Segflow: A New Object-Oriented Load Flow Which Uses Trace Methods and Affiliation Objects

by

Larry V. Trussell

Dissertation Submitted to the Faculty of the Virginia
Polytechnic Institute and State University in Partial
Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Approved:

Robert Broadwater
Dr. R. Broadwater, Co-Chairperson

A G Phadke
Dr. A. Phadke, Co-Chairperson

Jaime De la Ree Lopez
Dr. J. De La Ree Lopez

John S. Bay
Dr. J. Bay

Terry L. Herdman
Dr. T. Herdman

Lamine Mili
Dr. L. Mili

Ali H. Nayfeh
Dr. A. Nayfeh

August, 1994

Blacksburg, Virginia

Abstract

Segflow: A New Object-Oriented Load Flow Which Uses Trace Methods and Affiliation Objects

by Larry V. Trussell

Committee Chairmen: Dr. R.Broadwater & Dr. A.Phadke - Electrical Engineering

This dissertation presents a new alternative type of object-oriented load-flow called Segflow. Segment objects are used to support the modeling of individual types of power system equipment. Current and voltage trace techniques are used by Segflow to bind Segment models in support of Kirchoff's laws. Affiliation objects and the Target Voltage method are used to bring the solution of the Segflow model to the solution of the power system network. An example for modeling a simple transformer is given along with the solution results for a collection of common power system models.

Analysis of the Target Voltage method and simulation results show that Segflow is capable of consistently solving the nonlinear load-flow problem. The object-oriented layout of Segflow provides very distinct modeling advantages. Power system modeling is from the perspective of each equipment's model. Rather than force all equipment models into large sets of equations to be solved simultaneously by an equation solver, Segflow allows component models to remain intact and independent as Segment objects. Each Segment has a one-to-one correspondence with some piece of equipment in the respective power system which is maintained in the system solution. The behavior of a Segment can be evaluated throughout the load-flow process.

An important feature of Segflow is the object-oriented design enabling new equipment models to be added into the Segflow environment as autonomous objects. New models are created from the Segment class and their inputs and outputs are always relative to the same Segment attributes. Because all Segment objects are derived from a Segment modeling class and all Segments are treated the same by Segflow, many types and varieties of power system equipment models can be easily created.

A Segflow application is a collection of interacting objects whose interactions lead to a load-flow solution. Segflow is an interesting alternative to classical approaches for solving the load-flow problem with large sets of simultaneous nonlinear equations. It also adds new aspects to the application of object-oriented design in load-flow analysis.

Acknowledgments

Thanks Prof. Kersting, Dr. Ranade, and Dr. Smolleck for the outstanding power system engineering education at New Mexico State.

Thanks Pennzoil, EUMP, Edison Electric, Southwestern Public Service, Baltimore Gas & Electric, and PES for believing in me.

Thanks Dr. Broadwater, Dr. Phadke, and Dr. Mili for putting up with and allowing me to learn from my own mistakes.

Thanks Mom for cheering me on all of these years.

And Thanks Tamela. We made it and words can't describe what it feels like to have done all of this together. I love you and look forward to the rest of our life.

Table of Contents

Figure List

Chapter 1

Segflow: A New Type of Load-flow

1. Segflow: A New Type of Load-Flow

Segflow is a new type of load-flow based on objects. An "object" can be simply thought of as a collection of functions called methods and data called attributes[1]. These methods and attributes represent some real or abstract item or entity within the respective problem domain[2]. In our case, the problem domain is a power system and an object corresponds to a software representation of power system equipment such as transformers, capacitors, and switches.

An object's behavior is its action and reaction in terms of state changes and information passing[2]. Methods and attributes within Segflow objects describe the behavior of the object in a manner consistent with the operation of the corresponding power system equipment. Methods within a transformer object, for instance, use attributes regarding impedances and turns ratios to relate voltages and currents at either end of the device's model. Each piece of power system equipment modeled within Segflow has its own object with tailored methods and attributes.

In Segflow, objects are arranged and connected in a manner similar to the actual power system equipment. That is, they are given specific avenues of communication that generally correspond to the connection points of physical equipment. These objects are all treated exactly the same by Segflow and are signaled to send and request information about voltages, currents, and other items. This interaction is referred to as information passing between objects and is a definitive attribute of an object-oriented design[3]. In Segflow, this interaction leads to a load-flow solution.

The load-flow solution in the Segflow design is that state in which all objects signal satisfaction with their internal state. This corresponds to the validation of Kirchoff's laws within each and all objects. A Segflow structure and process has been developed to impel objects to pass messages and respond in a manner to reach the load-flow solution. Selected aspects of these items are, in fact, the topic of this paper.

The remaining portion of this chapter discusses the benefits and advantages of Segflow as well as the contribution of others in the area of object-oriented load-flow. The next chapter presents the modeling base class called the Segment. Chapter 3 discusses the radialized model and trace techniques used for solving radial collections of Segments. Chapter 4 presents Affiliation objects and the Target Voltage method. Chapter 5 analyzes the Target Voltage method and Chapter 6 discusses the overall Segflow process.

1.1 The Object-Oriented Paradigm

Programming languages are paradigms that provide techniques to apply to the design and implementation of a problem. These languages are generated by people's thinking about understanding and solving a problem [4]. Segflow is based on languages directed towards the concept that things and ideas can be expressed as autonomous entities called objects.

Classical load-flow applications are based on procedural languages. These programs consist of series of operations on data structures to implement some algorithm [4]. For involved problems, the procedures are usually long and complicated. They are thus difficult and expensive to maintain.

If the only tools available are procedural programming languages then we will naturally solve every problem with sequential algorithms. And even though object-oriented techniques have been around since the sixties, most of the work has been focused on procedural programming until the late 1980's [3]. Perhaps power engineers have not paid attention to the capabilities of object-oriented designs due to the strict concentration on load-flow speed. Faster computers can now compensate for the loss of speed with object-oriented designs. We can now begin to search for object-oriented load-flow methods and may find that they provide outstanding benefits.

1.1.1 Objects

Schryver stated, "Any [system] is composed of objects that have their own internal calculus. The objects are constrained by a lawful set of interactions, or a topology that connects the objects." [5]. And Alvarado wrote, "The most useful computer programs are those which solve problems in a manner closer to human thinking." [6]. Segflow follows these modes of thinking. Object-oriented design for analysis is an alternative perspective on power systems.

In this dissertation, object-oriented design is presented as a method of analysis. To the extent of this research, object-oriented design does not provide solutions that current techniques of power system analysis does not already provide. At the most fundamental perception, the object-oriented design provides a method of organizing data. It allows us to classify and to group so that power system models are much simpler to understand and construct. As a further consequence, analysis is also easier to perform on very accurately constructed power system models.

This dissertation lays out an object-oriented design of the power system model based on my research and my understanding of power systems. This model is not unique in my recognition of similarities and abstractions. The formulations and proofs that are presented herein are a result of my discernment of power transmission systems represented in my object-oriented model. The behavior that my model exhibits to converge is rooted in its foundation. Someone else may be able to derive a behavior that will allow a completely different modeling foundation to converge. This dissertation does, however, demonstrate that a completely object-oriented power system model can converge to the load-flow solution. This is the first such demonstration and is thus a fundamental contribution to object-oriented load-flow research.

1.1.2 Classification

Classification is considered to be the most difficult part of object-oriented analysis and design [2]. It is difficult to break down complicated systems into groups of related components or classes. It is believed, however, that a system which is clustered into interacting objects is more pleasing to human cognition, more meaningful to human understanding, and more closely related to the actual system. Michalski and Stepp state, "An omnipresent problem in science is to construct meaningful classifications of observed objects or situations. Such classifications facilitate human comprehension of the observations and the subsequent development of a scientific theory [7]." Perhaps the classification that a designer has developed at the point of design is most responsible for the product of his efforts. It could be the case that our most fundamental ideas about conducting load-flow analysis or other power system simulations are not unique but only as static as our way of thinking about the universe. Newton, Galileo, Einstein are the famous examples of the simple solutions that result from a shift in perspective. They stopped, stood back, and reclassified their model of the world.

Traditionally, the object-oriented model has been classified as a system of simultaneous linear and nonlinear equations[8]. As a result of this classification, sophisticated equation solvers, data base interface schemes, and sparsity techniques have been developed. Work with this power system classification has continued for 40 years and resulted in very productive and accurate methods for solving the power flow problem. Segflow has resulted from standing back and taking another look at the power system model. It is a reclassification of the power system model as something other than simultaneous equations.

1.2 What Does Segflow Have to Offer

Segflow enables the users to focus their efforts towards power system component modeling and away from the techniques for solving large sets of simultaneous equations. It allows users to easily create models of specific types or classes of power system equipment. These models are built separately from the load-flow application and are dynamically drawn in for use when the particular model is needed to represent a specific piece of equipment.

1.2.1 Segment Independence

Segments and Segflow have no distinct relationship. That is, all component models look the same from the outside. They all exchange the same information about flows, potentials, and impedances. The load-flow algorithm that will be discussed in Chapter 6 never differentiates between a transformer, capacitor, or line section model. All components are treated the same by Segflow. New models or model modifications have no bearing on the load-flow program.

Segflow encompasses a careful layout of basic building blocks supporting the flows and topology of general power systems. New models are fabricated from these basic building blocks enabling users to model the behavior and attributes of particular types of power system equipment. The purpose of Segflow is to persuade these new models to interact and converge to a load-flow solution. The tactics used by Segflow to drive the collection of component models to the load-flow solution are, in fact, the topic of this dissertation.

1.2.2 Black Boxes

Complex systems such as power systems can be shown to be hierarchies of interrelated subsystems. These subsystems can be decomposed until, at last, the most elementary components remain [7]. In Segflow, the elementary component is called the Segment. Segments can be thought of as "Black Boxes." Segments hold connection and communication information about generic components. Segflow is created to manipulate these Segments to interact and converge to a steady state representing a balance of potential and flow (Kirchhoff's current and voltage laws). Segments are created to support the fundamental property of Segflow -- New models never impact the load-flow algorithm.

Any new component is constructed from the Segment model and therefore has all of the characteristics of a Segment. Segflow treats any component, whether a transformer or generator or line, like a Segment. The component modeler has a responsibility to override the Segment behavior and create a behavior in his component model that accurately represents the associated equipment and is conducive to the solution of a very large collection of Segment objects.

Component models can be quickly constructed and linked with the object-oriented load-flow. They can incorporate curves, anomalies, and special characteristics that a power engineer might wish to have reflected in the load-flow results. The object-oriented modeling platform allows him to model a piece of equipment in any manner he chooses. The load-flow can have its component models defined and redefined to fit the criteria of each specific application. Many different types of models can exist for a general types of equipment such as a transformers, line sections, or generators.

1.2.3 Message Passing

Objects are characterized as dormant until they are activated by other objects. This activation is invoked with a message. The message is sent to activate a very specific response and is therefore considered to have an associated method. That is, every message is associated with a method. A *sender* object sends a message to a *receiver's* particular method. The receiver then sends a message back to the sender describing the methods results. There are four things that a message sender needs to know:

- ✓ *How to contact the receiver*
- ✓ *The name of the correct receiver's method*
- ✓ *Arguments and parameter information to send*
- ✓ *Generally what type of information that will be received back*

The Segflow foundation defines the above aspects. Modelers in Segflow follow the resulting rules. Message passing within the Segflow envirionment allows Segment autonomy and thus more modeling freedom.

1.3 The Segflow Perspective

In both the academic and industrial realms, load-flow packages are used to solve intricate models of power systems containing various representations of power system equipment. Prefabricated load-flow programs usually offer diverse models for specific components. The load-flows use data pertaining to the models to construct a collection of equations describing the power system. These equations are all solved simultaneously using some method such as Newton-Raphson or Gauss-Seidel. The figure below delineates the representation of a power system's model as a collection of nonlinear, linear, and other equations such as constraints. It shows how these equations are all solved together with a load-flow algorithm to produce the solution of the power system.

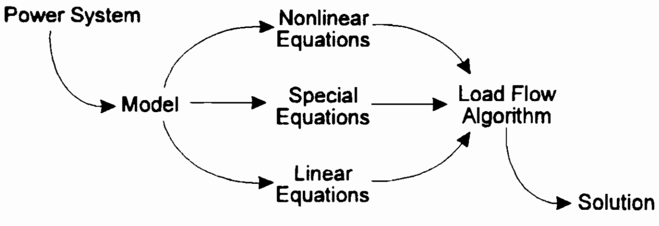


Figure 1.1 - The Classical Load-flow Approach

Since the component models are merged into large sets of simultaneous equations, they are inflexible. Engineers must make their desired model conform to the models available in the load-flow package. Manufacturer software modifications are complex, time consuming and expensive. Some engineers or engineering students may contemplate creating a specialized load-flow program from scratch to achieve certain specifications. Often, however, engineering time, expertise, or motivation, will not warrant the construction of a tailor made load-flow. The studies requiring load-flow results may be compromised, with the abandonment of more ambitious options, by inadequate models.

Segflow is different. Component models in Segflow are not integrated into a global format with all other component models. In Segflow, all component models remain distinct through the stages of model construction, solution, and result output. The figure below presents a visualization of Segflow. Component models communicate through Segflow and with other models.

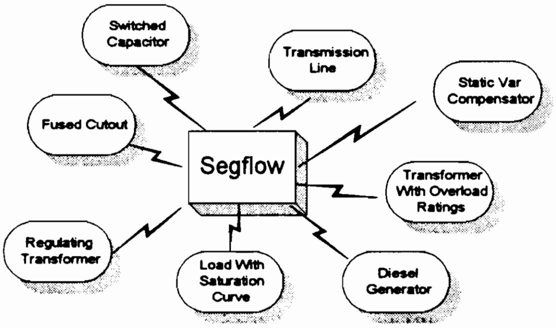


Figure 1.2 - A Conceptualization of the Object-Oriented Load-flow

The components shown above are examples of the models that Segflow users can create. There are a vast range of possibilities for component models provided with this type of approach. A component model could be a simple resistor -- or it could be the model for an entire power system. A generator model could be made having precise allowances for excitor, governor, and prime mover behaviors or it could be modeled as a voltage behind a transient reactance. The modeling capabilities provided by the object-oriented load-flow enhance the creativity of the user.

1.4 The Investigation

This dissertation presents the results of an investigation to determine a method for finding the load-flow solution with the completely object-oriented design described above. This is a new way to solve power system models.

There are a number of points of investigation presented as contribution within this dissertation. First, the object-oriented load-flow must have a method of converging to the load-flow solution. The load-flow is not allowed to distinguish between component types and must therefore treat all component models exactly the same. Due to the design of the object-oriented load-flow, the objects, through their interaction, should eventually reach a state corresponding to the load-flow solution of the meshed, nonlinear power system.

Second, a comfortable modeling environment should be supported. Objects, which are collections of data and functions that describe some physical device, are matched with actual power system equipment. These objects are connected and allowed to interact. To provide a comfortable modeling environment, this interaction consists of the passing of information about current flows and voltages. Segflow behaves as the collective behavior of all Segment objects when they are interacting. If the object interaction settles to some state then the load-flow has converged. Since the user constructs the equipment model within the Segment objects, the modeler has the ultimate flexibility and responsibility towards the convergence of the object-oriented load-flow. Future investigations into the uniqueness of solutions and multiple solutions is discussed in Chapter 7.

This dissertation presents an object-oriented load-flow that supports the success of new models in individual behavior accuracy and the overall system load-flow solution. This interaction described by the interface of the fundamental component model sets the information that will be passed between instances of component models. Their are a number of interface criteria:

- The interface must support the behavior of all power system equipment ranging from lines to generators to sources or infinite machines.
- Adequate information must be passed through component interfaces to allow the load-flow solution to be found.

Modern load-flow algorithms are very fast. They are essential to modern power system analysis. They are, however, built to emulate very specific types of power system models. They are inflexible to users and difficult to change for programmers. The contribution described in this dissertation should provide a power system analysis tool designed for modeling flexibility and not speed.

The desired outcome of this research was to develop an object-oriented load-flow design that is capable of accurately representing the details of modern power systems with relatively little programming effort. The load-flow is to help support the sophisticated modeling demands of the user. The component modeling was to be the fundamental aspect of the object-oriented load-flow so that new components are easy to introduce and integrate into the load-flow application.

This outcome of my research is a modeling philosophy which exhibits the following features:

- Segflow allows engineers to create new component models by simply describing the behavior of components in an actual power system[9].

- The object-oriented design provides a strong cohesion between real world equipment and their respective software objects [10].
- The model converges to a load-flow solution by object interaction. No algorithms incorporating global data structures or algorithms are constructed.
- Segflow is relatively non-heuristic and supported with a mathematical and engineering basis.
- All data and functionality is contained within component models and other design objects.

Segflow is not designed to calculate values or solve problems that cannot already be solved with modern load-flow techniques. I doubt that there is anything that Segflow can model that cannot be modeled with a persistent engineer writing a Gauss-Seidel or Newton-Raphson algorithm. Segflow is, however, a move from simulation based on algorithms to simulation based on component modeling. The Target Voltage method created through this research work allows the creation of a power flow algorithm that is completely transparent to component models. Users of this method can model to high detail and precision power system equipment without having to integrate their models into a classical load-flow platform.

1.5 What Have Others Done

The solution of global, nonlinear, simultaneous equations representing the power system has been used almost exclusively for over thirty years. The emergence of object-oriented programming techniques has not greatly moved the power engineering community towards object-oriented analysis. In fact, a lot of object-oriented concepts were developed long before object-oriented programming came around. H.H. Happ [11] was very interested in independently solving subsections of power systems and then using a transformation to combine the solution of the groups into a global solution for the entire system. Carre' [12] considered another method in which a power system model was broken into trees. He found that particular tree partitions could mimic an optimal elimination scheme used in block iterative Gauss-Seidel type load-flows.

More recently, attention has been given to the modeling capabilities of a truly object-oriented power system modeling design and to the effectiveness of using trace methods for solving radial systems. These two ways of thinking have diverted into two generally separate approaches which are presented in the following section.

1.5.1 Efforts in Object-Oriented Load-flow

Two tendencies toward object-oriented power system model design have emerged in the literature. One group has established a method for solving lightly meshed systems with trace methods used for solving distribution systems. They have not embraced an object-oriented approach and violate the object-oriented way of thinking with a global breakpoint impedance or sensitivity matrix. Another group uses object-oriented programming as a buffer between data and traditional Jacobian matrix based approaches to solving the set of load-flow equations.

1.5.1.1 Hakavik & Holen[10]

Hakavik and Holen took a similar approach in object oriented programming. They considered object-oriented programming a tool for data abstraction and encapsulation. An algorithmic approach was used on network topology to drive components to form the bus admittance matrix with respect to a Tinney sparsity technique. A solver was used to determine the solution to the application's linear and non-linear differential equations. They showed that object-oriented programming can be powerful for custom techniques for handling sparsity. They also showed that execution speed is not hampered with object oriented data handling. These authors (as was Neyer indirectly) are using object-oriented programming to handle data into and out of Ybus and the system Jacobian. The object oriented design proposed in this dissertation circumvents this need by avoiding any global equations.

1.5.1.2 Neyer, Wu, & Imhof [9]

This group showed their design philosophy for an object oriented load-flow application. They pointed out a number of problems with the present methods of creating load-flow applications:

- The effort for developing and particularly for debugging the [power flow] software seems to increase exponentially with the size of the project.
- Completed code is not flexible.
- Modifying and adapting computer code to changing requirements is costly.
- Discrepancies in data and algorithms lead to hard to detect programming errors.
- A strong coupling between data structures and procedures means that minor changes propagate through the entire load-flow program.

Their approach is object oriented in data handling. A modified polar Newton-Raphson algorithm and specialized sparsity scheme harness the power of data abstraction and encapsulation. Because a global load-flow and data handling algorithm are required to handle the classic non-linear power flow equations the approach is not object-oriented as is the contribution of this dissertation. Node models are constrained to PV, PQ, or Slack.

They did find that object-oriented programming leads to flexibility and modularity. Unfortunately, the characteristic loss of speed associated with most object oriented programs was a problem. Their contributing work provides a powerful new approach to writing a load-flow application.

The solution of the load-flow problems using optimal ordering and an upper triangularization technique is actually embedded into the way that components interact (pass information.). An object in their power flow constitutes a node and associated branches. A local portion of the Jacobian matrix is formed for each node and a "localPowerFlow" is initiated so that the system is solved as if it were factorized triangularly.

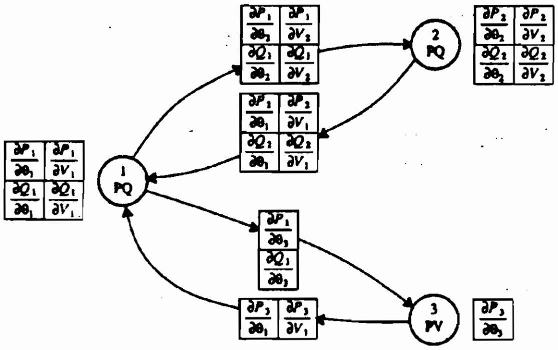


Figure 1.3 - Node Objects Pass Sections of the Jacobian Matrix

Very interesting results were found from their work. First, the speed of the load-flow algorithm is slowed by 30-50% due to message passing. Message passing, in fact, consumed 40.6% of the total run time for the IEEE 30-Bus test system. They also pointed out that:

- Programming was more natural since the computer model corresponds to the real world. Programs are much more readable since special purpose data structures are hidden from the user of a class.
- Inheritance made it very easy to incrementally build up, test and extend the existing program.
- Object-oriented programming emphasized programming on a higher conceptual level.
- Message passing in object-oriented programming supports descriptive programming.

The paper introduced the basic elements of object-oriented programming that can be used by the power engineering community with the example of a load-flow algorithm.

Like the work of Neyer, Wu, & Imhof, a heavy reliance is place on object-oriented modeling methods. Unlike their work, the power system is solved with an object-oriented method that is object-oriented by supporting component modeling flexibility. Component models are not constrained into a global set of equations to be solved by Newton-Raphson.

Chapter 2

The Segment and System

2. The Segment and System

The Segment class is used for equipment modeling. Objects created from this class specifically model unique power system components. The System class provides containment[13] for the collection of Segments representing a power system model. Each Segment in the load flow platform is responsible for establishing and maintaining a behavior that simulates a respective power system equipment. Segments are self serving; acting as they will in response to Segflow messages.

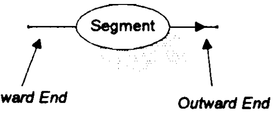
2.1 The Segment Class

The Segment is the most generalized class, the base class[13], for equipment modeling within Segflow. The Segment is an abstract[13] class from which all classes corresponding to specific equipment models are derived. Classes derived from the Segment class and their instances[13] are called Segment objects.

The Segment is the intrinsic component model of Segflow. It can be correlated to an arc in graph theory and is defined below:

Definition 2.1 - Segment

A Segment, denoted by Ψ, is the fundamental model in Segflow. It is a directed component analogous to an arc. The segment is connected only in rooted trees and the end toward the root end of the tree is called the *Inward End*. The other end is called the *Outward End*. The indegree[17] of every segment is exactly one except for the source Segment, corresponding to the root of the tree, which has indegree of zero.



The Segment class provides abstraction[2] for trace pointers[14], loads, as well as node and connection information. Abstraction is also provided within the Segment class for database access, windowed output and graphical symbol manipulation. Several virtual methods[13] are provided to support distinct equipment modeling within those classes derived from the Segment class. The methods are essential because they support the modeling of specific power system equipment. The virtual methods may be overridden (replaced) with new methods which explicitly describe the behavior of the particular equipment type being modeled. These virtual methods govern Segment behavior associated with current, voltage, and impedance information passing among other things.

Conceptually, the non-virtual protected [13] member methods and attributes of the Segment class provide a cast for derived model classes. The representation below shows how this cast provides connection points for the Segment, orientation with an inward and outward side, and a reference direction for flows:

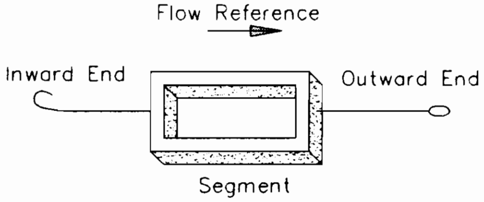


Figure 2.1 - A Representation of the Segment Class

Member attributes maintain current and potential values at the ends of the Segment and pointers support the connection of the Segment within the Segflow model.

The virtual methods allow derived classes to insert a model within the framework of the Segment to create an object specialized for modeling some type of power system equipment. This figure depicts a new model class being created from the Segment cast:

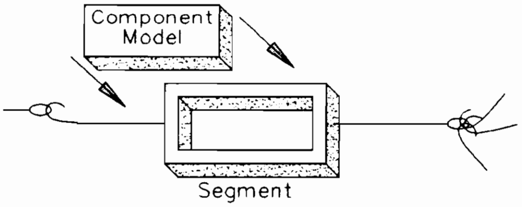


Figure 2.2 - A Derived Modeling Class

The role of the virtual methods in equipment modeling can be made more clear with examples of transformer and generator modeling presented in Chapter 6. The examples should help demonstrate how the Segment virtual functions fulfill very specific roles within Kirchoff's laws by relying upon attributes of the Segment class.

As discussed in Chapter 1, an objective of Segflow research is to produce a method of load-flow analysis that will allow software packages to be easily modified and maintained. There are several characteristics of the Segment object which allow the addition of new component models and the maintenance of old ones in Segflow.

- New Segment models are incorporated into Segflow with the registration of the Segment class's name into a database read by Segflow. (Dynamic linking of Segment classes is possible with new frameworks.[15]) No other modifications to the load flow are necessary.
- The load flow algorithm treats all Segment objects exactly the same. The load flow algorithm, as previously suggested, never has to be altered with the incorporation of new component models.
- Each Segment, has specific and unique data and functionality that allows it to model some piece of power system equipment. The modeling can be as detailed or general as the model builder likes.
- From the modeling viewpoint, Segments are connected at exactly one point and have a load at one point. Models are therefore created to respond to conditions at the connection point and the load point. One Segments has no responsibility to any other Segment (Unless the modeler creates a relationship).

2.1.1 Segment Attributes and Methods

The Segment contains attributes which describe its state and methods which describe its behavior. The following is a list of three essential attributes and methods which are briefly described. They are important to the development of current and voltage traces, Affiliation objects, and the Target Voltage method that is done later in this dissertation.

$\Psi \rightarrow_M Z$	Model Impedance -- This is a virtual method which allows the modeler to enter an expression for the Segment's series impedance. This method is used in the calculation of the source trace impedance discussed in Chapter 3.
$\Psi \rightarrow_M V_{Out}$	Outward Voltage -- The voltage at the Segment's outward node. The value of this attribute is determined from the method *OnVoltageTrace* discussed in Chapter 3.
$\Psi \rightarrow_M I_{In}$	Inward Current -- The current entering the Segment is determined from the method *OnCurrentTrace* discussed in Chapter 3.

The model impedance is essential to constructing the radialized model of Chapter 3. The inward current and outward voltage are commodities of voltage and current traces and are used to determine the solution of the radial collection of Segment objects.

2.1.2 Information Passing

Shown below is a tree of Segments representing component models:

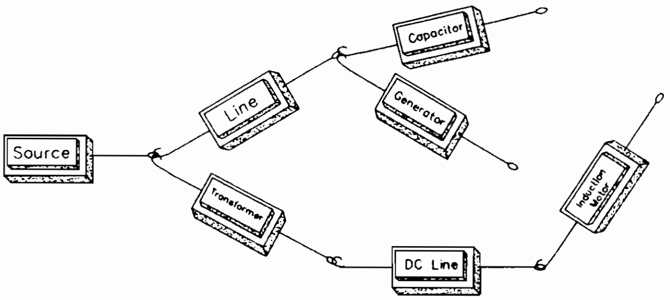


Figure 2.3 - Rooted Tree of Segments

Segments are connected into rooted trees so that information can be exchanged between a component and its single upstream connected Segment. That upstream Segment is called the feeder Segment and is defined as follows:

Definition 2.1 - Segment Feeder

The feeder of Segment Ψ_i is the single Segment, Ψ_j, in a Segflow rooted tree whose outward node is incident to the inward node of Ψ_i. A containment[13] attribute pointing to Ψ_j is maintained within Ψ_i and is noted as $\Psi_i \rightarrow_P \Psi_{Feed}$

Within the collection of Segments, information is only passed between a Segment and its feeder. The two primary types of information passed are voltage and current. Current information is sent from a Segment to its feeder during voltage traces (discussed in Chapter 3). During these traces, information about flows is accumulated from the ends of the radial model to the source. Here is a system of Segment objects, each representing some power system equipment:

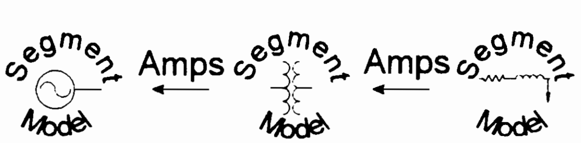


Figure 2.4 - Passing of Current Information

Information about voltage is requested by a Segment from its feeder. Another Segflow process called the voltage trace is used to propagate voltages from the source Segment to the end of the rooted tree of Segments:

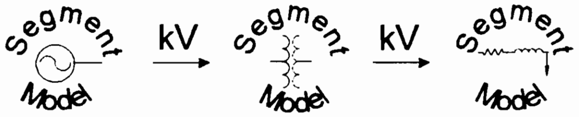


Figure 2.5 - Passing of Voltage Information

The powerful feature of the object-oriented design of Segflow is that any analytical model can be inside of the Segment model as long as current, voltage, and other types of information passing are supported. Detailed characteristics of adjacent Segments are unknown to a Segment. Each Segment simply expects a voltage and is responsible to provide a current to its feeder. The remaining chapters of this dissertation describe how the load-flow solution can be determined from this open object-oriented design.

Other types of information can be passed and other avenues of information passing can be established by the Segflow modeler. Links between Segments can be created to simulate remote controlling or coupled components. Dynamic modeling (briefly discussed in Chapter 6) can also be modeled with other types of information passing.

2.1.3 Segment Load

Models derived from the Segment class can utilize any sort linear or nonlinear injection internally. Segflow does, however, encapsulate the general modeling of constant power, current, or impedance loads as the following diagram suggests:

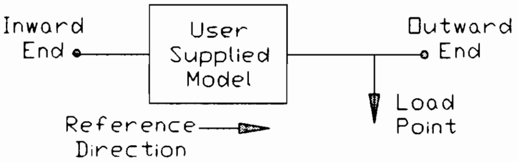


Figure 2.6 - A Segment Diagram Showing Encapsulated Load Point

Load at the load point is handled by the Segment class in two ways. Current injection from the load point is propagated to a current demand at the inward end and by message to the feeder Segment. The constant current and constant impedance loads are also evaluated and converted to constant current loads with the linearization method discussed in Chapter 6.

2.1.4 Compensation Current

Conpensation current is an attribute of the Segment class and represents a constant current load at the load point of the Segment model. The current is collected as a load by the Segment.

Definition 2.1 - Compensation Current

Compensation current is a Segment attribute and represents a constant current load. Compensation current for Segment i is noted as: $\Psi_i \rightarrow {}_A I_{Comp}$

Affiliation objects, presented in Chapter 4, contribute to this current in order to alter the outward voltage of Segments. Models for equipment such as generators might also contribute to the compensation current of a Segment. An example of generator modeling is presented in Chapter 6.

2.1.5 Getting Data to Segments

When a Segflow program is used, data has to be transferred from a database into appropriate Segments. A very important aspect of Segflow is the segment's ability to handle data input and output. A Segment method can be overridden by the modeler to intercept model information as Segflow reads an appropriate database. This diagram represents the process:

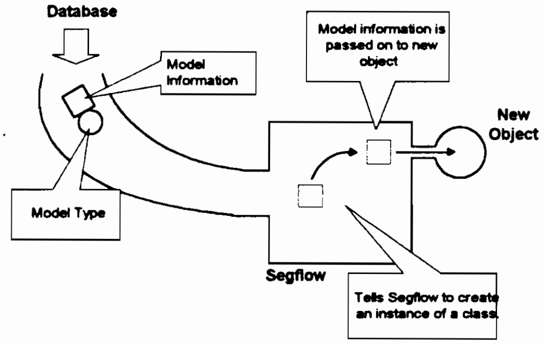


Figure 2.7 - Data Transfer to a Segment

Information is moved in groups from a data file or database into Segflow. From the information, Segflow creates instances of the appropriate class and passes the information on to that object. The data is interpreted by the new Segment with a method created by the model builder. This layout promotes object independence and maintains the fact that Segflow does not have to be altered with the incorporation of new component models.

2.1.6 Deriving New Models

As was previously mentioned, new equipment models are derived from the Segment class. These derived classes which are also called Segments exhibit all of the base class characteristics. Virtual functions are over-ridden to provide the specific behavior required for modeling their particular equipment. The figure below depicts classes derived from the Segment class:

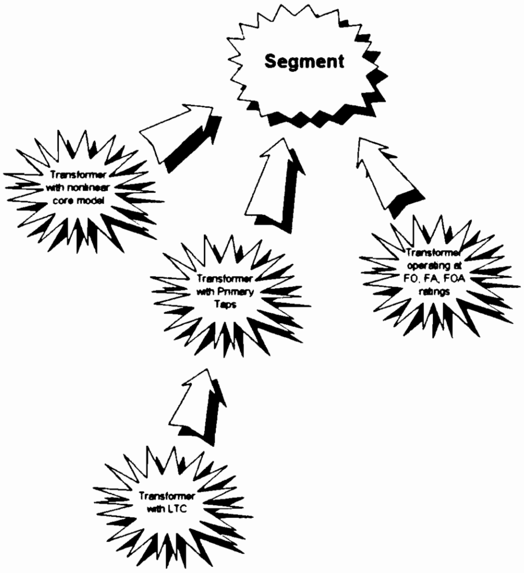


Figure 2.8 - Segments Derived from the Base Class

This diagram shows that three types of transformer models can be derived from the Segment class. The first transformer has a detailed core model that demonstrates core losses and saturation. The second transformer is an ideal transformer with series winding losses and a tap selector on the primary side. The third transformer derived directly from the segment class can operate with forced oil, forced air, or forced oil and air [16] depending on the loading. A transformer with a load tap changer can be derived from the transformer with the high side tap selector.

2.2 The System Class

The power system is represented with the System class. The collection of Segments used for component modeling is contained within the System class. The System class orchestrates the message passing that is used to invoke the traces and the Affiliation objects which will be discussed in Chapters 3 and 4.

The system class is essential to Segflow. The scope of all Segment objects is within the System object. It is the only object with influence transcending over all segments. It serves four fundamental functions:

- **Database I/O**: A database is used to store information about power system components. Chunks of information are grasped by the System object and used to construct the collection of Segment objects.
- **Algorithm Emulation**: The Segments in Segflow are connected for use with the trace methods discussed in the next chapter. The System class directs traces and activates Affiliation object methods. Linarization and Adjustment messages are also sent by the System class. All of these topics are presented later in the dissertation.
- **System Structure**: The System class is responsible for construction and maintaining the collection of Segment objects used for power system modeling.
- **Segment Access**: Segments may need access to other Segments other than their feeder. A Segment can make a request to the System object for the address of another Segment.

Chapter 3

The Radialized Model and Traces

3. The Radialized Model and Traces

Object independance in a load flow analysis scheme refers to the identical treatment of all component models. The object-oriented load flow is created to support object independance. This support is possible because of a fundamental model or template having a well established interface. Other models are created from this fundamental model and therefore inherit the common interface. The interface is sustained with a radial representation of the power system.

A power system model is solved when validation of Kirchoff's laws has been achieved throughout the system. Since Kirchoff's laws correspond to current flows and voltage potentials, the information to be passed in the object-oriented load flow is voltage and current. The reference direction of current flow is along the outward direction of a component (to be discussed below). This direction is always from left to right in the figures of this chapter.

3.1 The Radial Model

Components in power system network models are typically connected between or to nodes. Current flow through these components is assigned a reference direction. Arcs[17] are therefore valid representations of power system components in these models. The component model in the object-oriented load flow is analagous to an arc. The indicative arc of a power system with loops or meshes has multiple other components connected to either of its nodes. Both the indegree and outdegree[17] of the arc are greater than one. In a radial system with arcs referenced away from the source, the indegree of every arc is one and the outdegree is zero or more for connecting arcs.

Broadwater, Thompson, and McDermott showed the advantages in speed, storage, and elegance of using pointers in radial distrubution system analysis[14]. Another advantage of their methods, and in fact the contribution to this research, is the clear definition of the type and path of exchanged information. Here is a line section related to their work in radial systems:

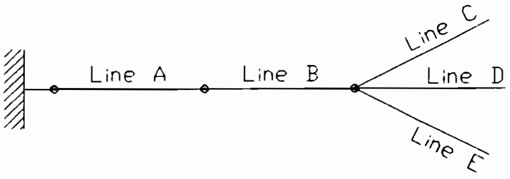


Figure 3.1 - A Sample Rooted Tree

Look at line section "B". There are two pieces of information handled by the line.

- First, all of the line flows for "Line C", "Line D", and "Line E" are combined to form a flow through "Line B". This information is added to the flow through "Line A". This type of information passing corresponds, in their work, to current and is analagous to the implementation of Kirchoff's current law.
- The second type of information passing corresponds to voltage. A voltage at the right end of "Line A" can be recieved by "Line B" and altered with some sort of voltage drop. The voltage at the right of "Line B" can be formed and subsequently be used by "Line C", "Line D", and "Line E". This information passing corresponds to Kirchoff's voltage law.

3.1.1 Some Examples

In the radial circuit above, Line B takes a value of current or voltage and supplements or alters it with an internal voltage drop or load. This revision takes place completely within the bounds of "Line B"; without the influence of any other line section. This demonstrates that the line sections are independant and that the actions are object-oriented ones.

Lets replace "Line B" with a transformer:

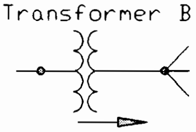


Figure 3.2 - A Transformer Example

The first KCL type operation occurs when the current demand from all Segments connected to the transformer's outward end are divided by the turns ratio of the transformer and added to the transformer's feeder Segment. The second KVL type operation occurs when the transformer multiplies the voltage at its inward node by the transformer turns ratio and supplies that resulting voltage to all of its outwardly connected components.

Here is a capacitor replacement of “Line B”:

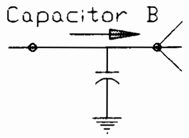


Figure 3.3 - A Capacitor Example

During the KCL operation, the demand current from the capacitor component’s outwardly connected components is combined with the capacitor injection and supplied as a demand on the capacitors inward component. The KVL operation takes the voltage at the capacitors inward node and supplies it to the outward node.

The purpose of these examples is to explicity show that

1. The type of component does not effect the general KVL and KCL operations.
2. Voltage and currents are commodities exchanged with a components inward connection. These commodities can be altered by a component without concern for other components in the system.

3.1.2 The Advantages

It is advantageous for a component model to be connected in a rooted tree for a number of reasons. First, a component passes and retrieves information from exactly one component -- its innode connected component. Second, the topology of the network can be maintained with a single linked list. This is a very simple and efficient method for containing a circuits topology. Finally, the linked list(s) that are used to emulate the KVL and KCL patters of operation provide a natural data base structure. It is therefore concievalble that a power system’s database, memory structure, and load flow algorithm can be the same.

3.1.3 Who else uses a rooted tree

Authors Shirmohammadi, Semlyen and Luo have developed a method for solving power system network models with a radial representations. The authors present a robust and efficient method for solving “weakly meshed distribution systems.” An infinite bus or swing machine is chosen as the root of their radial model. They break the weakly meshed system at “breakpoints” to form a radial system. The source in the radial model is the swing bus and each break point is the intersection of exactly two line sections.

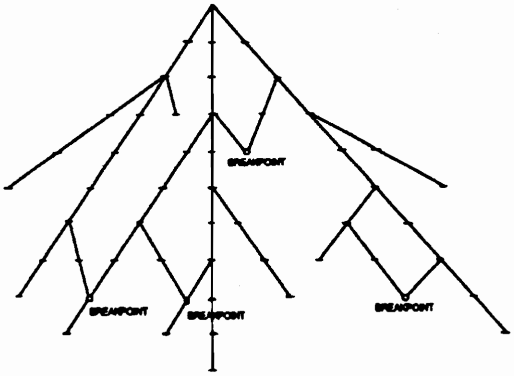


Figure 3.4 - A Weakly Meshed System With Breakpoints

That radial system is solved with forward and backward "sweeps" corresponding to the general trace methods that will be discussed below.

3.1.3.1 Shirmohammadi, Semlyen and Luo[18]

The authors use a multi-port compensation method to load the breakpoint nodes with constant current loads. The values of the loads are determined from a breakpoint impedance matrix which is global:

$$\begin{Bmatrix} \hat{V}_1 \\ \vdots \\ \hat{V}_j \\ \vdots \\ \hat{V}_p \end{Bmatrix} = \begin{bmatrix} \hat{Z}_{11} & \cdots & \hat{Z}_{1j} & \cdots & \hat{Z}_{1p} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \hat{Z}_{j1} & \cdots & \hat{Z}_{jj} & \cdots & \hat{Z}_{jp} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \hat{Z}_{p1} & \cdots & \hat{Z}_{pj} & \cdots & \hat{Z}_{pp} \end{bmatrix} \begin{Bmatrix} \hat{J}_1 \\ \vdots \\ \hat{J}_j \\ \vdots \\ \hat{J}_p \end{Bmatrix} \qquad (3.1)$$

The dimension of the impedance matrix corresponds to the number of breakpoints. All breakpoints have exactly two branches and the combined load added to each section is zero. The above impedance matrix is small for weakly meshed systems regardless of the system size. A system with 20,000 branches and two loops will have a 2X2 breakpoint impedance matrix. The matrix does, however, have to be constructed globally. Diagonal terms (Thevenin Impedances) do correspond to trace impedances. Off diagonal terms must be determined from the network topology.

While a radial representation of a power system network is used in the research represented by this dissertation, there are fundamental differences between this contribution and that of Shirmohammadi, Semlyen and Luo.

- First, the load flow presented here is object-oriented since no global matricies or evaluations have to be performed.

- Second, the object-oriented load flow supports densly meshed power system networks and breakpoints having more than two incoming sections.
- Finally, the series impedance of components is not needed by the object-oriented load flow in support of component modeling freedom. (Some representative number of a series impedance is requested for the construction of the radial model. This number is not used during the load flow convergence process.)

3.1.3.2 The later work of Luo & Semlyen [19]

The work here is built from the work of Shirmohammadi, Semlyen, & Luo discussed previously. The primary modification is the use of active and reactive flows along the forward and backward sweeps. This modification works much better with PQ busses but doubles the size of the breakpoint transformation matrix. The matrix was previously called the breakpoint impedance matrix. It is now called the sensitivity matrix. It relates changes in active and reactive compensation power loading to changes in bus voltage and angle. They apply a detailed method for network flow tree labeling to determine impedance values for use in their sensitivity matrix.

It is conceivable that power flows could be used in the object-oriented load flow instead of current flows. Although this does simplify the arithmetic associated with the modeling of PQ loads and generators, it increases the complexity of modeling voltage drops across lines, transformers, or constant current or impednac loads. The information associated with Kirchoff's laws, current and voltage, is maintained in the object-oriented load flow. It is the responsibility of the component modeler to emulate the behavior of aspects of a component model associated with the nonlinear quantity, power.

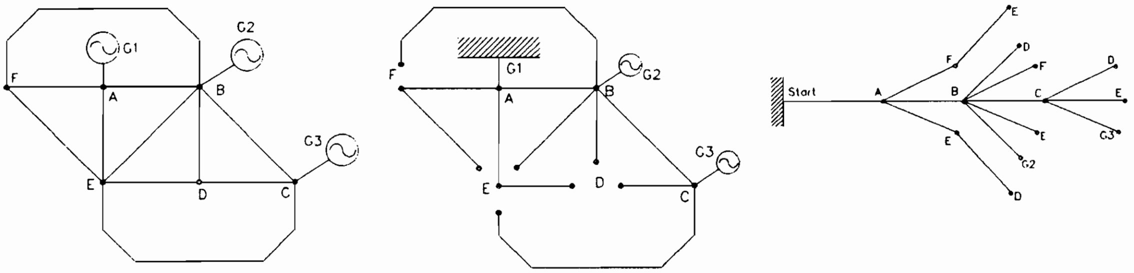
3.2 Constructing the Radialized Model

The discussion above is supportive of an object-oriented load flow layout. The root, or substation in radial distribution system analysis, corresponds to a node with both know voltage magnitude and voltage angle. In the analysis of power system networks, the swing bus has a constant and predetermined value for both of these quantities. Power system networks analyzed by the object-oriented load flow are therefore rendered into a radial system with the swing bus assuming the position of tree root.

Definition 3.1 - Radialization

A power system network model is Radialized particular arcs are disconnected from certain nodes so that the indegree of every arc (except the arc cooresponding to the source) is one. A radialized system is delimeted as Γ and the root or infinite bus Segment is delimited $\Gamma \rightarrow Start_{\Psi}$. The magnitude of the source trace impedance (discussed later) for each Segment in a radialized model is a minimum.

Shown below is the Wood & Wollenberg power system network and the radialized model:



1. Power Network *2. Breaks for Radialization* *3. Radialized Model-* Γ

Figure 3.5 - The Radialization Process

The purpose of this radialization process is to allow a clear avenue of communication between a component and another. Component models in the radialized system communicate exclusively with the single upsream component.

Construction of the radial representation of the power system model is done with a network combinatorial optimization technique[17]. The resulting model must be radial and the source trace impedance of any Segment has to be a minimum. Using the combinatorial technique, the radialized system model is created by starting with the source Segment (representing the swing bus). Each time a Segment is added to the growing rooted tree, all Segments that could possibly be connected are evaluated. The Segment that, when connected, would have the least source trace impedance is actually connected. All other Segments remain for evaluation during subsequent connections.

Pointers for the inward, outward, and feeder Segments are created and maintained as the rooted tree is constructed. The process of constructing the rooted tree is the only operation requiring global operation in the object-oriented load flow. The process of constructing the rooted tree with each Segment having a minimum source trace impedance is time consuming. The IEEE 300 bus test system took approximately three minutes to construct. However, the tree only needs to be constructed once because it can be saved to a database with all tree related pointers intact. The system can be quickly read in as a radialized model during consecutive system loading.

3.3 Trace Methods for Solving the Radialized Model

Up to this point, two objectives necessary for the presentation of the object-oriented load flow have been discussed. First, the meaning of object independance has been defined in the last chapter and established as the Segment previously in this chapter. Second, a structure for object communication has been generated with the incorporation of a radial representation of a power system network. Their are two objectives remaining.

The next chapter will present a method for realizing the solution of a power system network with its radialized representative. This section, however, will present a previously secured method for solving radial circuit models. This method referred to as the Trace Method supports object independence and the object communication ideas developed earlier. The Trace Method has been developed by R. Broadwater and others[14,20,21,22] and is a descendent of the Ladder Method created by W. Kersting[23,24]. The Ladder Method and the Trace Method have become valuable tools for radial system analysis. The difference between the conceptualization of the two methods (not necessarily the application) is subtle but distinct.

3.3.1 The Ladder Method

The Ladder Method. was derived from a popular method to solve non-recurrent and linear ladder type circuits such as the one below.

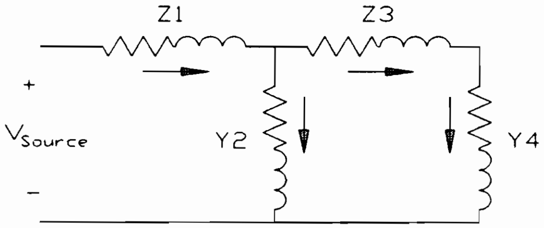


Figure 3.6 - Non-recurrent Linear Ladder Type Circuit

The source voltage to the left is known as some constant value. The solution process is started by assuming a voltage across the admittance Y4. This assumption allows the calculation of the current through Y4 and Z3. A voltage drop can be calculated across Z3 to give the voltage across the admittance Y2. This process is continued all the way back to the source. The calculated source voltage is compared to the actual source voltage. The ratio of the two is used to multiply the voltages and currents throughout the rest of the circuit. After the multiplication, the solution of the entire linear circuit is known.

The solution of a system with PQ load is different from the case of the linear circuit. Here is the same figure except that the admittances have been replaced with PQ loads:

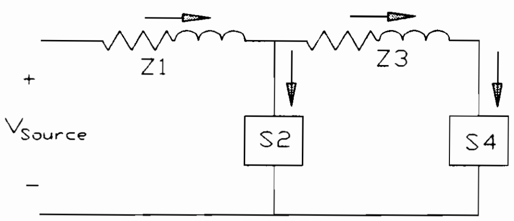


Figure 3.7 - Ladder Type Network with PQ Loads

Once again, a voltage is assumed at the end of the "ladder" across the load S4. Load current is developed from the assumption and used to get the voltage drop across Z3. A voltage can be determined for the load S2 and the calculation of a corresponding load current. The process continues all the way back to the source. At the source, the difference between the actual source value and the calculated value is determined. That difference is then added to the voltage guess across S4[1] and the process is repreated until the currents evaluated at the PQ loads remain stationary within toleration. The process usually takes four to six iterations. In a radial distribution system, the process is applied to the laterals and then to the main feeder in some independent fashion.

3.3.1.1 A Look at the Ladder Method on a Simple Voltage Divider

This simple example is given to contrast the effectiveness of the ladder method with more general trace methods. The ladder method is very effective for solving circuits with constant impedance loads like the one that follows:

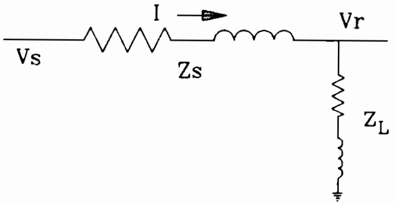


Figure 3.8 - Circuit with Constant Impedance Load

Implementing the Ladder method begins with a guess of the receiving end voltage:

$$V_r^{(1)} = Guess \tag{3.2}$$

The next step is to determine the load current:

---

[1] The per-unit system is used with the Ladder method

$$I_L^{(1)} = \frac{V_r^{(1)}}{Z_L} \tag{3.3}$$

The load drop across the series impedance can be considered to find the calculated source voltage:

$$V_S{}' = V_r^{(1)} + \frac{V_r^{(1)}}{Z_L} Z_S \tag{3.4}$$

The error in the calculated source voltage is found as:

$$\varepsilon = \frac{V_S}{V_r^{(1)} + \frac{V_r^{(1)}}{Z_L} Z_S} \tag{3.5}$$

And the new receiving end voltage is scaled:

$$V_r^{(2)} = \varepsilon V_r^{(1)} = V_S \frac{Z_L}{Z_L + Z_S} \tag{3.6}$$

The ladder method converges after one iteration on a circuit with constant impedance load.

3.3.1.2 The Ladder Method with Constant Current Load

Repeating the above analysis for a constant current load results in an iterative expression for the receiving end voltage:

$$V_r^{(i+1)} = \frac{V_S^i}{\left(V_S + I_L Z_S\right)^{i-1} + \frac{I_L^i Z_S^i}{V_r^{(1)}}} \tag{3.7}$$

A binomial series expansion can be used twice on the denominator to show that:

$$\lim_{i \to \infty} V_r^{(i)} = V_S - I_L Z_S \tag{3.8}$$

3.3.2 The Trace Method

The Ladder method that was just described is driven by inward motions from the ends of a radial system to the source. Errors are evaluated at tie points between laterals and the main feeder as well as at the source. These errors are used to adjust the voltages at the ends of the system for the next inward motion.

The Trace Method has an inward motion corresponding to the collection of load current and the satisfaction of Kirchoff's current law. (The Ladder Method attepts to satisfy both KCL and KVL along the inward motion.) Unlike the Ladder Method, the Trace Method incorporates an outward propagation starting at the radial systems source and extending to the outer edges of the system. Kirchoff's voltage law is applied during this propogation. After iterations of these inward and outward motions, both the current and voltage law of Kirchoff will be simultaneously satisfied for every component in the model signifying the solution of the collective system.

The Trace Method is well suited for an object-oriented power flow design. It is implemented by recognizing each element in a power system's model in an organized manner. Segments in the object-oriented load flow pass information in a manner consistent with the topological layout used for the solution of radial systems using the Trace Method.

3.3.2.1 Some Definitions

The Trace Method is the motivating force behind the solution of a power system model with the object-oriented load flow. The terminology associated with the description of the load flow will, therefore, be presented here with the introduction of the primary component -- the Trace Method.

The first definition is used to describe the recognition of Segments which are independant and isolated except for their methods of message passing:

Definition 3.2 - Focus

When control or attention is given to the state, methods, or data of a particular Segment, Ψ, in the object-oriented load flow via a particular method of the load flow the Segment is said to have been ***given focus***.

The source trace is not explicitly used in the object-oriented load flow. It is, however, important to the understanding of the inward and outward trace that will be defined shortly.

Definition 3.3 - Source Trace

The source trace of Segment, Ψ, is the set of Segments not including Ψ such that each Segment in the set occurs once and each Segment belongs to the unique path between Ψ and the system source or start $\Gamma \rightarrow Start_{\Psi}$.

An inward trace represents the order in which Segments are focused during the load currrent collecting portion of the Trace Method.

Definition 3.1 - Inward or Current Trace

A Segment, Ψ, in a radialized model Γ can be given focus during the inward trace only if the Segment is an end or all other Segments having Ψ in their source trace have been focused during the same inward trace. Each Segment in Γ is given focus exactly once during an inward trace. The first Segment focused during an inward trace is delimeted $\Gamma \rightarrow End_{\Psi}$. The inward trace is alwo known as the current trace because it is used to collect load current towards the radialized model's source.

Inward or current traces are used to collect currents from the outskirts of a circuit, through the circuit, and back to the source.

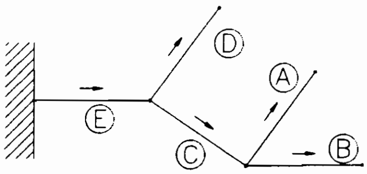


Figure 3.9 - Current Trace Circuit

An inward accumulates through current from component **B** and **A** to component **C**. That current is combined with **D** and added to **E**. The current demand from the source is then known.

Segments are focused along an outward trace such that upstream voltage and drop information can be propogated from the source of the radial model to the outer segments.

Definition 3.1 - Outward or Voltage Trace

A Segment, Ψ, in a radialized model Γ can be given focus during the outward trace only if Ψ is the source or starting Segment $\Gamma \rightarrow Start_{\Psi}$ or all other Segments in the source trace of Ψ have been focused during the same outward trace. The outward trace is known as the voltage trace because it is used to propogate voltages from the source of the radialized model throughout the circuit.

An outward trace is the ordered progression from component to component starting from the radial system's source to all other components. The outward trace is used for voltage propagation. Consider the source voltage in the circuit shown below:

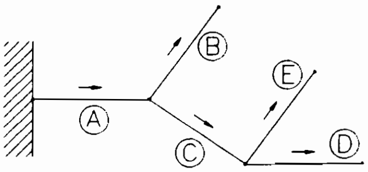


Figure 3.10 - Voltage Trace Circuit

Voltages are propagated from the source to sections **A**, **B**, **C**, **E**, and **D** in that order[2]. This movement among the components is called an outward trace [17] since components gaining control are gradually farther or more outward from the source. Broadwater refers to the process as a forward trace in his work [14].

3.3.2.2 Solving circuits with trace methods

The inward and outward propogations briefly presented above are used to solve circuits like the example presented earlier:

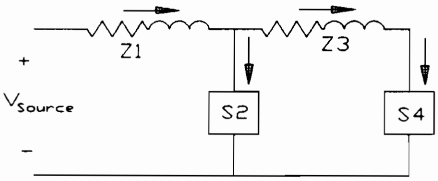


Figure 3.11 - Ladder Circuit with PQ Loads

An outward trace is performed which propagates the precise source voltage to each of the nodes atop the loads S2 and S4. Next, the load current of S4 is evaluated with the loads voltage. That current is used as the current flow through Z3. The load current through S2 is added to the current through Z3 to get the current through Z1. An outward trace is started at the source. The current through Z1 is used to evaluate the voltage drop and get a new voltage across S2. The current through Z3 is used to further drop the voltage to S4. The process is reiterated with the new voltage estimates and eventually converges.

3.3.2.3 Trace methods on a constant current circuit

The trace methods inherently solve radial systems with constant current loads with exactly one inward and one outward trace. Here is a small radial system with a single constant current load:

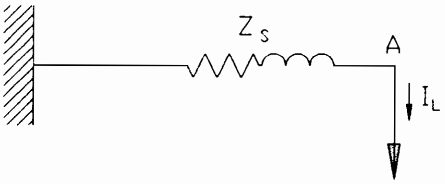


Figure 3.12 - Circuit with Constant Current Load

---

[2] Other orders are possible.

During the inward trace, the load current is summed back to the source. The voltage drop across the line can then be used from the source voltage to get the voltage at **A**. Since the current remains constant, the system is solved with the single inward and outward trace set.

3.3.2.4 Trace Methods with Constant Impedance Load

The behavior of the trace method will be investigated on this circuit:

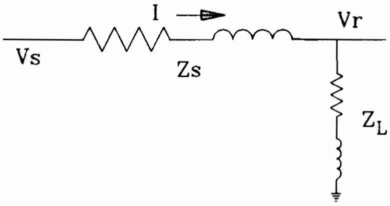


Figure 3.13 - Circuit With Constant Impedance Load

This is a linear problem since the source voltage, line impedance, and load impedance will be given. The following derivation will show how the trace method reaches the correct solution for the receiving end voltage. It will also be shown that the method will not converge if the line impedance is greater than the load impedance.

Given the line and load impedance, a trace method iteration can be defined by a forward and backward trace. After a forward trace is used to propagate the source voltage, a backward trace is implemented:

$$I = \frac{V_r}{Z_L} \tag{3.9}$$

Use a forward trace to determine the segment downstream voltage:

$$V_r = V_s - IZ_s \tag{3.10}$$

To start the iterative process, the downstream voltage should be estimated as the source voltage. A backward and forward trace used in succession constitutes an iteration and yields a new guess at the downstream voltage:

$$V_r^{(1)} = V_s - \frac{V_s}{Z_L} Z_s \tag{3.11}$$

Another backward and forward trace can be induced on the above results:

$$V_r^{(2)} = V_s \left(1 - \frac{Z_s}{Z_L} + \frac{Z_s^2}{Z_L^2} \right) \tag{3.12}$$

And after more backward and forward traces:

$$V_r = V_s\left(1 - \frac{Z_s}{Z_L} + \frac{Z_s^2}{Z_L^2} - \frac{Z_s^3}{Z_L^3} + \cdots\right) \tag{3.13}$$

The downstream voltage calculated with Ladder method iterations can be written as:

$$V_r = V_s \sum_{n=0}^{\infty}\left(-\frac{Z_s}{Z_L}\right)^n \tag{3.14}$$

This equation demonstrates that the trace method will only converge if:

$$\text{Series converges if } \left|\frac{Z_s}{Z_L}\right| < 1.0 \tag{3.15}$$

This result shows that the total equivalent series impedance between a segment and its source should always be less than the fixed impedance representing the load and downstream demand.

Equation (3.14) is useful to show that the trace method does converge to the correct solution. Use this substitution in (3.14):

$$A = \frac{Z_s}{Z_L} \tag{3.16}$$

to get:

$$V_r = V_s\left(1 - A + A^2 - A^3 + \cdots\right) \tag{3.17}$$

which represents a binomial series in A:

$$1 - A + A^2 - A^3 + \cdots = (1 + A)^{-1} \tag{3.18}$$

So, if we revert the substitution of A then:

$$V_r = V_s \frac{1}{1 + \frac{Z_s}{Z_L}} = V_s \frac{Z_L}{Z_L + Z_s} \tag{3.19}$$

which is the correct solution to the voltage divider circuit.

3.3.2.5 The Application of the Trace Method by Broadwater & Rahman [21]

The Distribution Analysis and Economic Evaluation Workstation (DANE) is presented in the respective paper. The product is intended for student use but does contain state-of-the-art concepts about openarchitecture and object-oriented analysis. The application programmer interface (API) is constructed with objects (groups of data). There are a number of characteristics of DANE that are directly related to this dissertation work:

- Forward and Backward traces are used to solve radial systems.
- Different object types respond differently during traces.
- Dynamic memory allocation is used to support user modeling.

DANE is a precursor to the EPRI Distribution Engineering Workstation (DEW).

3.3.2.6 The Work of Broadwater and others [22]

The archetecture of the Distribution Engineering Workstation is designed around the heavy reliance on inward and outward traces (forward and backward traces). The support of object independance although a truely object oriented design is not persued in liu of a relational data scheme. The paper shows how classes with similar characteristics can be grouped together into a superclass. A subclass is a specialization of a higher level class. It includes the attributes and methods of its parent class. These characteristics are said to be inherited from the parent class. An example of the class hierarchy concept can be found in the design of the EPRI Distribution Engineering Workstation [22]:

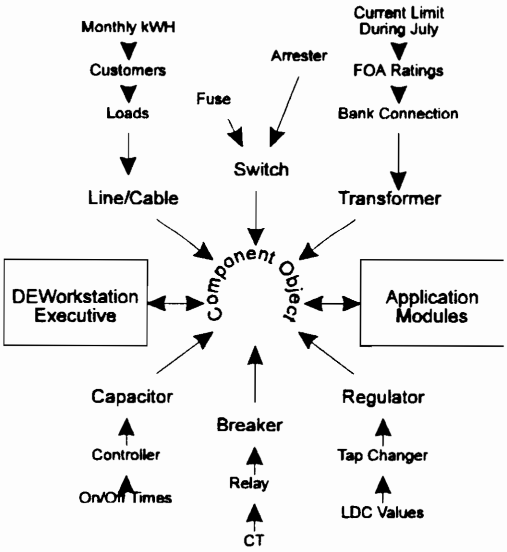


Figure 3.14 - DEWorkstation Component Objects

Object-oriented concepts are used with DEWorkstation even though object-oriented programming languages are not. A component object is the fundamental structure (analogous to class) describing a power system. Power system components such as lines, transformers, or breakers are derived from the component object. Components such as switched or fixed capacitors can be further derived from the capacitor object.

3.4 The Source Trace Impedance

The source trace impedance is the Thevinin impedance for a Segment. It is, that is, the impedance lying along the path between a Segment and the source of the radialized model. The accumulation of source trace impedance for a segment is an object-oriented technique. A global or localized algorithm is not needed to compute the impedance. The impedance is instead propagated from component to component along an outward trace.

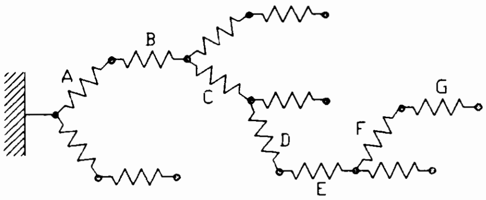


Figure 3.15 - Diagram Showing Source Trace Impedance

The source trace impedance for **G** can be found by adding the impedances of **A**, **B**, **C**, **D**, **E**, **F**, and **G**. A change in current at **G** will bring about a proportional change in voltage at **G** with the proportionality constant being the source trace impedance of **G**.

Definition 3.1 - Source Trace Impedance

The source trace impedance for a Segment is simply the source trace impedance of the segment's feeder supplemented with the modeler supplied impedance of the Segment. This impedance is found for each Segment during outward propogations of the object-oriented load flow algorithm. It is the impedance between a segment and the radialized network source along the unique path separating the two. The source trace of impedance of Segment, Ψ, is the impedance seen from the outward end of Ψ along the source trace. The source trace impedance of a Segment is denoted as $\Psi \rightarrow_A Z_S$.

3.4.1 Minimal Voltage Drop

The source trace impedance for each Segment is minimized during the constrution of the radialized model to insure that the voltage drop due to a load is minimal.

Loads in the radialized model are fed from a single path from their respective Segment back to the model's source. If the radialized model is not constructed correctly, this path may be one of unnaturally high impedance. Consider the simple power system network with the infinite bus, two 115/13.8kV transformers, and a very large load shown:

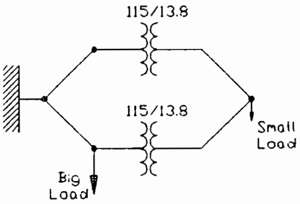


Figure 3.16 - A Power System with Sub-transmission and Distribution voltages

If the system is radialized as follows:

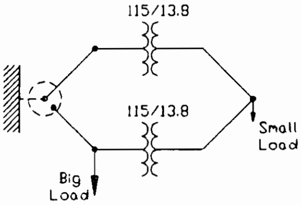


Figure 3.17 - A Poor Radialization of the Network

then an enormous load current would be forced to travel through the 115/13.8kV transformer and through the 13.8kV portion of the model. This is an unnatural path because it is a high impedance path. The source trace impedance for all segments in the above system is not a minimum. Clearly the best path for the load current is along the 115kV line section Segment and into the source.

In this radialization all segments have a minimum source trace impedance:

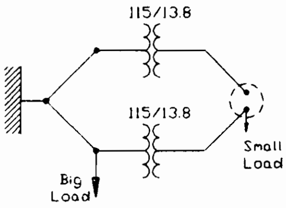


Figure 3.18 - The Best Radialization of the Network

All loads travel along a path of least impedance within the radialized model. The Affiliation objects will alter current flow so that the state of the radialized model corresponds to the solution of the power system network.

3.4.1.1 Load Placement

Loads in a power system network may have many possible locations for placement in the radialized model. The correct place to put the load is with the Segment having the least source trace impedance. The load is therefore fed along a path with minimal loss and voltage drop.

3.5 The Move to Networks

The radialized model and the Trace Method are the basis of the object-oriented load flow because they support object independance from the load flow as well as distinct avenues for message passing. In order to solve power systems network models, a new object has to be introduced called the Affiliation object. Affiliation objects communicate with Segments whose outward nodes correspond to some common node in the power system network. Each Affiliation object communicates with its member Segments until the outward voltage of all of the members is the same. If these voltages match for the collection of Segments belonging to each Affiliation object then the solution of the radialized model will match the solution of the power system network. Affiliation objects and their internal methods called the Target Voltage method are the topic of the next chapter.

Chapter 4

Affiliation Objects & The Target Voltage Method

4. Affiliation Objects & The Target Voltage method

The previous chapter showed that Segflow power system model consists of a collection of radially connected Segments. If each Segment is connected so that its source trace impedance is minimal that the collection is called the radialized model of the power system. A set of trace techniques were presented called the current and voltage traces which are used to find the solution of the radialized model with constant current loads.

Some components are broken apart when a power system network model is radialized. The connection information that is lost through the radialization process is maintained with Affiliation objects. This chapter presents the Affiliation object which is used to drive the solution of the radialized system to match the solution of the power system network model with corresponding current loads. The algorithm used by the Affiliation object is the Target Voltage method.

After the successful implementation of Affiliation objects and their inner mechanism, the Target Voltage method, the solution of the power system network with constant current injections will have been found. Chapter 6 will discuss the Segflow process and the linearization methods of Segments which allow the solution of the nonlinear network model and even dynamic simulation of power system behavior.

4.1 Affiliation Objects

The connection information that is lost through the radialization process is maintained with Affiliation objects. Segments whose outward nodes are disconnected to form the radialized model become members of a container attribute [25] of a particular Affiliation object corresponding to the respective outward node. This Affiliation object simulates the behavior of the respective network node from within the radialized model. The purpose of an Affiliation object is to bring the outward voltages of its member Segments to the same value. When this purpose is fulfilled, the Affiliation object forms a “virtual node”.

Affiliation objects are considered container objects because they associate or have access to a collection of Segment objects. They are also a vehicle for the implementation of the Target Voltage method that will be presented later in this chapter.

Definition 4.1 - Affiliation objects

Segments of the set $\Psi_1 \ldots \Psi_N$ of the radialized model Γ are members of Affiliation object Λ if superimposing $\Psi_1 \ldots \Psi_N$ onto the model of the power system network reveals that the outward ends of the Segments form a node. The Affiliation object retains, as its own, the name of the common outward nodes of the associated Segments. This common outward node is called the identity node of the Affiliation object. The degree of a Affiliation object corresponds to the number of associated Segments.

4.1.1 A Clarification of Affiliation object Members

As the above definition implies, all Segments whose outward nodes are the identity node of an Affiliation object are members of that object. The inward node of some Segments correspond to the identity node of an Affiliation object. The outward nodes of the Segments in the figure below are marked with arrows. The inward node is unmarked. The dot in the middle of the diagram refers to the identity node of the Affiliation object to which the figure corresponds:

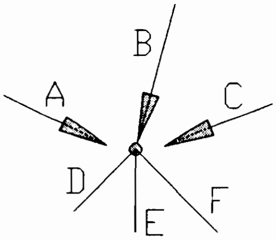
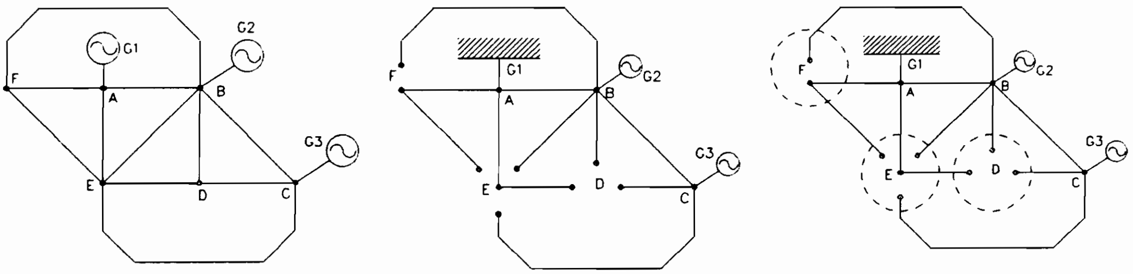


Figure 4.1 - A Node in a Network and Affiliation object Member Segments

The system has been radialized so that Segments **A**, **B**, and **C** are member of a Affiliation object. Segments **A** and **C** were disconnected from the node to radialized the model. Segment **B** has an outward node corresponding to the Affiliation object's node. Segments **D**, **E**, and **F** are fed from Segment **B** during current and voltage traces. They are not members of the Affiliation object and only interchange information with Segment **B**. The Affiliation object drives the outward voltage of Segments **B**, **A**, and **C** together. Segments **D**, **E**, and **F** bind their inward voltage to the outward value of B during voltage traces. Thus, the corresponding inward and outward voltage of all Segments involved become the same value by the action of the Affiliation object and voltage traces and the voltage of the emerging virtual node is found. The section below shows the Affiliation objects of an example system.

4.1.2 Return to the Six Bus Example System

Shown in Figure 4.2 (A) is the six bus example system presented in Chapter 3. The second frame, Figure 4.2 (B) shows the breaks created during the radialization process. Affiliation objects are introduced by Segflow to bind those Segments together in the radialized model. The Affiliation objects are shown in the last figure:



(A) Power Network *(B) Radialized Model* *(C) Affiliation objects* $\Lambda_F, \Lambda_E, \Lambda_D$

Figure 4.2 - Three Affiliation objects

The locations of Affiliation objects can be found from inspection after a network model has been radialized. The connectivity of the radialized model cannot be found without the application of an algorithm which minimizes the source trace impedance for every Segment. The radialization process and the number and degree of Affiliation objects are therefore dependent. This dependency is rendered in the next section.

4.1.3 The Number of Affiliation objects

The radialization of a power system network model may not produce a unique radial model. (The constraint that all Segments have a minimum source trace impedance that was given previously with the definition of radialization does tend to produce unique radial models.) For a given power system model, however, their is always a relationship to any radialized model and its Affiliation objects. The relationship involves the number of simple loops or meshes in the network model and the number of Affiliation objects and Segments that are members of Affiliation objects in the corresponding radialized model.

From Linear Network Theory [17, 26] it is known that the number of branches in the power system network's cotree is related to the number of loops:

$$N_{Cotree} = N_{Loops} \tag{4.1}$$

The set of branches in the cotree is characterized by two items:

1. The cotree voltages are a basic set of variables from which all other variables can be determined. That is, the cotree voltages contain complete information about the state of a network.
2. Breaking or removing all of the cotree branches plus one non-cotree branch splits the network into two disconnected trees.

Here is a network that has its cotree and one tree branch marked by the intersection of a dotted line:

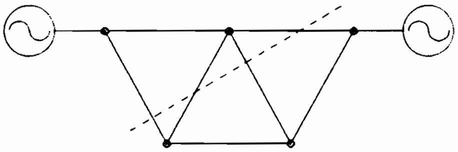


Figure 4.3 - The Cotree and One Tree Branch

During the radialization of a power system network, some branches are broken away from a node at their outward end. These branches are called "broken branches" and become Segments belonging to Affiliation objects in Segflow. A set of broken branches is created when the power system model is radialized to form a rooted tree. This set is a cotree set. The breaking of any other branch in the radialized model will separate the tree, and thus the original network, into two trees. It follows therefore[17], that:

$$N_{Breaks} = N_{Cotree} = N_{Loops} \tag{4.2}$$

$$where$$

$$N_{Breaks} = \text{Number of Branches Broken From a Node}$$

$$N_{Cotree} = \text{Number of Branches in Network's Cotree}$$

$$N_{Loops} = \text{Number of Simple Loops or Meshes in Network}$$

An Affiliation object is formed for each set of broken branches whose outward end is incident with the same node. Since the radialized model consists of a single rooted tree, the identity node of the Affiliation object will have connected to it the outward node of a single branch. This node is, by definition, a member of the Affiliation object. An Affiliation object has a number of Segments associated with it that correspond to branches having outward nodes incident to the Affiliation object's identity node. Exactly one Segment in each Affiliation object will correspond to a branch that was not broken to form the radial model and still has an outward node incident to the identity node. So, the total number of Affiliation object member Segments correspond to the number of cotree branches summed with the number of Affiliation objects.

Each Affiliation object will have a collection of broken-branch type Segments and one Segment that does not correspond to a broken-branch. Thus using equation (4.2):

$$N_{\Lambda S} = N_{Loops} + N_{\Lambda} \tag{4.3}$$

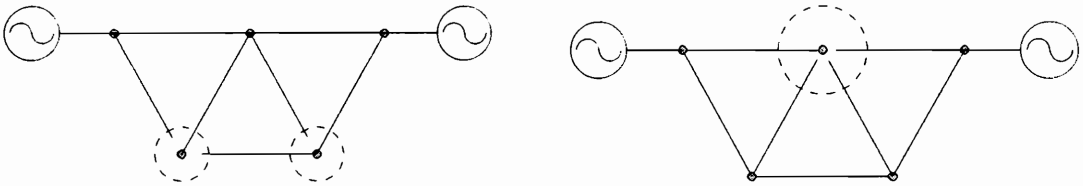
$$where$$

$$N_{\Lambda S} = \text{Number of Segments Belonging to Loop Objects}$$

$$N_{Loops} = \text{Number of Simple Loops or Meshes in the Network}$$

$$N_{\Lambda} = \text{Number of Affiliation objects}$$

Consider two radializations of the simple network presented above. The Affiliation objects of each radialization are governed by the above equation:



5 Segments Belonging to Affiliation objects
3 Loops
2 Affiliation objects

4 Segments Belonging to Affiliation objects
3 Loops
1 Affiliation object

Figure 4.4 - Two Groups of Affiliation objects

4.1.4 Affiliation object Statistics

Experimentation has been performed on a number of popular test systems. Although the details of these experiments will be given in Chapter 6, a chart with information regarding Affiliation object degree is given on the following page:

		Number of Affiliation objects with Specified Number of Segments			
	# Busses	2 Seg.	3 Seg.	4 Seg.	5 Seg.
Stevenson	5	2	-	-	-
Anderson & Fouad	9	1	-	-	-
Wood & Wollenberg	6	2	2	-	-
IEEE 14	14	7	-	-	-
Gross	16	1	-	-	-
IEEE 30	30	12	-	-	-
NMPP	36	11	-	-	-
New England	57	8	-	-	-
IEEE 300	300	73	18	1	-
Frequency		84%	15%	1%	0%

Table 4.1

The models listed above have been properly radialized with each Segment having a minimum source trace impedance. The chart shows that these power system networks contain mostly degree-two Affiliation objects. This is just an interesting result pertaining to the expectations of Affiliation objects within a properly radialized power system model.

4.2 The Action/Reaction Predictability Problem

Affiliation objects support the object-oriented layout of Segflow because they are autonomous and they send messages to member Segments requesting specific information and actions. The primary internal method of Affiliation objects is the Target Voltage method. This method operates over the information available to Affiliation objects and produces output that can be sent to the member Segments of an Affiliation object. This section does review the action/reaction behavior of a linear and radial system with constant current loads. This concept is fundamental to the derivation of the Target Voltage method in the next section and to its analysis in Chapter 5.

4.2.1 The Action/Reaction Predictability Problem

Shown below is the radialized model for the six bus network presented in Chapter 3. The source trace for $\Psi_{C,D}$ is shown as a solid line:

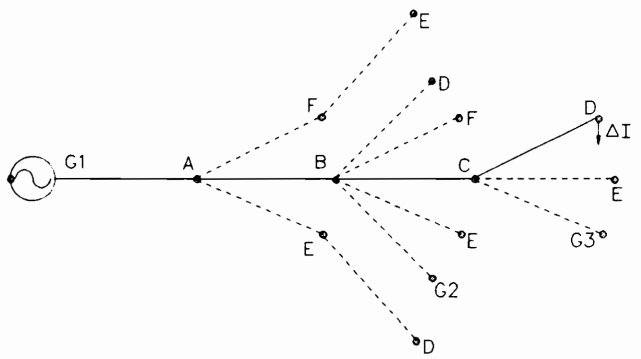


Figure 4.5 - Radialized Model with Source Trace and Compensation Current Shown

A change in compensation current (discussed in Chapter 2) for $\Psi_{C,D}$ is depicted as ΔI. During an outward trace , the source trace impedance for all Segments could be found. The value for $\Psi_{C,D}$ is:

$$\Psi_{C,D} \rightarrow_A Z_S = \Psi_{G1,A} \rightarrow_M Z + \Psi_{A,B} \rightarrow_M Z + \Psi_{C,D} \rightarrow_M Z \qquad (4.4)$$

This equation states that the source trace impedance attribute of Segment C-D is comprised of the sum of the impedance methods of Segments G1-A, A-B, and C-D. Since the overall figure of the radialized model is visible, Equation (4.4) can be written. Note, as was discussed in Chapter 3, that the process of determining the source trace impedance is within a specialized outward trace and not a closed form equation.

Now, back to Figure 4.5. A change in the current $\Psi_{C,D} \rightarrow_A I_{Comp}$ causes a voltage change proportional to the series source impedance. That is:

$$\Delta\left(\Psi_{C,D} \rightarrow_A V_{Out}\right) = \Delta\left(\Psi_{C,D} \rightarrow_A I_{Comp}\right) * \Psi_{C,D} \rightarrow_A Z_S \qquad (4.5)$$

This equation shows how a change in the compensation current for $\Psi_{C,D}$ causes a predictable change in outward voltage. Of course, the outward voltage of other Segments are effected by the change in compensation current at this particular Segment. That change in outward voltage of other Segments cannot be predicted from within $\Psi_{C,D}$. In fact, if other Segments with access to the change in compensation current at $\Psi_{C,D}$ cannot predict their own change in voltage. This type of predictability requires an admittance matrix which would be created from the connection and impedance of all Segments in the model. This type of matrix is characteristic of traditional load flow methods like Gauss-

Seidel and Newton-Raphson. As an example, the next section presents a method for fulfilling the purpose of the Affiliation object if this matrix were available.

4.2.2 An Example Non-Object-Oriented Affiliation Object Method

The method presented in this section is an introduction to the Target Voltage method. The method of this section does not support the object-oriented design of Segflow because the method depends on network information that is not available to Affiliation objects. If a Affiliation object had access to a global admittance matrix then the compensation currents could be found so that the outward voltages of all member Segments matched with a single inward and outward trace following the operation of the Affiliation object. Look at the Affiliation object formed by the Segments having outward names of "D" in the six bus system of Figure 4.2(C) :

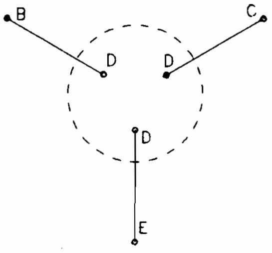


Figure 4.6 - The Degree-3 Affiliation object “D” of the Six Bus System

The dashed line in the above figure corresponds to the Affiliation object, Λ_D, binding the three Segments. If we could to make the outward voltages of $\Psi_{B,D}$, $\Psi_{C,D}$, and $\Psi_{E,D}$ become the same value then the solution of the radialized system would match the solution of the network containing the node "D". A global algorithm could be made to determine the impedance matrix from the perspective of the loop's member Segments. Since interest lies only in the changes in the outward voltage of these Segments with respect to changes in compensation current, the system could be written like this:

$$\left\{\begin{matrix} \Delta\left(\Psi_{B,D}\rightarrow_A V_{Out}\right) \\ \Delta\left(\Psi_{C,D}\rightarrow_A V_{Out}\right) \\ \Delta\left(\Psi_{E,D}\rightarrow_A V_{Out}\right) \end{matrix}\right\} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} \left\{\begin{matrix} \Delta\left(\Psi_{B,D}\rightarrow_A I_{Comp}\right) \\ \Delta\left(\Psi_{C,D}\rightarrow_A I_{Comp}\right) \\ \Delta\left(\Psi_{E,D}\rightarrow_A I_{Comp}\right) \end{matrix}\right\} \qquad (4.6)$$

The diagonal terms of the above impedance matrix are self-impedance values and the others are mutual-impedance values. To simulate the node voltage, changes are going to be made to the compensation currents of the three Segments. Two sets of constraints can be written. First, the new outward voltage for each Segment brought about by the change in compensation current should be the same.

$$\begin{gathered}\Psi_{B,D}\rightarrow{}_{A}V_{Out}-\Delta\left(\Psi_{B,D}\rightarrow{}_{A}V_{Out}\right)= \\ \Psi_{C,D}\rightarrow{}_{A}V_{Out}-\Delta\left(\Psi_{C,D}\rightarrow{}_{A}V_{Out}\right)= \\ \Psi_{E,D}\rightarrow{}_{A}V_{Out}-\Delta\left(\Psi_{E,D}\rightarrow{}_{A}V_{Out}\right)= \\ \Lambda_{D}\rightarrow{}_{A}V_{Node}\end{gathered}\tag{4.7}$$

The symbol $\Lambda_{D}\rightarrow{}_{A}V_{Node}$ is the Affiliation object voltage to which each Segment is driving the value of its outward voltage. In the second constraint, the sum of the changes in the compensation currents must total zero so that no net change is incurred on the loading of the system:

$$\Delta\left(\Psi_{B,D}\rightarrow{}_{A}I_{Comp}\right)+\Delta\left(\Psi_{C,D}\rightarrow{}_{A}I_{Comp}\right)+\Delta\left(\Psi_{E,D}\rightarrow{}_{A}I_{Comp}\right)=\vec{0}\tag{4.8}$$

The last three sets of equation can be combined to form the system:

$$\begin{Bmatrix}\Psi_{B,D}\rightarrow{}_{A}V_{Out} \\ \Psi_{C,D}\rightarrow{}_{A}V_{Out} \\ \Psi_{E,D}\rightarrow{}_{A}V_{Out}\end{Bmatrix}=\begin{bmatrix}Z_{11}-Z_{13} & Z_{12}-Z_{13} & 1 \\ Z_{21}-Z_{23} & Z_{22}-Z_{23} & 1 \\ Z_{31}-Z_{33} & Z_{32}-Z_{33} & 1\end{bmatrix}\begin{Bmatrix}\Delta\left(\Psi_{B,D}\rightarrow{}_{A}I_{Comp}\right) \\ \Delta\left(\Psi_{C,D}\rightarrow{}_{A}I_{Comp}\right) \\ \Lambda_{D}\rightarrow{}_{A}V_{Node}\end{Bmatrix}\tag{4.9}$$

When given the initial outward voltage of member Segments, the matrix can be inverted and used to solve for the change in compensation current required of each Segment and the Affiliation object voltage that will result from the alterations. The value of the third compensation current can be found with Equation (4.8).

This technique of finding the precise value of compensation voltage and resulting Affiliation object voltage depends completely upon the reduced system impedance matrix being available to the Affiliation object. In the Segflow, however, this information cannot be made available because a global algorithm for collecting and consolidating impedance information is not available. The best information available to Affiliation objects is the source trace impedance from each of the Segments. This information can be used to iteratively find the node voltage that was found previously in one step. This information can be used in the Target Voltage method which is presented next.

4.3 The Target Voltage method

The Target Voltage method is the technique within Affiliation objects that is used to find the values of compensation current needed to add to all member Segments so that deviation of outward voltage is reduced following a voltage and current trace. The Target Voltage method depends upon information that can be requested from Segments -- outward voltage and source trace impedance. Once compensation currents are found with the Target Voltage method, a Affiliation object sends messages to member Segments with regard to the compensation current alterations. The convergence capabilities of Segflow and the Target Voltage method are heuristic in nature when more than one Affiliation object is present.

The method is accurate when no path coupling is present between the member Segments of the Affiliation object. The use of the Target Voltage method has been tested on a variety of systems with good results. The method is developed by example from the degree-3 Affiliation object below.

4.3.1 The Target Voltage method development on a degree-3 Affiliation object

Shown below is an Affiliation object like the degree-3 one used earlier in this chapter. The Affiliation object is represented with a dashed oval:

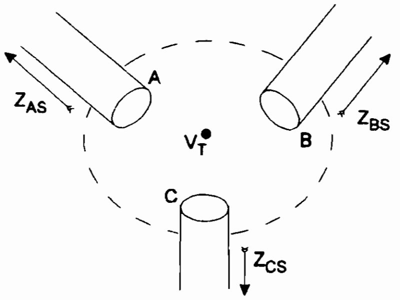


Figure 4.7 - General Degree-3 Affiliation Object

The Segments in the Affiliation object are Ψ_A, Ψ_B, and Ψ_C. The source trace impedances corresponding to each Segment are labeled Z_{AS}, Z_{BS}, and Z_{CS} and correspond to Segment source trace impedances $\Psi_A \to_A Z_S$, $\Psi_B \to_A Z_S$, and $\Psi_C \to_A Z_S$. Compensation current will be added to the outward ends of these Segments and will be noted as explained in Chapter 2. The target voltage is a weighted average of all Affiliation object member Segments and is defined as follows:

Definition 4.1

The target voltage is the anticipated voltage to which the outward voltage of all members of a Affiliation object will converge. The target voltage is based on:

1. Loop Segment voltages from the solution of the linearized radialized system model.
2. Segment source trace impedances.

The target voltage is represented as a dot and labeled V_T in Figure 4.7. Assuming that the target voltage is known, the change in a the compensation current of Ψ_A needed to make that Segment's outward voltage meet the target voltage can be approximated as:

$$\Delta\left(\Psi_A \to_A I_{Comp}\right) = \frac{\Psi_A \to_A V_{Out} - V_T}{\Psi_A \to_A Z_S} \qquad (4.10)$$

For the three Segments in the figure above, we have:

$$\Delta\left(\Psi_A \rightarrow_A I_{Comp}\right) = \frac{\Psi_A \rightarrow_A V_{Out} - V_T}{\Psi_A \rightarrow_A Z_S} \tag{4.11}$$

$$\Delta\left(\Psi_B \rightarrow_A I_{Comp}\right) = \frac{\Psi_B \rightarrow_A V_{Out} - V_T}{\Psi_B \rightarrow_A Z_S}$$

$$\Delta\left(\Psi_C \rightarrow_A I_{Comp}\right) = \frac{\Psi_C \rightarrow_A V_{Out} - V_T}{\Psi_C \rightarrow_A Z_S}$$

Adjustments must be made to the compensation currents of Ψ_A, Ψ_B, and Ψ_C such that there is no net change to the loading of the system.

$$\Delta\left(\Psi_A \rightarrow_A I_{Comp}\right) + \Delta\left(\Psi_B \rightarrow_A I_{Comp}\right) + \Delta\left(\Psi_C \rightarrow_A I_{Comp}\right) = \vec{0} \tag{4.12}$$

Equation (4.11) and Equation (4.12) can be combined to get:

$$\frac{\Psi_A \rightarrow_A V_{Out} - V_T}{\Psi_A \rightarrow_A Z_S} + \frac{\Psi_B \rightarrow_A V_{Out} - V_T}{\Psi_B \rightarrow_A Z_S} + \frac{\Psi_C \rightarrow_A V_{Out} - V_T}{\Psi_C \rightarrow_A Z_S} = \vec{0} \tag{4.13}$$

Now we can solve for the target voltage in terms of the known Segment outward voltages and source trace impedances:

$$V_T = \frac{V_A Z_{SB} Z_{SC} + V_B Z_{SA} Z_{SC} + V_C Z_{SA} Z_{SB}}{Z_{SA} Z_{SB} + Z_{SB} Z_{SC} + Z_{SA} Z_{SC}} \tag{4.14}$$

Where:

$$V_x = \Psi_x \rightarrow_A V_{Out}$$

$$Z_{Sx} = \Psi_x \rightarrow_A Z_S$$

The compensation currents from Equation (4.11) can be applied to all member Segments, Ψ_A, Ψ_B, and Ψ_C, with this closed form solution of the target voltage.

4.3.2 General Target Voltage method

Consider an Affiliation object having N member segments. The Affiliation object receives information about the outward voltage and series impedance of each Segment. The Affiliation object also has the capability to add constant current load to the outward end of each Segment through compensation current information passing. Shown below is that Affiliation object:

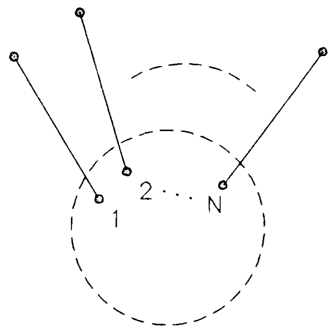


Figure 4.8 - A Degree-N Affiliation Object

The Affiliation objects has access to member Segments $\Psi_1, \Psi_2, \cdots, \Psi_N$ and their respective source trace impedances $\Psi_1 \rightarrow_A Z_S$, $\Psi_2 \rightarrow_A Z_S, \cdots, \Psi_N \rightarrow_A Z_S$. The outward voltages of the Segments represent the solution of the radialized system of constant current loads. Those voltages are also respectively $\Psi_1 \rightarrow_A V_{Out}$, $\Psi_2 \rightarrow_A V_{Out}, \cdots, \Psi_N \rightarrow_A V_{Out}$. The operation of the Affiliation object is to meet two goals.

- Add constant current load to Segments so that their voltages reach some common value called the target voltage ${}_T$.
- The net change in system loading due to Affiliation object operation is zero.

Since the circuit model is radial and presumed linear, a load current change to exactly one Segment, Ψ_k, in the Affiliation object causes a predictable change in that Segment's outward voltage:

$$\Delta(\Psi_k \rightarrow_A V_{Out}) = \Delta(\Psi_k \rightarrow_A I_{Comp}) * \Psi_k \rightarrow_A Z_S \qquad (4.15)$$

When changes are made to multiple member Segments of the Affiliation objects, the equality above can be loosely held:

$$\Delta(\Psi_k \rightarrow_A V_{Out}) \approx \Delta(\Psi_k \rightarrow_A I_{Comp}) * \Psi_k \rightarrow_A Z_S \qquad k = 1,2,\cdots N \qquad (4.16)$$

The desired result of changes in compensation current to Affiliation object member Segments is to bring them to the common target voltage. This implies:

$$\Delta(\Psi_k \rightarrow_A V_{Out}) = \Psi_k \rightarrow_A V_{Out} - V_T \qquad k = 1,2,\cdots N \qquad (4.17)$$

Equation (4.15) can be arranged and combined with Equation (4.17) to get:

$$\Delta(\Psi_k \rightarrow_A I_{Comp}) \approx \frac{\Psi_k \rightarrow_A V_{Out} - V_T}{\Psi_k \rightarrow_A Z_S} \qquad k = 1,2,\cdots N \qquad (4.18)$$

The total change in compensation current within the Affiliation object is constrained to zero:

$$\sum_{k=1}^{N} \Delta\left(\Psi_k \rightarrow_A I_{Comp}\right) = \vec{0} \tag{4.19}$$

Equation (4.18) can be substituted into Equation (4.19) to get:

$$\sum_{k=1}^{N} \frac{\Psi_k \rightarrow_A V_{Out} - V_T}{\Psi_k \rightarrow_A Z_S} = \vec{0} \tag{4.20}$$

The denominator of the fractions can be removed by multiplying both the denominator and numerator by the product of all source trace impedances, $\prod_{j=1}^{N} \Psi_j \rightarrow_A Z_S$:

$$\sum_{k=1}^{N} \left(\Psi_k \rightarrow_A V_{Out} - V_T\right) \prod_{\substack{j=1 \\ \neq k}}^{N} \Psi_j \rightarrow_A Z_S = \vec{0} \tag{4.21}$$

The value for the target voltage can now be isolated:

$$V_T = \frac{\displaystyle\sum_{k=1}^{N} \Psi_k \rightarrow_A V_{Out} \prod_{\substack{j=1 \\ \neq k}}^{N} \Psi_j \rightarrow_A Z_S}{\displaystyle\sum_{k=1}^{N} \prod_{\substack{j=1 \\ \neq k}}^{N} \Psi_j \rightarrow_A Z_S} \tag{4.22}$$

The target voltage can now be found from within a Affiliation object having any number of member Segments. Equation (4.18) used with Equation (4.22) determines compensation currents for all member segments.

4.3.3 Deriving the Target Voltage method With a Z Matrix Approach

The Target Voltage method was derived using information available to the member Segments of Affiliation objects. The effects of a compensation current change on a single Segment was determined. This behavior was used as an approximate behavior for all Segments in a Affiliation object when multiple changes to compensation current was made. When the total change to compensation current within a Affiliation object was constrained to zero an expression for the target voltage could be found. This section takes the Affiliation object impedance matrix used in Section 4.2.2 and solves it within an Affiliation object using a Z matrix. The result is very similar to the Target Voltage method results.

The Segflow notation from Chapter 2 that has been used in derivations up to this point will now be relaxed. The following assignments are made:

$$\begin{aligned} \Delta V_x &= \Psi_x \rightarrow_A V_{Out} \\ \Delta I_x &= \Psi_x \rightarrow_A I_{Comp} \end{aligned} \qquad (4.23)$$

This notation remains in effect for Section (4.3.3.1) and Section (4.3.3.2).

4.3.3.1 The Z Matrix Applied to Flows

The radialized network appears as a matrix of self- and mutual- impedances from within an Affiliation object. Here is the radialized network with respect to a degree-3 Affiliation object:

$$\begin{Bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta V_3 \end{Bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{12} & Z_{22} & Z_{23} \\ Z_{13} & Z_{23} & Z_{33} \end{bmatrix} \begin{Bmatrix} \Delta I_1 \\ \Delta I_2 \\ \Delta I_3 \end{Bmatrix} \qquad (4.24)$$

If we apply the following constraints:

$$\Delta I_1 + \Delta I_2 + \Delta I_3 = \bar{0} \qquad (4.25)$$

and

$$\Delta V_i = V_i - V_T$$

then the system can be factored and rewritten as:

$$\begin{Bmatrix} \mathrm{V}_1 \\ \mathrm{V}_2 \\ \mathrm{V}_3 \end{Bmatrix} = \begin{bmatrix} Z_{11} - Z_{13} & Z_{12} - Z_{13} & 1 \\ Z_{12} - Z_{23} & Z_{22} - Z_{23} & 1 \\ Z_{13} - Z_{33} & Z_{23} - Z_{33} & 1 \end{bmatrix} \begin{Bmatrix} \Delta I_1 \\ \Delta I_2 \\ V_T \end{Bmatrix} \qquad (4.26)$$

This matrix equation can now be written as a Gauss-Seidel iteration common in numerical methods[28]. Note that this application of the Gauss-Seidel technique is not directly related to the Gauss-Seidel load-flow common in power system analysis.

$$\begin{aligned} \Delta I_1^{i+1} &= \frac{V_1^i - V_T^i + (Z_{13} - Z_{12})\Delta I_2^i}{Z_{11} - Z_{13}} \\ \Delta I_2^{i+1} &= \frac{V_2^i - V_T^i + (Z_{23} - Z_{12})\Delta I_1^i}{Z_{22} - Z_{23}} \\ V_T^{i+1} &= V_3^i + Z_{33}\left(\Delta I_2^i + \Delta I_1^i\right) \end{aligned} \qquad (4.27)$$

Only the diagonal terms of the impedance matrix in equation (4.24) are known to an Affiliation object. To apply the above result, therefore, the off diagonal terms need to be neglected:

$$\Delta I_1^{i+1} = \frac{V_1^i - V_T^i}{Z_{11}}$$

$$\Delta I_2^{i+1} = \frac{V_2^i - V_T^i}{Z_{22}}$$

$$V_T^{i+1} = V_3^i + Z_{33}\left(\Delta I_2^i + \Delta I_1^i\right) \tag{4.28}$$

Some substitutions can be made in the above expression:

$$\frac{V_3^i - V_T^{i+1}}{Z_{33}} + \frac{V_1^{i-1} - V_T^{i-1}}{Z_{11}} + \frac{V_2^{i-1} - V_T^{i-1}}{Z_{22}} = 0 \tag{4.29}$$

and if we consider $V_T^{i+1} \approx V_T^{i-1} = V_T^i$ then:

$$V_T^i = \frac{V_1^{i-1} Z_{22} Z_{33} + V_2^{i-1} Z_{11} Z_{33} + V_3^i Z_{22} Z_{33}}{Z_{22} Z_{33} + Z_{11} Z_{33} + Z_{22} Z_{33}} \tag{4.30}$$

This expression is very similar to the target voltage value derived earlier and does reach the target voltage in the limit.

This section shows that the Target Voltage method can be derived using a Z matrix approach to solve the radialized system impedance matrix from the perspective of a Affiliation object. In that derivation, negligence is cast upon impedance values indistinguishable from within the Affiliation object. Approximations are also made over iterative intervals to create an expression for $_T^i$ that matches the target voltage.

4.3.3.2 Y Matrix Applied to Node Voltages

The nodal application of the Y matrix does not work well into the object-oriented layout of Segflow. Consider the modal for a radial system with constant current loads:

$$\{I\} = [Y_{Bus}]\{V\} \tag{4.31}$$

The matrix $\{I\}$ represents the current injections into the linear system. $\{V\}$ is the vector of node voltages. The current injection for a particular node can be given by:

$$I_j = \sum_{\substack{k=1 \\ \neq j}}^{N_{Bus}} Y_{jk} V_k + Y_{jj} V_j \tag{4.32}$$

This can be rearranged as:

$$V_j = \frac{1}{Y_{jj}}\left(I_j - \sum_{\substack{k=1 \\ \neq j}}^{N_{Bus}} Y_{jk} V_k \right) \qquad (4.33)$$

This method is applied iteratively at each node and therefore does not benefit from voltage and current trace methods. The equation above uses local impedances (admittances) only. The impedances of adjacent Segments probably do no represent the relative relationships of the source trace impedance. The equation above would tend to draw current from adjacent Segments with low impedances. Those Segments may, however, have very large upstream impedances.

Finally, there is no way to implement the suggested target voltage using the above equation. The nodal version of the Y matrix approach has to be used for all node voltages. In that case, users are forced to create their models to fit into the YBus admittance matrix. There is minimal use for object-oriented methods in that case and little modeling flexibility is achieved.

4.4 The Affiliation object Leader

This section shows that as an Affiliation object becomes more isolated from a systems source, the target voltage approaches the average of the outward voltage of the Affiliation object's member Segments. Shown below is the radialized six bus system presented in Chapter 3 and discussed earlier in this chapter. The Segments and their source traces of the "D" Affiliation object are highlighted:

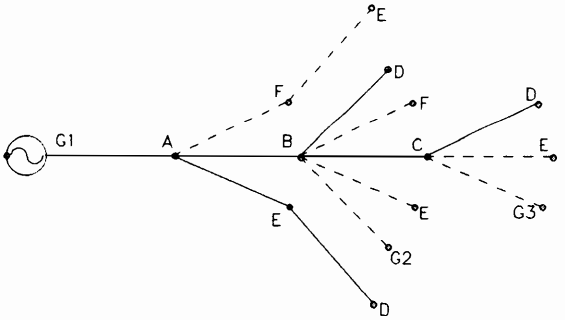


Figure 4.9 - Member Segment Traces for a Degree-3 Affiliation Object

It can be seen from this figure and verified with general network theory that any degree-3 Affiliation object can be represented as follows:

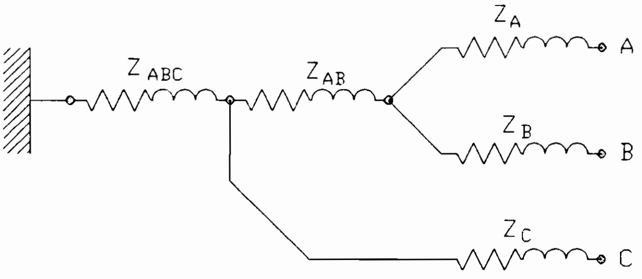


Figure 4.10 - Circuit with Degree-3 Affiliation Object

The outward nodes of three Segments, Ψ_A, Ψ_B, and Ψ_C correspond to nodes A, B, and C above. These assignments can be made from any degree-3 Affiliation object. The picture demonstrates that the source traces for Segments, Ψ_A, Ψ_B, and Ψ_C eventually join. The compensation currents of Ψ_A and Ψ_B join and travel through impedance Z_{AB}. That current combines with the load current of Ψ_C and travels through Z_{ABC} to the source. As the Target Voltage method is used to bring the outward voltages of Segments, Ψ_A, Ψ_B, and Ψ_C together, the constant current loads on those Segments change. Since the changes in those load currents always add to zero then the current through Z_{ABC} never changes due to action of the Affiliation object.

$$\Delta\left(\Psi_A \rightarrow_A I_{Comp}\right) + \Delta\left(\Psi_B \rightarrow_A I_{Comp}\right) + \Delta\left(\Psi_C \rightarrow_A I_{Comp}\right) = \vec{0} \qquad (4.34)$$

So

$$\Delta I_{ABC} = \vec{0}$$

The source trace impedances of Ψ_A, Ψ_B, and Ψ_C are used. Those impedances can be written as follows:

$$\begin{aligned} \Psi_A \rightarrow_A Z_S &= Z_A + Z_{AB} + Z_{ABC} \\ \Psi_B \rightarrow_A Z_S &= Z_B + Z_{AB} + Z_{ABC} \\ \Psi_C \rightarrow_A Z_S &= Z_C + Z_{ABC} \end{aligned} \qquad (4.35)$$

So even though Z_{ABC} has no effect voltage/current relationship within the Target Voltage method controlled Affiliation object, it is still used in determining the target voltage since:

$$V_T = \frac{V_A Z_{SB} Z_{SC} + V_B Z_{SA} Z_{SC} + V_C Z_{SA} Z_{SB}}{Z_{SA} Z_{SB} + Z_{SB} Z_{SC} + Z_{SA} Z_{SC}} \qquad (4.36)$$

Where:

$$V_x = \Psi_x \rightarrow_A V_{Out}$$

$$Z_{Sx} = \Psi_x \rightarrow_A Z_S$$

Now, lets consider the effect of isolating the Affiliation object from the source by increasing the impedance of Z_{ABC}:

$$Z_{ABC}^{New} = Z_{ABC}^{Old} + \Delta Z \tag{4.37}$$

Put the new value of Z_{ABC} into Equation (4.36):

$$V_T = \frac{V_A(Z_{SB} + \Delta Z)(Z_{SC} + \Delta Z) + V_B(Z_{SA} + \Delta Z)(Z_{SC} + \Delta Z) + V_C(Z_{SA} + \Delta Z)(Z_{SB} + \Delta Z)}{(Z_{SB} + \Delta Z)(Z_{SC} + \Delta Z) + (Z_{SA} + \Delta Z)(Z_{SC} + \Delta Z) + (Z_{SA} + \Delta Z)(Z_{SB} + \Delta Z)} \tag{4.38}$$

The function can now be factored with respect to powers of ΔZ to get:

$$V_T = \frac{\begin{array}{l} V_A Z_{SB} Z_{SC} + V_B Z_{SA} Z_{SC} + V_C Z_{SA} Z_{SB} \\ +\Delta Z\left[V_A(Z_{SB} + Z_{SC}) + V_B(Z_{SA} + Z_{SC}) + V_C(Z_{SA} + Z_{SB})\right] \\ +\Delta Z^2(V_A + V_B + V_C) \end{array}}{Z_{SA} Z_{SB} + Z_{SA} Z_{SC} + Z_{SB} Z_{SC} + \Delta Z(2Z_{SA} + 2Z_{SB} + 2Z_{SC}) + 3\Delta Z^2} \tag{4.39}$$

If ΔZ is small then the target voltage above has the same value as the one in equation (4.36). When the leader grows, however, and ΔZ becomes large, the target voltage reaches a limit at the average of the member Segment outward voltages.

$$\lim_{\Delta Z \to \infty} = \frac{V_A + V_B + V_C}{3} \tag{4.40}$$

Where:

$$V_x = \Psi_x \rightarrow {}_A V_{Out}$$

This implies that Affiliation objects attached to long leaders have insufficient information for the target voltage to be anything other than the simple average of the voltages of the member Segments.

4.4.1 Using the Average

The previous section demonstrated that a sizable leader pushed the target voltage to the average of the Affiliation object's member Segment outward voltages. It would seem, therefore, that the voltage average would be a sensible alternative to the target voltage. It has been experimentally shown that in many cases, the average value of loop voltages and the target voltage converge together. Here is an example:

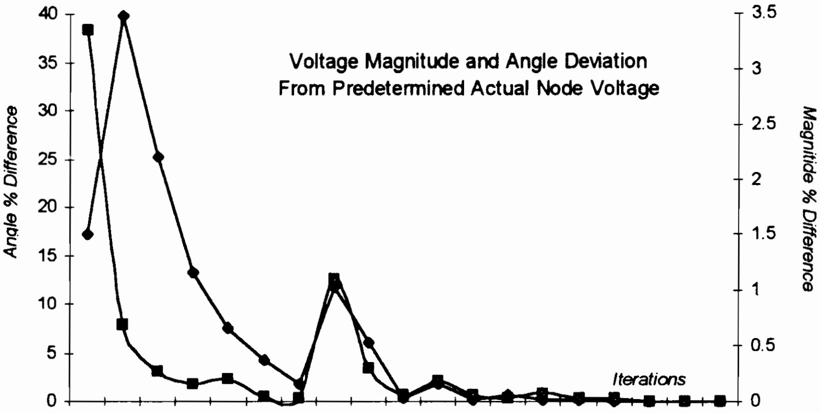


Figure 4.11- Convergence Using Averaging Technique

The error in voltage magnitude and angle of the average of member Segment outward voltages diminishes quickly. The bump in the middle of the graph is caused by responses of other Affiliation objects in the six bus system. (The six bus system has two other Affiliation objects whose behavior affects the voltage and angle of member Segments in the Affiliation object of Figure 4.11) Although the methods seem comparable in this example, there are two reasons why the average voltage cannot be used as a target.

First, the target voltage has, at it foundation, provisions to make the sum of all current adjustments add to zero. The voltage found with the Target Voltage method will always produce a zero net change in load when applied. If the average voltage is used, some sort of method for balancing the currents will have to be introduced. This provision will more than likely violate the coupling between the members of the Affiliation object.

Second, the voltage averaging technique does not incorporate the trace impedance information. In the cases in which the average is very close to the target voltage with long leaders, this is not a problem. Here, however, is a behavior plot for a Affiliation object that is close to the source:

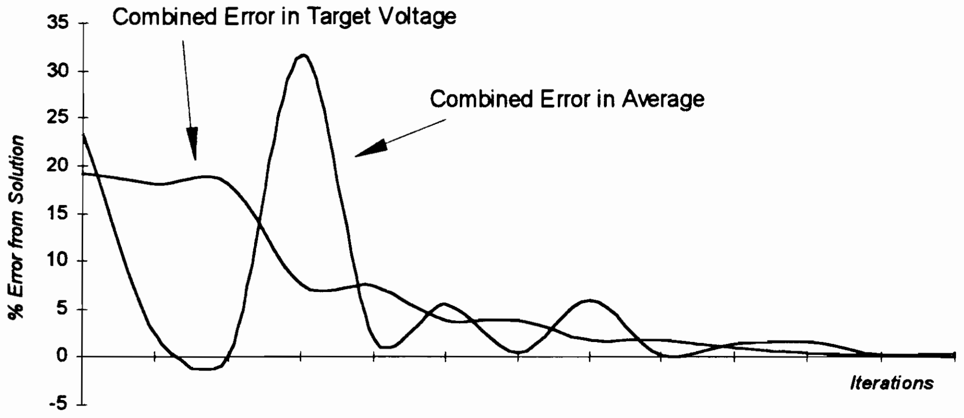


Figure 4.12 - Target Voltage Method vs. Averaging Technique

Notice how the error in target voltage is steadily decreasing to the voltage of the Affiliation object's identity node. The average oscillates and in many cases diverges the load-flow. This strengthens the claim that there is no foundation for using the average voltages of loop Segments. I used some heuristic methods in experiments with averages and found nothing to challenge the efficiency, simplicity, and robustness of the Target Voltage method.

4.5 An Example of Applying the Target Voltage Method

A four bus system was created and loaded so that an increase of one MW or one MVar would create an system with no load-flow solution. The system was then put into the Segflow environment and radialized. The figure depicts the radialized model, line Segment model impedances and their loading:

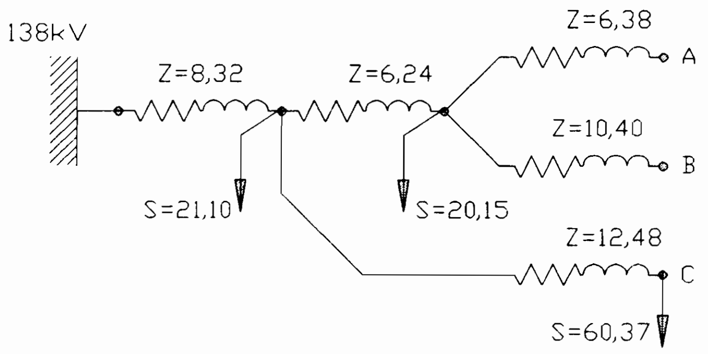


Figure 4.13 - Example Circuit With Degree-3 Affiliation Object

All line impedances are in Ohms, all loads are in kVA, and all voltages are in kV. Segments A, B, and C are members of a Affiliation object named 'Bus 3'. The 'Bus 3' load was placed onto Segment C because Segment C has the lowest source trace impedance of any of the loop's member Segments.

An outward trace is performed and essentially propagates the source voltage throughout the circuit as well as the source trace impedance for each Segment. The loads are also evaluated on the Segment voltages:

Segment	Source Trace Z (Ω)	Linear Load (Amps)	Outward Voltage (kV)
A	(20,94)	-	116.53∠-11.21
B	(24,94)	-	116.53∠-11.21
C	(24,96)	294.91∠-31.66	106.49∠-20.61

The standard deviation of the voltages is 0.0869. The target voltage can be calculated using the equation developed in the previous chapter:

$$\begin{aligned} V_T &= \\ &\frac{(116.51\angle-11.21)(24,96)(20,80)+(116.51\angle-11.21)(20,94)(20,80)+(106.5\angle-20.61)(20,94)(24,96)}{(24,96)(20,80)+(20,94)(20,80)+(20,94)(24,96)} \\ &= 112.5\angle-14.53 \end{aligned} \tag{4.41}$$

The compensation currents are:

$$\begin{aligned} \Delta I_A &= \frac{(116.5\angle-11.21)-(112.5\angle-14.53)}{(20,94)} = 80.5\angle-31.9 \\ \Delta I_B &= \frac{(116.5\angle-11.21)-(112.5\angle-14.53)}{(24,96)} = 78.2\angle-29.9 \\ \Delta I_C &= \frac{(106.5\angle-20.6)-(112.5\angle-14.53)}{(20,80)} = 158.6\angle149.1 \end{aligned} \tag{4.42}$$

The compensation currents are signaled into the Affiliation object's member Segments and a current and voltage trace is initiated. Here is a list of target voltage values and standard deviations for a few more iterations.

#	Target Voltage	Standard Deviation
1	111.77∠-15.21	0.0248
2	111.56∠-15.46	0.0071
3	111.51∠-15.46	0.0020
4	111.48∠-15.48	0.0002

So the standard deviation among the Affiliation object's member Segments has reached a very small value after a few iterations of the Target Voltage method.

4.6 Implementation of the Affiliation Objects

In a system with multiple Affiliation objects, focus is always given to the Affiliation object with the worst deviation among its member Segments. The chart below reflects the general solution process which will be discussed in more detail in Chapter 6:

DF036.XLS

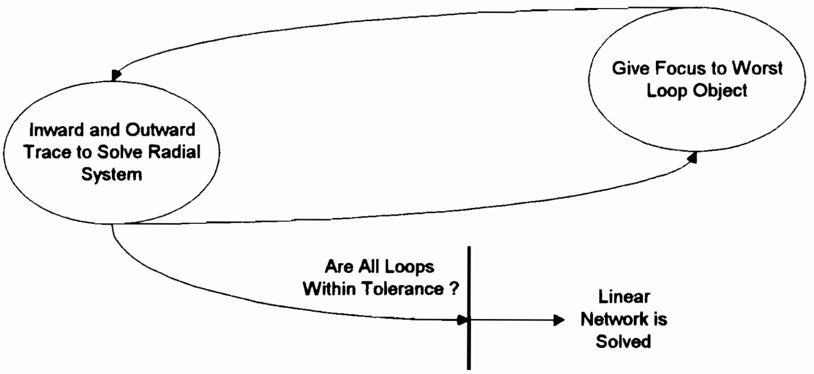


Figure 4.14 - Processing Loops to Get Linear Network Solution

The trace methods discussed in Chapter 2 are used to find the solution of the radialized system having constant current loads. At that point, the Affiliation object having the worst deviation in outward voltage between its member Segments is focused. That Affiliation object applies the Target Voltage Method to its member Segments. A few statements can be made:

- It seems appropriate to expect the Target Voltage method to converge in systems with multiple Affiliation objects and an existing solution. The next chapter shows that systems with single Affiliation objects have some properties that suggest a tendancy toward convergence. Chapter six states that the Affiliation having the worst spread in member Segment outward voltage is always focused. This may produce an algorithm that maintains some of those properties leading toward convergence.
- If iterations in a system with multiple Affiliation objects is destructive the convergence of one of those Affiliation objects then it will, as its deviation increases, become the focused Affiliation object. As that Affiliation object is focused, its deviation will decrease. If this is destructive to the deviation of another, particular, Affiliation object then an oscillation may be sustained in trying to reach the solution of the linear network. If this oscillation can be created then it will be

detected in the Segflow. A minor adjustment could be made to the behavior of one Affiliation object to bypass this oscillatory state.

4.7 Loops Objects in Action

Systems with multiple Affiliation objects are not explicitly analyzed in this dissertation. Extensive simulations have been performed on systems ranging from half-a-dozen busses to 300 busses. In each case, the solution has been found when it existed. When finding the solution of each power system model, the solution of various linear networks was found many times. The Target Voltage method was applied within Affiliation objects hundreds of times for each simulation. The method worked on the nine test cases studied (Chapter 6). Here is a graphical depiction of the transition of Target Voltage state during the solution of a test system:

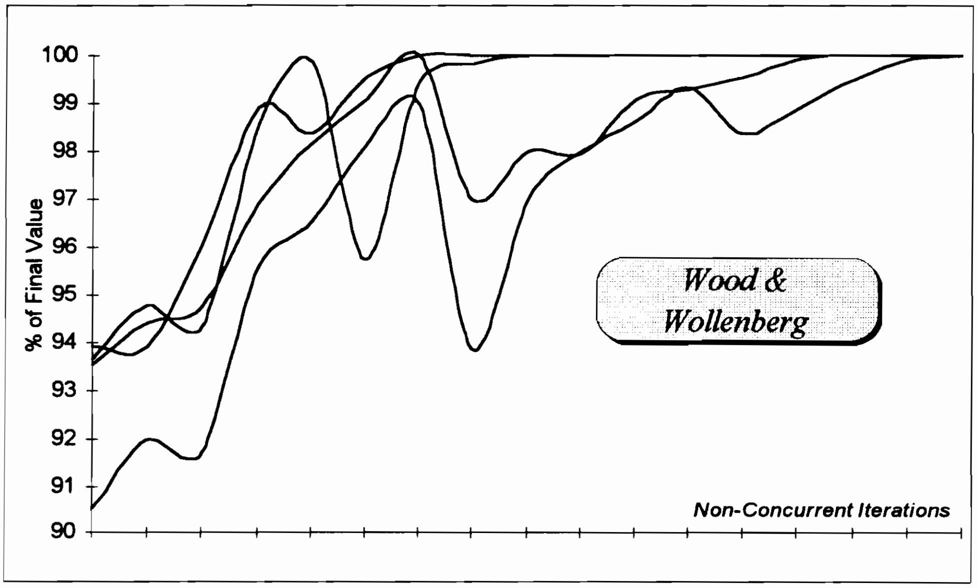


Figure 4.15 - Affiliation object Convergence Curves

Shown above are the target voltages of the four Affiliation objects in a test system. Two of the Affiliation objects are degree-2 and the other two are degree-3. Embedded in the figure are the re-evaluations of load or load linearizations which are a part of the total solution of the power system model. The iterations are non-concurrent. That is, the Affiliation objects were placed on an independent axis representing the local Affiliation object iteration number. Causes and effects seen in the pits and peaks are therefore skewed.

4.7.1 Implications of Loop Handling Methods

Using Affiliation objects supports object independence and maintains the over-all object oriented design of the power system model. The loop coupling scheme allows power system models to remain radial with respect to the swing bus. The Affiliation object also hides the radial modeling scheme from the user by providing loop flow information.

Chapter 5

Analysis of the Target Voltage Method

5. Analysis of the Target Voltage Method

The Target Voltage method presented in Chapter 4 is a method employed by Affiliation objects to bring the outward voltage of their member Segments to the same value. This goal is achieved by altering the compensation current of member Segments. The amount of compensation current change requested of each Segment is governed by the Affiliation object by employing the Target Voltage method.

A system of Segments, trace methods, Affiliation objects, and the Target Voltage method can be used to determine the solution of a power system model with constant current loads. The explicit proof of convergence for this system cannot be found for a number of reasons:

- First, Segment models are unknown. Users can build models using any combination of linear, or nonlinear approaches with continuous or discontinuous functions.
- Second, the Affiliation object with the worst deviation in member Segment outward voltage is always evaluated by Segflow. (This was discussed briefly in Chapter 4 and will be presented in more detail in Chapter 6.) Determining the convergence characteristics of a system with multiple Affiliation object is therefore nearly impossible without actually running the Segflow calculations.
- Finally, a closed form representation of the Target Voltage method cannot be created because the method as well as current and voltage traces are compositions of functions. This is unlike traditional methods which solve large systems of simultaneous equations. For the following illustration, assume that the outward voltage and demand current can be defined with these two functions:

$$
\begin{array}{lll}
 & \mathit{Where} & \\
V_{Out} = V(I_{In}, S, V_{Feed}) & I_{In} = \mathit{Current\ Entering} & V_{Out} = \mathit{Outward\,Voltage} \\
I_{In} = I(V_{Out}, S, I_{Dem}) & S = \mathit{State} & I_{Dem} = \mathit{Total\,Current\ Dem} \\
 & V_{Feed} = \mathit{Voltage\ of\ Feeder} &
\end{array}
\qquad (5.1)
$$

Then voltage and current traces over N Segments are defined by the two compositions listed below:

$$
\begin{gathered}
V_1(I_1, S_1, V_2(I_2, S_2, V_3(I_3, S_3, V_4(\cdots = 0 \\
I_N(V_N, S_N, I_{N-1}(V_{N-1}, S_{N-1}, I_{N-2}(V_{N-2}, S_{N-2}, I_{N-3}(\cdots = 0
\end{gathered}
\qquad (5.2)
$$

This type of mathematical representation is difficult to manage and is avoided in this chapter.

Analysis can be performed to help understand the stability, consistency, and speed of the Target Voltage method. Four sets of analysis are presented in this chapter. Each analysis concerns a radialized model with exactly one Affiliation object and Segments modeling series impedances. As always, Affiliation objects and thus the Target Voltage method act on radialized systems with constant current loads. The analysis is presented to show that in some situations, the Target Voltage method does have some characteristics which might be beneficial to to determining the network solution from the Segflow radialized model.

5.1 Notation For Chapter 5

The following notation changes are made for this chapter to allow more clarity within the rather complex formulations:

Segflow Notation	Meaning	Chapter 5 Representation
$\Psi_A \to {}_A V_{Out}$	Outward Voltage	V_A
$\Psi_A \to {}_A Z_S$	Source Trace Z	Z_{SA}
$\Psi_A \to {}_M Z$	Segment Z	Z_A
$\Psi_A \to {}_A Z_S \cap \Psi_B \to {}_A Z_S$	A & B Common Z	Z_{AB}
$\Psi_A \to {}_A I_{Comp}$	Compensation Current	I_A

Table 5.1

A matrix representation is used to simulate the behavior of series impedance Segments during a single inward and outward trace. The matrix representation is developed by example from the following figure:

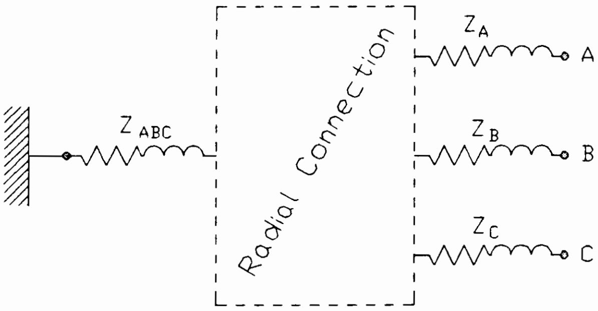


Figure 5.1 - View From an Affiliation object

An Affiliation object encapsulates Segments with outward nodes A, B, and C. Within the dashed rectangle lies an unknown radial connection which converges through Z_{ABC} and to the source. Changes in the compensation current of the member Segments (noted as Z_A, Z_B, and Z_C) causes voltage changes predictable with knowledge of the radial connections as follows:

$$\begin{aligned} -\Delta V_A &= \Delta I_A(Z_A + Z_{AB} + Z_{AC} + Z_{ABC}) + \Delta I_B(Z_{AB} + Z_{ABC}) + \Delta I_C(Z_{AC} + Z_{ABC}) \\ -\Delta V_B &= \Delta I_A(Z_{AB} + Z_{ABC}) + \Delta I_B(Z_B + Z_{AB} + Z_{BC} + Z_{ABC}) + \Delta I_C(Z_{BC} + Z_{ABC}) \\ -\Delta V_C &= \Delta I_A(Z_{AC} + Z_{ABC}) + \Delta I_B(Z_{BC} + Z_{ABC}) + \Delta I_C(Z_C + Z_{AC} + Z_{BC} + Z_{ABC}) \end{aligned} \tag{5.3}$$

By the design of the Target Voltage method, Affiliation objects are not allowed to induce a net load change into the power system model. The following equation is therefore implied:

$$\Delta I_A + \Delta I_B + \Delta I_C = 0 \tag{5.4}$$

Applying this constraint to Equation (5.3) yields:

$$\begin{Bmatrix} \Delta V_A \\ \Delta V_B \\ \Delta V_C \end{Bmatrix} = -\begin{bmatrix} Z_A + Z_{AB} + Z_{AC} & Z_{AB} & Z_{AC} \\ Z_{AB} & Z_B + Z_{AB} + Z_{BC} & Z_{BC} \\ Z_{AC} & Z_{BC} & Z_C + Z_{AC} + Z_{BC} \end{bmatrix} \begin{Bmatrix} \Delta I_A \\ \Delta I_B \\ \Delta I_C \end{Bmatrix} \tag{5.5}$$

Study of the intersections contained in each of the diagonal terms will allow the following simplification:

$$\begin{Bmatrix} \Delta V_A \\ \Delta V_B \\ \Delta V_C \end{Bmatrix} = -\begin{bmatrix} Z_{SA} & Z_{AB} & Z_{AC} \\ Z_{AB} & Z_{SB} & Z_{BC} \\ Z_{AC} & Z_{BC} & Z_{SC} \end{bmatrix} \begin{Bmatrix} \Delta I_A \\ \Delta I_B \\ \Delta I_C \end{Bmatrix} \tag{5.6}$$

This formulation can be repeated for Affiliation object 2, 4, or more member Segments. The diagonal terms of this matrix are known by an Affiliation object's member Segments and thus by the Affiliation object for use in the Target Voltage method. The off-diagonal terms are not known by Affiliation objects. Equation (5.6) will be used to simulate the effect of a single current and voltage trace to solve a radial network of series impedances and constant current loads.

5.2 Behavior of Single Degree-2 Affiliation Object

This section evaluates the behavior of a single Affiliation object of degree-2. The object exists in a Segflow radialized model comprised of all linear Segments. All loads are constant current. The evaluation results in the eigenvalues of the iteration matrix for the Affiliation object. Here is a representation of the subject system:

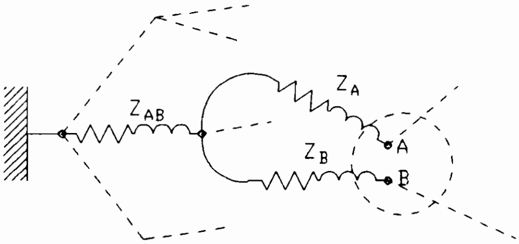


Figure 5.2 - A Radialized Model with a Single Degree-2 Affiliation Object

The stray dashed lines show that many Segments can exist in the system and are not important to the analysis of this section. There is, for this analysis, only one Affiliation object and it is represented by the dashed circle. The member Segments are labeled with an **A** for Ψ_A and a **B** for Ψ_B. Representation of current, voltage, impedance and source trace impedance values are consistent with Section 5.1.

The operation of the radial system with respect to changes in the compensation current of Ψ_A and Ψ_B is given by this equation, also from Section 5.1:

$$\begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i+1} = \begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i} - \begin{bmatrix} Z_{SA} & Z_{AB} \\ Z_{AB} & Z_{SB} \end{bmatrix} \begin{Bmatrix} \Delta I_A \\ \Delta I_B \end{Bmatrix}^{i} \tag{5.7}$$

The changes in the value of compensation current are found with the Target Voltage method as follows:

$$\Delta I_A^i = \frac{V_A^i - V_T^i}{Z_{SA}} \qquad \Delta I_B^i = \frac{V_B^i - V_T^i}{Z_{SB}} \tag{5.8}$$

The target voltage, V_T^i, will be substituted shortly. Combining Equations (5.7) and (5.8) yields:

$$\begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i+1} = \begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i} - \begin{bmatrix} Z_{SA} & Z_{AB} \\ Z_{AB} & Z_{SB} \end{bmatrix} \begin{Bmatrix} \dfrac{V_A^i - V_T^i}{Z_{SA}} \\ \dfrac{V_B^i - V_T^i}{Z_{SB}} \end{Bmatrix} \tag{5.9}$$

The impedances of the right-hand vector can be moved over into the multiplying matrix:

$$\begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i+1} = \begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i} - \begin{bmatrix} 1 & \dfrac{Z_{AB}}{Z_{SB}} \\ \dfrac{Z_{AB}}{Z_{SA}} & 1 \end{bmatrix} \begin{Bmatrix} V_A^i - V_T^i \\ V_B^i - V_T^i \end{Bmatrix} \tag{5.10}$$

The target voltage for a loop object of degree-2 is given from Chapter 4:

$$V_T = \frac{V_A Z_{SB} + V_B Z_{SA}}{Z_{SB} + Z_{SA}} \tag{5.11}$$

This value can be substituted into Equation (5.10) to get the following factored expression:

$$\begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i+1} = \begin{bmatrix} 0 & -\dfrac{Z_{AB}}{Z_{SB}} \\ -\dfrac{Z_{AB}}{Z_{SA}} & 0 \end{bmatrix} \begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i} + \frac{1}{Z_{SA} + Z_{SB}} \begin{Bmatrix} 1 + \dfrac{Z_{AB}}{Z_{SB}} \\ 1 + \dfrac{Z_{AB}}{Z_{SA}} \end{Bmatrix} \begin{Bmatrix} Z_{SB} & Z_{SA} \end{Bmatrix} \begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i} \tag{5.12}$$

These results can be combined to form a single transition matrix:

$$\begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i+1} = \frac{1}{Z_{SA} + Z_{SB}} \begin{bmatrix} Z_{SB} + Z_{AB} & Z_{SA} - Z_{AB} \\ Z_{SB} - Z_{AB} & Z_{SA} + Z_{AB} \end{bmatrix} \begin{Bmatrix} V_A \\ V_B \end{Bmatrix}^{i} \tag{5.13}$$

The eigenvalues for this matrix have been determined as:

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= \frac{2Z_{AB}}{Z_{SA} + Z_{SB}} \qquad \leq 1 \end{aligned} \tag{5.14}$$

Note that the second eigenvalue is less than one because both the source trace impedance values in the denominator contain the common impedance in the numerator. This analysis implies, that a radialized system with just line impedances and constant current loads and a single degree-2 Affiliation object will not diverge.

5.3 Behavior of Single Degree-3 Affiliation Object

An Affiliation object with three member Segments is analyzed in this section. A radialized system having three loop objects is figured below:

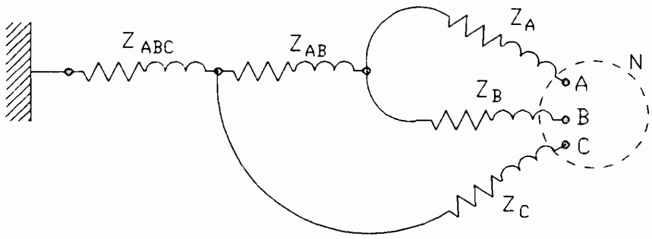


Figure 5.3 - A Radialized Model with a Single Degree-3 Affiliation Object

The member Segments of the Affiliation object are labeled within the representative dashed circle as Ψ_A, Ψ_B, and Ψ_C . As in the previous discussion of the degree-2 Affiliation object, laterals can emerge from any node in the above figure. Only one Affiliation object may exist, however. Notation for Segment voltages, currents, and source trace impedances are consistent with Section 5.1.

In the radialized system of Figure 5.3, the change in voltage of Affiliation object member Segments with respect to changes in compensation current can be represented as follows:

$$\begin{Bmatrix} \Delta V_A \\ \Delta V_B \\ \Delta V_C \\ \vdots \\ \vdots \\ \Delta V_\Omega \end{Bmatrix} = -\begin{bmatrix} Z_{TR} & B \\ & \\ C & D \end{bmatrix} \begin{Bmatrix} \Delta I_A \\ \Delta I_B \\ \Delta I_C \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \tag{5.15}$$

Segments in the system number from Ψ_A to Ψ_Ω but there is only a single Affiliation object and it contains just Segments as Ψ_A, Ψ_B, and Ψ_C. Hence, there are only compensation current changes for those Segments. All other Segments have a change in compensation current of zero. The Segments of the Affiliation object are not effected by any other Segments in this restricted scenario. Since we are not concerned with the behavior of other Segments, the upper three rows can be isolated from Equation (5.15).

$$\begin{Bmatrix} \Delta V_A \\ \Delta V_B \\ \Delta V_C \end{Bmatrix} = -\begin{bmatrix} Z_{SA} & Z_{AB} & Z_{AC} \\ Z_{AB} & Z_{SB} & Z_{BC} \\ Z_{AC} & Z_{BC} & Z_{SC} \end{bmatrix} \begin{Bmatrix} \Delta I_A \\ \Delta I_B \\ \Delta I_C \end{Bmatrix} \tag{5.16}$$

When the Affiliation object focuses, the target voltage is given by:

$$V_T = \frac{V_A Z_{SB} Z_{SC} + V_B Z_{SA} Z_{SC} + V_C Z_{SA} Z_{SB}}{Z_{SB} Z_{SC} + Z_{SA} Z_{SC} + Z_{SA} Z_{SB}} \tag{5.17}$$

And changes in Segment compensation current are given by:

$$\Delta I_A = \frac{V_A - V_T}{Z_{SA}} \quad \Delta I_B = \frac{V_B - V_T}{Z_{SB}} \quad \Delta I_C = \frac{V_C - V_T}{Z_{SC}} \tag{5.18}$$

The application of the Target Voltage method for a single iteration yields:

$$\begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i+1} = \begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i} - \begin{bmatrix} Z_{SA} & Z_{AB} & Z_{AC} \\ Z_{AB} & Z_{SB} & Z_{BC} \\ Z_{AC} & Z_{BC} & Z_{SC} \end{bmatrix} \begin{Bmatrix} \dfrac{V_A - V_T}{Z_{SA}} \\ \dfrac{V_B - V_T}{Z_{SB}} \\ \dfrac{V_C - V_T}{Z_{SC}} \end{Bmatrix} \tag{5.19}$$

The above expression can be re-factored to get:

$$\begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i+1} = \begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i} - \begin{bmatrix} 1 & \dfrac{Z_{AB}}{Z_{SB}} & \dfrac{Z_{AC}}{Z_{SC}} \\ \dfrac{Z_{AB}}{Z_{SA}} & 1 & \dfrac{Z_{BC}}{Z_{SC}} \\ \dfrac{Z_{AC}}{Z_{SA}} & \dfrac{Z_{BC}}{Z_{SB}} & 1 \end{bmatrix} \begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i} + V_T \begin{Bmatrix} 1 + \dfrac{Z_{AB}}{Z_{SB}} + \dfrac{Z_{AC}}{Z_{SC}} \\ 1 + \dfrac{Z_{AB}}{Z_{SA}} + \dfrac{Z_{BC}}{Z_{SC}} \\ 1 + \dfrac{Z_{AC}}{Z_{SA}} + \dfrac{Z_{BC}}{Z_{SB}} \end{Bmatrix} \tag{5.20}$$

Further factorization yields:

$$\begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i+1} = \frac{1}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}} [M] \begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i} \tag{5.21}$$

Where the matrix M is given by:

$$M = \begin{bmatrix} Z_{SB}Z_{SC} + Z_{AB}Z_{SC} + Z_{AC}Z_{SB} & Z_{SA}Z_{SC} + Z_{AC}Z_{SA} - Z_{AB}Z_{SA} - Z_{AB}Z_{SC} & Z_{SA}Z_{SB} + Z_{AB}Z_{SA} - Z_{AC}Z_{SA} - Z_{AC}Z_{SB} \\ Z_{SB}Z_{SC} + Z_{BC}Z_{SB} - Z_{AB}Z_{SC} - Z_{AB}Z_{SB} & Z_{SA}Z_{SC} + Z_{AB}Z_{SC} + Z_{BC}Z_{SA} & Z_{SA}Z_{SB} + Z_{AB}Z_{SB} - Z_{SA}Z_{BC} - Z_{BC}Z_{SB} \\ Z_{SB}Z_{SC} + Z_{BC}Z_{SC} - Z_{AC}Z_{SB} - Z_{AC}Z_{SC} & Z_{SA}Z_{SC} + Z_{AC}Z_{SC} - Z_{BC}Z_{SA} - Z_{BC}Z_{SC} & Z_{SA}Z_{SB} + Z_{AC}Z_{SB} + Z_{BC}Z_{SA} \end{bmatrix} \tag{5.22}$$

The matrix is not diagonally dominant but all entries are less than one and each row sums to exactly 1. The eigenvalues for the system are:

$$\lambda_1 = 1$$

$$\lambda_{2,3} = \frac{Z_{AC}Z_{SB} + Z_{BC}Z_{SA} + Z_{AB}Z_{SC}}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}} \pm \sqrt{\left(\frac{Z_{AC}Z_{SB} + Z_{BC}Z_{SA} + Z_{AB}Z_{SC}}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}}\right)^2 + \frac{(Z_{AB} - Z_{AC})^2 - 2(Z_{AB}Z_{BC} + Z_{AC}Z_{BC} - Z_{BC}^2)}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}}} \tag{5.23}$$

The eigenvalues above usually lie within the unit circle because of small numerators from the off diagonal terms and the fact that $Z_{AB}Z_{AC}Z_{BC} = 0$. The analysis presented here for degree-3 Affiliation objects does not demonstrate convergence due to eigenvalues. It can be seen, however, that the eigenvalues are usually within the unit circle because the term $\frac{Z_{AC}Z_{SB} + Z_{BC}Z_{SA} + Z_{AB}Z_{SC}}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}}$ is usually small. Several radial circuits were found with eigenvalues outside of the unit circle. The radial circuits still converged with application of the Target Voltage method.

It should be made clear that the analysis of this section is performed on a very specific type of radialized model. It may not correctly represent the behavior of a general system within Segflow containing multiple Affiliation objects and user defined Segment models.

5.4 Target Voltage Convergence of a Degree-3 Affiliation object

This section deals with the value of the target voltage within an Affiliation object of degree-3. The analysis will show that, in the given radialized model, the target voltage of a focused Affiliation object will steadily approach a convergent value. The relatively slow convergence rate suggested by this analysis is verified in the Chapter 6 simulation results by the large number of Affiliation object focuses needed to solve a system in Segflow.

Consider a power system containing lines, constant current loads, and an Affiliation object like the one below:

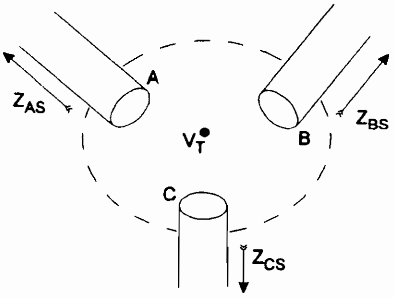


Figure 5.4 - A Degree-3 Affiliation object

Constant current load increments are to be added to the ends of each of the lines so that their respective voltage approaches V_t.

From the Chapter 4, the expression for the target voltage at the ith iteration can be written as:

$$V_T^i = \frac{V_A^i Z_{SB} Z_{SC} + V_B^i Z_{SA} Z_{SC} + V_C^i Z_{SA} Z_{SB}}{Z_{SA} Z_{SB} + Z_{SA} Z_{SC} + Z_{SB} Z_{SC}} \tag{5.24}$$

The changes in load currents for the Segments (which add to zero) are given as:

$$\Delta I_A^i = \frac{V_A^i - V_T^i}{Z_{SA}} \quad \Delta I_B^i = \frac{V_B^i - V_T^i}{Z_{SB}} \quad \Delta I_C^i = \frac{V_C^i - V_T^i}{Z_{SC}} \tag{5.25}$$

The power system model with the Affiliation object is radial. As demonstrated in Section 5.1, a reduced impedance matrix can be found which relates the changes in voltage at each of the loop Segments as their currents change:

$$\begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i+1} = \begin{Bmatrix} V_A \\ V_B \\ V_C \end{Bmatrix}^{i} - \begin{bmatrix} Z_{AS} & Z_{AB} & Z_{AC} \\ & Z_{BS} & Z_{BC} \\ & & Z_{CS} \end{bmatrix} \begin{Bmatrix} \Delta I_A \\ \Delta I_B \\ \Delta I_C \end{Bmatrix}^{i} \tag{5.26}$$

The impedance matrix is diagonally dominant and symmetrical and only valid if the sum of current changes is zero. Equation (5.26) can be substituted into equation (5.24) to get the following:

$$V_T^{i+1} = \frac{\begin{matrix} \left(V_A^i - \Delta I_A^i Z_{SA} - \Delta I_B^i Z_{AB} - \Delta I_C^i Z_{AC}\right) Z_{SB} Z_{SC} + \\ \left(V_B^i - \Delta I_A^i Z_{AB} - \Delta I_B^i Z_{SB} - \Delta I_C^i Z_{BC}\right) Z_{SA} Z_{SC} + \\ \left(V_C^i - \Delta I_A^i Z_{AC} - \Delta I_B^i Z_{BC} - \Delta I_C^i Z_{CC}\right) Z_{SA} Z_{SB} + \end{matrix}}{Z_{SA} Z_{SB} + Z_{SA} Z_{SC} + Z_{SB} Z_{SC}} \tag{5.27}$$

The sum of the voltage terms in the numerator divided by the impedance in the denominator matches the expression for the target voltage in equation (5.24). These terms can be factored out and (5.25) can be substituted into the result to get the following simplification:

$$V_T^{i+1} = V_T^i - \frac{\begin{matrix} \left[\left(V_A^i - V_T^i\right) Z_{SA} Z_{SB} Z_{SC} + \left(V_B^i - V_T^i\right) Z_{AB} Z_{SA} Z_{SC} + \left(V_C^i - V_T^i\right) Z_{SA} Z_{SB} Z_{AC}\right] Z_{SB} Z_{SC} \\ + \left[\left(V_A^i - V_T^i\right) Z_{AB} Z_{SB} Z_{SC} + \left(V_B^i - V_T^i\right) Z_{SB} Z_{SA} Z_{SC} + \left(V_C^i - V_T^i\right) Z_{BC} Z_{SA} Z_{SB}\right] Z_{SA} Z_{SC} \\ + \left[\left(V_A^i - V_T^i\right) Z_{AC} Z_{SB} Z_{SC} + \left(V_B^i - V_T^i\right) Z_{BC} Z_{SA} Z_{SC} + \left(V_C^i - V_T^i\right) Z_{SA} Z_{SB} Z_{SB}\right] Z_{SA} Z_{SB} \end{matrix}}{Z_{SA} Z_{SB} Z_{SC} \left(Z_{SA} Z_{SB} + Z_{SA} Z_{SC} + Z_{SB} Z_{SC}\right)} \tag{5.28}$$

Equation (5.24) is also useful in simplifying the “diagonal” terms of the above equation to zero:

$$\frac{\left(V_A^i - V_T^i\right) Z_{SB} Z_{SC} + \left(V_B^i - V_T^i\right) Z_{SA} Z_{SC} + \left(V_C^i - V_T^i\right) Z_{SA} Z_{SB}}{Z_{SA} Z_{SB} + Z_{SA} Z_{SC} + Z_{SB} Z_{SC}} = 0 \tag{5.29}$$

The expression for the new target voltage is now:

$$V_T^{i+1} = V_T^i - \frac{\begin{matrix} \left(V_A - V_T^i\right)\left(Z_{AB} Z_{SC} + Z_{AC} Z_{SB}\right) + \left(V_B - V_T^i\right)\left(Z_{AB} Z_{SC} + Z_{BC} Z_{SA}\right) \\ + \left(V_C - V_T^i\right)\left(Z_{AC} Z_{SB} + Z_{BC} Z_{SA}\right) \end{matrix}}{Z_{SA} Z_{SB} + Z_{SA} Z_{SC} + Z_{SB} Z_{SC}} \tag{5.30}$$

A substitution can be made for the outward voltages in terms of the target voltage:

$$\begin{aligned} V_A^i &= \alpha_A V_T^i \\ V_B^i &= \alpha_B V_T^i \\ V_C^i &= \alpha_C V_T^i \end{aligned} \tag{5.31}$$

so that:

$$V_T^{i+1} = V_T^i \left[1 - \frac{\begin{array}{c}(\alpha_A - 1)(Z_{AB}Z_{SC} + Z_{AC}Z_{SB}) + (\alpha_B - 1)(Z_{AB}Z_{SC} + Z_{BC}Z_{SA}) \\ +(\alpha_C - 1)(Z_{AC}Z_{SB} + Z_{BC}Z_{SA})\end{array}}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}} \right] \tag{5.32}$$

The following substitution is introduced between the source trace impedances and the intersections:

$$\begin{aligned} Z_{AB} &= \zeta_1 Z_{SA} \\ Z_{AC} &= \zeta_2 Z_{SC} \\ Z_{BC} &= \zeta_3 Z_{SB} \end{aligned} \tag{5.33}$$

These substitutions are made to get:

$$V_T^{i+1} = V_T^i \left[1 - \frac{\zeta_3 Z_{SA}Z_{SB}(\alpha_B + \alpha_C - 2) + \zeta_1 Z_{SA}Z_{SC}(\alpha_A + \alpha_B - 2) + \zeta_2 Z_{SB}Z_{SC}(\alpha_A + \alpha_C - 2)}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}} \right] \tag{5.34}$$

Finding a common denominator:

$$V_T^{i+1} = \\ V_T^i \frac{Z_{SA}Z_{SB}[1 - \zeta_3(\alpha_B + \alpha_C - 2)] + Z_{SA}Z_{SC}[1 - \zeta_1(\alpha_A + \alpha_B - 2)] + Z_{SB}Z_{SC}[1 - \zeta_2(\alpha_A + \alpha_C - 2)]}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}} \tag{5.35}$$

The target voltage can be approximated as its value in the limiting case:

$$V_T \cong \frac{V_A + V_B + V_C}{3} \tag{5.36}$$

The following can easily be shown from Equations (5.31) and (5.36):

$$\alpha_A + \alpha_B + \alpha_C \cong 3 \tag{5.37}$$

This result can now be used in equation (5.35) to get:

$$V_T^{i+1} = V_T^i \frac{Z_{SA}Z_{SB}[1 + \zeta_3(\alpha_A - 1)] + Z_{SA}Z_{SC}[1 + \zeta_1(\alpha_C - 1)] + Z_{SB}Z_{SC}[1 + \zeta_2(\alpha_B - 1)]}{Z_{SA}Z_{SB} + Z_{SA}Z_{SC} + Z_{SB}Z_{SC}} \tag{5.38}$$

Where $0 \le \zeta_k \le 1$

$$\alpha_k \approx 1$$

This equation shows the transition of the target voltage of a Affiliation object during application of the Target Voltage method. The factor multiplying the target voltage is not static and therefore cannot be used to show strict convergence. The factor iterating the target voltage is usually very close to one because the factors $1+\zeta(\alpha-1)$ are typically very small. This suggests a slow and steady convergence rate. The changes invoked upon the target voltage through iterations are well behaved within the nine test systems solved with Segflow. As the Target Voltage method reaches convergence, each of the factors, α_A, ,α_B, and ,α_C becomes one and thus $V_T^{i+1}\big|_{i\to Large} = V_T^i$.

5.5 Target Voltage Convergence of a Degree-N Affiliation object

This section reflects the analysis of Section 5.4 for a general Affiliation object. Consider a power system with linear lines and loads containing a single Affiliation object named Λ. Λ has access to N Segments numbered $\Psi_1 \cdots \Psi_N$. Each Segment has an outward voltage tagged with a matching subscript $V_1 \cdots V_N$. Each Segment, j, also has a source impedance $\Psi_j \to_A Z_S$. The target voltage is expressed as:

$$V_T^i = \frac{\sum_{j=1}^{N} V_j^i \prod_{\substack{k=1 \\ k\neq j}}^{N} Z_{KK}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ k\neq j}}^{N} Z_{KK}} \qquad \textit{where:} \quad \begin{aligned} Z_{XX} &= \Psi_X \to_A Z_S \\ Z_{XY} &= \Psi_X \to_A Z_S \cap \Psi_Y \to_A Z_S \end{aligned} \tag{5.39}$$

Changes in the outward current for $\Psi_1 \cdots \Psi_N$ are calculated respectively with:

$$\Delta I_j^i = \frac{V_j^i - V_T^i}{Z_{jj}} \tag{5.40}$$

Our iterations of the target voltage correspond to a single loop Λ. This Affiliation object is solely iterated in this system. Since the system we are considering is linear and radial, changes to the Segments constant current load belonging to Λ have the following effect:

$$V_j^{i+1} = V_j^i - \sum_{l=1}^{N} \Delta I_l^i Z_{jl} \tag{5.41}$$

Substituting this into the target voltage of (5.39) we get:

$$V_T^{i+1} = \frac{\sum_{j=1}^{N} \left(V_j^i - \sum_{l=1}^{N} \Delta I_l^i Z_{jl} \right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} \tag{5.42}$$

The target voltage expression of Equation (5.39) can be used to factor out the voltage term of the numerator above to yield:

$$V_T^{i+1} = V_T^i - \frac{\sum_{j=1}^{N} \left(\sum_{l=1}^{N} \Delta I_l^i Z_{jl} \right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} \tag{5.43}$$

The compensation current expression of Equation (5.40) can be substituted above to get:

$$V_T^{i+1} = V_T^i - \frac{\sum_{j=1}^{N} \left(\sum_{l=1}^{N} \frac{V_l^i - V_T^i}{Z_{ll}} Z_{jl} \right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} \tag{5.44}$$

The above expression can be factored to get:

$$V_T^{i+1} = V_T^i - \frac{\sum_{j=1}^{N} \left(\sum_{\substack{l=1 \\ \neq j}}^{N} \frac{V_l^i - V_T^i}{Z_{ll}} Z_{jl} \right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} - \frac{\sum_{j=1}^{N} \left(V_j^i - V_T^i \right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} \tag{5.45}$$

The target voltage of equation (5.39) can be used to easily show that the third term in the above expression is zero:

$$\frac{\sum_{j=1}^{N} \left(V_j^i - V_T^i \right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} = 0 \tag{5.46}$$

The following simplification of equation (5.45) results:

$$V_T^{i+1} = V_T^i - \frac{\displaystyle\sum_{j=1}^{N}\left(\sum_{\substack{l=1\\ \neq j}}^{N}\left(V_l^i - V_T^i\right)Z_{jl}\prod_{\substack{k=1\\ \neq j \neq l}}^{N} Z_{kk}\right)}{\displaystyle\sum_{j=1}^{N}\prod_{\substack{k=1\\ \neq j}}^{N} Z_{kk}} \qquad (5.47)$$

all impedances are symmetrical such that $Z_{xy} = Z_{yx}$. Equation (5.47) can therefore be refactored:

$$V_T^{i+1} = V_T^i - \frac{\displaystyle\sum_{j=1}^{N}\left(V_j^i - V_T^i\right)\sum_{\substack{l=1\\ \neq j}}^{N} Z_{jl}\prod_{\substack{k=1\\ \neq j \neq l}}^{N} Z_{kk}}{\displaystyle\sum_{j=1}^{N}\prod_{\substack{k=1\\ \neq j}}^{N} Z_{kk}} \qquad (5.48)$$

The following substitutions are introduced:

$$\begin{aligned} V_j &= \alpha_j V_T \\ \underset{i \neq j}{Z_{ij}} &= \zeta_{ij} Z_{jj} \end{aligned} \qquad (5.49)$$

The voltage substitution is applied to Equation (5.48) to get:

$$V_T^{i+1} = V_T^i \left(1 - \frac{\displaystyle\sum_{j=1}^{N}\sum_{\substack{l=1\\ \neq j}}^{N} Z_{jl}\left(\alpha_j^i - 1\right)\prod_{\substack{k=1\\ \neq j \neq l}}^{N} Z_{kk}}{\displaystyle\sum_{j=1}^{N}\prod_{\substack{k=1\\ \neq j}}^{N} Z_{kk}}\right) \qquad (5.50)$$

The second summation sign was shifted left to make the following steps more clear. Since all impedances are symmetrical, the second summation can be broken into two parts:

$$V_T^{i+1} = V_T^i \left(1 - \frac{\displaystyle\sum_{j=1}^{N}\sum_{l=j+1}^{N}\left(Z_{jl}\left(\alpha_j^i - 1\right)\prod_{\substack{k=1\\ \neq j \neq l}}^{N} Z_{kk} + Z_{lj}\left(\alpha_l^i - 1\right)\prod_{\substack{k=1\\ \neq j \neq l}}^{N} Z_{kk}\right)}{\displaystyle\sum_{j=1}^{N}\prod_{\substack{k=1\\ \neq j}}^{N} Z_{kk}}\right) \qquad (5.51)$$

Now, Equation (5.51) can be factored to get:

$$V_T^{i+1} = V_T^i \left(1 - \frac{\sum_{j=1}^{N} \sum_{l=j+1}^{N} Z_{jl}\left(\alpha_j^i + \alpha_l^i - 2\right) \prod_{\substack{k=1 \\ \neq j \neq l}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} \right) \qquad (5.52)$$

The second substitution from Equation (5.49) is now made:

$$V_T^{i+1} = V_T^i \left(1 - \frac{\sum_{j=1}^{N} \sum_{l=j+1}^{N} \xi_{jl}\left(\alpha_j^i + \alpha_l^i - 2\right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} \right) \qquad (5.53)$$

A common denominator can now be found and the expression factored once again to yield:

$$V_T^{i+1} = V_T^i \frac{\sum_{j=1}^{N} \left(1 - \sum_{l=j+1}^{N} \xi_{jl}\left(\alpha_j^i + \alpha_l^i - 2\right) \right) \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}}{\sum_{j=1}^{N} \prod_{\substack{k=1 \\ \neq j}}^{N} Z_{kk}} \qquad (5.54)$$

Each term in the numerator has a matching term in the denominator of the target voltage multiplication factor. As the outward voltages of Affiliation object member Segments get closer to each other, the factors $\alpha_j \rightarrow 1$. The value of the target voltage therefore converges to some constant value.

The values of α typically range as $.8 \leq \alpha \leq 1.2$ and $0 \leq \zeta \leq 1$ so that the quantity multiplying the target voltage during iterations is around 1. It appears that the multiplier for Affiliation objects with high degrees can have a higher deviation from 1 resulting in a faster convergence or perhaps a greater oscillation around the final target voltage value.

Once again, this analysis does not imply that Segflow will converge with systems having user defined Segments and multiple Affiliation objects. This analysis is only an investigation into the behavior of a single Affiliation object in a radialized model with constant impedance line sections.

Chapter 6

The Segflow Process

6. The Segflow Process

The Segflow process is the overall method that brings the system of radially connected Segment objects to represent the solution of a nonlinear power system network model. Segments, current and voltage traces, and Affiliation objects with the Target Voltage method are all incorporated into the Segflow solution process. This diagram below represents, as rings, the levels of the Segflow process:

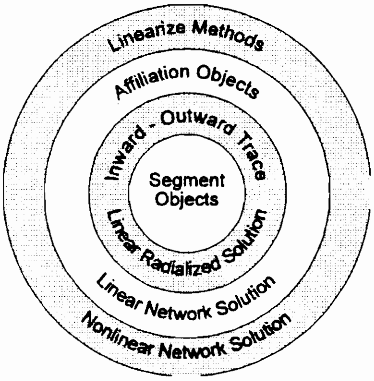


Figure 6.1 - Layers of the Segflow Process

The diagram represents Segments as the fundamental aspect of Segflow. In the second ring, voltage and current traces are used to solve the radial collection of Segments to determine the solution of the radialized model having constant current loads. Affiliation objects are used to bring the solution of the radialized model to match the solution of the corresponding power system network having the same constant current loads. Finally, linearizations which are discussed by example in this chapter, are used to drive the state of the Segment collection to match the solution of the nonlinear power system network.

6.1 The Segflow Flowchart

Many aspects constitutional to this dissertation have been discussed in the last five chapters. These aspects are organized in Segflow such that the solution of nonlinear power system models can be found. This organization is contained in a method of the System class used for load-flow analysis. The figure below represents the Segflow organization as a flowchart:

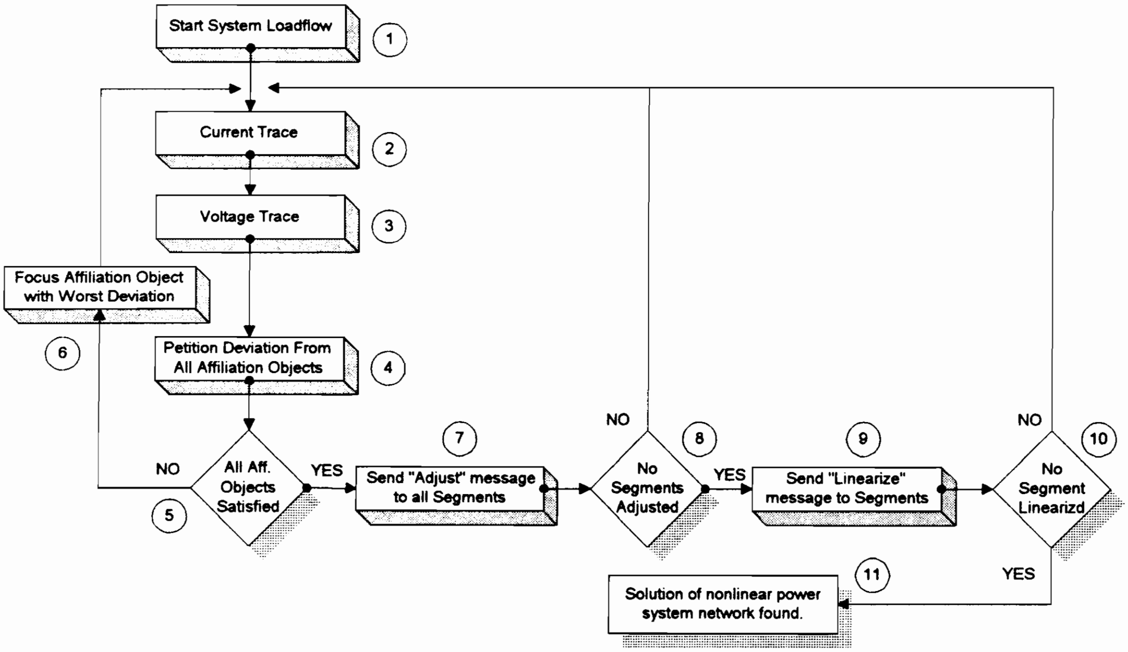


Figure 6.1 - The Segflow Process

The table below describes each of the boxes in the above flowchart. Each of the steps in this tables corresponds to the number next to the particular box in the figure above. This description is a general overview of the Segflow process.

Step 1 Start	The load-flow solution was called for by the System object discussed in Chapter 2. At this point, an initial voltage trace would have been performed to propagate the voltage of the system's source Segment (swing bus) throughout the radialized model. Each Segment would have also been sent a message to Linearize (also discussed in Chapter 2) so that all loads are represented as constant current values with respect to the outward voltage of each Segment.
Step 2 Current Trace	A current trace (Chapter 2) is evoked by the System. Through an inward propagation, this process collects load current from the ends of the circuit and to the source Segment. Each Segment has a value of current entering its inward node.

Step 3 Voltage Trace	A voltage trace is used to propagate the source Segment outward voltage through the model supplied drop of each segment.

Since Segments are not supposed to alter injections within the voltage and current trace methods, steps 2 and 3 of the Segflow process determine the solution of the radialized model. This solution corresponds to a radial circuit having constant current loads.

Step 4 Get Affiliation Object Deviation	A message is sent to all Affiliation objects requesting their status. An Affiliation object can respond with a message that its member Segments are in satisfactory condition. Alternatively, it can respond with a factor representing a normalized deviation in the outward voltages of its member Segments.
Step 5 Check Affiliation Objects	If all of the Affiliation objects interrogated in Step 4 responded in satisfaction with their member Segments then the Segflow process is continued to the next step. Otherwise, Affiliation objects need to be focused so that they can correct their internal discrepancies.
Step 6 Invoke Worst Affiliation Object	The Affiliation object having the worst deviation from Step 4 is focused. This means that a message is sent to that Affiliation object signaling it to invoke the Target Voltage method. Alterations are made to the value of compensation current and so current and voltage traces are made after this step.

The completion of Steps 1-6 brings the System's Segments to states that represent the solution of the power system network having constant current loads.

Step 7 Adjust Segments	Messages are sent to Segments calling for them to "adjust" their state in lieu of the solved, constant current load, network. Segment outward voltage and input current are used as a basis for state changes. A current and voltage trace is invoked each time a Segment responds to the "adjust" message. Segments modeling lines and transformers probably do not respond to the "adjust" message. Generator Segments, however, may alter their current output to bring up terminal voltage. A generator modeling example is discussed later in this chapter.
Step 8 Check Segment Adjustment	If any Segments responded to the "adjust" messages of the previous step then the Segflow process returns to the solution of the radial system and Affiliation object invocations.

Steps 7 and 8 are primarily used to balance loads and supplies in the radialized model. Generators, SVCs, and other devices whose output may vary are adjusted to a satisfactory output. After Step 8, loads in the radialized model should generally be fed from a proper location and Segment voltages should be reasonable. These steps prepare the system for the linearization steps in which load currents may be determined from voltages and Segment states.

Step 9 Linearize Segments	"Linearization" messages are sent to all Segments. Segments may respond by re-evaluating their loads and injections based on the Segment state and outward voltage. These loads are converted to constant current loads in preparation of restarting the Segflow process. Each Segment returns a message regarding its state as satisfactory or not.
Step 10 Check Linearization	If any Segment responded to the "linearization" message with a return message of "unsatisfactory" then the Segflow process is returned to the voltage and current traces used to solve the radial model (Step 2). Otherwise, the Segflow process is completed.
Step 11 Process Complete	The completion of the Segflow process signifies the solution of the nonlinear power system network model. Voltages, flows, and model states are all available within each Segment.

This section has presented an overview of the Segflow load-flow process. Certainly there are many details of the prototype Segflow that have not been discussed here. These details do not, however, pertain to the concept and contribution of the object-oriented load flow which is the subject of this dissertation.

6.2 Some Segment Model Examples

This section introduces, by example, the method for modeling power system components. As has been discussed previously, specific component models are derived from more general ones. The following is an example of a transformer class.

6.2.1 A Transformer Class Example

An equipment modeling class, such as a transformer class, is created in Segflow by deriving it from the Segment class. Some of the virtual methods of a modeling class are presented in this example. These methods are essentially all which differentiate the derived class from its parent, the Segment. The transformer class presented here is based on the following model:

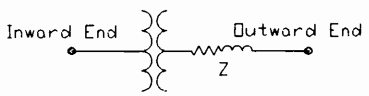


Figure 6.2 - A Simple Transformer Model

Information about the transformer voltage ratings, tap ratio, and impedance are stored as attributes of the derived class. The exerpt below shows a portion of an exemplar[13] for the class modeling the transformer. Notice how it is derived from CSegment, the Segment class, and stores equipment data as protected[13] attributes for classes derived from CTransformer:

```
class CTransformer : public CSegment
{
  protected:
    Complex Z;            // ** Impedance
    float RatedHighKv,    // ** kV Ratings
          RatedLowKv,
          TapRatio;       // ** Fixed Tap
  public:
    CTransformer() : CSegment(){};
    void OnCurrentTrace();
    void OnVoltageTrace();
    void OnTraceImpedance();
    void OnModelDisplay();
    void OnReadDataBase(CRecord *record);
};
```

Figure 6.3 - A Transformer Class Exemplar

The methods *OnModelDisplay* and *OnReadDataBase* and the constructor *CTransformer*[13] are not further discussed in this example but are used for creation and input/output into instances of the CTransformer class. The remaining three methods are discussed below.

6.2.1.1 OnCurrentTrace()

This method is a direct application of Kirchoff's current law within the transformer model. In all Segments, its purpose is to determine the inward current demand when given the outward end current demand. The method has access to the total of all downstream current demand and load current with the Segment attribute *IOut*. From the perspective of this method, the transformer appears like this:

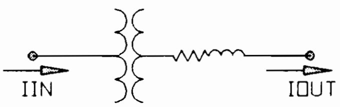


Figure 6.4 - The Transformer Within OnCurrentTrace Method

IOut is known, *IIn* is found with the *OnCurrentTrace* method by transferring *IOut* through the transformer as follows:

```
void CTransformer::OnCurrentTrace()
{
  IIn = IOut * RatedLowKv / RatedHighKv / TapRatio;
};
```

Figure 6.5 - The OnCurrentTrace Method for CTransformer

6.2.1.2 OnVoltageTrace()

The purpose of this method is always to find the outward end voltage when given the inward end voltage. This method represents the application of Kirchoff's voltage law within the transformer model. In this case, the inward voltage *KVIn* is available to the method as a Segment attribute. Here is a diagram:

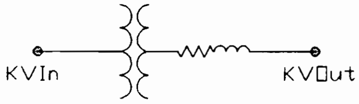


Figure 6.6 - The Transformer Within OnVoltageTrace Method

The goal of the method is to determine the outward voltage KVOut by transferring KVIn through the transformer and taking a drop through the impedance. Here is the method:

```
void CTransformer::OnVoltageTrace()
{
  Complex KVDrop = IOut * Z / 1000.;
  KVOut = KVIn * RatedHighKv / RatedLowKv *
       TapRatio - KVDrop;
};
```

Figure 6.7 - The OnCurrentTrace Method for CTransformer

6.2.1.3 *OnTraceImpedance()*

The purpose of this method is always to approximate the series impedance from the outward end when given, as a Segment attribute, *ZIn*, the series impedance from the inward end. Here is the model in the case of the transformer class:

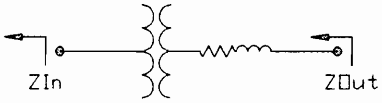


Figure 6.8 - Transformer Within OnTraceImpedance Method

The following is the method which reflects *ZIn* through the transformer and adds the transformer impedance:

```
void CTransformer::OnTraceImpedance()
{
   float a = RatedHighKv / RatedLowKv * TapRatio;
   ZOut = ZIn / a / a + Z;
};
```

Figure 6.9 - The OnCurrentTrace Method for CTransformer

This method is important to the construction of the radialized model which will be discussed later.

6.2.1.4 *A New Modeling Class Emerges*

In most cases, the over-riding of the above virtual methods is all that is necessary to create a new class for modeling power system equipment. Models for single- or three-phase lines, regulating transformers, and self-regulating capacitor banks can be handled in a fashion similar to the above example. Models for more extensive components such as generators or static var compensators require the incorporation of another virtual method dealing with iterative linearization called the *OnAdjust* method. The method is discussed in the following example dealing with a generator class.

6.2.2 A Generator Class Example

Segflow treats the generator model just as it would line models, transformer models or any other models. The internal functionality of the generator class must provide an accurate representation of the generator object to neighboring Segments. The generator model of this section represents generator real power output with a negative load. The Target Voltage method is used within the *OnAdjust* method of the generator class to alter the generator's current output so that its terminal voltage matches the voltage magnitude setting. Here is a generator object attached to a general radialized system:

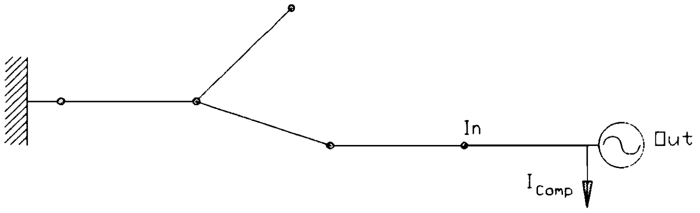


Figure 6.10 - A Generator Segment within a Radialized System

The generator Segment is marked with an inward and outward end. The load point marked with the compensation current is also shown on the generator Segment at the outward end. The exemplar for the generator class is similar to all Segment derived classes and is shown below:

```
class CGenerator : public CSegment
{
  protected:
    Complex ITotal;  // ** I added to IComp
    float KvSet,    // ** Terminal kV Setting
          KwSet,    // ** kW Output
          QTotal; // ** kVar added in Adjust
  public:
    CGenerator() : CSegment(){};
    int OnAdjust();
    int OnLinearize();
    void OnModelDisplay();
    void OnReadDataBase(CRecord *record);
};
```

Figure 6.11 - A Generator Class Exemplar

Four methods are missing from this exemplar that were present for the CTransformer one. The *OnTraceImpedance* method is missing because the generator modeled here has no internal impedance. Thus, the default Segment method for *OnTraceImpedance* is sufficient for the generator class. The *OnVoltageTrace* method of the base class (CSegment) is also sufficient since their are no drops from the inward end to the outward end of the generator Segment. Finally, the *OnCurrentTrace* method of the base class propagates current, including compensation current, through the Segment and to the inward end of the Segment. The generator class does not need to over-ride this function.

Two methods have been added to the generator class. The *OnAdjust* method handles alterations in generator output current at the solution of the power system network with linear loads. The *OnLinearize* method converts output current added in the *OnAdjust* method to a kVar to be evaluated at the outward voltage of the Segment. These methods are described in more detail shortly.

Like the transformer class, the methods *OnModelDisplay* and *OnReadDataBase* are used for displaying and retrieving Segment data. These methods are not discussed in this dissertation. It should be noted, however, that the *OnReadDataBase* method is responsible for converting the output kW setting of the generator to a negative constant power load.

6.2.2.1 OnAdjust()

The OnAdjust method is evoked after all Segments and Affiliation objects are satisfied with their state with respect to constant current injections. In this method, the generator will increase its output current to bring its voltage magnitude up the kV setting. The change in output current is registered by a change in compensation current. The total compensation current change as well as the total intended correction to vars are stored in generator attributes. Here is the method:

```
int CGenerator::OnAdjust()
{
  // ** See if adjustment needed
  if(fabs(kVSet-kVDn.mag())<TOL) {
    return NO_CHANGE;
  }
  float dQ, angle;
  Complex kVTarg, dI;
  // ** Get operating angle
    angle = kVDn.degree();
  // ** Get target voltage from angle
  // ** and voltage setting.
    kVTarg.Phasor(kVSet, angle);
  // ** Get current change to reach
  // ** target voltage
    dI=(kVDn-kVTarg)/Zs*1000.;
  // ** Find corresponding kVar change
    dQ=( kVDn * dI.conjg() ).imag();
  // ** Keep track of added values
    QTotal+ = dQ;
    ITotal+ = dI;
  // ** Add current to Compensation
    AddToCompensationI(dI);
  // ** Signal that Change has occured
    return CHANGED;
};
```

Figure 6.12 - The OnAdjust Method for CTransformer

This method makes alterations to the compensation current of the generator in an effort to bring the terminal voltage magnitude to match the setting. The first step in the method is to see if an adjustment is necessary. If an adjustment is needed then a target voltage is constructed from the present voltage angle of the generator and is kV magnitude setting. A change in compensation current is calculated along with a corresponding change in reactive output that should be generated. The compensation current change is added and the generator Segment signals Segflow that it has made changes.

6.2.2.2 OnLinearize()

This method is not essential to the simple generator being modeled here. It is important, however, to generators modeled with reactive power limits or other details. In this example, the method is used to convert the total output current accumulated in the compensation current attribute to constant kVar load. Here is the method:

```
int CGenerator::OnLinearize()
{
  // ** See if linearization needed
    if( GetIComp().mag() )<TOL) {
      return NO_CHANGE;
    }
  // ** Remove Q current from IComp
    AddToCompensationI(-ITotal);
    ITotal = Complex(0.,0.);
  // ** Add Q to load
    AddPQLoad(0.,-QTotal);
    QTotal = 0.;
  // ** Call Linearize of Base Class
    return CSegment::Linearize();
};
```

Figure 6.13 - The OnAdjust Method for CTransformer

At the convergence of the Segflow load-flow, the output active and reactive power of generators using this model will be contained as the opposite of the Segment load. Generator Segments similar to the one above were used in experiments with a prototype Segflow program. The generators in the experiment supported reactive power upper and lower limits as well as a range on the generators output voltage. The experiment results are presented in the next section.

6.3 Experimentation

Several systems have been tested with the Segflow object-oriented modeling scheme. The Segments, voltage and current traces, and Affiliation objects inherent to Segflow have operated as designed and allowed the object-oriented models to consistently converge. The chart below shows the results which are presented to show that Segflow does solve the models of nonlinear power system networks in a reasonable amount of time. The size of each network is given in terms of busses, meshes, and generators. Segflow statistics regarding Affiliation objects are also given.

Segflow Test Results from Simulation on 80486/66Mhz. Using Microsoft Visual C++ 1.5 and Microsoft Foundation Class Library 2.5	Stevenson	Anderson & Fouad	Wood & Wollenberg	IEEE 14	Gross 16	IEEE 30	NMPP	New England	IEEE 300
Solution Time From Flat Start (Sec)	0.05	0.05	0.33	0.94	0.05	1.26	7.91	10.22	362.11
Number of Busses	5	9	6	14	16	30	36	39	300
Number of Meshes	2	1	6	7	1	12	13	8	112
Number of Generators	2	3	3	5	3	6	8	10	69
Number of Segments	8	12	14	23	20	47	57	57	480
# 2 Segment Affiliation Objects	2	1	2	7	1	12	11	8	73
# 3 Segment Affiliation Objects	0	0	2	0	0	0	1	0	18
# 4 Segment Affiliation Objects	0	0	0	0	0	0	0	0	1

Figure 6.14 - Some Test Systems Analyzed With Segflow

The time required for Segflow to reach a solution exponentially increases as the size of the respective power system increases. The 300 bus system is a relatively large network and requires six minutes by Segflow to reach a solution. This is probably a much longer period than would be required by a comparable Newton-Rhapson routine on the same computer. Segflow does, however, allow the extensive modeling and quick model alteration of any component model in the system. That type of modeling flexibility is not found with the Newton-Rhapson method.

Chapter 7

Conclusion

7. Conclusion

The simulation results of Chapter 6 show that a new type of object-oriented load-flow called Segflow, is capable of solving the nonlinear load-flow problem. As discussed in Chapters 1 and 2, Segflow has a modular object-oriented design with Segment objects representing the various types of power system equipment. The radialized model, presented in Chapter 3, is created when these objects are connected in a radial fashion. Current and voltage traces are used to solve this radial collection of Segments. Affiliation objects using a technique called the Target Voltage Method are used to bring the solution of the radialized model to the corresponding solution of the network model as discussed in Chapters 4 and 5. A process involving Segments, traces, Affiliation objects, and linearization methods is used to determine the solution of the nonlinear power system network model. This process is presented in Chapter 6.

The object-oriented layout of Segflow provides very distinct modeling advantages. Power system modeling is from the perspective of each equipment's model. Rather than force all equipment models into large sets of equations to be solved simultaneously by an equation solver, Segflow allows component models to remain intact and independent as Segment objects. Each Segment has a one-to-one correspondence with some piece of equipment in the respective power system which is maintained in the system solution. The behavior of a Segment can be evaluated throughout the load-flow process.

An important feature of Segflow is the object-oriented design enabling new equipment models to be added into the Segflow environment as autonomous objects. New models are created from the Segment class and their inputs and outputs are always relative to the same Segment attributes. Because all Segment objects are derived from a Segment modeling class and all Segments are treated the same by Segflow, many types and varieties of power system equipment models can be easily created.

A Segflow application is a collection of interacting objects whose interactions lead to a load-flow solution. Segflow is an interesting alternative to classical approaches for solving the load-flow problem with large sets of simultaneous nonlinear equations. It also adds new aspects to the application of object-oriented design in load-flow analysis[9,10].

This dissertation demonstrates that a power system modeled with an object-oriented design can be made to converge to its load flow solution. Since equipment models don't have to be merged into an overall model such as a Jacobian, they are simpler to understand and construct.

The Target Voltage method allows, the solution of large densely meshed power systems with voltage and current trace methods designed for the analysis of radial systems. This allows for very flexible modeling capabilities and maintains memory efficiency. A conceptual simplification of the load-flow problem has been achieved since trace methods and the Target Voltage method are simple concepts to understand.

Segflow supports modeling capability and flexibility. The method for solving the load flow problem does not have to be altered as new component models are added. The object-oriented modeling scheme will allow engineers to spend more time thinking about the actual modeling of equipment and less time thinking about algorithms. Users can spend more time working on realistic component models and less time solving systems of equations.

To summarize, the contributions of this work are:

- An object-oriented load-flow called Segflow was found. Segflow utilizes the Target Voltage method to find the solution of looped power system models using radial trace methods. The validity of Segflow was verified on nine test systems.
- An expression for the Target Voltage was found for Affiliation objects with any number of member Segments.
- The radialized model was defined in Chapter 3 and the need for minimal source trace impedance values for all Segments was explained.
- A relationship between Affiliation objects, member Segments, and system loops was found in Chapter 4.
- An exploration into the convergence qualities of the Target Voltage method was performed in Chapter 5.

The simulation results of Chapter 6 suggest that Segflow is capable of solving power system models with a small number of loops. Segflow appears to be competitive in finding the solution of small systems and less competitive in finding the solution of larger systems. Although the convergence characteristics of specific scenarios involving single Affiliation objects was investigated, a proof of convergence for the Segflow process could not be found. This leaves the characterization of the method as heuristic and implies that power system configurations could be found such that the Segflow algorithm would not converge while other methods such as Newton-Rhapson would converge.

7.1 Applications

Segflow has applications in power system education. The trace methods used to run Segflow are simple. A beginning power student can be taught how and why the techniques are implemented. As power system components such as lines, transformers, and generators are discussed in class, the student can construct his own models and add them to a Segflow application. The nature of the modeling would provide good insight into and understanding of the operation of power system equipment. The ease of adding component models to Segflow would allow the student to spend more time exploring and learning about power system equipment and less time struggling with complicated load flow algorithms. Kersting has reported that students are very receptive to this type of straightforward computer modeling [24].

Segflow also has industrial applications. Software developers will find that Segflow is cheaper and easier to maintain due to its object-oriented design [2]. They will find that they can be much more responsive to user design requests since models are easier to create and change. Power engineers using Segflow should be able to order component models from various venders that match power system equipment in their power system.

7.2 Future Work Investigating Multiple Solutions

Power system models containing two or more generators or constant power loads may have multiple solutions to the load-flow problem. Classical solution methods based on a system wide Jacobian matrix, such as Newton-Rhapson, are capable of finding these multiple solutions. The solutions consist of physically achievable states called stable equilibrium points, and abstract states called unstable equilibrium points and saddle points. These different solutions are found by altering the initial conditions of the Newton-Rhapson algorithm.

It is assumed that the Segflow process presented herein reaches a stable equilibrium point representing a physical state of the respective power system. This assumption is made because the process starts from an unloaded system and load is added until the solution of the actual system is found. Since load is incrementally added, the final solution should not have deviated from the initial stable solution of the unloaded system.

It may be possible for the Segflow process to find multiple load-flow solutions for a given system. The creator of generator models could allow different starting angles within his model. These different starting angles might lead to the convergence of different states within the collection of Segflow objects -- thus representing different solutions of the power system model.

Although the action to reach various solutions may be valid and possibly valuable, it seems improbable that a Segflow user could intelligently seek particular solutions. The problem with seeking multiple solutions in Segflow is that objects can only access local information. Jacobian information of classical methods which guide the alteration of generator initial conditions is not available in Segflow. Information local to a single generator is probably not sufficient to guide the selection of starting conditions for that generator. If multiple load-flow solutions could be reached by Segflow, it seems difficult to guide the process from one solution to another.

7.3 Future Work With Dynamic Simulation

Research has been and will be performed with dynamic simulation within the Segflow environment. Another layer can be added to the Segflow diagram presented in Chapter 6. The layer wraps around the solution of the nonlinear network as this figure implies:

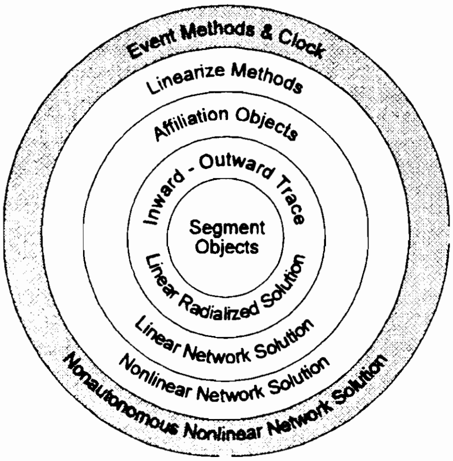


Figure 7.1 - The Outer Rings of Segflow

An object called a clock keeps track of a simulated time within Segflow. After Segflow determines the solution of the nonlinear power system network model, each Segment is sent a message seeking response for the event of a clock tick. Segments representing components with time dependent features such as fuses or reclosers may respond by blowing or opening and thus interrupting current flow from their inward to outward end. Other Segments may have internal representations for dynamic characteristics. Numerical integration or differentiation can be performed.

The following diagram represents an excitor for an emergency diesel generator that was simulated along with a governor, prime mover, and regulator to study generator start up behavior at Calvert Cliffs Nuclear Power Plant[27]:

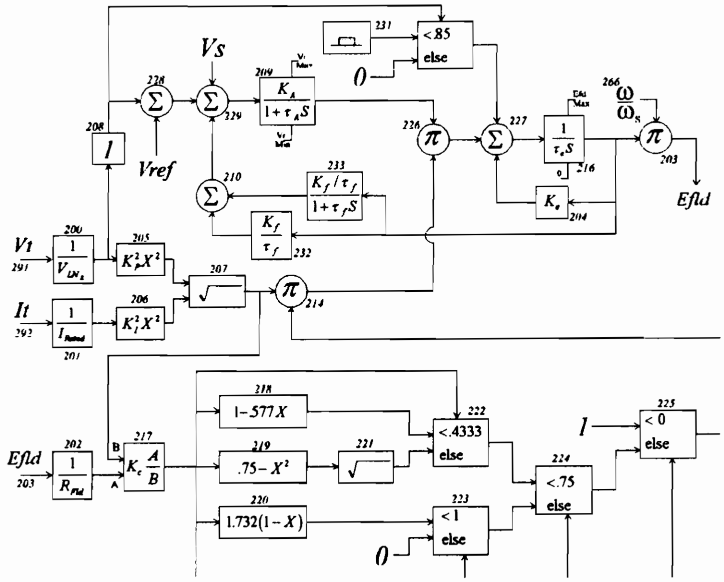


Figure 7.2 - Excitor Modeled Within Segflow

This entire excitor model was constructed dynamically in a Segflow prototype and contained within a single Segment object. Segflow offers powerful dynamic modeling capabilities.

Work is also being planned to investigate the behavior of Affiliation objects at points near power system instability. The required dynamic studies can be performed with very detailed machine models such as the one above.

7.4 Future Work With Parallel Processing

The object-oriented design of Segflow has a natural application to a parallel processing machine. A valuable contribution to the speed of the Segflow process would be such an application. Possibly the main Segflow algorithm described in Chapter 6 could occupy one processor while Segment and Affiliation objects occupied other processors.

Bibliography

[1] I. Grahm, *Object Oriented Methods.* 2nd Ed., Wokingham: Addison-Wesley, 1993.

[2] G. Booch, Object-Orieted Analysis And Design, 2nd Ed., Benjamin/Cummings, Redwood City, CA, 1994

[3] E.V. Berard, *Essays on Object-Oriented Software Engineering.* New Jersey: Prentice Hall, 1993.

[4] S.C. Dewhurst, K.T. Stark, *Programming in C++.* New Jersey: Prentice Hall, 1989

[5] J.C. Schryver, "Object-Oriented Qualitative Simualtion of Human Mental Modes of Complex Systems". *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 22, No. 3, 19993, 526-541.

[6] F.L. Alvarado, Y. Liu, "General Purpose Symbolic Simulation Tools for Electric Networks". *IEEE Transactions on Power Systems,* Vol. 3, No. 2, 1988, 689-697.

[7] R. Michalski, R. Stepp, *Learning from Observation: Conceptual Clustering in Machine Learning: An Artificial Intelligence Approach.* Palo Alto, CA: Tioga, P. 332, 1992.

[8] B. Stott, "Review of Load-Flow Calculation Methods". *Proceedings of the IEEE,* Vol 62, No. 7, 1973, 916-29.

[9] A.F. Neyer, F.F. Wu, K. Imhof, "Object-Oriented Programming for Flexible Software: Example of a Load Flow". *IEEE Transactions on Power Systems,* Vol. 3, No. 3, 1990, 689-695.

[10] B. Hakavik, A.T. Holen, "Power System Modelling and Sparse Matrix Operations Using Object-Oriented Programming". IEEE PES Summer Meeting paper #93 SM 491-1 PWR, Vancouver, B.C., Canada, July 18-22, 1993.

[11] H.H. Happ, "Diakoptics -- The Solution of System Problems by Tearing". *Proceedings of the IEEE,* Vol. 62, No. 7, 1973, 930-940.

[12] B.A. Carre', "Solution of Load-Flow Problems by Partitioning Systems into Trees". *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-87, No. 11, 1968, 1931-1938.

[13] J.O. Coplien, *Advanced C++ - Programming Styles and Idioms.* Massachusetts: AT&T/Addison-Wesley, 1992.

[14] R. Broadwater, J. Thompson, T.E. McDermott, "Pointers and Linked Lists in Electric Power Distribution Circuit Analysis", *Proceedings of 1991 IEEE PICA Conference,* Baltimore, Maryland, May 7, 1991, pp. 16-21.

[15] D.J. Kruglinski, *Inside Visual C++.* Microsoft Press: Redmond, 1993.

[16] *Electric Utility Engineering Reference Book,* Vol. 3, Distribution Systems. East Pittsburgh, PA: Westinghouse Electric Corporation, 1965.

[17] R.K. Ahuja, T.L. Magnanti, J.B. Orlin. *Network Flows- Theory, Algorithms, and Applications.* New Jersey: Prentice Hall, 1993.

[18] D. Shirmohammadi, H.W. Hong, A. Semlyen, G.X. Luo, "A Compensation-Based Power Flow Method for Weakly Meshed Distribution and Transmission Networks". *IEEE Transactions on Power Systems*, Vol. 3, No. 2, 1988, 753-762.

[19] G.X. Luo, A. Semlyen, "Efficient Load flow for Large Weakly Meshed Networks". *IEEE Transactions on Power Systems*, Vol. 5, No. 4, 1990, 1309-1315.

[20] R.P. Broadwater, A. Chandrasekaran, C.T. Huddleston, A.H. Khan, "Power Flow Analysis of Unbalanced Multiphase Radial Distribution Systems". *Electric Power Systems Research*, Vol. 14, 1988, 23-33.

[21] R.P. Broadwater, S. Rahman, H.E. Shaalan, R.E. Lee, "A Distribution Engineering Workstation for Undergraduate and Graduate Education". IEEE PES Winter Meeting paper 93 WM 124-8 PWRS, Columbus, January 31-February 5, 1993.

[22] R.P. Broadwater, L.V. Trussell, M.V. Ellis, J.C. Thompson, H.W. Ng, N.M. Singh, D.L. Loyd, "Application Programmer Interface for the EPRI Distribution Engineering Workstation.", submitted to 1994 T&D conference Chicago IL.

[23] W.H. Kersting, D.L. Mendive, "An Application of Ladder Network Theory to the Solution of Three-Phase Radial Load-Flow Problems". IEEE PES Winter Meeting paper #A 76 044-8, New York, January 25-30, 1976.

[24] W.H. Kersting, "A Method to Teach the Design of a Distribution System". *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-103, No. 7, 1984, 1945-1952.

[25] N.C. Shammas, *Advanced C++*. Sams: Carmel, 1992.

[26] R.H. Galloway, J. Taylor, W.D. Hogg, "New Approach to Power System load-flow analysis in a digital computer". *Proceedings of the IEE*, Vol. 117, No. 1, 1970, 165-169.

[27] L. Trussell, Final Report Submitted to Calvert Cliffs Nuclear Power Plant. August 1993.

[28] M.L. James, G.M. Smith, J.C. Wolford, *Applied Numerical Methods For Digital Computation*. Harper & Row: New York, 1985.

Vitae

Larry Trussell
3317 Circle Brook Dr. SW
Apt # J
Roanoke, VA 24014-1901
(703)772-7615 (H,Fax)

SUMMARY OF EXPERIENCE

EPRI Distribution Engineering Workstation at Virginia Tech *1992 - 1994*
Blacksburg, Virginia

DESIGNED AND CREATED IMPEDANCE, LOAD ESTIMATION, CAPACITOR DESIGN APPLICATIONS AND UTILITIES FOR GENERATING REPORTS, VIEWING SYSTEM MESSAGES AND CREATING STATISTICS. INITIAL WORK PERFORMED ON GAUSS-SEIDEL LOAD-FLOW AND MOTOR START SIMULATOR.

Baltimore Gas & Electric - Calvert Cliffs Nuclear Power Plant *1992 - 1994*
Lusby, Maryland

CREATED SOFTWARE TO SIMULATE THE DYNAMIC BEHAVIOR OF EMERGENCY DIESEL GENERATORS DURING ESSENTIAL LOAD STARTUP DURING LOSS OF COOLANT EMERGENCIES. DYNAMICS OF MECHANICAL AND ELECTRICAL SYSTEM WERE MODELED FROM COLD-START TO STEADY STATE.

Research of Nonlinear Oscillations in Power Systems *1992 - 1994*
Blacksburg, Virginia

INITIAL INVESTIGATION INTO THE ROLE OF AUTOPARAMETRIC RESONANCE IN SIMPLE POWER SYSTEM MODELS AND THE POSSIBLE RESULTING LOSS OF STABILITY.

Baltimore Gas & Electric - Calvert Cliffs Nuclear Power Plant *1992 - 1994*
Lusby, Maryland

CREATED SOFTWARE TO MODEL 200 BUS PLANT SYSTEM. DYNAMIC MODELS FOR STATIC VAR COMPENSATORS, FAST ACTING VOLTAGE REGULATORS, FAST (OR SOLID STATE) SWITCHED CAPACITORS, AND TRANSFORMER LOAD TAP CHANGERS WERE MODELED TO DEMONSTATE A MEANS TO MAINTAIN PLANT VOLTAGE SECURITY DURING VARIOUS SCENARIOS.

Los Alamos National Labs (Through NMSU) *1992 - 1994*
Las Cruces, New Mexico

CREATED SOFTWARE AND USED THE EPRI EMTP PACKAGE TO DETERMINE AND EVALUATE SETTINGS FOR A PROPOSED 60 MVAR STATIC VAR COMPENSATOR AND THE LOS ALAMOS LABS PARTICAL ACCELERATOR.

Southwestern Public Service Company *1992 - 1994*
Amarillo, Texas

CREATED SOFTWARE AND USED A COMMERCIAL ANALYSIS PACKAGE TO MODEL AND ANALYZE VOLTAGE PROBLEMS WITHIN SEVERAL THREE PHASE DISTRIBUTION SYSTEMS.

Pennzoil Sulphur Company Plant & Distribution Maintenance *1992 - 1994*
Pecos, Texas

ALTERNATING SUMMER POSITIONS ON PLANT ELECTRICIAN AND FIELD LINEMAN CREWS. EXPERIENCE BUILDING AND TROUBLESHOOTING MOTORS AND THEIR STARTERS, AND BUILDING AND MAINTAINING FIELD DISTRIBUTION NETWORK OVER A FIVE SQUARE MILE AREA.

EDUCATION

Virginia Polytechnic Institute & State University *1991 - 1994*
Blacksburg, VA

EARNED PH.D. IN ELECTRICAL ENGINEERING.

New Mexico State University *1990 - 1991*
Las Cruces, NM

EARNED MASTERS DEGREE IN ELECTRICAL ENGINEERING THROUGH THE ELECTRIC UTILITY MANAGEMENT PROGRAM.

New Mexico State University *1986 - 1990*
Las Cruces, NM

EARNED BACHEARLORS DEGREE IN ELECTRICAL ENGINEERING. GRADUATED WITH HONORS.

Carlsbad High School *1983 - 1986*
Carlsbad, NM

GRADUATED FROM HIGH SCHOOL WITH HONORS AND AS SALUTATORIAN.

HONORS

Eagle Scout	*1983*
Brotherhood - Order of the Arrow	*1983*
Deacon - First Presbyterian Church Carlsbad New Mexico	*1983*
Pennzoil Four Year Engineering College Scholarship	*1986*
A.J. Crawford Award for Saving the Life of Another	*1990*
Tomilson Fort $500 Power Engineering Paper Contest	*1991*
1991 EEI/PES Forgivable Loan Receipient	*1991*