Interaction Notes

Note 449

June 1985

EMP Hardening Topology Expert System (Hard Top)

Michael A. Messier

JAYCOR
Santa Barbara, California, 93105

Abstract

The architecture for an expert system for use as an aid in the EMP hardening of complicated systems, such as aircraft and ships, is developed. It is determined that the best form for the topology knowledge base is one that mimics the topology data base: graphic. A specific graph model is explored as a candidate data base structure. Examples of the use of this data base are shown.

INTRODUCTION

It is sometimes necessary to retroharden military systems to the effects of EMP. These systems, e.g., ships or aircraft, are generally quite complicated from an electronics viewpoint, perhaps having an electrical and electromagnetic topology which has evolved in a semi-random fashion over long periods of time. It is difficult to analyze the hardening and shielding levels of a system which has evolved in such a manner; it is even more difficult to design "fixes" which are compatible with the existing system, i.e., cause no undesirable side effects, cross couplings, interference, etc. Ideally, we would like to be able to identify all incomplete shielding topologies, ground loops, cable cross-coupling paths, and other EMP coupling points. We would like to be able to prescribe fixes in terms of additional (or better designed) shielding, appropriate protective devices (active or passive), cable re-routings, etc. Ideally, we would like to expose the entire system to threat level EMP environments in order to measure all important responses and confirm the effectiveness of the prescribed fixes. In lieu of full scale environment exposure, it would be desirable to accurately analyze the system responses to several lower level tests, e.g., current injection or dipole radiation and extrapolate the resulting measurements to the responses which would have been obtained had a full scale test been conducted.

The development of microcomputer and expert system technology has reached the point where it now appears possible to approach this ideal situation. Specifically, it should be possible to develop an expert system which can perform the desired analysis of a real system and which can even be run on one of the newly introduced lap top computers in real time at the site of the test. Such a real time system would provide the instant feedback which is so desirable in analyzing the effectiveness of a recent hardening fix. It is probably safe to say that the major factor hindering the development of such a system is not the hardware or software technology, but the development of an appropriate knowledge base which can be used by the expert system. In order to facilitate the development of such a knowledge base, it will be necessary to develop a prototype system which we can place into the hands of hardening experts. This will provide an initial context in which they can develop the necessary base. As they develop their capability and discover the limitations of the expert system shell, they will be able to provide the feedback necessary to provide a more functional shell. In addition to developing the basic shell structure, this part of the testing phase will be critical to the development of an optimum user interface. Graphical displays will be vital to the rapid and comprehensive display of a large amount of data concerning the tested system's physical architecture, its electrical topology, and the electrical response of individual components. The development of the user interface is a major project in its own right.

It is reasonable to ask why we should move in the direction of an expert system rather than search for exact solutions. There are many reasons for this. They are all based upon the complexity of the problem and the fact that it is not always that well defined. First of all, the term "expert system" might better be replaced by "heuristic system" or "artificial intelligence", since we plan to go beyond a simple rule based system to one which recognizes topological patterns, can learn from experience, and create new shielding and protection schemes based on the knowledge that it has acquired. It will not be limited to solving an interaction matrix and spitting out a bunch of numbers for someone to interpret. If it ever needs to compute an interaction matrix, it will do so after first using heuristic rules to narrow down the matrix to something more managable and solvable within the lifetime of the operator. Similar methods are used in chess programs, for example. In addition, an intelligent heuristic system can operate with incomplete data. Given the complexity of real systems, the so-called exact solution can easily be an exact solution of an inexact problem: highly misleading at best. For example, what good is the exact solution for the distribution of currents in a cable bundle when the exact termination impedances are in question, or, for that matter, the current and voltage sources themselves. An intelligent set of general rules would be more appropriate, given the overall uncertainty, and probably faster and more efficient.

SYSTEM ARCHITECTURE

Figure 1 shows the architecture of the HARD TOP system as presently conceived. In its full blown glory, it is a fairly complicated system, composed of several modules. Each of these modules will be fairly independent and can be developed separately, so long as the interfacing protocols are developed first. Most people think of an expert system as a heuristic machine operating with a set of "IF-THEN" rules. The rules are obtained by interviewing experts in the field of interest and discovering how they reason their way through problems. In a high level system, the rules have associated confidence factors, so that the resulting set of conclusions can be rated according to some sort of confidence level. Fuzzy sets and fuzzy logic are mathematical tools which can be applied to situations which are not "black or white". These rule based systems are very powerful in many situations. Our own attempts at building a rule based hardening analysis system have shown that the results can be quite inadequate. Other forms of knowledge representation, as well as an integrated numerical capability, are beneficial (or absolutely required) for practical application.

The form of the knowledge representation should closely approximate the form of the data representation. In this case, the data structures which represent the electromagnetic and electric topologies are in the form of graphs (nodes and branches). To whatever extent possible, the knowledge base, which describes desirable and undesirable topologies, should be expressed in the form of graphs. This does not mean that all of the knowledge must be in this form; certain types of knowledge are best expressed in the form of rules. The analysis sequence, for example, is best controlled in the form of rules. Most IF-AND-THEN rules can be expressed in the form of graphs, anyway, so this is not a serious restriction. Once a rule is expressed in a graphical form, it is possible to use pattern matching techniques to compare knowledge against data.

The architecture described by Figure 1 has provision for both sequential analysis and pattern mathcing analysis. In addition, there is provision for numerical analysis. All together, this arrangement resembles that of the human brain, in which the left half is primarily concerned with sequential, precise, reasoning and the right half is

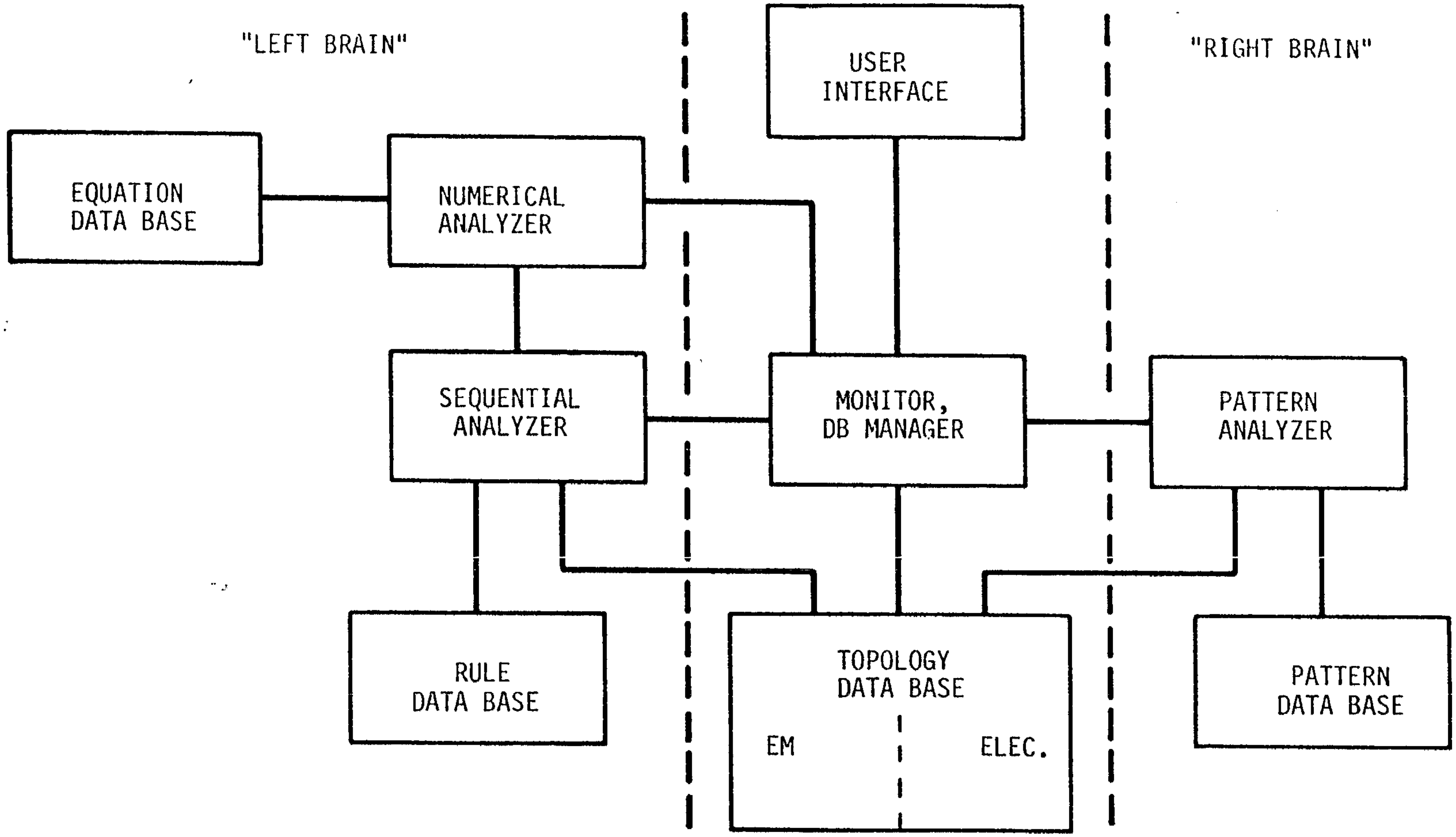


Figure 1. Architecture of complete hardening topology expert system.

optimized for parallel, spatially oriented types of thinking. The core of the system is the center section of the diagram. It consists of the user interface, the monitor/data base manager, and the topology data base. This is the part of the system which will be constructed first because (1) we know basically how to do it, and (2) it performs an immediately useful function even without all of the auxilliary intelligence. The remainder of the system can evolve from the core over a period of time with succeeding generations acquiring more capability and intelligence. The monitor/data base manager (M/DBM) is the heart of the system. It is essentially an operating system with the minimal amount of intelligence required to maintain and allow easy user access to the topology data base. All user access is through the graphic user interface. The graphic interface allows the display of topological data at several levels and in several forms. For example, a user could call up a symbolic display, in the form of a node-branch graph, of the overall electromagnetic (shielding) or electrical (cable) topology, or a sublevel of it, or a small physical section. Alternatively, the user could call up a wiring (circuit) diagram of a section of the system, with certain wires or bundles emphasized in contrasting colors, or he can trace cables in high resolution through the use of arrow keys (or a mouse, if available).

The monitor will eventually be responsible for coordinating the activities of the other modules, which will probably operate as co-processes. Thus, as microcomputer technology advances to the point where separate processors can be used for each module, each with the optimum characteristics for its assigned function (rule base evaluation, high speed numerical processing, and parallel pattern processing), the basic architecture will not require massive changes. In the meantime, all of the coprocessing will have to be simulated in software.

All activities center about the topology data base so that the proper functioning of the DBM is crucial to the operation of the entire system. As will be described in more detail below, the electromagnetic and electrical topologies of the system being tested will be stored in the form of graph structures. Separate structures will be used to record the data corresponding to each form of topology; these will have to be linked so that we know which cable bundle is passing through which shield, enclosure, or bulkhead. The data will probably be entered in a semi-random fashion, so that it will be up to the DBM to not only organize it, correctly renumber the nodes and branches and provide the proper connectivity relations, but to recognize improperly entered data (resulting in physically unrealizable or improbable topologies) and request assistance from the user. Thus, the DBM will require a minimal topological knowledge base to perform its task. Since this knowledge is best expressed in the form of graphs acting as template examples of good and bad structures, the preliminary expert system design will need to include a rudimentary form of the pattern analyzer module.

The monitor will have a basic, fixed set of instructions for the performance of its housekeeping and interfacing functions. In its more advanced stages, the expert system will be controlled primarily by the sequential analyzer. The sequential analyzer is the rule based system usually associated with this type of activity. It will probably operate in the backtracking mode, i.e., it will start with a goal and work back through all of the sub-goals required to satisfy it. The primary goal or "prime directive" of the system is to prove that the topology is EMP hard. In order to prove this, it will have to satisfy several sub-goals, such as the absence of magnetic coupling loops, the absence of unprotected apertures, proper protection of shield penetrations, etc. In order to prove these sub-goals, it will frequently have to call on the pattern analyzer to test certain aspects of the topology and on the numerical analyzer for specific numbers ranging from something as simple as a skin depth calculation to something as complicated as a scattering matrix calculation. Of course, it will not always be necessary or desirable to start from the primary goal and run the entire 100 years. The user may desire a check on only the adequacy of the aperture protection. In such a case, he would simply start with one of the sub-goals. It will always be possible to interrupt the sequential analyzer and use the DBM in order to investigate specifics of the topology or modify the model.

The programming language which is best capable of performing all of the functions on a single processor system appears to be LISP. It is available in several dielects for the major microcomputer systems. Several of these microcomputer versions include compilers and floating point support, a necessity for fast operation, as well as turtle graphics support for the user interface. Another feature which may prove useful is the ability to handle objects and classes, originally introduced as part of the SMALLTALK programming environment. These features are available for both the popular MSDOS line of computers (IBM PC and compatibles) and the Apple Macintosh, with its superior graphics capability.

In the remainder of this note, we will discuss the topology representation. The graph representation was developed independently from that developed by Baum, Tesche, Vance and others (References 1 through 13) and takes a somewhat different viewpoint. Naturally, it is equivalent to theirs, since the same objects and relations are being represented. Baum has developed his representation to a much higher degree than we have and it is quite possible that we will ultimately convert to his (or at least translate the results of his investigations to our notation). For the purpose of this note, however, we will retain our notation and concepts. We feel that it is healthy to present alternate viewpoints, since each viewpoint tends to emphasize different aspects of the problem. As we describe our model, we will show how it relates to Baum's. The description will also include discussions of the various operations which will be performed on the data, such as node/branch renumbering and the recognition of good and bad topologies through pattern matching techniques.

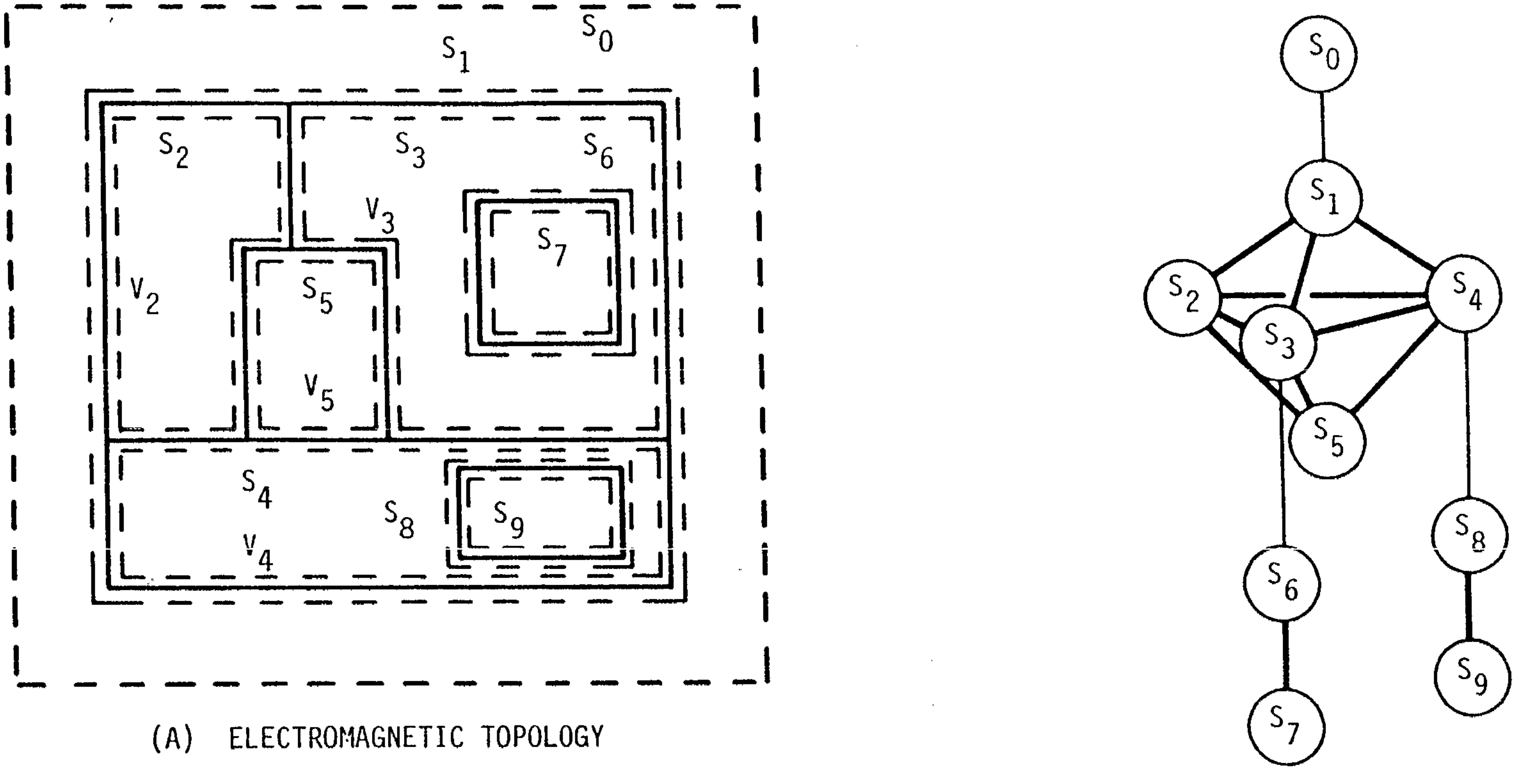
Following Baum, we define two types of topology: electromagnetic and electrical. Electromagnetic topologies deal with the question of shielded enclosures and, as such, describe closed surfaces and their connection and penetration. Electrical topologies are concerned with the connection of cables and wires. These cable bundles can be characterized by enclosing surfaces or "tubes" and junctions. Technically, electrical topologies can be considered subsets of electromagnetic topologies, and it is, after all, the combined topology that we are concerned with in the hardening analysis of a system. For practical reasons, however, it is more convenient to treat the two topologies separately and provide an interface so that the connections between the two can be traced. It is somewhat arbitrary where one treats the connection between the two. For example, a description of the penetration of a cable through a shield can be recorded either in the data structure describing the shield or in the data structure describing the cable tube, or both. In our scheme, protective devices and shield groundings associated with the penetration of tube through an aperture are recorded in the electrical topology. The physical characteristics of the aperture itself, e.g., size and shape, are recorded in the data structures associated with the electromagnetic topology. The electromagnetic data structures also contain pointers to the nodes describing any cables associated with a penetration or connection between surfaces. These pointers complete the interface structure. It is possible to follow the tubes through the shield surfaces or, conversely, to identify all tubes penetrating a surface and follow the cables from there.

The major difference between electromagnetic and electrical topologies, and the one which makes it convenient to treat them separately, is that electromagnetic topologies involve closed surfaces and electrical topologies involve open ended tubes. Electrical connections must pass through sequential shield levels in the electromagnetic topology; rules are easily formulated which define "good" connections and deviations from these can be spotted by the expert system. On the other hand, the open ended tube structure of the electrical topology allows surfaces to be bypassed by electrical connections. This is compensated for by other rules, analogous to Kirchoff's laws, which govern the connections (Reference 4). We first describe the electromagnetic topology representation and then the electrical topology representation. Examples showing the interfacing between the two will be described in the last section.

The graph structure used in HARD TOP might be described as "surface based". In other words, we treat closed surfaces as the primary objects and think of surfaces as being separated by volumes. The volumes can be either air or the metal within a physical shield. In the graph, numbered nodes represent closed surfaces and branches represent the logical connections between them. The graph requires two types of branch. A "penetration" branch represents possible signal paths between the two surfaces on either side of a metallic shield; a "connection" branch represents the same for two surfaces separated by an air volume. Penetration branches contain data about apertures and/or the diffusion properties of the shield; they also contain pointers to any cable bundles which penetrate the apertures. Connection branches usually point to data concerned with cable type connections between surfaces exposed to air, e.g., ground straps or grounded cable shields. Connections can also represent radiative transfer, as would be the case with an antenna on the outside of an aircraft and its connection to the surface at infinity.

The graph structure used by Baum, et al., could be considered "volume based". The primary object, represented by numbered nodes, is the air volume. The branches represent logical connections between these volumes, each of which passes through a surface. Baum's graph structures contain two types of nodes: air nodes and surface nodes. Since the branches between volume nodes pass through surfaces, he must include some type of data structure that describes the penetration. This appears in the graph as a surface node which bisects each branch.

These structures, and their correspondence, will be made more clear if we consider an example. Figure 2(a) shows a shielded system with the surfaces numbered in a simple manner. In our notation, volumes are given the same numbers as the surface which encloses it. Volume numbers do not appear in the graph structure, they exist only for descriptive convenience. Figure 2(b) shows the corresponding graph, using our surface node model. The branches drawn as thick lines represent penetration branches. The branches drawn as thin lines represent connection branches. Note that the surface nodes connected by penetration branches form clusters. These clusters will form the basis for a more



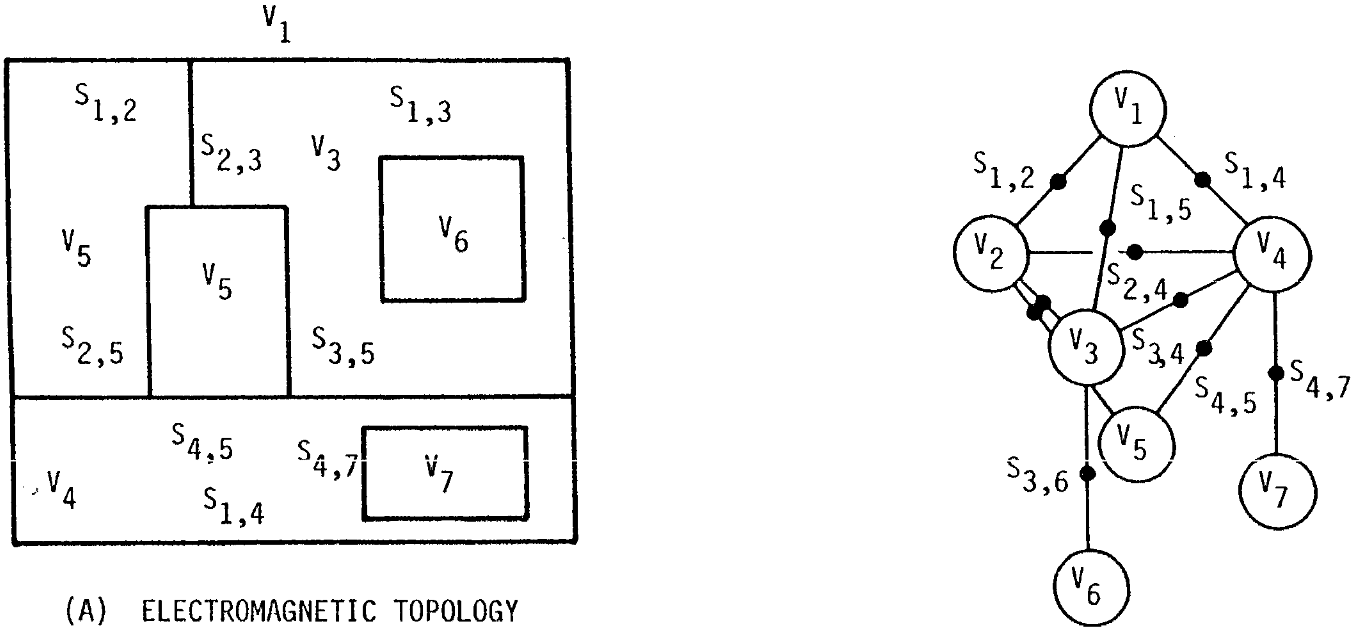
(A) ELECTROMAGNETIC TOPOLOGY

(B) GRAPH

Figure 2. Example of surface based topology and resulting graph structure.

sophisticated numbering system. Figure 3(a) shows the same shield system with the volumes numbered in a simple way as the primary objects. In this representation, surface segments are given double subcripts which take on the values of the two volumes they connect. The corresponding volume based graph is shown in Figure 3(b). Note that the overall structure of the graph is the same as that of Figure 2(b). Even the node numbers are the same through the first cluster. This is because the closed surfaces correspond to the same volumes at this level. The volume based graph has an additional node on each of the branches. Each of these represents the surface segment through which the branch penetrates. This is the last time we will discuss the volume based graph. Further details can be found in References 6, 8 and 9.

We temporarily divert now to the subject of node numbering. For the basic functions of pattern matching and graph tracing, numbers are not necessary; the only requirement is that each node in the graph have some sort of unique name or pointer. On the other hand, advanced analyses often involve matrix operations and some appropriate multidimensional indexing scheme. It is desireable to have this scheme reflect the electromagnetic or shielding relationship between the nodes. We, will discuss one such scheme below, one which follows the spirit of Baum's, but is based upon the surface based graph structure discussed in this note. Before that, and just to illustrate the variety of numbering schemes possible when one sets his mind to it, we point out a scheme which can be easily applied to tree-like structures. Trees are recursively defined structures. Every node in a tree, except the last "leaf" nodes, has a sub-tree associated with it. It, in turn, is a child of a parent node (except for the first or "root" node). It is possible, of course, for a node to have more than one parent, in which case we are not dealing with an ordinary tree. In this case, which unfortunately is the situation in which we find ourself, recursive numbering cannot be used since the node can acquire more than one name. When that is not the case, a node can be numbered by attaching another integer to the number of its parent (or another name to the name of its parent). Since the root node is unique and common to all, it is not necessary to assign an integer to it. Each node below it is defined by a single integer arbitrarily assigned (perhaps left to right in the tree). Each node attached to these, assumes this integer as the first descriptor and adds a second integer using the same arbitrary scheme as the first, but local to the subtree. This procedure



(A) ELECTROMAGNETIC TOPOLOGY

(B) GRAPH

Figure 3. Example of volume based topology and resulting graph structure.

continues until at the n-th level of the tree, below the root node, each node is described by n integers. As we already mentioned, this scheme will not work for our graphs in their present form. Future research may show that the graphs can be rearranged by duplicating nodes, so that the numbering system can be applied. The resulting sequences may be viewed as an infinitely dimensioned orthogonal space which might have some beneficial metric and ordering qualities not available in the more common array representations.

The node numbering system devised for this graph system is based upon a partitioning of the nodes by clusters. The results are similar in concept to the sublayer partitioning of Baum (References 8 and 9). We define a cluster as a set of adjacent surface nodes, all of which are connected by penetration branches. In Figure 2(b), for example, the highest level cluster consists of nodes S_1 through S_5. There are two clusters at the next and final level: S_6-S_7 and S_8-S_9. The surface at infinity (S_0) might be considered a special case, if that proves convenient. The clusters form a graph of their own, so we can begin by numbering them. Within each cluster, the surface nodes form a similar graph. Thus, each surface node inherits the cluster number and attaches a local number. Though we will ignore the subject for now, each surface node can be thought of as a set of surface sub-nodes in certain pattern matching operations. Each of these are connected to all other sub-nodes by connection branches. The numbering system can be extended down to this level in the same way, adding a third set of indices. This is not an operation we intend to perform very often, so we will ignore it for now.

The clusters are numbered first by level in the graph (below S_0) and then by an arbitrarily assigned integer. The symbol "C" will be used to denote a cluster much as "S" was used to denote a surface node. The cluster just below S_0 is numbered C(1,1). If it had a sibling, also connected to S_0, the sibling might be numbered C(1,2), as shown in Figure 4. The two clusters connected to C(1,1), at the second level, are numbered C(2,1) and C(2,2). If there were any at the next level, they would be numbered C(3,n), where n is a member of a sequence of integers in the range of 1 to N, the number of clusters on that level. Note that N continues across all branches of the cluster graph. Thus if C(1,2) was the parent of a set of nodes on level 2, it would have siblings with

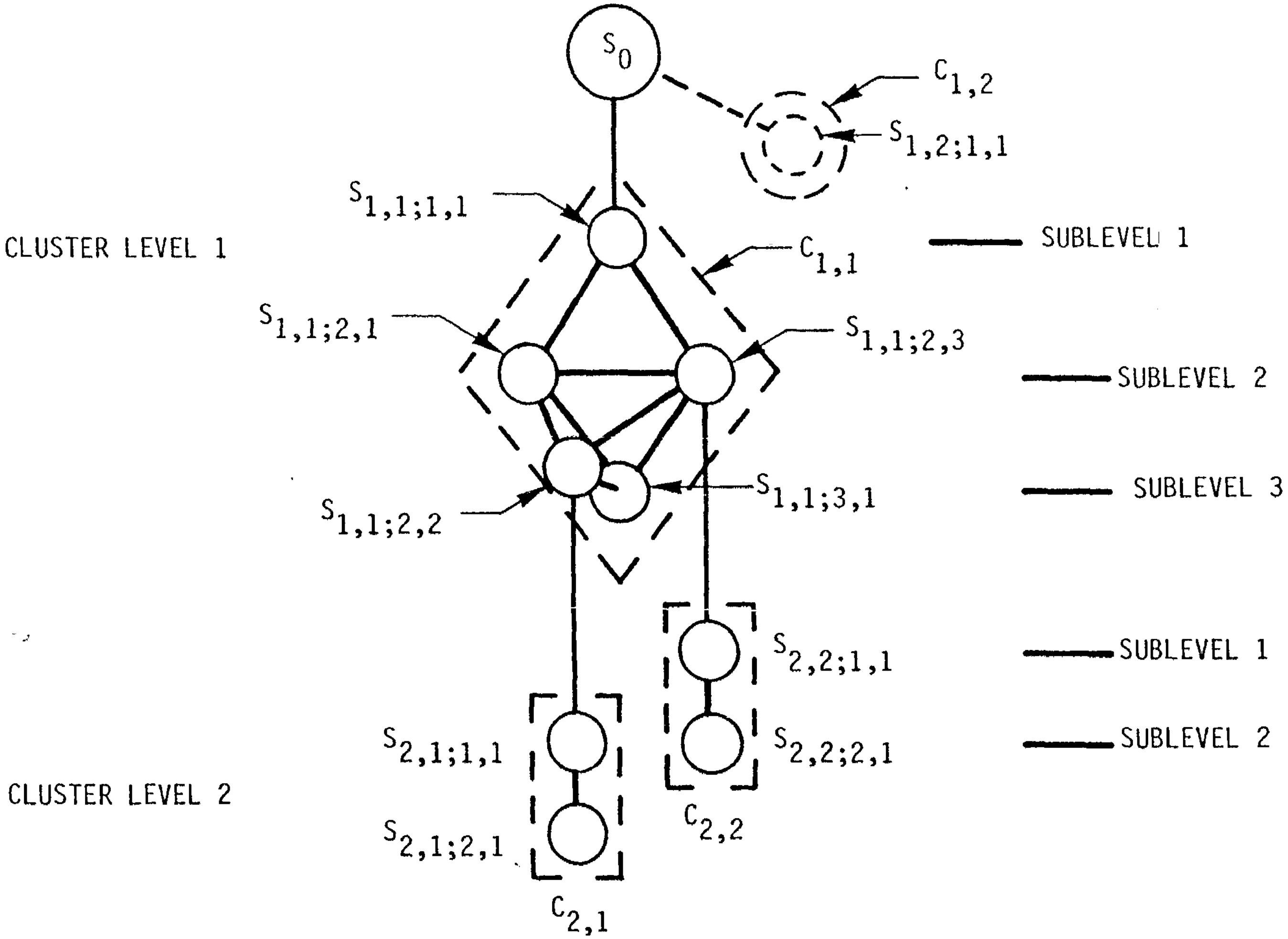


Figure 4. Cluster numbering system.

numbers starting with C(2,3). The surface nodes within a cluster inherit the cluster node number as the first set of indices. The second set of indices is again formed by the level number (sublevel within the cluster) and an arbitrarily assigned integer. The total surface node number is then of the form S(m,n;i,j) where m is the cluster level, n is a cluster level i.d., i is a cluster sub-level, and j is a sub-level i.d. The number (i-1) is the number of shields that a signal must penetrate to reach the node within that cluster (perhaps we should have started the numbering at 0 so that i equals the shield number). If that number is added to the (i-1) of a surface node in an attached cluster, we have the total number of shields traversed to reach that node, etc. The number of air crossing connections required to bridge two nodes in sequentially connected clusters is simply the difference in cluster levels, m. If the clusters are not sequentially connected, the connection must pass through a cluster at a higher level (lower cluster level number, closer to the root S_0). The number of connections is then the sum of the differences between the levels of the two termination clusters and the common cluster through which the connection passes (don't worry if this sounds confusing, its probably not very important).

We now discuss how pattern matching techniques can be utilized to apply a knowledge base to a topological model. For the purposes of illustration, consider the problem of finding closed loop connections between surfaces. This is a common problem which one desires to avoid because such loops are very good at coupling to magnetic fields and producing large surface and cable shield currents. Figure 5 shows how we might define a closed loop in terms of node-branch connections. A closed loop is a surface node connected to one or more nodes, which are, in turn connected to a final node which is itself connected to the first. The branches in all cases are of the connecting variety (as opposed to the penetration type). This is complicated to say but easy to graph. Figure 6(a) shows a rather simple geometry in which two enclosures are connected to each other as well as being grounded to their enclosing surface. The loop is obvious to the casual observer. How does the expert system find it? Figure 6(b) shows the corresponding graph. A basic search procedure is involved:

1. A node is found with two connective branches

2. The branch structures are checked for the presence of physical connections

3. If these are not found, an orderly search through the remainder of the structure is continued, otherwise

4. A node-by-node pattern match is attempted

5. A success or failure is reported and the search continues if desired.

Two variations on this procedure need to be considered. If more than three surface nodes are involved in the possible loop, the intermediate template nodes can be continually split into multiple node branch pairs until a complete match either succeeds or fails. This is a standard "wild card" pattern matching technique. Similarly, if a system node is associated with a branch which indicates two or more physical connections, that surface node can be split into that many subsurface nodes connected to each other by connective branches. The search then continues on the expanded graph. The last situation might appear if an enclosure is grounded at two or more points.

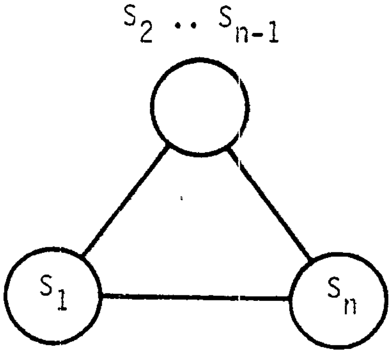
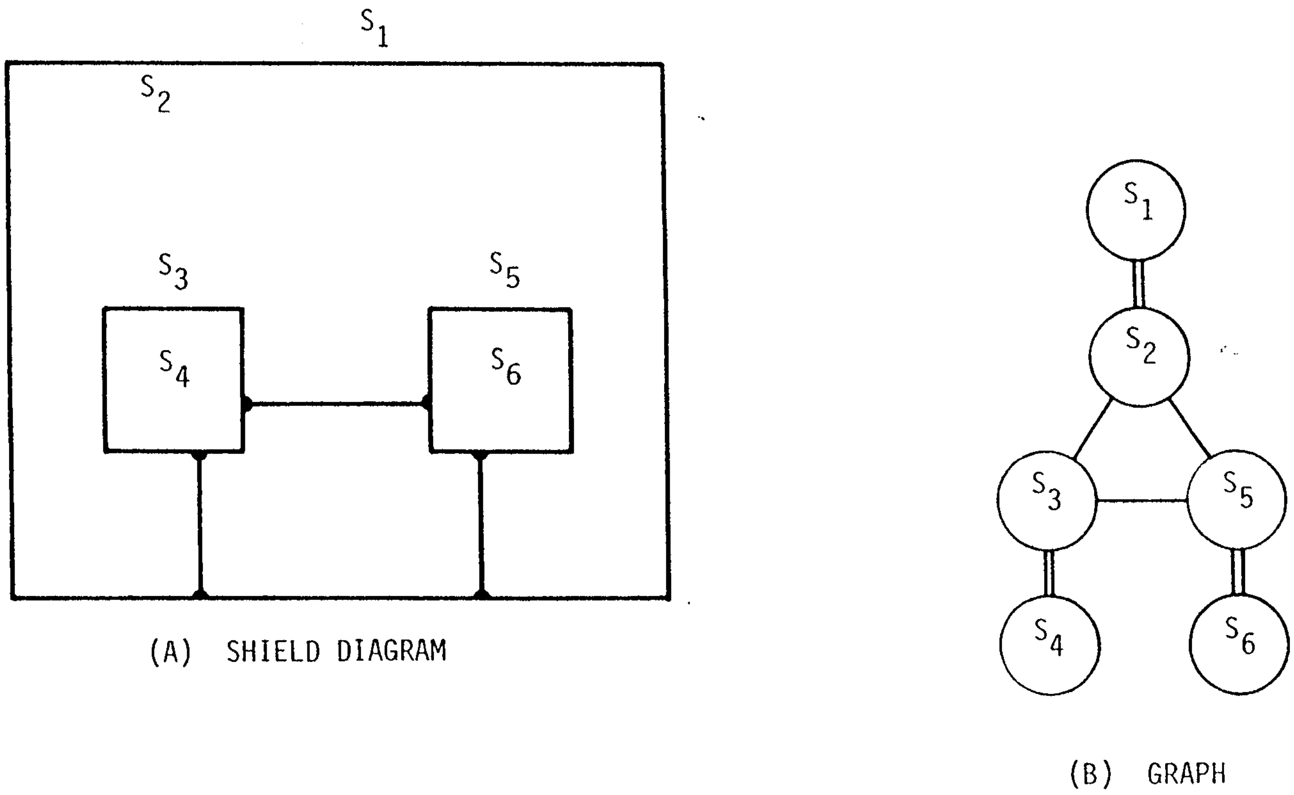


Figure 5. Graph representation of a closed loop.



(A) SHIELD DIAGRAM

(B) GRAPH

Figure 6. Electromagnetic topology which includes an undesireable B-field coupling loop.

The concept of a surface node as a set of subsurface nodes is very important and can be applied in several ways. When this is done, the surface node can be thought of as a nucleus composed of many nucleons, each connected to the other through connection branches. The nucleus can be "fissioned", as in the example of pattern matching when a branch contains two cables connected at two different points. It may sometimes be desirable to have each nucleon represent a physical segment of the surface. This is analogous to the procedure used in numerical integral equation codes, where one desires to solve for the currents running on a surface. In this case, the connection branches might represent subsurface interactions via scaler and vector potentials. In any case, such subdivision concept allows for future interfacing between the expert system and numerical surface current calculations.

ELECTRICAL TOPOLOGY

The electrical (cable) topology data base is primarily responsible for storing information regarding the connection and associations between cables and cable bundles. It must record this data in a form which is compatible with the electromagnetic topology data structures so that a picture of the overall topology can be constructed. Clearly, there must be some type of interface between the two data bases so that shield penetrations and cable shield connections to cavity surfaces (either direct or through protection devices) can be identified.

In the following discussion, it will be necessary to differentiate between the shields associated with cables and the shielded enclosures referred to previously. The former will be called cable shields and the latter will be called electromagnetic shields. They are described by cable bundle surfaces, or tubes, and electromagnetic surfaces respectively. Both topologies are represented by similar graph structures. In the case of electromagnetic surfaces, the nodes represent closed surfaces; the branches indicate all possible physical paths through which electrical signals may propagate. These branches are implemented as pointers to data structures which describe the connections or penetrations which actually occupy that path, if any. Thus, the branch may point to a representation of a cable which penetrates a shield, as well as to a description of the aperture through which it penetrates. The cable bundle, in turn, is represented by a tube or tubes connected by branches which indicate connections. They are implemented as pointers to data structures which describe the nature of the connection. This may be an ideal short circuit, a physical connector, or some type of filter or protective device. The open nature of the tubular surface surrounding a bundle complicates the representation significantly. In the electromagnetic surface representation, a strict hierarchy of levels was enforced by their closed nature. In the case of cables and their tubular enclosing surfaces, any level in the node structure can be connected by branches to any other level because the ends of any inner level wires can be connected to the ends of outer level wires. Needless to say, the rules describing a "good" cable configuration are different from the rules describing a "good" electromagnetic surface topology.

At this point, it may be prudent to define a "bundle". For our purposes, we define a bundle recursively as

1. A wire, or,

2. A set of parallel wires, or,

3. A set of parallel bundles.

It is assumed that the members of a bundle are close to each other so that they can be treated as a physical unit. They need not be electrically insulated from each other, although that is the usual case. There may be times, for example, when it is topologically convenient to treat the junction between three wires as the merging of two bundles.

There is bound to be some confusion between the terms bundle and tube, compounded by the fact that the term "tube", as used here, may conflict somewhat with the definitions and analyses used by Baum (Reference 4). Again, we point out that our concepts were developed independently and some reorganization is probably in order to reconcile the models. It will be sufficient at this time to simply consider bundles as the interior of tube surfaces. A pointer to a tube is a pointer to everything enclosed by the tube.

Bundles can be shielded or unshielded. An unshielded bundle is defined by a single tubular surface. A shielded bundle, on the other hand, is defined by concentric tubular surfaces. In terms of the nodal notation defined for closed surfaces, these two tubular surfaces are connected by penetrating branches.

There are several operations which can be performed on bundles:

1. They can merge or split. In a merge, the incoming sub-bundles lose their individual group identify. A split is the inverse of a merge; it takes on meaning when a direction is assigned to the bundle (such as when a cable run is being traced through a system).

2. They can join or separate. This is similar to merge and split, except that the combined sub-bundles retain their original group identities.

3. They can connect, end-to-end, becoming essentially a single bundle with the same topological structure as the original two, with the exception of the connector characteristics. If the connector is treated as ideal, it can be ignored.

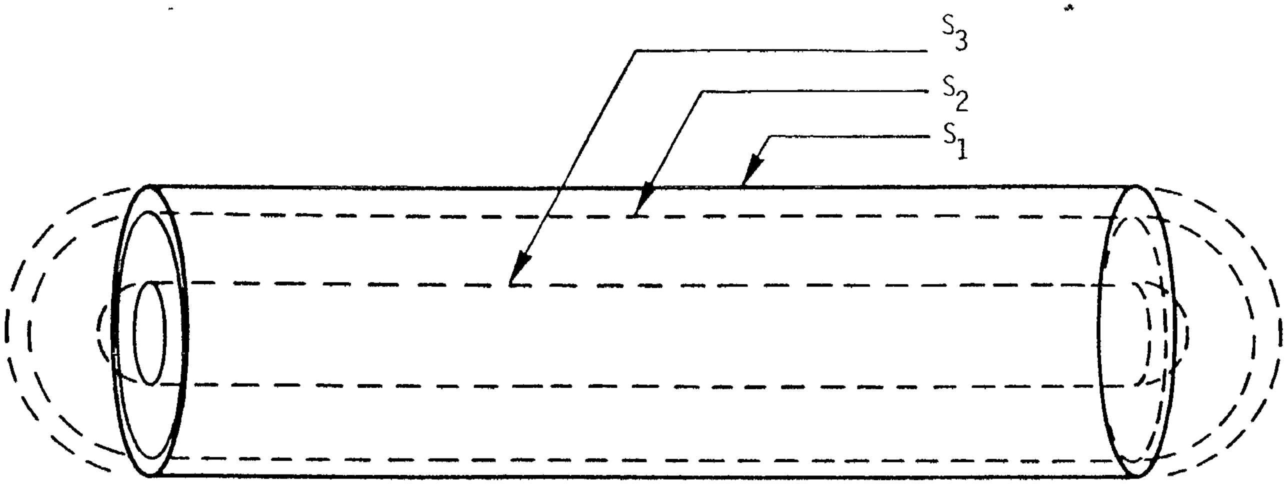
4. Bundles can penetrate shields. In the case of a shielded bundle, the possible attachment of the cable shield to the electromagnetic shield must be considered and accounted for. Such an attachment might be direct or indirect (capacitance, filter, or nonlinear protective device).

5. Bundles can attach to electromagnetic shields. A simple example is that of a ground strap. Cable shield grounding during penetration is a hybrid example.

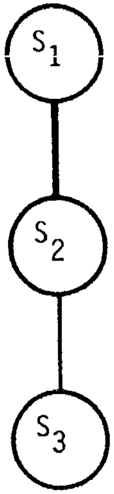
6. Bundles terminate in various ways. The most common is through an impedance or device (black box) to "ground". A bundle may also terminate with an antenna to free space.

Baum defines tubes as being connected by junctions. The operations described above constitute several types of junctions. This can be a further source of confusion. However, for the purpose of tracking the course of bundles or tubes, we feel that it is important to distinguish between merging, where the bundles take on a new identity, and joining, where the original bundles simply become subsets of a new set.

We use Figure 7 to introduce the graph representation of electrical topology. It shows an isolated coaxial cable segment and its corresponding topological graph. Of course, an isolated piece of cable has little value by itself, but serves as a useful starting point. Note that we have introduced tube end caps in order to conceptually close the tube surfaces. This is to indicate that we are not interested in any connections that might be imagined for an isolated inner conductor, such as a connection branch to infinity. Since the coax is shielded, it is defined by three concentric surfaces: S_1 defining the outer surface of the shield, S_2 defining its inner surface, and S_3 defining the outer surface of the inner conductor. Note that, unlike the electromagnetic



(A) COAX SEGMENT



(B) NODE REPRESENTATION

Figure 7. Nodal representation of an isolated segment of coaxial cable.

topology, nodes do not have to be paired through dense branches. In fact, since the lowest level of the graph represents the outer surface of a conductor rather than an inner surface, the last node cannot be paired. This characteristic distinguishes a correct electrical topology.

Even though we choose to represent the isolated cable bundle as though it were composed of closed surfaces, we must remember that each of the conductors can make its own connections independently of the others. For example, the center conductor of a coax might pass through an electromagnetic surface while the outer conductor is grounded to the surface. In order to represent a situation such as this, it will be necessary to allow branches from the center conductor node to bypass the penetration branch connecting the inner and outer surfaces of the cable shield. This is counter to the situation experienced with the electromagnetic graphs; there the branches represented the only possible paths for physically realizable connections. This difference in the use of branches is a consequence of the fact that tube surfaces are actually open at the ends.

The complexity of describing a bundle containing several sub-bundles with nodes explicitly showing all of the connection branches forces us to provide a shorthand notation: one which indicates the manner in which the data structure is actually implemented. The data structure will not contain information on all of the connection branches, unless there is some explicit reason for doing so. Rather, the sub-bundles will be treated more like the nucleons of the electromagnetic shield representation. Their individual branches only become important during operations analogous to fission. Splitting and separation are such operations. Merging and joining are analogous to fusion. These cable bundle nucleons are not to be confused with the nucleons which result from dividing a surface into segments. If a bundle surface were so divided, the resulting segment surface nucleons would each contain copies of the sub-bundle nucleons.

As an example, Figure 8 shows an unshielded bundle consisting of two sub-bundles. These can be individual wires or bundles of wires, or even bundles of bundles. We are only interested in the outermost surfaces. The true model diagram consists of three nodes, all connected by connection branches. This structure can be collapsed into a single node containing two bundle nucleons. If there are many of these, the

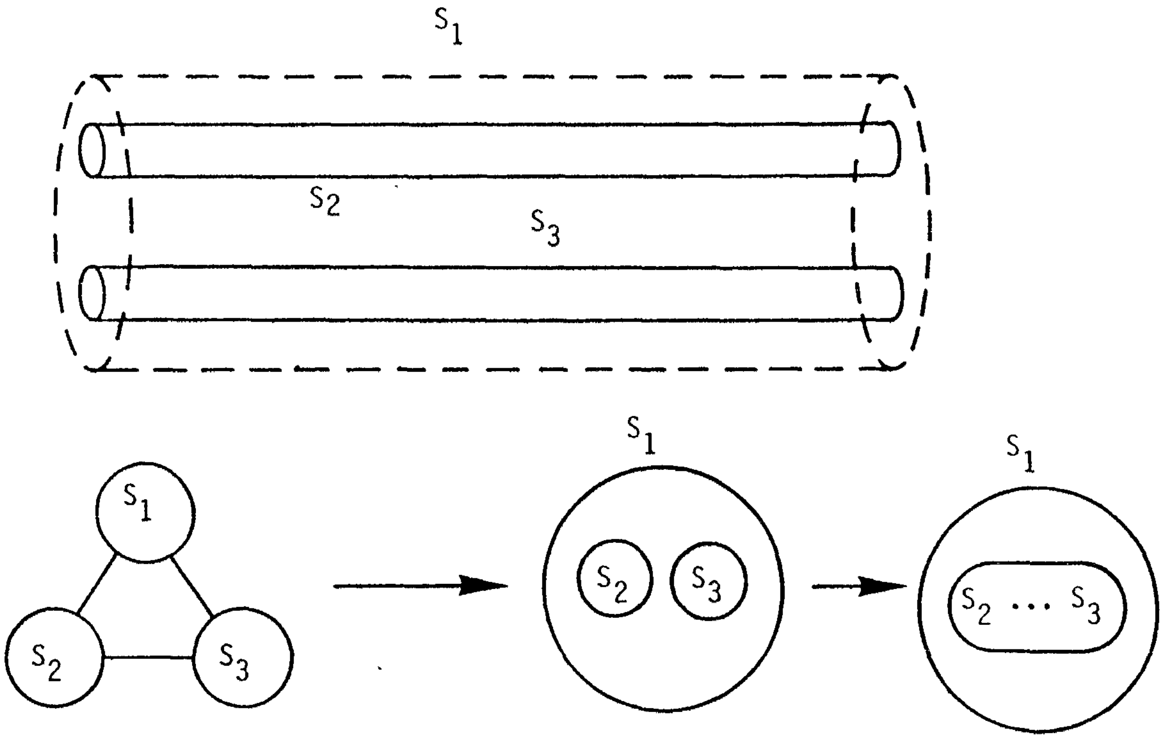


Figure 8. Unshielded bundle segment containing two sub-bundles. Nodal diagram collapsed into shorthand structures.

individual nucleons can be represented by a single structure, as shown, which contains a list of the member nucleons. In essence, we are simply representing a cross-section through the cable, so we will refer to this shorthand notation as a cross-section diagram. Shielded bundles can be collapsed in a similar manner, with the penetration branch between the outer and inner surfaces of the shield being present. Figure 9 shows an example of the development of the cross-section diagram for a shielded bundle with two sub-bundles.

Let us proceed to see how the graph works in practice by looking at some of the operations mentioned above. In Figure 10, we use cross-section diagrams to illustrate the difference between joining and merging two unshielded cable bundles. In the first case, the original surfaces, S_1 and S_2, are retained as nucleons within the surface of the combined bundle, S_3. When the bundles are merged, S_1 and S_2 lose their individual identities within S_3. The joining concept might be used when two sets of harnessed cables are themselves harnessed into a single bundle. The merging concept better describes the situation in which the original two harnesses are stopped at the point of the merge and a single harness is used to wrap all of the individual cables. Of course, there can be a mixture of joining and merging. For example, the harness about S_1 might be broken while that of S_2 is retained. In the case where one bundle is shielded, that shield would be retained in the joined bundle (if good shielding practices are observed). These merging and joining operations are easily implemented as list structures in LISP.

Repeating the previous discussion, bundle nodes are connected by branches which represent the connections between cables or cable segments. In many cases, these branches simply indicate a perfect short circuit. In other cases, they represent real connectors and/or protective devices. A branch may also represent a load. In a complete topology, all cables should terminate with either a branch to an electromagnetic surface or a branch to a cable shield surface in order to provide a current return path. Of course, an electromagnetic surface can be the outermost surface at infinity, representing an earth ground or an antenna. During the initial phases of data collection, however, it is quite possible to have a bundle surface node with a dangling branch, indicating a connection to an unknown load, or to have the branch connected to a surface node representing a "black box" termination with unknown ground return path. The expert system will be expected to tag such topologies and ask for more

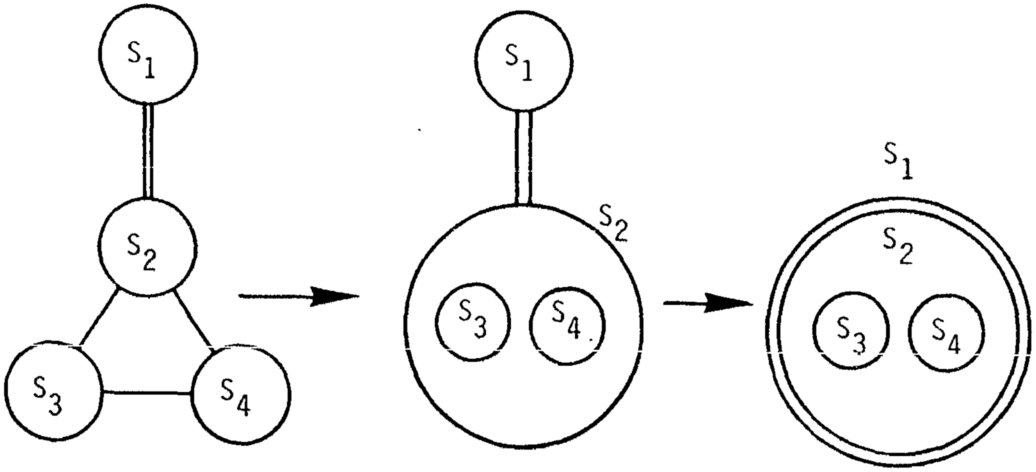
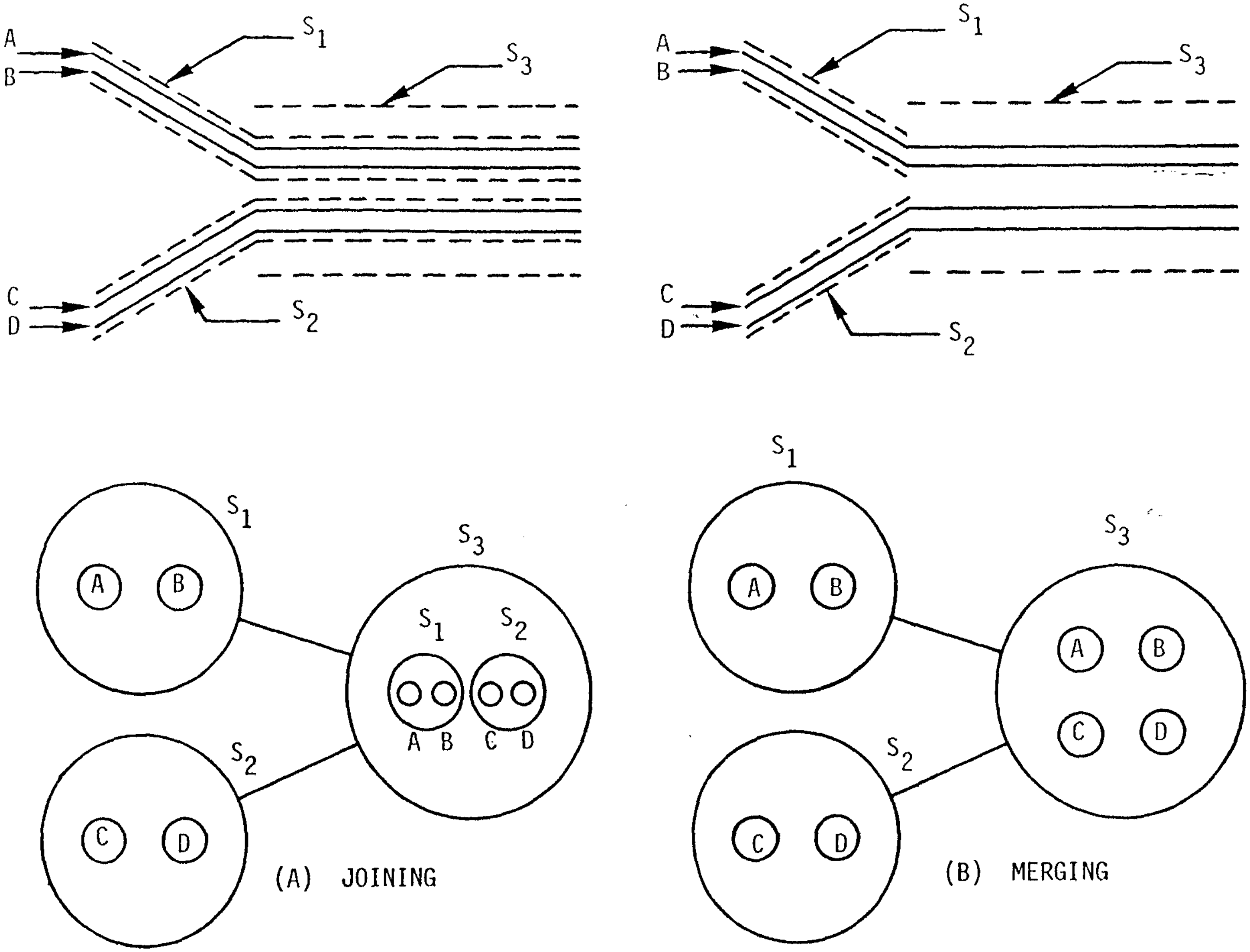


Figure 9. Shielded bundle segment containing two sub-bundles: nodal diagram collapsed into two levels of cross-section diagram.



(A) JOINING

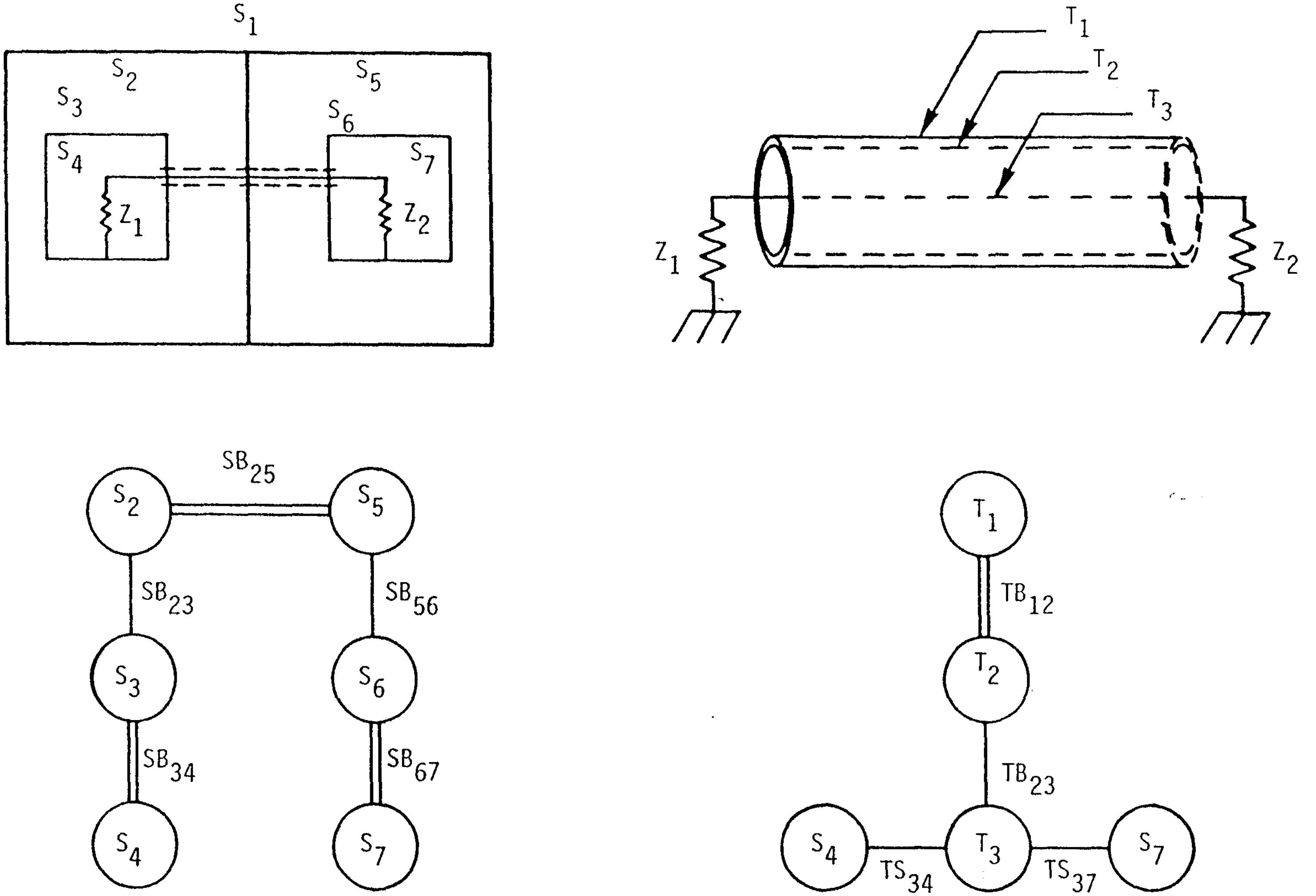
(B) MERGING

Figure 10. Cross-section diagrams illustrating the difference between joining and merging two unshielded bundles.

information, suggest probable existing return paths (based on available information), or suggest the proper placement of a return path if none exists. In the latter case, the inference machine would analyze the existing geometry in order to provide a connection which minimizes the possibility of a magnetic field coupling loop and cross-talk between cables.

Figure 11 shows a simple, but realistic, example which may aid in understanding some of the relevant concepts. Here we have a system which contains two main compartments. Each compartment contains a shielded enclosure. Each enclosure, which may represent an equipment rack, contains an electronic device. The devices are represented by impedances. The two devices are connected by the center conductor of a shielded cable and are "grounded" to the inner surface of the enclosure. For the moment, we ignore the grounding of the cable shield; there are several possibilities and we will want to explore some of the variations. Also shown in the figure are the graph representation of the electromagnetic surfaces and the representation of the cable. Note that the connections of the center conductor are displayed as branches to nodes representing the inner surfaces of the enclosures. In order to reduce confusion, we denote the electromagnetic surfaces by S_n, where n is an integer, and the tube surfaces by T_n. We also number the branches. The surface electromagnetic branches are of the form SB_{mn}, where m and n are integers. The cable branches are of two forms: TB_{mn} and TS_{mn}. The TB branches connect cable nodes; the TS branches connect cable nodes with electromagnetic surface nodes.

We have chosen to represent the cable as a single entity. This is not the only acceptable way. For example, we could have considered the cable to be composed of two individual cables connected at the partition between the two system compartments. It would also have been possible to consider the wire segments inside the enclosures as separate entities. The method chosen here yields the simplest graph. The data structures we use will record all cable connections as part of the cable graph. Connections to electromagnetic surfaces might be recorded as part of that nodal structure, or in a combination of both. The data partition used here is probably the cleanest, but it will take more experience to know for sure.



(A) VOLUME SHIELD REPRESENTATION

(B) CABLE BUNDLE REPRESENTATION (NOT SHOWING SHIELD CONNECTIONS)

Figure 11. Example of two shielded enclosures connected by a shielded cable.

Since the cable is represented by a single set of nodes, and since it passes through all of the electromagnetic surfaces, each of the electromagnetic shield branch structures (SB_{mn}) will contain pointers to it. More exactly, the pointers (or equivalent list structures) point to cable node T_1, which represents the outermost surface. From this node, we can trace the remainder of the node structure. In addition to the pointer to the cable, the penetration branches, e.g., SB_{25}, might contain pointers to information concerning the aperture through which the cable passes as well as any other apertures (doors, hatches, etc.). In addition, the connection branches might contain pointers to grounding straps or rods which connect the otherwise isolated surfaces. Such information is not only important in determining how other fields might radiate into the volume, but may also be needed to determine the possibilities for cable current return paths. While the representation we have chosen uses the cable node structure to record connections between electromagnetic and cable shields, it would be possible to use the electromagnetic penetration branch structures to accomplish the same goal.

The cable node branches TS_{34} and TS_{37} contain pointers to records which describe the load impedances Z1 and Z2. These terminate on surface nodes S_4 and S_7. As the cable node structure stands, without reference to any cable shield connections, there is no explicit cable current return path. Given no additional information, the expert system could explore the branches leading from S_4 to S_7, looking for apertures which would indicate a return path over the electromagnetic surfaces and through any other cables or ground straps. In this case, it would find that there are no connections indicated between S_2 and S_3 or between S_5 and S_6 except the cable itself. Since no attachment of the cable to these surfaces is shown, a lack of return path would be indicated. The expert system then prompts:

> "Hey, Stupid, you have a cable (T_3) connecting S_4 (the ladies washroom) with S_7 (the Command Control Center) with no return path indicated. Now its none of my business why you set up such a circuit, but it will be necessary to complete it. I see that you have a floating cable shield. Perhaps you should connect it to something. If I were you (and thank God I'm not) I would ground and seal at the penetrations SB_{34}, SB_{25} and SB_{67}. I would also be careful about adding any more ground connections at SB_{23} and SB_{56}; this could establish magnetic field coupling loops."

After brushing off the insult as the price to pay for modern technology, the technician informs the expert system that the appropriate ground connections have, in fact, already been made. The system generates the new graph shown in Figure 12. Since the cable terminates at the two enclosures, it assumes that the outer surface of the cable shield is connected to the outer enclosure surfaces (S_3 and S_6) while the inner surface of the cable shield is connected to the inner surfaces of the enclosures (S_4 and S_7). This results in the return path indicated by the branches TS_{24} and TS_{27} (where the data structures show a short circuit). Because the cable penetration apertures were sealed in the process, any reference to them in the data structures associated with branches SB_{34}, SB_{25} and SB_{67} would be removed.

The final example, shown in Figure 13, concerns a power line entering a shielded building. It is desirable to protect the systems inside the building from current surges induced on the power line by EMP. Two possible positions for the placement of a protective device are shown as G_1 and G_2. Placement of the device inside the building (G_2) is not good practice since the currents can penetrate the aperture and radiate fields while the ESA responds. The best position is outside the shield. The purpose of this example is to show how easily the expert system can distinguish between the two cases and provide a warning if necessary. The cable node diagrams corresponding to the two situations are shown in Figures 13(a) and 13(b). In Figure 13(b), the power line (T_1) is shown with branches to S_0 (surface at infinity) and S_2 (inside building) only. The protector is indicated by the pointer associated with branch TS_{12}, thus indicating that it connects the cable with the inside of the shield. The more desirable geometry is depicted by the graph of Figure 13(d). Here there is an additional branch, TS_{11}, which contains data showing the protector connected between the power line and the external surface of the building shield.

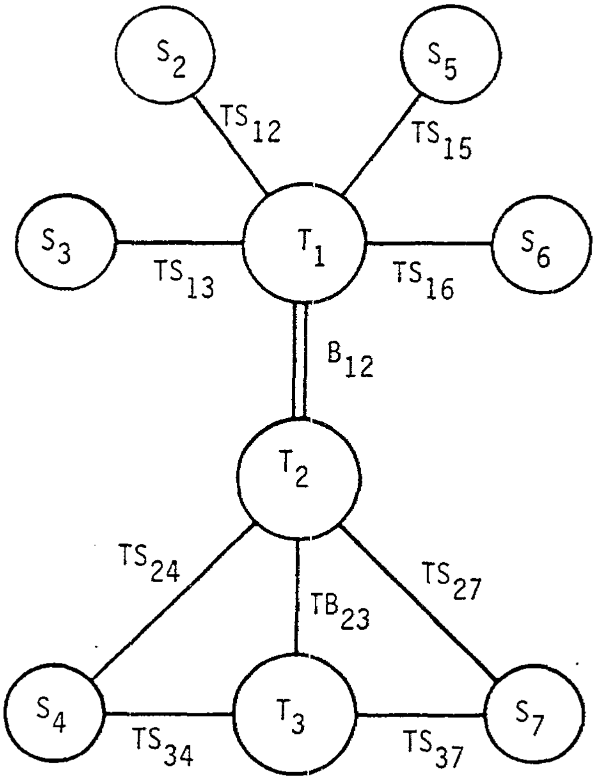
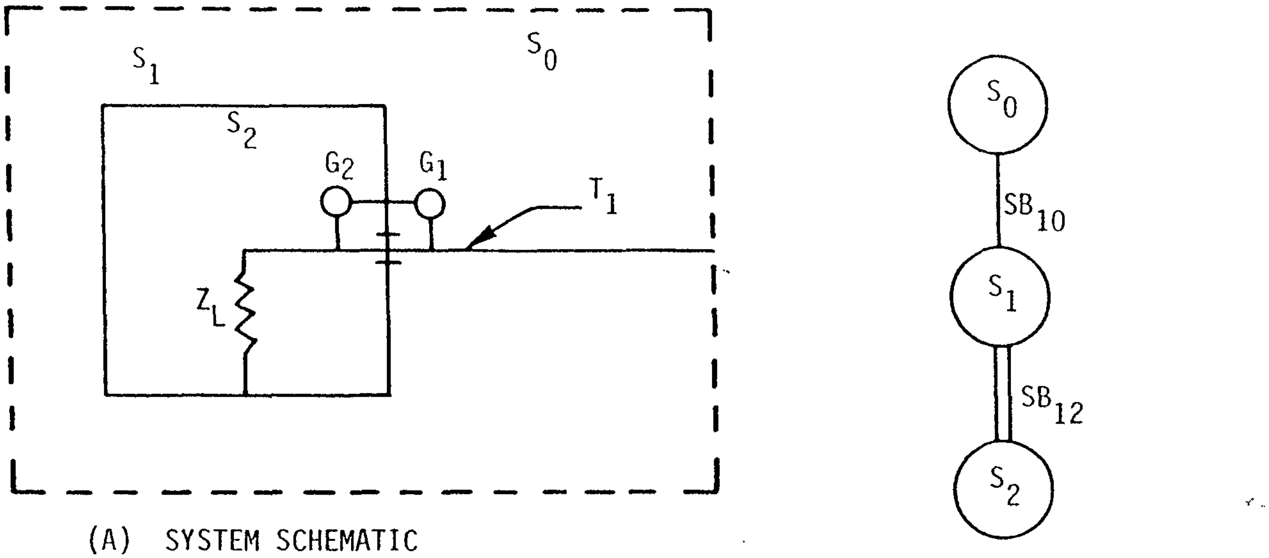
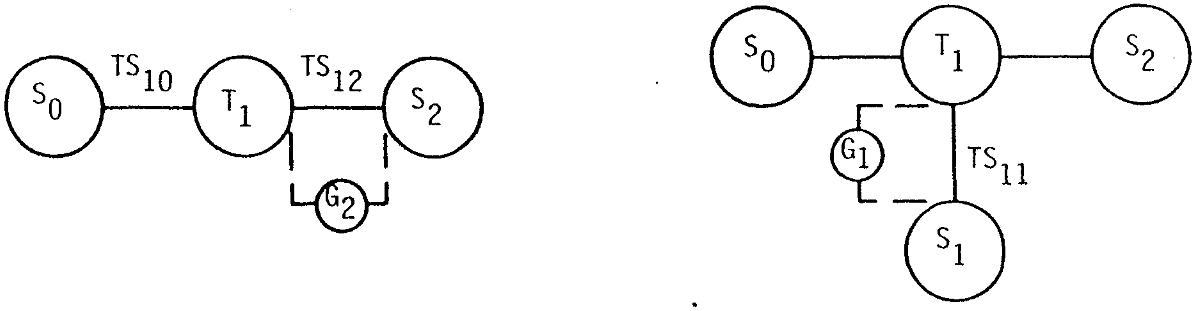


Figure 12. Cable node diagram with shield connections recorded.



(A) SYSTEM SCHEMATIC

(B) SURFACE GRAPH



(C) CABLE GRAPH (PROTECTOR INSIDE)

(D) CABLE GRAPH (PROTECTOR OUTSIDE)

Figure 13. Example showing protection of a power line entering a shielded building.

CONCLUSION

The general architecture of the proposed HARD TOP EMP hardening expert system has been described. We have noted that past experience indicates that the usual rule based system is not practical. Instead, one must use a knowledge base which reflects the structure of the topology data base (at least for that part of the system which evaluates the physical systems electromagnetic topology). Since the EM topology is expressed in the form of a graph, i.e., nodes and branches, the knowledge base should be in this form. We have described a candidate topology graph structure and shown how it might be used by an expert system which can, at a minimum, analyze the complicated electromagnetic topology of an aircraft or ship. Future expert systems should also be able to analyze the penetration of test signals and extrapolate to threat environments. Presently, we are constrained not by computer hardware or software capability, but by knowledge base limitations. This will continue to be the case until a prototype system has been built for which we can target a specific knowledge base.

REFERENCES

1. Baum, C.E., How To Think About EMP Interaction, Proceedings of the 1974 FULMEN Meeting, AFWL, April 1974.

2. Tesche, F.M., M.A. Morgan, B. Fishbine, E.R. Parkinson, Internal Interaction Analysis: Topological Concepts And Needed Model Improvements, IN 248, AFWL, July 1975.

3. Vance, E.F., Shielding And Grounding Topology For Interference Control, IN 306, AFWL, April 1977.

4. Baum, C.E., T.K. Liu and F.M. Tesche, On The Analysis Of General Multiconductor Transmission Line Networks, IN 350, AFWL, November 1978.

5. Vance, E.F., On Electromagnetic Interference Control, IN 380, AFWL, October 1979.

6. Baum, C.E., Electromagnetic Topology, A Formal Approach To The Analysis And Design Of Complex Electronic Systems, IN 400, AFWL, September 1980.

7. Lee, K.S.H., Editor, EMP Interaction: Principles, Techniques, And Reference Data, EMP Interaction 2-1, AFWL-TR-80-402, AFWL, December 1980, Chapter 3.1.

8. Baum, C.E., Sublayer Sets And Relative Shielding Order In Electromagnetic Topology, IN 416, AFWL, April 1982.

9. Baum, C.E., Topological Considerations For Low-Frequency Shielding And Grounding, IN 417, AFWL, June 1982.

10. Yang, F.C. and C.E. Baum, Use Of Matrix Norms Of Interaction Supermatrix Blocks For Specifying Electromagnetic Performance Of Subshields, IN 427, AFWL, April 1983.

11. Baum, C.E., Bounds On Norms Of Scattering Matrices, IN 432, AFWL, 1983.

12. Noss, R.S., Alternative Labelling Schemes In Electromagnetic Topology, IN 433, May 1983.

13. Noss, R.S., Planarity Criteria In Electromagnetic Topology, IN 434, AFWL, July 1983.