observations of the link's 'logical level' can be made by the nodes, for example by counting the number of packets successfully transmitted over the link; however, such an approach has an inherent time-lag and thus a loss of connectivity will not be seen until several transmissions have possibly been attempted. Another suggestion is to monitor those parameters of the channel which can be used to decide if the link is available, for example signal power and signal-to-noise ratio on a radio channel.

Once the available links are found, the question arises as to whether they form an acceptable network for the applications considered. We might insist that every node have a certain minimum number of neighbours, or the users might impose a minimal performance requirement of some kind. The approach taken to this issue would probably fall between the extremes of, on the one hand, accepting the set of available links as given and attempting to fulfill the user needs without further adjustments to the network, and on the other hand taking the minimum requirements as 'hard' constraints and waiting until they are met before proceeding.

Routing strategies are broadly classified as either *flooding* techniques or *point-to-point* techniques. Flooding is a general broadcast aimed at a large number of nodes, and is useful when there is a need for high transmission reliability under uncertain connectivity or if the connectivity is changing too fast to be observed consistently by the nodes. Of course flooding is an inefficient use of the network, and so would probably be reserved for critical messages with wide circulation such as a load-shed command. Point-to-point methods differ mainly in the locations where connectivity information is stored, as discussed in Section 3.1.1. Generally speaking, if the connectivity is not changing rapidly, virtual circuit approaches are preferred, while in more dynamic networks higher channel efficiency is achieved by using a datagram-type approach because a virtual circuit cannot be guaranteed to exist for the entire transmission interval. Other issues to be considered are methods for disseminating routing information to the network and methods for controlling network congestion at peak traffic levels.

3. User/Network Interface

Decisions at this level concentrate on issues affecting the operation and maintenance of the packet network. This corresponds to the session, presentation and application layers of the OSI model. These layers use the end-to-end communications capabilities offered by the Network Management services to allow users to remotely access databases or mainframe computers as well as permitting inter-computer file transfers, information exchange via electronic mail, and so on. Questions which arise in connection with these layers include how software changes should be disseminated to the nodes, the tradeoff between hardware reliability and redundancy, and responses to node failures. The interfaces to other networks are also addressed by the services at this level, including control of access to the network and the operation of the inter-network gateways. In general the software which supports the services at this level is proprietary, which complicates integration with other networks. Thus there is a need for utilities to standardise their requirements for the inter-network interface.

3.2 Performance metrics for communications systems

In designing a DAS, there are usually several possible communications systems that will support the traffic generated by the proposed DAS functions. In order to be able to compare the different systems, we need to characterise their performance at the various network layers introduced in 3.1.2. This is done through the use of *performance metrics*, which can be estimated for each of the communications systems under consideration. Of course, the relative weights given to the metrics described in this Section are up to the utility, so we concentrate on a description of the metrics and how they reflect the performance of the communications network.

Recall from 3.1.2 that a communications network is divided into three levels, namely the **Physical, Network Management** and **User/ Network Interface** levels, and we identified each level with layers of the OSI model of a packet network. We now introduce metrics by which the performance of the network at each of these levels may be quantified.

3.2.1 Physical level

The Physical level of a communications network is concerned with the transfer of digital data from one node to another. Thus at this level we are interested in the performance of a single link in the network. The two most important measures of link performance are the *bit-rate* it can support and the *probability of bit error* for transmissions over the link.

More specifically, we can define the **transfer rate of information bits**, or TRIB, on a link from node A to node B as the rate at which bits from A are received without error at B[27]. TRIB is measured in information bits per second, where information bits are defined in Figure 3.4. We only measure the rate of transfer of address and data bits, and the *overhead* represented by the message preamble and error-protection bits is ignored. Thus the TRIB is independent of the methods used for synchronising communications or error control, and networks in which these functions are done differently can still be fairly compared. We can write

$$TRIB_i = \frac{N_i}{T_i} \tag{3.1}$$

for the transfer rate of information bits on the i^{th} link, where N_i is the average number of information bits accepted by the receiver and T_i is the average transmission time for those bits. The average time taken to send dead spaces (Fig. 3.4) and possible re-transmissions is included in T_i, and thus the TRIB is sensitive to the quality of the link.

The link quality can be specified by the **residual error rate**, which is probability that a bit in error is accepted by the receiver. The residual error rate for the i^{th} link, RER_i,

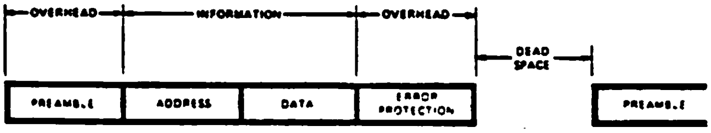


Figure 3.4: Message components

is given by

$$RER_i = \frac{E_i}{B_i} \tag{3.2}$$

where E_i is the average number of bits in error accepted by the receiver and B_i is the average total number of bits accepted. The RER will be lower than the link's probability of bit error (usually much lower) if the receiver has some means of detecting and correcting bit errors, such as that provided by Reed-Solomon error-correcting codes [24]. Taken together, $TRIB_i$ and RER_i give a good indication of the rate at which information may be reliably transferred over the i^{th} link. The advantages of some of the overhead bits which are ignored in calculating the TRIB. are taken into account in the RER, and so we get a balance between the rate at which information may be transferred over a link and the likelihood that it contains a certain number of errors.

An alternative way of defining link quality is in terms of the likelihood that a given packet is transmitted free of error over the link [28]. Suppose node A transmits a packet intended for node B. As noted in 3.1.2, other nodes within range of B can potentially interfere with the correct reception of the packet by B's receiver if they transmit during the packet interval. This attempt to transmit a packet from A to B is said to be *successful* if there are no errors in the packet at the output of B's receiver, and unsuccessful otherwise. The **probability of a successful transmission**, denoted P_s, is associated with a single attempt and does not distinguish between the first attempt to send the packet from A to B and subsequent retries (if needed). The probability of unsuccessful transmission, $P_u = 1 - P_s$, depends on the number of transmissions from other nodes within range of B, and on the type of signalling and error-correction coding used.

We assume the following operational constraint on the network : P_u^{max}, the maximum P_u over the set of links in the network, is less than some specified value. How is this value specified ? Note that, although P_u for a link of the network is a measure of *local* performance, it does affect *global* performance. For example, as P_u increases, a larger number of retries will be necessary along this link, increasing the delay for packet arrivals at their intended receivers. Thus the specified value for P_u^{max} must take into account the requirements on global performance.

The collection of nodes within range of B is called the *local population* at B, and the *local traffic* at B at a given time is the number of transmissions from such terminals taking place at that time. With respect to the packet being transmitted from A to B, the remainder of B's local traffic is referred to as *interference traffic*. Let the average local traffic be denoted by m, while the mean value μ of the interference traffic is called the *average interference level*. We assume that transmission processes for different nodes are independent and identically-distributed (iid). If the local population at B is small, both the local traffic and the interference traffic are binomially-distributed; if the local population at B is large enough, we can model both kinds of traffic by a Poisson random variable and $m = \mu$.

The maximum average local traffic allowed subject to the constraint that the packet error probability is not greater than P_u^{max} is denoted $m^*(P_u^{max})$. Suppose the network is homogeneous, in the sense that the probability distribution of the interference traffic is the same at each receiver. Then the *local throughput*, S, is the expected number of successful transmissions per packet interval by the local population, or in other words

$$S = m \cdot P_s = m \cdot (1 - P_u) \tag{3.3}$$

Since the probability of a successful transmission decreases as the average local traffic increases, S is the product of increasing and decreasing functions of m and is of the form shown in Figure 3.5. In particular, this implies that *we cannot increase the local throughput at B indefinitely just by increasing the local traffic at B.* The maximum local throughput that can be obtained under the constraint that the packet error probability is not greater

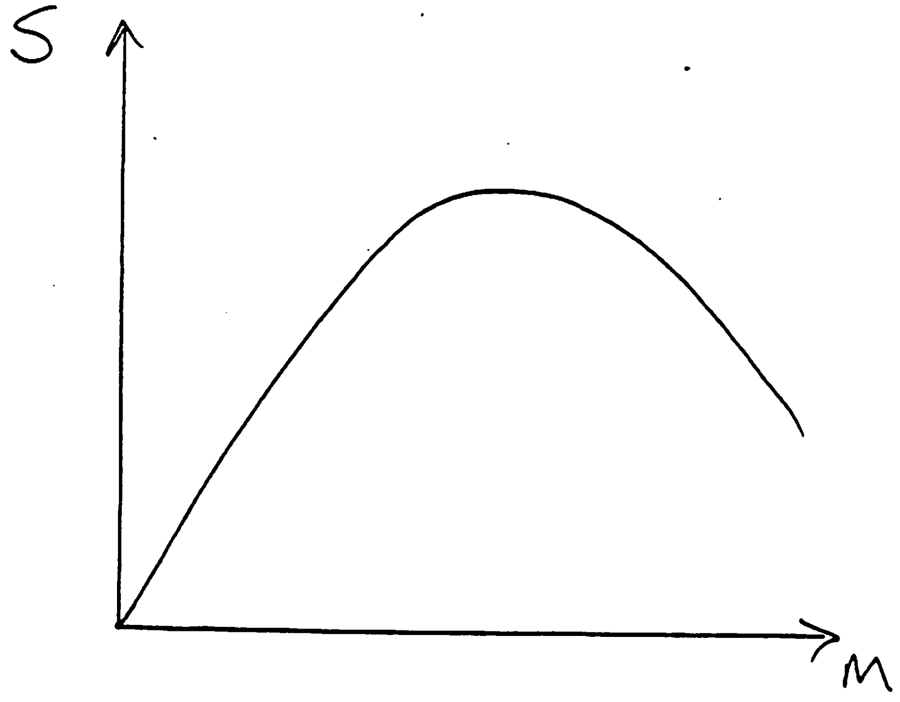


Figure 3.5: Local throughput as a function of local traffic

than P_u^{max} is denoted by $S^*(P_u^{max})$. This can be found from the formula

$$S^*(P_u^{max}) = \max_{m \leq m^*(P_u^{max})} \{ m \cdot (1 - P_u(m)) \} \tag{3.4}$$

Other quantities which indicate link performance include **link reliability**, **link availability** and **the probability of a lost packet**, P_{lp}. Link reliability refers to the likelihood that the link is operational - in a sense defined by the link's TRIB and RER, perhaps - for a specified period of time. Usually link reliability is measured by the link's estimated mean time before failure (MTBF). Availability refers to the portion of the time the link *could* be used, regardless of its actual usage, and is measured by

$$A = \frac{MTBF}{MTBF + MTTR} \tag{3.5}$$

where MTTR is the mean time taken to repair (or reinstate) a non-operational link. Thus MTTR includes device failures at the link nodes *and* link unavailability due to unacceptably high interference traffic. P_{lp} is the probability that the transmitting node does not receive an acknowledgement (ARQ) from its intended receiver within a specified time after transmitting the packet. P_{lp} depends on the balance between ARQ and forward error-correction coding (FEC) for the link, as mentioned in 3.1.2, so if this balance can be varied in time the probability of a lost packet will also be time-varying. Note that if the allowed response time is long enough that a re-transmission of the packet is accepted by the receiver, the packet is not designated as lost; this would be the case for background tasks such as remote meter reading, where meters are read only once a month and so the response time can be on the order of hours or days.

3.2.2 Network Management level

The Network Management level of a communications network is concerned with the end-to-end transfer of digital data through the network. In the light of the discussion in Section 3.1, we restrict consideration to packet data networks. We regard the end-to-end transfer of data from transmitter A to receiver B as occurring over a 'virtual' link between A and B. Then the metrics of 3.2.1 can be applied to this virtual link in order to characterise its performance. This approach to measuring network performance does not need the details of how packets are routed from A to B, since the metrics are computed for the virtual link connecting A and B.

Other performance metrics that can be applied to the end-to-end transfer of data from A to B are the **response time**, **first attempt success percentage** (FASP) and the **probability of a false alarm**, denoted by P_{fa}. The response time is the average time between the transmission of a packet by A and the reception by A of an acknowledgement from B that the packet has been correctly received. Thus re-transmission and propagation times are included in the calculation of the response time. The FASP is a measure of the frequency of re-transmissions and thus indicates the delay incurred by packets due to channel errors which result in the packet not being received correctly at B. However this indication of delay takes no account of packet propagation time, ie. no distinction is made between short and long virtual links. It is important to know how the average FASP for a network was computed when two networks are being compared. For example, measuring the number of receivers who correctly received a broadcast message and measuring the FASP of the virtual links over a given time period will usually give different answers for the network FASP. P_{fa} refers to the probability that a node receives a packet not intended for it, and so is a measure of how many packets which do not reach their intended receivers are *not* lost from the network. Note that with datagram packet routing 3.1.1, P_{fa} is actually greater than the probability of a packet being lost to a receiver other than its intended one, because the destination address contained in the packet header allows re-routing of the packet. Virtual-circuit routing techniques, on the other hand, do not permit such re-routing unless the routes are known to all nodes.

3.2.3 User/Network interface level

At this level the details of data transfer through the network are hidden and so the metrics of the previous two sections are not appropriate. The metrics used to describe network performance at this level are *qualitative*, in the sense that we usually do not need to accurately estimate them but instead we just rank various proposed networks as best, next best, and so on with respect to the various metrics.

Perhaps the most important characterisation of a network is the **cost** of using it. This cost includes start-up cost, operational costs and the incremental cost of expanding the system. Another important criterion from the point of view of a utility weighing up the installation of an extensive communication network is the **state of development** of the technology used in the candidate networks. As we will see in the next Section, some of the techniques suggested for use in a DAS are well-proven while others are still undergoing field testing. Thus there are risks associated with certain choices of network due to possible technical problems only coming to light in a large-scale implementation. Other factors influencing performance of the network from the user's perspective are its **maintainability** and **security from catastrophe**, which refer to the network's ease of repair and maintenance and dependence on individual components, respectively. For example, a network using a satellite link scores poorly on both counts, since the satellite transponder cannot be repaired and the system crashes if the transponder fails. Finally, the **average delay** experienced by an application may or may not be important, depending on whether the application needs to be carried out in real-time or not.

We note in conclusion that, regardless of which performance metrics are used to describe network performance or how they are calculated, there exists a need to standardise the metrics so that different communication systems may be compared as objectively as possible.

3.3 The design of a communications system

In this Section, we discuss some of the issues involved in designing the backbone communications system for a DAS. We first consider the possible communications media that have been suggested, then the possible signalling methods for the system, and finally some of the network design issues. The discussion is not intended to be a comprehensive review of utility experience with various communication schemes, though some implementation details are given where appropriate.

3.3.1 Communications media

In [29] the following classes of communications techniques are listed :

- physical media
  - metallic wires
  - coaxial cable
  - fibre-optic cable
- distribution line carrier
  - power harmonics
  - injected carrier
- broadcast media
  - AM/VHF radio
  - UHF/VHF radio
  - satellite radio
  - packet radio network
- common-carrier
  - dedicated line
  - dial-up line

These techniques are described in [29] and their equipment requirements for a 'generic' DAS tabulated. The generic DAS is a hierarchy consisting of a master station, submaster stations (ie. repeaters), remote terminal units (RTU's), transducers, and the communications links between them. This is an example of a *centralised* system, and is only one of a number of possible network configurations (as we will see later in this Section). The operational attributes of the various techniques are compared using some of the metrics defined in 3.2.3

along with issues such as system expandability in the event of traffic growth. To supplement the treatment in [29], we discuss briefly some practical issues faced by utilities that have implemented these techniques.

Distribution line carrier communications

The use of the power lines in the transmission system as a communications medium is well-established, where the need is for point-to-point communication and injected-carrier frequencies greater than 30 khz may be used to provide acceptable data rates. However, in the distribution system these carrier frequencies lead to excessive propagation and coupling losses through distribution transformers [30]. At lower frequencies, in the 5 – 10 khz range, these losses are reduced to feasible levels, but now harmonic noise causes unacceptable losses. Two solutions to this problem have been implemented : the use of specially-designed *receive filters* and *modulation of the 60 hz power signal*.

In [30] the use of a receive filter with a $\frac{\sin x}{x}$-type characteristic is suggested to combat the harmonic interference problem. See Figure 3.6 for the frequency response of such a receiver, where we note the nulls at the power frequency harmonics and the concentration of energy at the carrier frequency f_c, where f_c is given by

$$f_c \;=\; 60n \,+\, 30 \qquad hz \tag{3.6}$$

for n an integer. In practice, receivers phase-locked to the carrier frequency can provide reliable communications for signal powers 20dB below the adjacent harmonic peaks by operating in between the harmonic frequencies, as in equation 3.6 [30].

The other possibility is to modulate the 60 hz power signal directly. Such a scheme is the basis of the TWACS system developed for Emerson Electric Co. [31], where the outbound signal is derived from shifting the phase of the feeder voltage waveform to cause zero-crossing deviations and the inbound signal (from the remote devices) is due to load current pulses synchronised with the voltage signal. The phase-shifts are produced by injecting one cycle of the 60 hz wave, with voltage magnitude approx. 1% of the system voltage, in series with the voltage to be modulated so as to give two consecutive zero-crossing shifts typically around 40 μsec. each. The inbound current pulse is located around a zero-crossing of the voltage waveform, which is used as a reference, and is typically a half-cycle with peak 60 – 70 A superimposed on the bus or neutral current waveform.

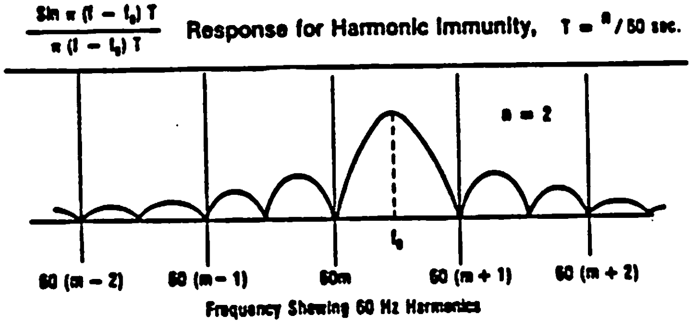


Figure 3.6: Frequency response of a receive filter to combat harmonic interference

Such direct-modulation (of the power-frequency signals) schemes have the advantage that the power system is tuned to allow transmission at this frequency, which simplifies the processing required to transmit data. The problem with this approach is that communication is lost with a remote device if the power is switched off, in contrast to the injected-carrier technique which remains operational unless the power line is physically broken. For example, in an emergency where switches are opened to isolate faulted equipment, neither technique provides communication to the devices downstream of these switches; but injected-carrier schemes allow us to communicate with the switches whereas direct-modulation schemes cannot work due to the absence of power. Since we need this remote communication capability for many DAS functions like feeder reconfiguration, service restoration and load control, we consider only the injected-carrier distribution line carrier methods from here on.

A major advantage of distribution line carrier schemes is that, ideally at least, *complete coverage* of the distribution system is provided. However, these schemes have some serious disadvantages which have limited their use by utilities up to now. One important limitation of such schemes is the low data-rates supported : the present ceiling among operational systems is about 100 *bits / sec.* Distribution line carrier systems require signal injection and conditioning equipment, and are dependent for their operation on the quality of the communication channel provided by the distribution feeder. Unfortunately the characteristics of this channel vary with time and location, since they are determined by momentarily-connected loads, the length and type of cable used in the feeder (ie. overhead or underground) and the distribution network topology [32]. In addition, the channel noise is *not* uniformly distributed over frequency, time or location. As shown in Figure 3.7 certain frequencies may be attenuated more than others, which makes it impossible to derive a systematic relationship between frequency and signal attenuation. Indeed, experiments in [32] show that the frequency with the highest signal-to-noise ratio (SNR) is not necessarily the one with the highest signal power. Thus, in a multi-receiver system, each receiver will see different signal and noise power distributions, implying that no single frequency is best for all remote locations. In [30] this problem is addressed by *frequency diversity*, where unsuccessful transmissions are re-transmitted on different frequencies to move out of communications 'holes'; in 3.3.2 we will see another way of overcoming this problem through the use of *spread-spectrum signalling.*

The Department of Energy, in conjunction with JPL, has developed a model for distribution line carrier systems which accurately describes the propagation and loss characteristics of radial distribution networks observed in practice [30]. The model also confirmed the experimental evidence that the natural signal propagation mode on three-phase feeders is zero-sequence (mode 3) in which the signal is coupled to all three phases in parallel with respect to neutral. For example, a signal coupled from one phase to neutral at a substation is a mode 3 signal $3 - 4$ miles down the line, due to mutual coupling between the phases. This is desirable since most harmonic noise is not mode 3 and may be cancelled by zero-sequence coupling - noise reductions of 15 or 20 dB are obtained in practice. The model predicted the presence of *standing waves* which have been confirmed in field tests : a voltage maximum occurs at an open-circuit, the distance between successive voltage minima is $\lambda / 2$ where the speed of propagation of light is $c = \lambda \cdot f_c$, and voltage minima and maxima are equally-spaced [33]. Thus voltage nulls will occur if the line-length is an integer multiple of $\lambda / 4$, which for $f_c = 10 \; khz$ is about 7 miles. The solution to the standing-wave problem proposed in [34] is to terminate the line in its characteristic impedance, since this will force the reflected wave to zero and the cancellation between injected and reflected waves that gives rise to voltage nulls will be eliminated. Since the structure of the feeder will change after reconfiguration, load control, and so on, the terminating impedance must be variable to match the line's characteristic impedance accurately. This technique does result in a 'flatter' voltage profile with no nulls, as shown in [34], but also reduces the magnitude of

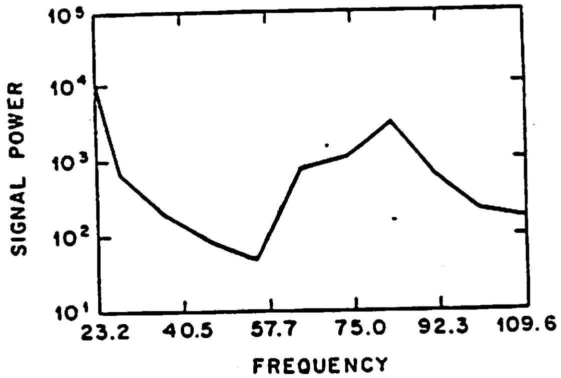


Figure 3: Typical Signal Power Variations vs Frequency as Measured on Laboratory 220 Volt Line

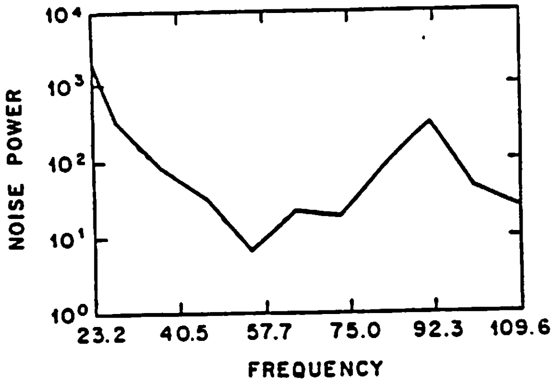


Figure 4: Typical Noise Power Variations vs Frequency as Measured on Laboratory 220 Volt Line

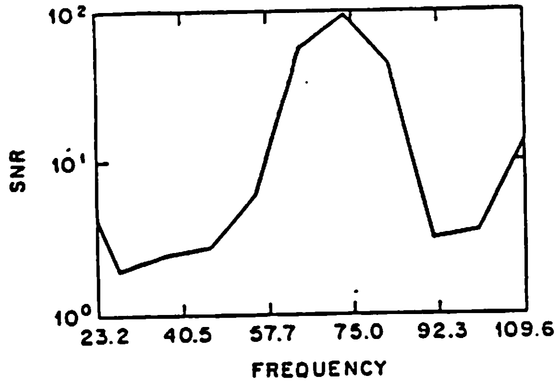


Figure 5: Typical SNR vs Frequency as Measured on Laboratory 220 Volt Line

Figure 3.7: Typical signal and noise power variations with frequency

the line voltage; so we have a tradeoff between improving the quality of the links to receivers at the nulls and decreasing the quality of the links to the other receivers on the feeder.

Broadcast radio communications

The use of broadcast radio techniques provides communications among nodes distributed over a broad geographical area in cases where direct physical connections are impractical. There are two basic types of broadcast radio techniques possible : *centralised* and *packet radio.* Actually, a centralised radio network is just a special case of the more general packet radio network design, but since it is the only technique with which utilities have practical experience, we discuss it separately. Note that satellite radio schemes are prone to catastrophic failure and appear too costly to be practical, despite their high data-rates and excellent system coverage, so we do not consider them in this report.

Centralised schemes may only support one-way communications, but in general we wish to monitor the system behaviour in real-time so we consider systems that permit two-way communication between the central station (Master) and the remote terminal units (Remotes) located on the feeder. In general such systems use different RF bands for the outbound (Master→Remote) and inbound (Remote→Master) links, since the outbound link usually requires mass-addressing capability while the inbound links individually need much less bandwidth but must be able to operate under the heavy traffic conditions prevailing when many Remotes are sending data to the Master.

For AM/VHF radio schemes, the outbound signal is multiplexed onto the broadcast signal of a radio station. Since the audible portion of this signal only uses about one-third of the station's carrier power [35], the outbound signal does not affect the quality of the broadcast as detected by the listener. In [35] the superposition of the utility transmissions on the AM station output is done using small-angle quadrature-phase-modulation, and the power-line is used as an antenna to simplify implementation. The return link uses VHF transmitters which employ narrow-band signalling with bandwidths of only $50 - 100$ hz. Of course, practical frequency-controlling devices do not permit such an assignment to each VHF transmitter without large guard bands between the subchannels, and this would be an inefficient use of the RF bandwidth. The solution to this problem proposed in [35] is to synchronise the frequency of each Remote to the AM station carrier frequency, with each Remote offset by a different amount so that overlap is avoided. This type of *frequency-*

division multiplexing permits the inbound signals to be sent in closely-spaced narrow-band subchannels and achieves the required SNR for reliable communications at relatively low VHF transmitter power. In addition, the digital synchronisation circuits for the forward-link receiver are shared by the return-link transmitter, so the system design is simplified and precise timing can be obtained. A further advantage of this method is that the entire group of transmitters drift together when the outbound carrier frequency drifts, so this carrier drift does not introduce any mutual interference and network performance is unaffected.

Further experience with AM/VHF systems is presented in [36], where it is shown that the VHF return link performance is poorer than that of the AM forward link with the modulation methods used, and the use of UHF return links is suggested as a possible solution to this problem.

A UHF/VHF system is now a feasible alternative for utilities since the FCC has set aside 20 pairs of UHF frequencies in the 928−952 Mhz band for use by electric utilities. This type of system can achieve higher data rates than an AM/VHF system, but in both cases the system is classified as low data rate from the point of view of a DAS. Another problem with these systems is that the VHF return link is limited to line-of-sight (LOS), restricting the coverage area or requiring the installation of repeaters. The centralised radio network will fail if the Master fails, but otherwise is secure from catastrophic failure since the loss of a Remote means only one link is non-operational. Compared to the other communications techniques listed in 3.3.1, radio networks are very reliable and easy to maintain or repair.

Packet radio networks are of the form already discussed in Section 3.1 with a radio transceiver at each node. These radios share the broadcast channel, giving rise to the need for channel access protocols to reduce contention among the radios. Among the advantages of this type of network are :

- easy to install and deploy - no physical connections need to be made or broken between the packet radio units
- easy to reconfigure the network - a radio can be given new routing instructions without interrupting communications
- modularity - a failed unit can be substituted with a backup unit and service resumed quickly

The packet radio units control the movement of packets through the network. Each unit consists of a radio, an antenna, and a digital controller; the radio acts like the modem and line in a wire-based terrestrial network by establishing connectivity with neighbouring radios, while the digital controller provides the packet-switching functions [37]. In the most general case where no unit is directly connected to all the others, we have a *multi-hop* packet radio network in which packets must be relayed through intermediate units to reach the receiver. Thus the digital controller must provide store-and-forward operation, receiving packets from neighbouring radios, making routing decisions, and forwarding packets based on these decisions. A comprehensive review of the issues involved in the design of packet radio networks is given in [26].

Multi-media systems

In the backbone communications system of a DAS, there may be widely-varying performance requirements (as measured by the quantities introduced in Section 3.2). If this is the case, no one medium is likely to be the 'best' choice, since no technique in the above list has overriding advantages. Thus we are lead to consider mixing these techniques, matching the characteristics of the medium at a particular level in the system to the performance requirements at that level. For example, Southern California Edison (SCE) has commissioned a communications system called NetComm to allow implementation of an ambitious Distribution Automation program which will provide a reliable, interactive link between SCE and its 3.6 million customers [38]. The NetComm system consists of a packet radio network with 500,000 nodes, each of which is connected to a group of customers by a two-way distribution line carrier link. The packet radios are close enough to each other that signal power levels can be kept well below the FCC threshold for licensing, and use spread spectrum signalling for interference suppression and multiple-access capability. The distribution line carrier transceiver is on the secondary side of the distribution transformer, so higher frequencies can be used resulting in higher data rates. Customer data is sent through the network to the appropriate processing centre, and then returned to the local packet radio and transmitted down the power line to the customer meter. This meter is microprocessor-based and can do simple processing, as well as easily interfacing with the distribution line carrier system. The system is about to undergo full-scale field testing.

3.3.2 Spread spectrum signalling

Recall from 3.3.1 that, for a distribution line carrier system, no single frequency is best for all remote devices. Thus we can expect the performance of any narrow-band signalling method to vary from excellent to unacceptable over the set of links of the network. One way of solving this problem has already been discussed, namely frequency diversity; in this section we introduce another type of signalling scheme which solves this problem by spreading the signal bandwidth over a large frequency range so that there will always be some allowed frequency within this range at which a given link can perform adequately. As we will see, there are other advantages to be gained from the use of spread spectrum methods, some of which cannot be obtained in packet networks by narrow-band schemes. Narrow-band signalling is the conventional approach and so we will not discuss it in this report, concentrating instead on the recent developments in spread spectrum techniques.

Spread-spectrum is a means of transmission in which the transmitted signal occupies a bandwidth in excess of the minimum required to transmit the information, the bandwidth expansion being determined by some code that is independent of the message data but known to both the transmitter and receiver. The expansion of the message bandwidth prior to transmission is referred to as *spreading*, while the recovery of the modulated message from the transmitted spectrum is referred to as *despreading*. The use of spread spectrum has three main purposes :

- energy density reduction
- identifiability
- interference suppression

Because a spread spectrum system distributes the transmitted energy over a wide bandwidth, the signal-to-noise ratio (SNR) at the receiver is lower than with narrow-band signalling. Despite this low SNR, the receiver is able to operate successfully since the transmitted signal has certain characteristics which distinguish it from the noise. The capability for energy density reduction is important when regulatory constraints force the signal power below a certain level, and helps maintain message privacy since the signal is harder to detect.

The spread spectrum signal is easily identified by its intended receiver since the transmitter and receiver both know the code used to spread the bandwidth. The transmitter

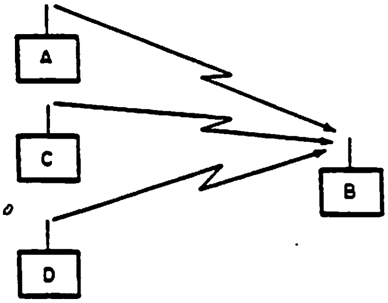


Figure 3.8: Signal capture

and receiver are said to be synchronised in the code domain, which leads to the necessity for additional signal acquisition and tracking functions that increase the cost of the system.

Interference refers to any signal which hampers the correct reception of the transmitted signal. The properties of spread spectrum which enable the system to combat the various types of interference found in a packet radio network are [39]

1. Signal capture
2. Multiple-access capability
3. Anti-multipath capability
4. Narrow-band interference rejection

1. Signal capture refers to the ability of a unit to demodulate at least one of a number of overlapping packets, all of which are addressed to it. Recall from 3.1.2 that we assume a packet radio unit can deal with only one packet at a time, whether sending or receiving; when a packet is locked onto by a receiver it is said to have 'captured' that receiver. The capture effect is illustrated in Figure 3.8. Typically the first-arriving packet, or the one with the strongest signal arriving in a certain time period, captures the receiver. The other signals constitute interference which may cause the receiver to lose the packet intended for it.

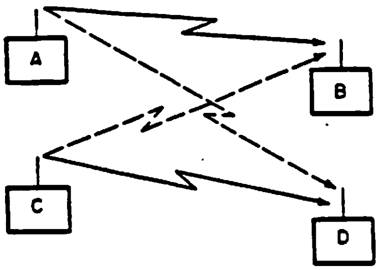


Figure 3.9: Multiple-access capability

2. The multiple-access capability of spread spectrum refers to the ability of two or more units to simultaneously receive packets from different transmitters even though these packets overlap in time and each of the transmitters is within range of each of the receivers. As with signal capture, multiple-access is fundamentally a multiple-transmission issue where the communication channel has two or more inputs corresponding to the transmitters within range of the particular receiver. The multiple-access capability is illustrated in Figure 3.9, and is a measure of the performance of a subset of the packet radio network which includes multiple transmitters and receivers. A given receiver chooses among the incident packets primarily on the basis of properties of the packets' spreading codes rather than their arrival times, and ideally packets with identical signal powers and arrival times can be received correctly once the codes have been chosen properly.

3. The anti-multipath capability refers to the ability of a pair of nodes to communicate reliably over a link which has multiple transmission paths for the transmitted signal. This form of interference is common in urban areas, where buildings provide reflected paths for the spread spectrum signal and so give rise to multipath interference. The anti-multipath capability is illustrated in Figure 3.10, and since the multipath interference arises from delayed versions of the transmitted signal the receiver discriminates among the packets according to the autocorrelation properties of the signal's spreading code.

4. Narrow-band interference rejection refers to the ability of a unit to correctly

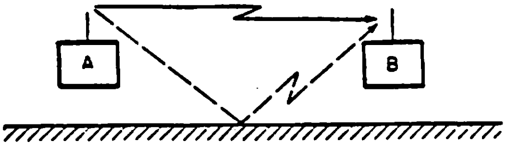


Figure 3.10: Anti-multipath capability

receive a packet in the presence of deterministic narrow-band interference. This interference may be due to hostile jamming or it may be 'unintentional' interference due to the transmissions of neighbouring radios. This capability of spread spectrum is illustrated in Figure 3.11, and can be explained by considering the Hartley-Shannon formula for the capacity of a bandlimited Gaussian Noise channel :

$$C = B \cdot log_2 (1 + \frac{P_s}{2N_0B}) \quad bits/second \tag{3.7}$$

where

C = maximum error-free bit-rate

B = signal bandwidth in hz

P_s = signal power

N_0 = noise power spectral density

Assume that the narrow-band interference power P_j is finite, as indeed it must be if the interference is caused by a hostile jammer or other radio units. If we further assume that the narrow-band noise is white over the signal bandwidth B, then the noise spectral density is given by

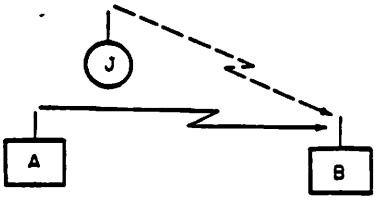


Figure 3.11: Narrow-band interference rejection

$$N_0 = \frac{P_j}{2B}$$

Plugging this into equation 3.7 and solving for P_j yields

$$P_j = \frac{P_s}{2^{\frac{C}{B}} - 1} \tag{3.8}$$

Thus as the bandwidth increases for fixed signal and interference power levels, the channel capacity increases; or put another way, as the bandwidth is increased for a fixed signal power, the interference power must also increase in order to hold the capacity constant. If the baud interval is denoted by T, it can be shown [24] that a pulse time-limited to T with bandwidth B lies in a subspace of signal space of dimension approximately 2BT (this is approximate since no pulse can be both time-limited and band-limited). Since the interference is presumably distributed evenly over all dimensions, the noise seen by the receiver is decreased relative to the case where the transmitted pulse is also spread over all of signal space. The issue of how to choose the subspace so that it is unknown to the jammer is addressed by the use of pseudorandom spreading codes which give the transmitted signal 'noise-like' characteristics.

The capture and multiple-access capabilities of spread spectrum have the greatest influence on network design and performance, since they affect the network's throughput-delay characteristic and the choice of a channel-access protocol. The features of spread spectrum which provide for good signal capture also give good anti-multipath performance, and some packet radio networks operate at frequencies which are subject to significant multipath interference. On the other hand, the ability of spread spectrum to reject narrowband noise, and the improvement in message security due to the energy density reduction possible with spread spectrum (as seen above), play no special role in a packet radio network compared to other narrow-band systems.

The two most common spread spectrum techniques are **direct-sequence** (DS) and **frequency-hopping** (FH). In DS spread spectrum, the data signal is multiplied by a spreading signal prior to modulation, while in FH spread spectrum the carrier frequency of the modulated data signal is varied over a range of allowable frequencies during the transmission. In either case the spreading signal should be random, but then the receiver will not be able to distinguish the transmitted signal from noise unless the SNR is increased to narrow-band levels. The solution is to make the spreading signal *pseudorandom*, which means it is deterministic and periodic but with sufficiently long period that its autocorrelation function is similar to that of white noise.

A comprehensive introduction to spread spectrum signalling is provided by [40], which also includes a review of the properties of pseudorandom sequences required in this type of application. Direct-sequence spread spectrum is analysed in [39] and a simple model developed to illustrate its advantages over narrow-band schemes in packet radio networks. Bounds on the bit error probability are also presented. Frequency-hopping spread spectrum is discussed in [28], where its superiority to narrow-band techniques is demonstrated by estimating the increase in local throughput allowed by frequency-hopping as compared to narrow-band schemes. The dependence of the performance of this form of spread spectrum on the error-correcting code used and the number of allowed frequencies can be observed from the results. The role of spread spectrum techniques in packet radio networks is addressed in [39], and the effects of choosing this type of signalling on the various network protocols discussed.

3.3.3 Network design

The first issue to be addressed in network design is that of topology, namely the configuration of the communicating devices with respect to one another. Network topologies fall into two broad categories : **centralised vs. distributed.** By centralised, we mean there exists one node (the Central node) which controls the flow of data through the network, while in a distributed network the control function is divided up among geographically separated nodes so that decisions are taken locally rather than centrally. Of course, practical communications networks will most likely be somewhere in between these two extremes, but in order to introduce the issues involved in deciding on the network topology, we consider them first.

Note first of all that we will be referring to *links* between devices, where such links may or may not be physical connections. For example, a Remote Terminal Unit (RTU) and a distribution line carrier transmitter are actually connected (by the power line), but a radio receiver locked onto a particular transmitter also forms a link between the radios.

Typical centralised topologies are shown in Figure 3.12. The simplest is the *star* configuration, in which every node is connected directly to the Central node via a two-way link. The nodes send data to the Central node, either on request from the Central node, which is called 'polling', or when triggered by some event in the coverage area of the node, which is called 'event-driven'. The Central node then processes the data and takes control decisions on the basis of information about the whole network. Nodes affected by such decisions are sent messages by the Central node, and since it is not necessary for any outlying node to have information about the other nodes, these output messages will typically be control signals (on/off commands). The advantages of the star topology from a communications point of view include the lack of contention on a link, because only one node is connected to the central node by each link, and the simplified traffic flow pattern, which means complex routing strategies are not required. However, when the number of nodes is large, the delays inherent in this configuration (all communication must pass through a central node which can thus become a bottleneck) and the inefficient use of links (each link is only used a small fraction of the time) render the star configuration an unsuitable choice. There is also no provision for an alternative route to a node if its link to the Central node fails, without first setting up a new connection.

One possibility for centralised control of a large network is a *tree* topology, in which the outlying nodes are organised into hierarchies, perhaps corresponding to distance from the central node or degree of processing power installed. Each node in the tree has a unique path back to the source. Usage of the communications links is higher; this configuration is ideal for broadcast messages; and since each node in the tree has a unique path to the

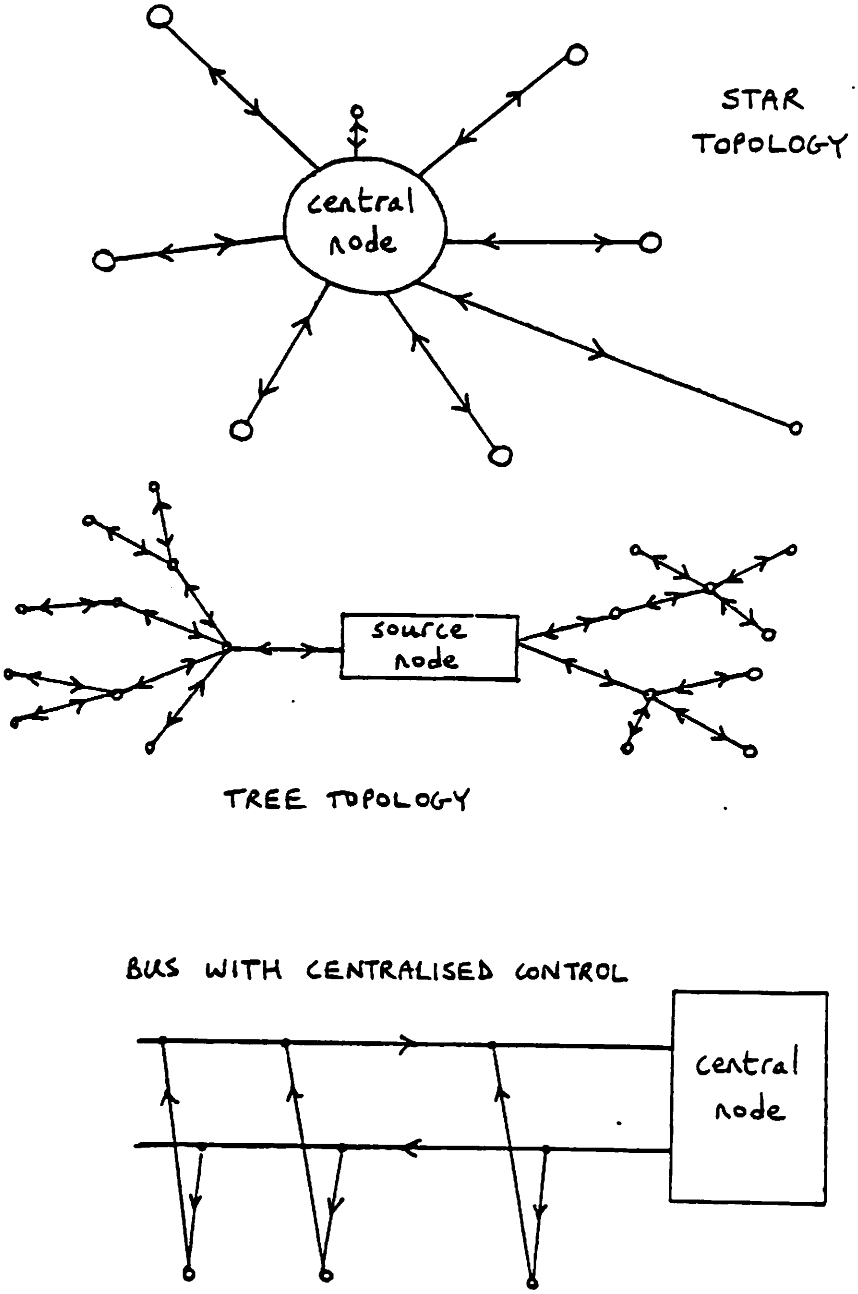


Figure 3.12: Centralised network topologies

source node, no routing decisions are necessary. However, the failure of a link removes all the downstream nodes from service, and the delay and bottleneck problems encountered with the star topology are still present. An improvement on the tree topology is the *bus with centralised control*, in which all nodes are connected to a 2-way bus and their actions controlled by a Central node. Inter-node message delays are reduced and the failure of one node does not affect the operation of the rest of the system; however the central node must cope with contention by the nodes for access to the bus, and again in heavy traffic conditions the delay in sending a message to the central node and receiving a response may be unacceptable. There are many other centralised configurations which we will not go into here, but they all share the common features of simple message routing, good broadcast capability, inherent delay due to the requirement for centralised processing, and the problem of bottlenecks at high traffic levels.

In a distributed configuration, by contrast, all nodes are functionally equivalent and - in a large network - assumed to be connected to a subset of the network which we referred to in Section 3.2 as the local population. The burden of deciding on the intended receiver, establishing a link to the intended receiver and sending the message, and receiving messages from other nodes is devolved onto the individual nodes. Among the advantages of a fully distributed network are decreased delays (because no message routing through a central node is required) and the capability of avoiding known bottlenecks, thus increasing throughput. However, since we may wish to transmit to a node not connected to the transmitter, more complicated routing strategies than in the centralised case are needed and they must be based only on information available to the transmitter. Since these decisions are taken locally, they will in general be suboptimal for the network as a whole, introducing delays which act to offset the gains obtained by not having to go through a central node. The processing power required of each node is considerably more than for the centralised case, and in large networks this increase in each node's computational burden may outweigh the advantage of not requiring a central processor.

One popular example of a distributed configuration is the *ring* topology, as shown in Figure 3.13. Each node is directly connected to only its two nearest neighbours, and collisions are usually avoided by some form of hub polling. Under this protocol, referred to as *token-passing* in the context of local-area networks, there is a unique 'token' which resides in exactly one node of the ring at any time. This token is an authorisation to transmit over one of the two available channels, and at the end of the transmission the token must

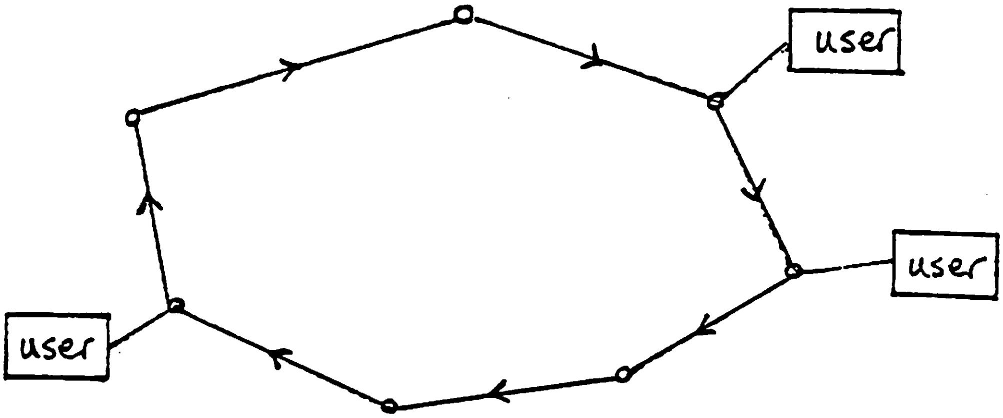


Figure 3.13: Ring-type network

be passed to the next node. As the token circulates through the ring each node has an opportunity to transmit a message. Broadcast messages are not allowed in ring topologies since the information would circulate indefinitely. A similar problem can arise if a node cannot properly receive a message intended for it. To overcome this problem, some of the nodes are given additional control capability and are referred to as *active nodes.* A user inserts and removes messages at an active node. However, the message delay grows linearly with the number of nodes, and better distributed configurations have been developed [41].

One possible compromise between these different philosophies is a *hierarchical* topology, an example of which is shown in Figure 3.14. Nodes on the same level are functionally equivalent and control their downstream nodes much like the central node in a centralised system. We also permit communication between nodes on the same level, which reduces inter-node delays and allows for the possibility of re-routing to avoid known bottlenecks or link failures. Events which only require local action can now be processed faster than in a centralised system, while still allowing for co-ordinated decisions based on system-wide knowledge. There will be contention at the interfaces between the levels which will lead to delays compared to (say) a tree structure, and messages sent to nodes further up the hierarchy may suffer longer delays than in a distributed system.

Once the network topology is chosen, the focus shifts to the design issues mentioned in 3.1.2. The decisions made depend strongly on the desired configuration, so we will not

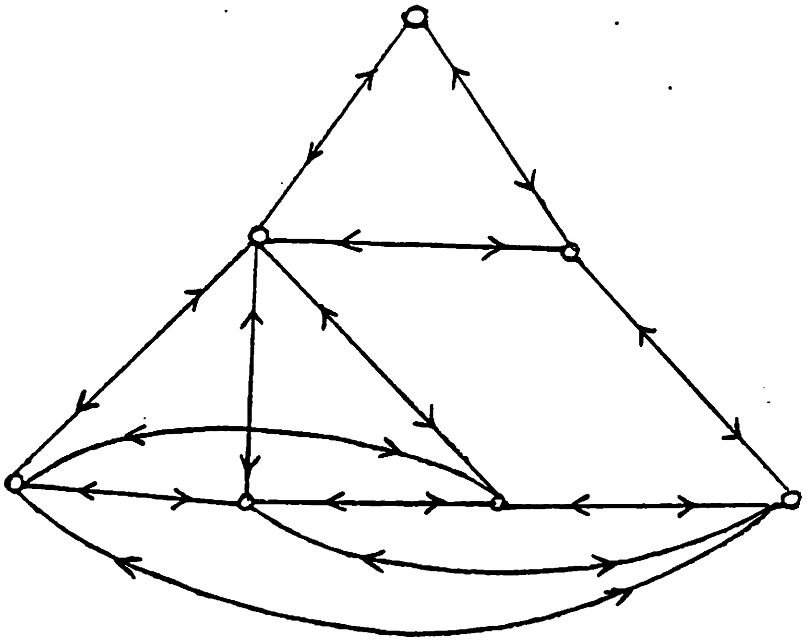


Figure 3.14: Hierarchical network

discuss them further in this section.

There is no 'best' network topology for a DAS, since, depending on the functional requirements, we put different emphases on properties such as broadcast capability or complexity of message routing. The fundamental point in network design is that **the communications network cannot be designed in isolation**, at least not if we want an integrated automation system with the minimum of interface difficulties. Among the factors which must be taken into account are the degree of distributed processing required, bulk system communications requirements, and the ratio of real-time to background traffic. It may well be that these requirements will vary with time and location throughout the system, which further complicates the design process. We will consider this 'coupling' between the communications system and the computation and control functions in more detail in Chapter 5.

Chapter 4

Computation and Control

The communications network of a Distribution Automation System (DAS) addresses the question of how data is transferred between the nodes of the DAS. No less important is the issue of processing the data, and in this Chapter we turn to the computation and control systems which address such questions as what data is needed to operate the DAS, where it is to be sent, and how it is to be used to control the system.

Current practice in EMS's is to centralise the data processing, typically in a mainframe computer or supercomputer. However, this centralised approach may not be suitable for the computation system in a DAS. One reason for this is the *increased diversity* (both in type and location) and *huge volume* of data needed to describe the operation of a typical distribution system as opposed to the bulk power system. In addition, many of the DAS functions discussed in Chapter 2 have significant components which can be carried out at a local level, and to keep the communications overhead down this property should be exploited. Thirdly, even if we discount the nature of the underlying distribution system and of the processing, the necessity for doing much of the processing in real-time (on the order of minutes) indicates that a centralised processor, with its inherent access delays, might not be acceptable. Much work has been done recently in the area of parallel processing, in which the computational burden is spread among many similar computers rather than one central processor. Of course, the computational requirements are not diminished - in fact, due to the inevitable communications between processors, the total load will probably increase - but the work is done *more quickly*, and in a real-time environment such as a DAS this is often an overriding advantage. We focus in this Chapter on the characteristics of parallel processing machines that make them realistic candidates for coping with the complex computational requirements associated with a large-scale DAS.

The operator of a large-scale DAS must cope with many kinds of data, possibly conflicting requirements from the DAS functions, and several control variables, all in real-time. Because of the desire to avoid costly mistakes, and in order to take advantage of the computer-based nature of a DAS, we consider the use of so-called expert systems to assist DAS operators (and in some instances replace them). In essence, an expert system is a program which shifts the problem-solving load from the operator to the computer. However, existing expert systems are unsuited to Distribution Automation applications. Consequently, our treatment concentrates on the issues involved in designing an expert system specifically for power systems applications.

4.1 Distributed and Parallel Processing

Distributed processing involves dividing the processing requirements for a given task between two or more processing units. Applications to large-scale systems, such as power systems, range from fairly simple to very complex processing schemes. Consider by way of example the task of reading in data from Remote Terminal Units (RTUs) to a central computer. Instead of the central computer taking 'raw' data from the RTUs and operating on it, we could distribute the processing among several data concentrators by having them read in the RTU data and summarise it (by statistical techniques) for the central computer, whose programs can then be written at a higher level. At the other end of the scale are suggestions for carrying out load flows in a distributed manner, taking advantage of the local steps involved to break the computations down into their basic components [42]. We note that the individual processors may be at the same location, which is referred to as *horizontal processing*, or at different locations, in which case we have *vertical processing*. The essential point is that more than one processor is used to carry out the task.

Parallel processing involves running two or more processing units simultaneously while maintaining communications links between the units. There are two basic forms of parallel processing, *synchronous* and *asynchronous*. Formally, a synchronous parallel algorithm is one in which there exists a process such that some stage of this process is not executed until another process has completed a certain portion of its task. Loosely speaking, in a synchronous parallel algorithm the computations must be done in an orderly way; and since the time taken for a stage of a process is unknown in general, there must be *synchronisation points* at which processors wait until the other units with which they are synchronised catch up. Synchronising processors imposes a communication overhead, and possibly results in blocking delays and processor idle times which grow with the number of processors.

In an asynchronous parallel algorithm, by contrast, there are no synchronisation points : processors do not wait for other processors but use the most recent information available to continue working. Of course, in cases where a task can be divided into independent components which can then be assigned one per processor, nothing is lost by using an asynchronous algorithm. However, when the processors need to exchange information, performance is degraded due to the increase in communications overhead. For example, most of the computational tasks which arise in power systems are solved by *iterative* methods. In this case processors may be working with out-of-date data, which results in requiring stronger assumptions to get the same results as in the synchronous case. In [43] an asynchronous parallel algorithm is presented to solve fixed-point problems of the type $x = F(x)$, $x \in R^n$. A natural division of processing is to update the n components of x separately. In order to guarantee convergence, we require that any particular iterate can only be used a finite number of times in the update process for any x_i, and that no x_i can be left 'un-updated' forever. These constraints are not overly restrictive : in fact, stronger conditions are usually met in practice, for instance where the most recent iterate is chosen and all processing times are finite. Under these conditions, the asynchronous parallel algorithm converges provided slightly stronger conditions are imposed on F than in the synchronous case (namely that F is a component-wise contraction mapping). In general, then, asynchronous parallel algorithms are more complicated and less widely applicable than synchronous parallel algorithms, but they offer simpler implementations, reduced inter-processor communications requirements, and increased savings in computation time.

The notions of distributed processing and parallel processing are closely related but, strictly speaking, not equivalent. To see this, note that (using the above definitions) distributed processing is a proper subset of parallel processing. This is because any distributed processing application fits into one or other of the parallel processing categories. Indeed, unless the individual processors are given independent portions of the task, the distributed processing application is an example of synchronous parallel processing (even if the processors execute sequentially rather than at the same time !). On the other hand, parallel processing may not involve several processors working on parts of the same task, so that, technically at any rate, processing has not been 'distributed'. For our purposes this distinction is merely terminology, and we take distributed and parallel processing to be interchangeable for the rest of this report.

Why use parallel processing ?

Many reasons have been put forward for using parallel rather than sequential processing methods, but from a Distribution Automation point of view perhaps the two most important are related to **cost** and **speedup**.

The communications, control and computation systems needed to operate a power system are closely linked and are often referred to collectively as a *3C system*. For example, depending on the DAS functions we wish to implement, the data requirements to control the system can be calculated; then by fixing the locations of the control and computation functions, we can calculate the communications traffic due to these DAS functions. The availability of cheaper and more powerful microprocessors has resulted in a decrease in computational costs relative to the costs of communication. In practice, this means that computations should be done and control decisions taken as close as is feasible to the measurement points in order to minimise the communications requirements. Thus to take advantage of these trends, **we would like the operation and control of the power system to be as distributed as possible.** In addition, some DAS functions such as voltage regulation are inherently local in their application, while others have significant 'local' components, such as digital relaying or load-shedding. In such cases, it may be cheaper to install more processing power at the affected locations than to incur the increased volume of communications traffic associated with a more centralised approach.

The primary motivation for considering parallel processing as an alternative to the conventional sequential approach is the possible reduction in the time taken to execute the process. The ratio of the time taken for a single processor to carry out the task to the time taken for N processors to do the same task is called the *speedup*. We might hope that speedup increases linearly with N, but unfortunately it can easily be shown that not only is the speedup sublinear in N, but that it is limited to the reciprocal of that fraction of the process which must be executed sequentially. This result is known as **Amdahl's Law**, and the proof can be outlined as follows : define

T_1 = time to execute the process on 1 processor;
T_N = time to execute the process on N processors;
δ = fraction of process that is 'inherently sequential';
then the speedup using N processors instead of 1 is given by

$$S = \frac{T_1}{T_N} \tag{4.1}$$

Suppose that the fraction of the process that is *not* inherently sequential, namely $1 - \delta$, can be done 'ideally' in parallel, then we have

$$T_N = \delta \cdot T_1 + (1 - \delta) \cdot \frac{T_1}{N} \tag{4.2}$$

which means the speedup is at most

$$S = \frac{N}{1 + \delta \cdot (N - 1)} \tag{4.3}$$

and thus S increases less than linearly with N for all nonzero δ. Also, for all finite N, $\delta \neq 0$ implies

$$S < \frac{1}{\delta} \tag{4.4}$$

and in the limit as the number of processors goes to infinity, $S \rightarrow \frac{1}{\delta}$, which completes the proof. Thus we can quantify the tradeoff between the speedup obtained by distributing the processing over a larger number of processors and the additional costs corresponding to the extra processors.

Of course, Amdahl's Law is stated for a given problem, and it may happen that the inherently sequential fraction of a process is itself a function of the 'size' of the process. If δ is a decreasing function of the dimensionality of the process, then by applying parallel processing methods to large-scale problems we can still achieve significant speedup over the case where one processor (or a small number of units) is used. One example where this phenomenon is observed would be the case where δ is a constant in absolute terms, for instance in averaging a number of quantities which must first be computed. The sequential part of such a process is just taking the expected value, so - assuming this calculation is much faster than the computation of the quantities to be averaged - the sequential fraction of the process decreases as the number of quantities increases.

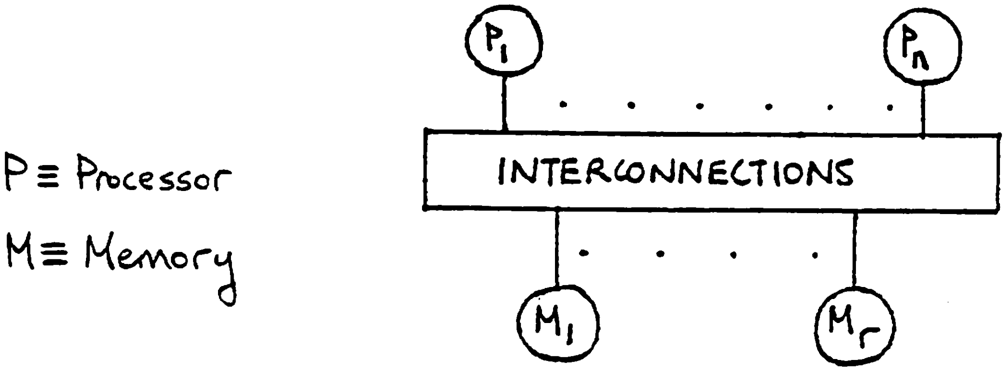


Figure 4.1: A multiprocessor system

4.1.1 Multicomputer issues

Systems for parallel processing can be divided into two broad classes [44], **multiprocessors and multicomputers.** The essential difference between these two categories is in the level at which the individual processors interact. A multiprocessor system allows all processors to directly share a common memory, as shown in Figure 4.1. In a multicomputer system, on the other hand, each processor has its own private memory, and a processor cannot directly access another processor's memory (Figure 4.2). We present a (necessarily incomplete) discussion of the important features of each type of parallel processor, and then go on to address some of the issues involved in multicomputer systems since these are the more likely candidates for Distribution Automation applications.

Multiprocessor systems are *shared memory* machines in which at least part of the main memory is accessible to all processors. Note that we do not require processors to be directly connected to the main memory, as long as the operating system presents the image of shared memory to the user (as for example in the IBM 3090, where processors have local cache memories but the operating system implements cross-cache validation). An important property of shared memory systems is that the access time to a piece of data is independent of the processor making the request [45]. Thus if the code running on any two identical processors can be swapped without affecting system performance, the system

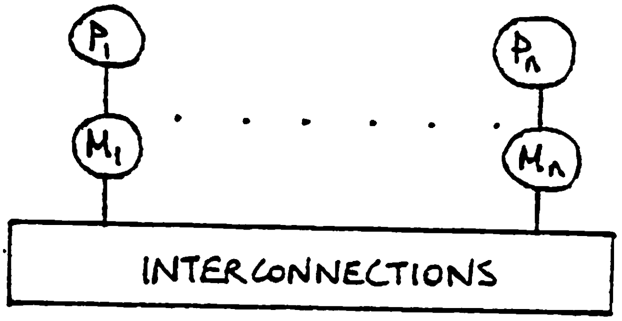


Figure 4.2: A multicomputer system

exhibits true shared memory.

The architecture of multiprocessor systems ranges from a single shared-bus configuration to a fully-connected (or 'crossbar') network. However, shared-bus networks only allow one processor at a time to access the main memory, which for a large number of processors implies a long wait for the bus, while crossbar-type networks require $O(N^2)$ links to connect N processors and so the cost soon becomes excessive. Currently the best balance between these two extremes is given by *multistage switching networks (MSNs)*, a simple example of which is shown in Figure 4.3. Let $N = 2^m$ for some integer m. An N-by-N MSN connects N processors to N memories through $log_2(N)$ stages of 2-by-2 switches, with N/2 switches per stage. The requesting processor supplies the binary representation of the desired memory module, and the connection at the i^{th} stage is determined by the i^{th} bit, counted from the most significant bit. Thus MSNs are *self-routing*, avoiding the need for a central controller and making distributed applications feasible. Many processors can simultaneously access many memories, because multiple paths exist through the network. However, since there is only a single path between an input and an output, these types of systems are not robust to link failures, and the incorporation of fault-tolerance into these networks is an active research area.

Multiprocessor systems are synchronised by the use of programming techniques called *atomic operations* [46], such as locks or semaphores, which are also used on unipro-

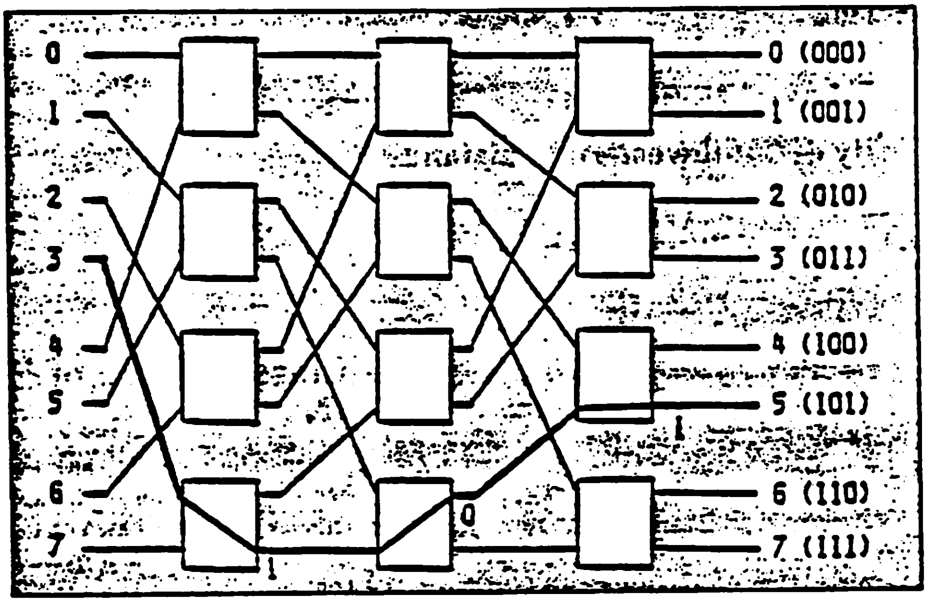


Figure 4.3: An 8-by-8 omega multistage switching network

cessor machines. Although these operations must be specified explicitly, processors need synchronise only once, after which there is no memory contention.

In a multicomputer system, some memory is local to each processor and no memory is globally accessible. Applications which require data to be shared among several processors must thus explicitly move data from one memory module to another. Because processors communicate with each other by sending messages to the appropriate memory locations, multicomputers are referred to as *message-passing* systems. A message from a source node may pass through a number of intermediate nodes before reaching its intended destination, using the store-and-forward capability of the intermediate processors. As with packet communications networks, the performance of multicomputer systems depends to a large extent on the *routing strategy* used to control the passage of a message through the network. Therefore, in contrast to the shared memory systems mentioned above, performance is tied to how well the location of data coincides with where it is used.

Message-passing synchronises a multicomputer system implicitly [46]. A processor requesting data from another processor's memory waits until the data is received, and so processes cannot get out of step. This form of implicit synchronisation simplifies the programming needed in a parallel processing application, but increases the scheduling overhead since processors must be stopped and started every time they request data.

Fully-connected systems where each processor is directly connected to every other processor are impractical when the number of processors is large. The simplest practical approach is to connect the processors in a ring; each processor 'talks' to its two nearest neighbours, although certain nodes may have additional control functions to prevent data circulating indefinitely in the event of an error at the intended receiver. However, the message delay in a ring network rises linearly with the number of intermediate processors. An improvement is the mesh, in which processors are connected in a 2-dimensional grid and talk to their four nearest neighbours. In a mesh configuration it takes $O(\sqrt{N})$ time to send data to all N processors, which is a considerable reduction from the $O(N)$ time in a ring topology when N is large.

Regarding processors which can access each other's memories as being connected allows us to characterise multicomputer systems in terms of their *degree* and the *network diameter* [44]. The degree is the average number of links per node, and reflects the cost associated with the particular configuration. The network diameter is the maximum number of links a message has to traverse along the shortest path between any source and any destination. Networks which have low degree - such as the ring - often have high diameter and so the communications overhead is relatively high. A higher degree means less message delay but incurs higher costs.

The best compromise between the goals of low degree and low network diameter is the **hypercube**, which we discuss in more detail below. The reason we concentrate on multicomputer systems is that they model a DAS more closely than do shared memory architectures. The size of a fully-implemented DAS for a typical distribution system rules out multiprocessor-type networks because of the large number of intermediate switches and links needed. In addition the aim of localising the control of the network to the areas affected is in opposition to the concept of a single main memory which all processors attempt to access. However, some of the advantages of shared memory systems can be preserved through the use of *hybrid* systems. Memory is local to each processor, as in the multicomputer case, but the operating system presents the image of a global memory to the user. Thus programs are written as if for a shared memory system, although the data must be held as for message-passing systems if the best performance is to be obtained.

4.1.2 The hypercube multicomputer

A hypercube of dimension n has $N = 2^n$ processors arranged so that each node is directly connected to n other processors. In three dimensions this corresponds to a cube, with the edges representing the links and the vertices representing the nodes (Figure 4.4). The memory addresses for the nodes are denoted by the binary equivalents of the decimal numbers between 0 and $N-1$, where connected nodes differ by exactly one bit. Thus the distance between two nodes is simply the number of bits in which their binary representations differ, and consequently the diameter of a hypercube is equal to its dimension, n. There are n disjoint paths between any source and any destination (though not necessarily of equal length), and hence the network is highly fault-tolerant. There are two additional facts that explain much of the usefulness of the hypercube architecture.

Fact 1. The higher the dimension of a hypercube, the higher its communication capacity is relative to its computational capacity.

This follows directly from the observation that the computational capacity of a hypercube with $N = 2^n$ processors is N, while there are $N \cdot n$ available communications

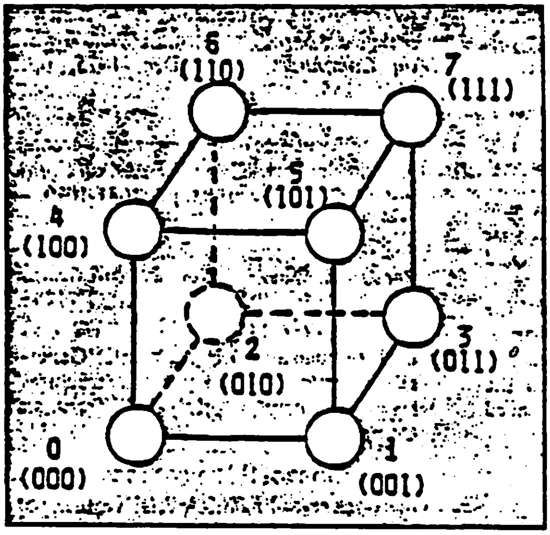


Figure 4.4: Three-dimensional hypercube

links. Thus the ratio of communication to computational capacity is n, the hypercube's dimension. This fact, coupled with each node's individual memory, allow expansion beyond most parallel architectures. For example, shared memory systems can accomodate up to 200 processors, whereas hypercube structures already exist with 1024 nodes and machines with many thousand nodes are in development [47].

Fact 2. The average number of intermediate nodes in a hypercube's communications path increases slowly relative to the increase in computational capacity.

To see this, note that the number of nodes at distance p from a given node is

$$m = \binom{n}{p} \tag{4.5}$$

so that in a 6-dimensional hypercube, there are 6 processors at distance 1 from the given processor (ie. directly connected), 15 at distance 2, 20 at distance 3, and so on. Then the average number of links messages emanating from the given node have to traverse is

$$E(l) = \frac{m \cdot p}{N-1} \tag{4.6}$$

Thus in a 6-dimensional hypercube, the average communications path has length l = 3 (*approx*), while an 8-dimensional hypercube with 4 times the number of processors - and hence 4 times the computing power - has an average path length of about 4, an increase

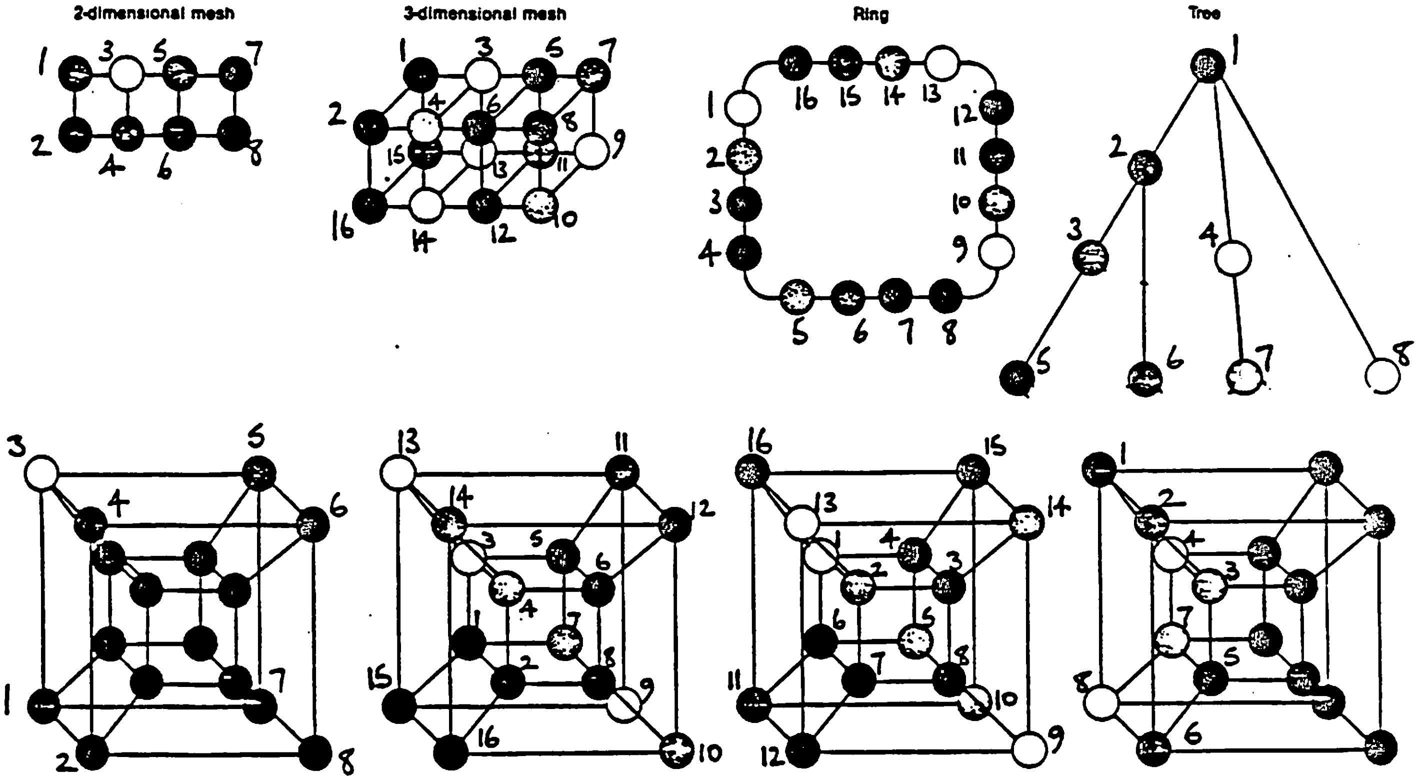


Figure 4.5: Flexibility of the hypercube architecture

of only 33%. Obviously this is a desirable characteristic of a parallel processing system consisting of a large number of nodes.

Another advantage of the hypercube network is the flexibility allowed by its interconnection scheme [47]. When the hypercube's dimension is four or greater, several possible topologies are available and one can be selected that matches the problem under study. This property is illustrated in Figure 4.5, where the numbering scheme is used to identify those nodes that are to be thought of as directly connected. As shown in the Figure, by directing communications appropriately between nodes, a 4-dimensional hypercube can be used as a 2- or 3-dimensional mesh, a ring, or a tree, among others. This feature of the hypercube configuration is useful in applications such as Distribution Automation, where different functions impose different requirements on the processing system. For example, the search techniques in network reconfiguration or capacitor switching are more suited to a ring topology, where a threshold can be applied to each of the candidate nodes in turn to decide the next step. On the other hand, functions such as load shedding or spot pricing require mass-addressing capability and so a tree topology is more appropriate.

We consider next some of the issues to be addressed in designing parallel algorithms.

4.1.3 Issues in parallel algorithm design

The first question to be answered is how to formulate the task so that it matches the structure of the parallel processor as closely as possible. In 'mapping' a problem onto a parallel architecture, we first divide it into distinct segments that will be executed in parallel, and then determine how the individual processors will communicate and synchronise with each other. Many problems can be expected to be such that there is a natural way of dividing them between multiple processors, typically by being composed of subtasks which run independently of each other. An example is direct load control : say we decide on a control strategy for load groups $1, \ldots, J$ which is to be implemented over the next H hours. Then the details of how the loads in a particular group are adjusted to satisfy the overall strategy are local and are not needed by the other groups, suggesting that our chosen control strategy can be carried out in parallel by assigning the computation of the individual strategies to separate local controllers.

How can the amount of 'parallelism' in a particular problem be described ? One qualitative measure of parallelism is *granularity*, which indicates of how much computing processors can do independently relative to the time they spend exchanging data with other processors [46]. Coarse-grained applications are characterised by subtasks consisting of lengthy independent processing sequences with little inter-processor communications. With fine-grained applications, in contrast, relatively few instructions are executed between communication events. The granularity of an application is sometimes determined by the degree of interaction between the subtasks, but it can also be determined by the nature of the parallel processor itself. A system composed of a small number of powerful processors connected via low data-rate links is more suited to coarser-grained problems, while a system consisting of many small processors communicating over faster links lends itself to finer-grained problems.

Useful parallel algorithms typically fall into one of two categories of parallelism, *explicitly-parallel* or *perfectly-parallel* [47]. An explicitly-parallel process is one that can be executed on multiple processors only with communication between neighbouring processors, while a perfectly-parallel process can be executed on multiple processors without communication between them. An example of an explicitly-parallel application is the fixed-point problem outlined at the start of the chapter, where we had to impose requirements regarding data exchange between the processors in order to guarantee convergence. The implementation of a load control strategy as mentioned above is an example of a perfectly-parallel application in a DAS.

Another concern in designing a parallel algorithm is *balancing the computational load.* If processors are assigned subtasks with widely differing computational requirements, the idle times of those units given less substantial jobs may increase to unacceptable levels in a synchronous algorithm, or the assumptions made about the currency of certain variables may not be realistic and degrade the performance of an asynchronous algorithm. One solution to this problem, in the case where there are many more subtasks than processors, is to estimate the computational requirements of the subtasks and then assign them one at a time in rotation through the processors until all subtasks have been assigned (rather like dealing cards !). This is of course a static solution and may be defeated by changing conditions as the process executes. In such cases we may find it necessary to use dynamic load balancing, though this imposes its own communication and computation requirements which must be weighed against the benefits derived from it. Such automatic redistribution of the computational load of a parallel process is best handled by the operating system.

In assessing the message delay for a particular multicomputer configuration, the network diameter provides an upper bound which may or may not be realistic. A better measure of the expected delay is the *mean internode distance*, as defined in equation 4.6. Unlike the network diameter, the mean internode distance depends on the message routing distribution. For symmetric multicomputer networks there are three basic message routing strategies [48] :

- uniform
- sphere of locality
- decreasing probability

Uniform message routing requires that the probability of node i sending a message to node j is the same for all i, j. No assumptions are made as to the computation generating the message, and as we expect most computations to generate at least some local communication traffic, this should provide an upper bound on the mean internode distance. A more realistic model is the **sphere of locality**, in which a given node sends messages to those 'sufficiently close' neighbours with high probability p and to nodes outside the sphere with probability $1 - p$. This model is founded on the assumption that most subtasks generate mainly local communication traffic and only rarely require global communication facilities. If there is no clear cutoff in message frequency as we move away from the centre of the sphere, or if the sphere itself is too large to be useful, we might assume that the probability of sending a message to a node decreases as the distance between it and the source node increases. This **decreasing probability** model can be used as a compromise between the previous two models, in the sense that close to the source the probability distribution is similar to a nearest-neighbour traffic pattern, while further away the probability drops off until it reaches a uniform level. More details on these routing models, as well as other performance measures for multicomputer networks, are discussed in [48], and various configurations presented.

4.2 Expert Systems

There is no widely-accepted definition of what constitutes an expert system. This is due in part to the difficulty people have in articulating what makes one person an 'expert' and not another, though the difference may be clear. Another source of confusion is that many conventional programs can perform certain kinds of expert tasks. Large and/or complex problems in which the computational requirements form the major source of difficulty can be solved by a computer in much less time than that taken by someone familiar with the subject. The distinction between the computational efficiency of conventional programs and whether they exhibit expert behaviour is not always so obvious, and so we first address those properties considered essential to an expert system.

In [49] three characteristics common to all experts are identified. First, an expert knows the concepts relevant to the problem domain and the principles on which they are based, and understands how these concepts interact with each other. Thus, faced with a new problem, an expert can come up with a solution by reasoning from first principles. Second, an expert has experience, in the form of solutions to previous problems or knowledge of how problems have previously been approached, and can recognise similarities between the current problem and their stored experience that allows them to short-cut the problem-solving process. If experience gained in the past is available to an expert system when faced with a similar problem now, so that the solution method might differ from that originally set by the designer, the system is said to have a *learning* capability. Third, an expert solves problems significantly faster than the average person familiar with the subject area, through a combination of short-cuts, rejecting irrelevant approaches, and the ability to focus on the key issues posed by the problem.

Any computer-based system which is to be used as an expert system must possess the above properties, though of course they are not necessarily a complete specification of expert systems. For example, a crucial element of any expert system is its **knowledge representation capability**. This refers to how data is classified, where and in what form it is stored, and when it is used by the system. The data may be input/output information or may describe the models used by the expert system to formulate the problem (ie. how the system 'sees' the problem domain). The above properties of an expert give no indication as to how such knowledge should be represented.

Another important feature of expert systems not covered by the above properties is the incorporation of **heuristics**. These are guesses used to produce a solution in cases where formal reasoning methods are unknown or cannot be applied. For instance, we often use an analogy between a problem whose solution is known and a slightly different problem to infer the new problem's solution, or at least some of the solution's characteristics. This approach relies on a 'continuity of solutions' argument and so can sometimes fail. Another example of the need for heuristics is in large-scale systems, where we wish to predict the behaviour of the system as a whole; by looking at a smaller system we might be able to solve the problem analytically, and then apply the results to the larger system taking the effects of its increased size into account. This often allows us to make qualitative statements about the behaviour of large systems without incurring the computational burden of an exact solution. Again, the association between the solutions of the small-scale and larger problems is usually not direct and can fail. This characteristic of heuristic approaches - that they 'usually' work but cannot be *guaranteed* to produce the correct solution - must be allowed for by the designer of an expert system.

4.2.1 Knowledge representation issues

Our concern in this section is the kinds of knowledge required in expert systems for DAS applications. The situation in electricity distribution systems is different to other areas where expert systems have been applied for the following reasons [42] :

- analytical models exist for the distribution system, and since this portion of the electricity system is widely distributed with respect to location and level of service, these models tend to be extensive. Efficient ways of representing the data used in these models are thus essential if the efficiency of the expert system is not to be degraded;
- based on these analytical models, several analytical tools have been developed to determine the behaviour of the distribution system (such as load flow packages), and these should be integrated with the expert system.

Unfortunately, general-purpose expert systems are weak in these areas, primarily due to their unstructured and thus inefficient data representation methods. In order to determine the desirable features of an expert system for Distribution Automation applications, we must first examine the various kinds of knowledge needed to operate and control a distribution system. These can be divided into **data knowledge** and **problem-solving knowledge** (or reasoning knowledge).

The data knowledge in a distribution system refers either to knowledge about the power system model or about the effects of external factors on the system. Knowledge of the power system model is naturally divided into two classes, one associated with physical entities such as buses and transformers, and the other associated with conceptual entities such as nodes and islands. Note that in general, knowledge about the physical elements of a system is *static*, while the conceptual entities usually change over time and so require *dynamic* knowledge representations. This knowledge takes many forms; for example, the network structure is described by connectivity knowledge which can either refer to physical elements or to nodes, branches, and so on. Another form of knowledge of the system model defines the relationships between groups of entities and their members. Since such groups are conceptual entities, their memberships may change (through network reconfiguration, for example) and so this form of knowledge is dynamic. Most DAS functions work with these entities rather than the actual system elements. One important point is that whether data is static or dynamic depends on the time-scale used, and data which some functions regard as static may have to be treated as dynamic by other functions which run less frequently. For example, for the purposes of capacitor switching for loss reduction we usually assume that customer loads consume constant average power, as in 2.1.2, yet direct load control requires a dynamic model of the controllable load (2.3.1) because load control strategies are calculated for periods of several hours. The effects of external factors such as weather conditions or holidays can usually be associated with the power system model.

Reasoning knowledge specifies how new information about the power system may be obtained from the available input and processed data. In power systems, reasoning knowledge ranges from well-defined analytical knowledge, such as is used in load flow calculations, to heuristics used by operators to control the minute-by-minute operation of the system. Any classification of this knowledge should depend only on the structure of the knowledge and should be independent of how the knowledge is represented. In [42] the operational requirements are divided into three classes of tasks, each of which has its own knowledge structure :

- Analysis - concerned with processing the available data to produce a concise description of the state of the system (for example contingency evaluation);
- Synthesis - concerned with finding the control actions necessary to achieve some desired objective (for example loss minimisation);
- Learning - the acquisition of problem-solving knowledge for both the analysis and synthesis task, which currently is stored in the operator's experience.

Current classifications of reasoning knowledge are purpose-oriented, whereas the above classifications focus more on the underlying knowledge structure. For instance, capacitor switching and service restoration are classified separately in Chapter 2 since they run under different operating conditions; but they are both synthesis tasks and hence require their logical structures to be represented in the same format.

Reasoning knowledge has conventionally been represented as *procedural knowledge*, in which a sequence of commands is specified to solve a given problem. It is suggested in [42] that reasoning knowledge is better stored as *declarative knowledge*, which has to be interpreted before it can be applied. This contrasts with procedural knowledge which only has to be executed to be applied. Declarative knowledge representation has the advantage that it is highly descriptive and thus easily understood, and also increases program flexibility since the knowledge is represented in a more modular form. Note that the distinction between declarative and procedural knowledge depends on the representation rather than on the knowledge itself. In general all knowledge can be represented procedurally or declaratively. However, knowledge concerned with algorithms is more efficiently stored as procedural knowledge, since the steps to the solution are already expressed as a procedure. On the other hand, if the solution to a problem involves heuristic knowledge, or if the sequence of steps cannot be determined in advance, the ease of developing and modifying a declarative knowledge base more than makes up for the computational inefficiency caused by the need for interpretation.

4.2.2 The design of an expert system

Practical expert systems represent knowledge in the form of rules, which enable the system to respond to its inputs according to the wishes of the designer. These rules are usually *production rules*, which are in the familiar if-then-else format. Of course, systems which have a learning capability may produce different outputs for the same input at different times, depending on whether the decision is influenced by the experience amassed by the system in the interim; but in this case the system designer simply controls how the expert system uses past results to influence current decisions, rather than explicitly specifying what action to take. There are three basic components of current expert systems : an **inference engine**, which interprets the rules governing system behaviour given the system input; a **knowledge base**, which stores the rules in a representation useable by the inference engine; and a **user interface**, which allows an operator to monitor and perhaps modify the operation of the expert system.

The inference engine usually works on the recognise/act cycle outlined in [42], namely

1. Find all the rules whose condition parts are satisfied by the current inputs. This subset of the system's rule base is referred to as the *conflict set.*
2. Select one rule from the conflict set (*conflict resolution*).
3. Perform the action(s) called for in the action part of the rule chosen in step 2.
4. Go to step 1.

Each time the above cycle is executed, some of the current elements of the conflict set may become invalid and other rules may have their condition parts changed from False to True. Thus after each cycle the conflict set must be updated. This can be done by checking all the rules in the system knowledge base, but this may be too time-consuming.

An alternative is to keep track of all elements of the rule base whose condition parts are affected by step 3, and just check these rules for validity. In this case the increased overhead involved in keeping track of the affected rules should be less than that incurred by checking all elements of the rule base, and so can be expected to work better the smaller the size of the conflict set. In practice we could set a threshold on the number of elements allowed in the conflict set before having to check all rules.

The conflict resolution strategy represents the reasoning behaviour of the expert system. Among the many possibilities for such a strategy listed in [42] are

- rule ordering - rules are ranked from highest to lowest priority, and the highest rule whose condition part is True is chosen
- size ordering - the rule with the longest list of constraints whose condition part is True is chosen
- recency ordering - the rule whose constraints have most recently been established and whose condition part is True is chosen

Another possibility is to group the rules into classes, perhaps according to their actions if their condition parts are True, and process these classes in parallel.

We discuss briefly some of the more important lessons learned from attempts to apply expert systems to problems currently requiring significant human expertise [50]. In most cases, experts were not good at formulating the rules by which they made decisions, and found it easier to explain how they solved specific problems than to define a generic solution procedure. The lack of a standard knowledge elicitation process makes it difficult to compare different expert systems and will limit the 'portability' of such systems unless the manner in which expert knowledge is determined is standardised (at least for problems in the same field). An iterative process may perform best in this regard, where an expert imparts knowledge to an expert system, observes how it performs, and augments its knowledge base as problems arise which the system cannot solve. In general, expert systems are potentially useful in cases where the knowledge and problems are relatively widely-known and static, the primary source of an expert's performance is special knowledge which can be described in a form suitable for computer processing, and where algorithmic solutions do not exist or are too computationally intensive. Some experience with designing expert systems for power systems is presented in [51] and [52].

4.2.3 Real-time expert systems

The application of computers to problems which must be solved in real-time is an integral part of a full-scale DAS. The operational complexity of such computer systems increases with the number of functions, the rate at which control decisions must be taken, and the number of factors to be considered before making such decisions. The size and diverse composition of a typical distribution system imply that a computer-based control system would be very extensive. The requirement for real-time operation imposes additional problems for conventional control schemes :

- data may not remain static during program execution, due to events which change the state of the system, and this may invalidate conclusions based on the data;
- the system may be required to continue operation even if a subsystem fails;
- events may occur asynchronously, which means the system must be capable of being interrupted to accept input from unscheduled events without losing the original data;
- because of this possibility of multiple inputs, the system must be able to decide on their relative importance and focus attention on the most important current one;
- the order and times of occurrence of events is often critical;
- perhaps most importantly, the system must have a solution ready when the time allowed to generate a response expires. In practice we also require that the solution is acceptably close to the optimal result, since not much is gained by producing fast but incorrect solutions.

A survey of current expert system tools for a wide range of applications in the aerospace, communications, medical, process control, and robotics fields is presented in [53]. The results of the survey show that current expert systems are not suitable for real-time applications. The reasons for this conclusion include : the expert systems surveyed were not fast enough; they had little or no capability for temporal reasoning (past, present and future); they were not able to focus on the most important current input; asynchronous input was not always accepted; and worst of all, it was not possible to guarantee response times. Of course, since only two of the over 100 systems examined were specifically designed for real-time operation, these results are hardly surprising.

Two common ways of ensuring that an expert system produces the best solution in the given response time are discussed in [53], and we outline them briefly here. The first method, *progressive deepening*, involves providing the system with successively more detailed layers of analysis. The system analyses the problem to depth 1, then depth 2, then depth 3, and so on, until the time allowed for generating the solution is over, at which point the solution from the deepest completed layer is reported. In this way a solution is always available, and in addition some indication of the confidence that should be placed in it is given by how deep the analysis went in the allotted time. An alternative approach, called *variable precision logic (VPL)*, uses censored production rules as the underlying computational mechanism. Censored production rules are ordinary production rules augmented with exception conditions, and are intended for cases where the implication is usually true and the exceptions are known. Typically they have the form 'if A, then B, unless C' where A,B,C are logical expressions and C is the exception condition. The expert system first ignores C and then, if time permits, examines the implications of C on the result. Thus the certainty of the expert system's conclusions is variable and reflects the investment of computational resources used to generate them.

A more detailed treatment of these and other issues in real-time expert systems is presented in [53], where the desirable properties of such systems are listed and discussed.

Chapter 5

Distribution Automation System Design

A Distribution Automation System (DAS) is in general a complex, interconnected, computer-based network containing a large number of components and capable of implementing a wide range of operational and control functions. Its design and operation are complicated by the interactions between the system elements, both physical (such as the switches, feeders, loads, and so on) and conceptual (such as the various DAS functions). In this chapter we concentrate on the factors which must be considered when designing or modifying a DAS. Thus we will be mainly concerned with the choices at the conceptual level which face utilities evaluating whether or not to install a DAS. Since each utility will have its own priorities and constraints, we merely identify those factors common to all utilities and attempt to indicate how these factors affect the operation of the utility's distribution system.

5.1 DAS design parameters

A partial list of candidate DAS functions is shown in Figure 5.1. We emphasise that, as more experience is gained with Distribution Automation and given relevant technological advances, the list of applications can be expected to grow. Indeed, the widespread implementation of Distribution Automation may motivate such advances, in a similar way to those which followed the introduction of electricity in the first place (though presumably not as wide-ranging !). How is a utility to decide which, if any, of these functions to

- Feeder reconfiguration for loss minimisation and overload prevention
- Integrated volt/Var control
- Cold load pickup
- Protection functions
- Bus/Feeder switching and sectionalising
- Load-shed commands
- Direct Load Control
- Spot Pricing
- Priority Service Pricing
- Wheeling
- Remote metering
- Customer Services
- Power quality diagnostics
- Expert monitoring
- Load survey and forecast

Figure 5.1: Possible DAS functions

apply to its distribution system ? In the absence of regulatory pressures, the first step is to recognise the 'variables' over which the utility has control. Among the variables to be examined are

1. the number of DAS functions implemented;
2. the penetration of Distribution Automation into the distribution system;
3a. the communications system used;
3b. the degree to which DAS computation and control is distributed;
3c. the proportion of DAS operation that must be done in real-time.

In listing those DAS functions considered for implementation, care must be taken to exclude any that are currently in place. However, functions which have a large degree of overlap, such as service restoration and network reconfiguration for loss reduction (Chapter 2), should be counted as separate functions because different algorithms are required to run them. The penetration of Distribution Automation into the distribution system refers to how far down the feeder a particular DAS function is installed. Obviously this does not apply to all functions, but there are some where the utility has freedom in choosing the 'depth' to which the function is to be implemented. For example, we may decide to only install switchable capacitors at the substation - in which case the feeder presents an aggregate load to the capacitor switching algorithm - or alternatively we may wish to have a switchable capacitor at selected distribution transformers, in which case the decision spaces E and U of section 2.1.2 will be different. Note that this penetration may be a function of location in the distribution system.

Suppose now that the variables in 1 and 2 have been fixed. Thus we assume that the number of functions, and their implementation pattern throughout the system, are given. From these values we can calculate the data requirements to support the given DAS implementation. We wish to find the communications traffic generated by these requirements in order to design the backbone communications network. However, this traffic depends on where the data is to be processed and stored, and on the frequency with which the various functions are to be executed. Therefore we must first fix the locations of the computation and control functions used in the DAS to process the data. These issues are discussed in Chapter 4. Knowing the configuration of the multicomputer system, we can derive the communications requirements to be used in the selection of a communications system. But our choice of communications system will then have an impact on the computation and control functions, for example by allowing certain DAS functions to be run more often without requiring any increase in data rate. We might also find that a change in the communication system configuration, which requires a corresponding change in the multicomputer specifications, improves the overall DAS performance (as measured by some combination of the metrics discussed in the next Section). The selection processes can be iterated in this way until no significant improvement is obtained. Thus the communication, control and computation systems are interlinked, and we cannot design one without regard to the effects of our design on the others if we wish to minimise interface difficulties between the systems and optimise DAS performance.

Again we note that the fraction of DAS operation which is to be carried out in real-time is not a variable for all DAS functions. However, we could decide to run spot pricing updates every hour in the neighbourhood of system peak, and every few hours during off-peak operation, if this implementation matches well with observed customer responses.

Note also that, even if the control and computation functions were fixed and not subject to adjustment, details of their operation could still influence - and be influenced by - our choice of communications network. For example, in a distributed processing environment such as those described in Chapter 4, it would typically be impractical to connect every node to a global clock. Thus the multicomputer is an asynchronous system. However, some DAS functions may require different processors to have timebases within a small tolerance of each other. An example of such a function is power quality diagnostics, where the computed quantities used to assess the performance of the distribution system must be accurate although the inputs may be drawn from separate measurement points. The provision of a network sense of time is discussed in [54], where three basic methods are suggested to bound the differences in the senses of time on different processors :

- the use of a network time server which all processors access to have their messages time-stamped. This method relies on the service time of the master clock being much smaller than the average message delay, and on the resolution of contention by the processors for access to the master clock through the choice of an appropriate implementation of the server;
- the use of periodic synchronising signals broadcast to all processors in the system. In this case correct operation depends on processors being able to continue processing

when they are told by the synchronising signal that they have got out of step with the rest of the system, and on an appropriate choice of the frequency with which the synchronising signal is broadcast;

- processors calculate the time delay associated with the received message using knowledge of the system topology, and adjust their local clocks accordingly. Such a system is useful only when it is not possible or necessary to share an accurate sense of time between the processors.

Clearly the choice of method to be used in maintaining a network sense of time will be a factor in the traffic which the communications system is required to accomodate, while the choice of communications system may favour one of these methods over the other two. A more comprehensive discussion of timing issues in distributed processing systems is presented in [54].

5.2 Evaluating the performance of a DAS design

In practice a utility considering the implementation of some or all of the functions in Figure 5.1 should examine various combinations of the parameters discussed in the previous Section in order to arrive at the 'best' DAS under the particular constraints on the utility's operation. How many such combinations need to be considered depends on the sensitivity of the performance of the DAS to changes in these parameters. We envisage an iterative approach to the design of the DAS, in which the relation between system performance and parameter values is estimated through several observations and used as a guide to improve the actual system design. Clearly, the utility must have some way of comparing the advantages and drawbacks of the various designs. We do not expect that the performance of a candidate design can be quantified, at least not at the level we assume in this Section. Thus the metrics introduced below should be thought of as indications of system performance as opposed to exact measures of it. In addition, the relative weights given to the various metrics will vary from one utility to another. For these reasons, we keep our discussion as general as possible.

Among the ways in which alternative DAS designs may be compared are

1. Reliability of service;

2. Economic evaluation;
3. Communications system factors;
4. Computation and control systems factors;
5. Impact on bulk power system operation;
6. Qualitative evaluation.

This list of metrics is not intended to be comprehensive; it simply isolates some of the ways in which different DAS designs behave differently, which can thus be used as bases for comparison of the designs.

There are many different notions of distribution system reliability. We use reliability to indicate the availability and quality of service provided to customers' loads, as in Section 2.1. We have seen in Chapter 2 that many of the DAS functions listed in Figure 5.1 are expected to improve distribution system reliability. There are several different indices of reliability which can be used to estimate this improvement. These indices are all calculated over some period of time : typically a year or longer. Some of the more common ones are

- System average interruption frequency index (SAIFI), defined as the ratio of the total number of customer interruptions to the total number of customers served;
- Customer average interruption frequency index (CAIFI), the ratio of the total number of customer interruptions to the total number of customers interrupted;
- Customer curtailment index (CCI), the ratio of the total curtailment to the total number of customers interrupted;
- Customer average interruption duration index (CAIDI), the ratio of the sum of customer interruption durations to the total number of customer interruptions;
- Average service availability index (ASAI), the ratio of the total number of hours when service was available when demanded to the total number of hours when service was demanded.

Further reliability indices and a discussion of the relative merits of the various indices can be found in [5].

It is of course difficult to attribute the 'correct' amount of improvement in reliability to each of the functions in a given DAS design, but fortunately for comparison purposes we are primarily interested in the change in reliability when individual functions are added or removed. Short of actually implementing the alternative DAS designs and calculating the desired indices from the monitored data, utilities must rely on simulation results to assess the effects of the different designs on system reliability. Software tools are available to assist in this assessment, for example the EPRI-developed Predictive Reliability Assessment Model (PRAM) [55]. However, such tools often make unrealistic assumptions about the distribution system which lessen their usefulness. One possible improvement is for the utility to correct the indices determined by PRAM so that the new 'effective' reliability indices are more applicable to the utility's own system.

A framework for the economic evaluation of the impact of Distribution Automation on the distribution system was presented in [56]. The functional steps in the methodology used to do the evaluation are shown in Figure 5.2. The accuracy of the evaluation is dependent on how the benefits of Distribution Automation are calculated. In [56] these benefits are divided into investment-related, interruption-related, operational-related, and customer-related savings, though a fifth category, improved operation, is recognised but treated as impossible to quantify. The methodology is based on the principle of *present value of revenue requirements (PVRR)*, in which financial effects for future times are corrected to their present-day values by assumptions about average interest rates and rates of inflation in the interim. Recognising the fact that the effects of installing a DAS are spread over many years, the following types of economic analysis are discussed :

- continuing plant analysis, in which the PVRR is calculated to perpetuity, thus placing alternative designs on an equal basis by taking into account the possibly different times at which equipment is expected to be retired;

- short term analysis, in which the PVRR is calculated over, say, the next 10 years and revenue requirements at the end of each year of the study are calculated - the relatively short term nature of this analysis should allow the utility to place more confidence in the result;

- year-by-year analysis, in which the actual revenue requirements in each year of a longer-term period are calculated, giving some indication of the necessary cash flows;

- break-even analysis, in which the difference in cumulative PVRRs for the *base case* (ie. no automation) versus the automated case is calculated for each year in the future,

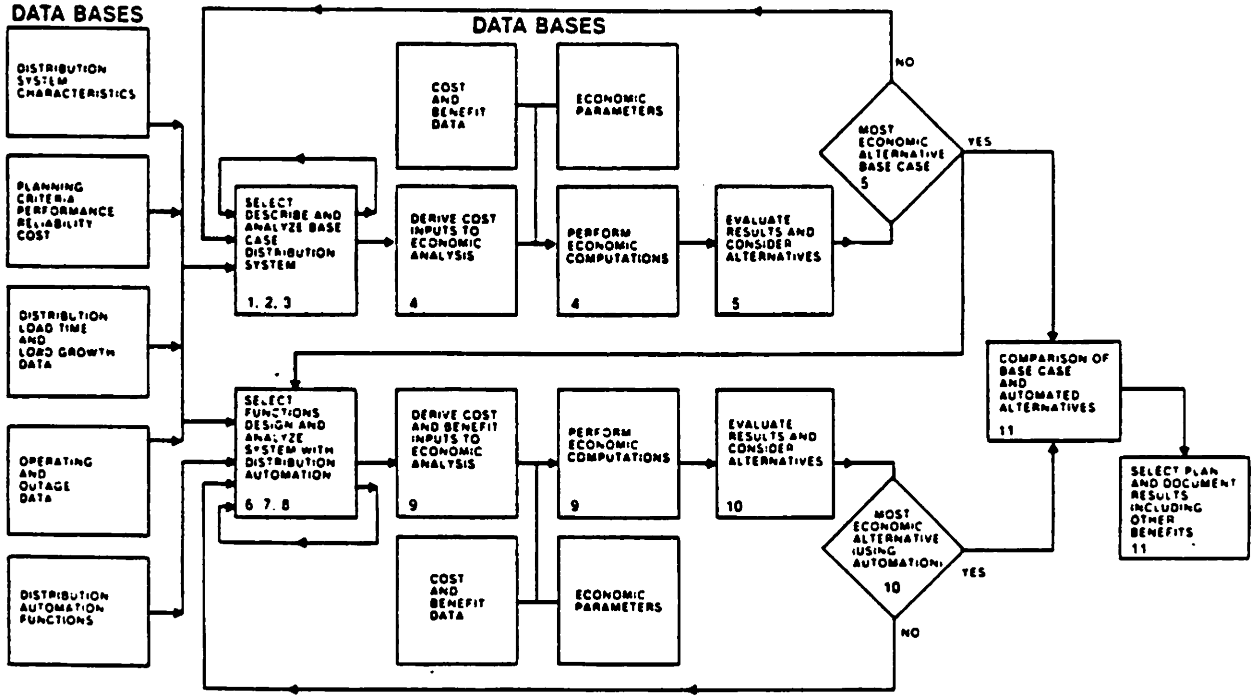

DATA BASES
DISTRIBUTION SYSTEM CHARACTERISTICS
PLANNING CRITERIA PERFORMANCE RELIABILITY COST
DISTRIBUTION LOAD TIME AND LOAD GROWTH DATA
OPERATING AND OUTAGE DATA
DISTRIBUTION AUTOMATION FUNCTIONS
DATA BASES
COST AND BENEFIT DATA
ECONOMIC PARAMETERS
SELECT DESCRIBE AND ANALYZE BASE CASE DISTRIBUTION SYSTEM
1, 2, 3
DERIVE COST INPUTS TO ECONOMIC ANALYSIS
4
PERFORM ECONOMIC COMPUTATIONS
4
EVALUATE RESULTS AND CONSIDER ALTERNATIVES
5
MOST ECONOMIC ALTERNATIVE BASE CASE
5
NO
YES
COMPARISON OF BASE CASE AND AUTOMATED ALTERNATIVES
11
SELECT PLAN AND DOCUMENT RESULTS INCLUDING OTHER BENEFITS
11
SELECT FUNCTIONS DESIGN AND ANALYZE SYSTEM WITH DISTRIBUTION AUTOMATION
6 7. 8
DERIVE COST AND BENEFIT INPUTS TO ECONOMIC ANALYSIS
9
PERFORM ECONOMIC COMPUTATIONS
9
EVALUATE RESULTS AND CONSIDER ALTERNATIVES
10
MOST ECONOMIC ALTERNATIVE (USING AUTOMATION)
10
YES
NO
COST AND BENEFIT DATA
ECONOMIC PARAMETERS

Figure 5.2: Functional steps in DAS evaluation methodology

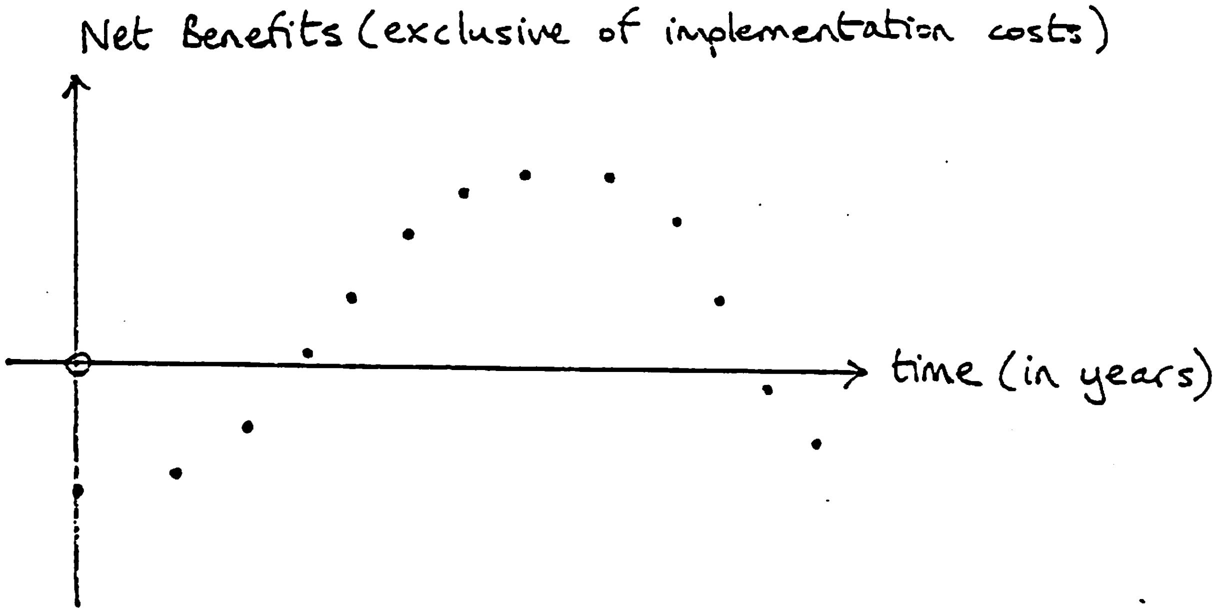


Figure 5.3: Net benefits of load control as a function of time

which gives an estimate of how long it takes the DAS to become more economical than the base case.

One important point is that sensitivity analysis can be very effective in evaluating the economic impact of changing DAS parameters. Data collection requirements can also be reduced, by approximating parameters that are difficult to obtain and then using a sensitivity analysis combined with some known results to estimate the economic impact of these changes.

Another advantage of an economic evaluation is that it can reveal counter-intuitive results. For example, in [57] it is shown that the net benefits of load control can behave as shown in Figure 5.3. This phenomenon can be explained as follows. The use of load control *reduces* the cost of new generation capacity, operation and maintenance costs, fuel costs, and transmission costs, among others, while it *increases* lost revenues and production costs. For example, controlling the Summer air conditioner load reduces the baseline generation capacity but may have no effect on the Winter load requirements; then more expensive peaking units will have to be used which act to offset the gains obtained by the reduced capacity requirements. Let the economic impact of the load control strategy be measured by the cumulative present worth of avoided costs less the decrease in revenue associated with the scheme. Then in the example shown in [57], the net benefit is initially negative since lost revenues outweigh the various savings. Later on the net benefit goes positive because a major generating unit was deferred, but eventually goes negative again due to the necessity of using more costly and less reliable generation. This example suggests once again that operating decisions should be made taking the system as a whole into account to obtain the best performance.

Metrics for comparing the performance of alternative communications systems are discussed in Section 3.2. Link performance is usually specified by the transfer rate of information bits and the residual error rate, while network performance is most often indicated by the throughput (average or maximum) and the average message delay. Multicomputer performance is considered in Chapter 4, where different structures may be compared on the basis of communication and computational capacity and the interconnection network is specified by such quantities as degree, diameter and the mean internode distance.

The distribution system does not exist in isolation but is connected to the bulk system of the utility. Typically the bulk system is controlled by an Energy Management System (EMS), which is a computer-based system that coordinates the real-time operation of the generation and transmission systems to perform various functions such as economic dispatch, unit commitment and state estimation. A DAS design must therefore take into account the issue of integration with the utility's EMS. Depending on the DAS functions chosen, the operation of the EMS can be improved since a more controllable and accurate model of the distribution system is available. For example, the volt/Var control function in the bulk system could deliver its strategy to the DAS at the substation level, and the appropriate action then taken in the distribution system to realise the desired adjustments. Since the potential benefits of integrating the DAS with the utility's EMS depend heavily on the nature of the power system, we will not discuss them further.

Among the qualitative factors which provide an indication of DAS performance, the most familiar are load and loss profiles, load factors, power factors, and participation factors. Definitions of these factors may be found in [5]. The implications of the effect on DAS performance of changes in these factors as the DAS design is varied can be deduced from experience. These metrics are of course implicitly accounted for in some of the earlier metrics, but it may be a useful guide to the validity of a heuristic approach to directly observe the changes in these quantities as DAS parameters are varied. Since many of the factors are currently derived from historical data, basing their calculation on monitored real-time data will allow greater confidence to be placed in them.

Chapter 6

Conclusions

In this report we have studied some of the issues involved in the design and operation of a Distribution Automation System (DAS^c). The emphasis has been on identifying the important considerations and indicating how they relate to one another, rather than on specific problems or implementation details. In addition, the fact that each utility has its own unique priorities and constraints makes a detailed analysis impossible. However, certain key observations relevant to all DASs may be made :

- the various components of a DAS are highly interlinked, and thus should be designed and operated in a coherent manner to obtain the best possible system performance
- the same is true for the DAS functions : identifying those functions which require the same data or affect the same system quantities leads to increased efficiency, while many functions which cannot be justified on their own may become feasible when added in an 'incremental' manner once a basic DAS structure is in place
- many of the benefits of a DAS arise from the more detailed knowledge a utility has about its distribution system and the more 'finely-tuned' control it can exercise once a DAS is installed; these benefits are difficult to quantify since there is no operational experience available with large-scale DASs, but can be expected to play an increasingly important role in the more competitive power supply markets of the future
- in a similar way, many DAS functions which directly involve the customer in the decision process lead to increased customer satisfaction with their electricity supply, which again is expected to figure more and more prominently in utility decisions

The discussion in this report is intended to motivate an investigation of the areas of Distribution Automation that require further study. Many of the problems addressed by the DAS functions have not been solved in general. Once a solution is available, an efficient algorithm which exploits the structure of the distribution system and the proposed DAS is needed. A study must be made of the various communication, computation and control systems suitable for Distribution Automation applications. The performance of these systems must be defined by metrics which will allow the designer of a DAS to compare alternative designs and choose the 'best' one. Perhaps most important of all, utilities must take a critical look at how they operate their distribution systems, in order to decide which operational considerations are relevant to them and in what order. It may be that, for some utilities, Distribution Automation is not a realistic option, but no utility can afford *not* to consider the possibility.

Bibliography

[1] Baran, M. E. and Wu, F. F., 'Network reconfiguration in distribution systems for loss reduction and load balancing', Paper 88 SM 556-3 presented at the IEEE Power Engineering Society 1988 Summer Meeting, Portland, Oregon, July 1988.

[2] Baran, M. E. and Wu, F. F., 'Optimal sizing of capacitors placed on a radial distribution system', Paper 88 WM 065-5 presented at the IEEE Power Engineering Society 1988 Winter Meeting, New York, New York, January 31-February 5, 1988.

[3] Shirmohammadi, D., 'Advanced Application Software for 3C', Work Paper (unpublished), January 1988.

[4] Natoli, D. R., 'A Distribution Automation Approach to Controlling Dispersed Storage and Generation Systems', Master's Report, EECS Department, UC Berkeley, August 1983.

[5] Gonen, T., *Electric Power Distribution System Engineering*

[6] Baran, M. E. and Wu, F. F., 'Optimal Capacitor Placement on Radial Distribution Systems', Paper 88 WM 064-8 presented at the IEEE Power Engineering Society 1988 Winter Meeting, New York, New York, January 31-February 5, 1988.

[7] Baldick, R., personal correspondence, December 1988.

[8] Baldick, R., 'Optimal On-Off Control of Capacitors in a Distribution System', Master's Report, EECS Department, UC Berkeley, May 1988.

[9] Rizy, D. T., Lawler, J. S., Patton, J. B. and Nelson, W. R., 'Measuring and Analysing the impact of voltage and capacitor control with high speed data acquisition', Paper

88 WM 098-6 presented at the IEEE Power Engineering Society 1988 Winter Meeting, New York, New York, January 31- February 5, 1988.

[10] Ohyama, T., Watanabe, A., Nishimura, K. and Tsuruta, S., 'Voltage dependence of composite loads in power systems', IEEE Transactions on Power Apparatus and Systems, PAS-104, pp. 3064-3073, November 1985.

[11] 'Application and Coordination of Reclosers, Sectionalisers and Fuses', IEEE Tutorial Course 80 EH0157-8-PWR.

[12] Bergen, A. R., *Power Systems Analysis*, Prentice-Hall (1986).

[13] 'Microprocessor Relays and Protection Systems', IEEE Tutorial Course 88 EH0269-1-PWR.

[14] Girgis, A. A. and Makram, E. B., 'Application of Adaptive Kalman Filtering in fault classification, distance protection and fault location using microprocessors', IEEE Transactions on Power Systems, Vol. 3, No. 1, pp. 301-309, February 1988.

[15] Cohen, A. I. and Wang, C. C., 'An optimisation method for load management scheduling', PICA 87 Proceedings, pp. 72-78.

[16] Larson and Casti, *Principles of Dynamic Programming*, Vols. 1 and 2, Marcel Dekker Inc., New York (1978).

[17] Runnels, J. E., 'Impacts of Demand-Side Management on T and D - Now and Tomorrow', IEEE Transactioas on Power Systems, Vol. PWRS-2, No. 3, pp. 724-729, August 1987.

[18] Geier, D. L. and Samaniego, G. M., 'Evaluation of Load Management as an Electric System Resource', Paper 85 SM 475-9 presented at the IEEE Power Engineering Society 1985 Summer Meeting, Vancouver, B. C., Canada, July 14-19, 1985.

[19] Schweppe, F. C., Tabors, R. D., Caramanis, M. C. and Bohn, R. E., *Spot Pricing of Electric Energy.*

[20] Crane, C. M., 'Demand-Side Real-Time Pricing 1987 Annual Report', PG&E Rate Department, March 1988.

[21] Kaye, R. J. and Outhred, H. R., 'A theory of electricity tariff design for optimal operation and investment', presented at the IEEE Power Engineering Society 1988 Summer Meeting, Portland, Oregon, July 1988.

[22] *Priority Service : Unbundling the quality attributes of electric power*, EPRI EA-4851, Project 2440-2, Interim Report, November 1986.

[23] Schweppe, F. C., 'Mandatory Wheeling : A Framework for Discussion', Paper 88 SM 690-0 presented at the IEEE Power Engineering Society 1988 Summer Meeting, Portland, Oregon, July 1988.

[24] Lee, E. A. and Messerschmitt, D. G., *Digital Communication*, Kluwer Academic Publishers (1988).

[25] Pandhi, S. N., 'The Universal Data Connection', IEEE Spectrum, pp. 31-37, July 1987.

[26] Leiner, B. M., Nielson, D. L. and Tobagi, F. A., 'Issues in Packet Radio Network Design', Proceedings of the IEEE, Vol. 75, No. 1, January 1987.

[27] O'Neal, J. B., Jr. and Hayden, L. E., 'Important Performance Criteria for Distribution Line Carrier System', IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, No. 7, July 1982.

[28] Pursley, M. B., 'Frequency-Hop Transmission for Satellite Packet Switching and Terrestrial Packet Radio Networks', IEEE Transactions on Information Theory, Vol. IT-32, No. 5, September 1986.

[29] Gaushell, D. J., 'Automating the Power Grid', IEEE Spectrum, pp. 39-45, October 1985.

[30] Tengdin, J. T., 'Distribution Line Carrier Communications - A Historical Perspective', IEEE Transactions on Power Delivery, Vol. PWRD-2, No. 2, April 1987.

[31] Mak, S. T. and Reed, D. L., 'TWACS, a new viable 2-way automatic communication system for Distribution networks : Part I', IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, No. 8, August 1982, AND Mak, S. T. and Moore, T. G., 'TWACS, a new viable 2-way automatic communications system for Distribution networks : Part II', IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 8, August 1984.

[32] Hagmann, W. 'A Spread Spectrum Communication System for Load Management and Distribution Automation', Paper 88 WM 060-6 presented at the IEEE Power Engineering Society 1988 Winter Meeting, New York, New York, January 31-February 5, 1988.

[33] Hemminger, R. C., Gale, L. J. and O'Neal, J. B., Jr., 'Signal propagation on single phase power distribution lines at power line carrier frequencies', IEEE Transactions on Power Delivery, Vol. PWRD-2, No. 1, January 1987.

[34] Borowski, D. C., Gale, L. J. and O'Neal, J. B., Jr., 'Effects of artificially loading distribution line carrier networks', IEEE Transactions on Power Delivery, Vol. 3, No. 1, January 1988.

[35] Holbrow, W. F. and Owen, R. E., 'Two-way utility system communications using AM broadcast radio', IEEE Transactions on Power Apparatus and Systems, Vol. PAS-104, No. 1, January 1985.

[36] Anderson, H. R., 'Measured data transmission performance for AM broadcast-VHF radio distribution automation communication system', Paper 88 WM 058-0 presented at the IEEE Power Engineering Society 1988 Winter Meeting, New York, New York, January 31-February 5, 1988.

[37] Tobagi, F. A., 'Modeling and Performance Analysis of Multihop Packet Radio Networks', Proceedings of the IEEE, Vol. 75, No. 1, pp. 135-155, January 1987.

[38] Holte, K. C., 'Netcomm : A utility/customer communication network', presented at the Pacific Coast Electrical Association, Inc. Engineering and Operating Conference, San Francisco, March 17, 1988.

[39] Pursley, M. B., 'The role of spread spectrum in packet radio networks', Proceedings of the IEEE, Vol. 75, No. 1, pp. 116-134, January 1987.

[40] Pickholtz, R. L., Schilling, D. L. and Milstein, L. B., 'Theory of Spread Spectrum Communications - A Tutorial', IEEE Transactions on Communications, Vol. COM-30, No. 5, May 1982.

[41] Schwartz, M., *Telecommunication Networks*, Addison-Wesley (1987).

[42] Neyer, A. F., Imhof, K. and Wu, F. F., 'An object-oriented data representation and task based reasoning system for energy management systems : a proposal', Memorandum No. UCB/ERL M87.75, October 1987.

[43] Baudet, M. G., 'Asynchronous iterative methods for multiprocessors', Journal of the Association for Computing Machinery, 25(2): 226-244, April 1978.

[44] Bhuyan, L. N., 'Interconnection Networks for Parallel and Distributed Processing (Guest Editor's Introduction)', IEEE Computer, June 1987.

[45] Karp, A. H., 'Programming for Parallelism', IEEE Computer, May 1987.

[46] Howe, C. D. and Moxon, B., 'How to program parallel processors', IEEE Spectrum, September 1987.

[47] Wiley, P., 'A parallel architecture comes of age at last', IEEE Spectrum, June 1987.

[48] Reed, D. A. and Grunwald, D. C., 'The performance of multicomputer interconnection networks', IEEE Computer, June 1987.

[49] Hong, S. J., Guest Editor's Introduction to IEEE Computer Special Issue on Expert Systems, July 1986.

[50] Adiga, S. Notes for IEOR 298-5, Fall 1988, IEOR Department, UC Berkeley.

[51] Talukdar, S. N., Cardozo, E. and Leao, L. V., 'TOAST : the power system operator's assistant', IEEE Computer, July 1986.

[52] Liu,C.-C. and Tomsovic, K., 'An expert system assisting decision-making of reactive power/voltage control', IEEE Transactions on Power Systems, Vol. PWRS-1, No. 3, August 1986.

[53] Laffey, T. J. *et al*, 'Real-time knowledge-based systems', AI Magazine, Spring 1988.

[54] Volz, R. A. and Mudge, T. N., 'Timing Issues in the distributed execution of ADA programs', IEEE Transactions on Computers, Vol. C-36, No. 4, April 1987.

[55] EPRI Report, *Development of Distribution System Reliability and Risk Analysis Models*, EL-2018, Research Project 1356-1, August 1981.

[56] EPRI Report, *Guidelines for evaluating Distribution Automation*, EL-3728, Project 2021-1, Final Report, November 1984.

[57] Nordell, D. E., 'An approach to conservation and load management economic analysis', IEEE Transactions on Power Systems, Vol. PWRS-2, No. 1, February 1987.