MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

Memo 433

SLICES
At the Boundary between Analysis and Synthesis

by

Gerald Jay Sussman

Abstract:

The algebraic difficulty of determining the component values in a circuit of known topology and specifications is large. Expert circuit designers use terminal equivalence and power arguments to reduce the apparent synergy in a circuit so that their computational power can be focussed. A new descriptive mechanism, called slices, is introduced. Slices combine the notion of equivalence with identification of parameters. Armed with appropriate slices, an automatic analysis procedure, Analysis by Propagation of Constraints can be used to assign the component values in a circuit.

Techniques of formation, notation, and use of slices are described. The origin of slices in the topological design process is indicated. Slices are shown to be of wider interest in scientific thought than just in circuit analysis.

Acknowledgements:

Contents:

We consider circuit analysis to be an indispensible part of the education of a circuit designer. We feel that analytic techniques are a necessary component of synthetic reasoning. Why? What is the role of analysis in the process of design?

At first glance the answer seems obvious. Given a topology (how we get that is a different story[Context]) the problem of determining component values which make a circuit behave as specified seems to be a problem of analysis. For example, the problem of finding a resistor which permits a given current to flow at a given potential is identical to the problem of determining the current that flows given a resistance and a potential.

This view is largely illusory. It is algebraically infeasible to determine the component values by solving the set of equations and inequalities which result from matching the symbolic analysis of a circuit with the given design parameters. The complexity can be considerably reduced by considering the teleology of the components of the circuit -- how each component contributes to the goals of the overall design. This information will partition the constraints and reduce the apparent synergy. I will present a descriptive mechanism, called slices by which this information can be notated and communicated to either a person or a machine. In a fully automatic design setting, slices can be left as annotation on the circuit by the topological planning process. They can then be used by a symbolic analysis program[EL] to guide the assignment of component values.

Expert circuit designers make heavy use of terminal and port equivalences for focussing their attention. At any instant an expert will only concentrate on a small portion of the circuit, assuming that the rest will "work as planned". This assumption is captured by summarizing the behavior of the part of the circuit peripheral to the area under attention as a set of equivalences. Equivalences are also used to summarize the specifications of the behavior of the part under attention. This allows the designer to reason about the details of one section of the circuit in isolation. Few assumptions about the nature of adjacent parts of a circuit are allowed to migrate across the boundary of equivalences. From the point of view of circuit analysis slices model the way an experienced designer uses terminal equivalences to provide him with multiple local views of a circuit.

An Example

Consider a problem which I have used in class:[Class]

> WEEI-AM (590 KHz) needs a "dummy load" for use
> in the maintenance of their 50 KW transmitter. A
> dummy load is a resistor, connected to the output of a
> transmitter in lieu of an antenna, to allow
> adjustments and measurements to be made on the

transmitter under controlled conditions without radiating the test signals (which might interfere with other services). The only resistor they have which is capable of dissipating 50 KW is a 1 Ohm graphite block in a drum of oil (used to carry off the heat). The transmitter, however, needs a 50 ohm resistive load -- it is to be adjusted to operate into a 50 Ohm antenna. Design a circuit to match the transmitter to the 1 Ohm resistor.
(Hint: You need just one inductor and one capacitor.)

The first difficulty here is understanding the problem. Ignoring the difficulties of natural language and the side issues which make the problem "real" to the student, how can we formally specify the desired construct?

Equivalent Circuits as Summaries of Behavior

Just as we have summarized the behavior of the antenna at 590 KHz as a pure resistance of 50 Ohms we specify the behavior of the desired network in terms of an equivalent circuit. We want to construct a 2-port network, N, such that if it has a 1 Ohm resistor connected to its second port, the resulting 1-port network, N', is equivalent to a 50 Ohm resistor at 590 KHz. A further reasonable constraint is that N dissipates no power itself.
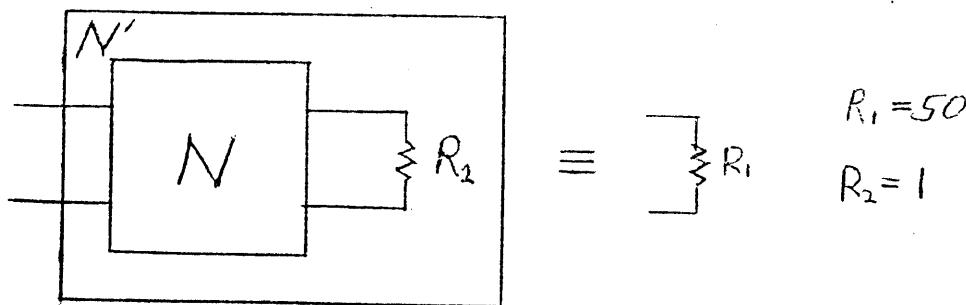
The Failure of Analysis

Let's ignore, for the nonce, the problem of finding a candidate topology. Suppose that somehow we are told that an appropriate circuit for N is an L-Network:
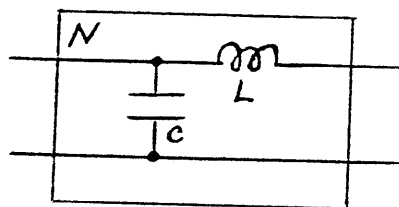
The problem is now reduced to one of determining values of L and C for which N displays the behavior desired.

Since the frequency is a constant of the problem, we can simplify things by working with the reactances rather than with frequency-dependent impedances. We define:

$$X_L = L\omega \quad , \qquad X_c = \frac{1}{C\omega}$$

Thus the real problem is one of finding the reactances.

We analyze the network N' to determine the impedance it presents.
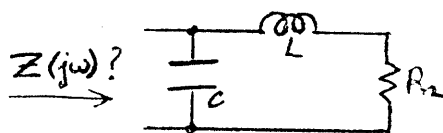
$$R_2 = 1 \, \Omega$$
$$\omega = 2\pi f \approx 3.7 \times 10^6 \, \frac{rad}{sec}$$

We can combine the impedances in series and parallel to get:

$$Z = \frac{-X_c j \, (X_L j + R_2)}{R_2 + (X_L - X_c)j}$$

Simplifying, we obtain:

$$Z = \frac{X_c X_L - R_2 X_c j}{R_2 + (X_L - X_c)j}$$

Separating the real and imaginary parts, we get:

$$Z = \frac{R_2 X_c^2}{R_2^2 + (X_L - X_c)^2} + \frac{-X_L^2 X_c + X_c^2 X_L - R_2^2 X_c}{R_2^2 + (X_L - X_c)^2} j$$

Our problem statement required the impedance to be R_1. Using our given numbers for R_1 and R_2, we get:

$$50 + 0j = \frac{X_c^2}{1 + (X_L - X_c)^2} + \frac{-X_L^2 X_c + X_c^2 X_L - X_c}{1 + (X_L - X_c)^2} j$$

Our complex equation reduces to two nonlinear equations in the two unknowns (X_L and X_C). Most engineering students can solve these, but we have only two undetermined parts. Suppose we were trying to design a T-section filter? The problem is hopeless, even with a powerful algebraic manipulator,[Algebra] without some better idea. Analysis for voltages and currents is often easy because of linearity but the component values are nonlinearly related to the specifications.

Knowing How It Works

The key to solving this problem lies in our knowledge of how an L-network accomplishes the goal of matching the impedances. Each part in the L-network contributes to accomplishing the goal.

Parallel Equivalent of a Series Circuit

Consider just the inductor-load combination:
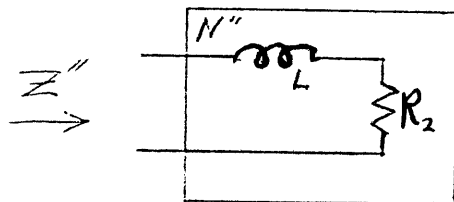
$$Z'' = R_2 + X_L j$$

If we look at the admittance rather than the impedance, we obtain:

$$Y'' = \frac{1}{Z''} = \frac{1}{R_2 + X_L j} = \frac{R_2}{R_2^2 + X_L^2} - \frac{X_L}{R_2^2 + X_L^2} j$$

This admittance is the sum of two terms. We can view this as a parallel combination of two admittances: a frequency-dependent conductance and a frequency-dependent inductive susceptance. At any particular frequency there is a resistor and an inductor whose parallel combination is equivalent to N":
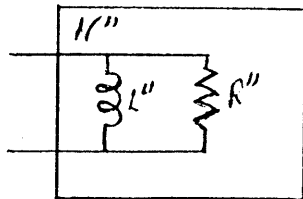

$$X_{L''} = \frac{R_2^2 + X_L^2}{X_L}$$

$$R'' = \frac{R_2^2 + X_L^2}{R_2}$$

Note that the equivalent parallel resistance, R", can be made larger than R_2 by adding in inductive reactance. In fact, we can make this apparent parallel resistance the 50 Ohms (R_1) required by the problem.

Now we see the _purpose_ of the parallel capacitor in the L-network. If we use it to resonate out the parallel inductance L" at the specified frequency, all that will be left is the 50 Ohm parallel resistance we constructed with the series inductor and load resistor combination.
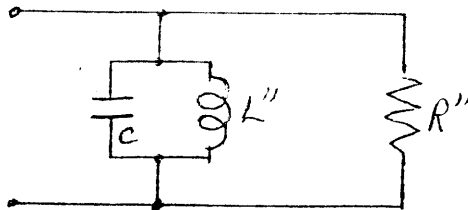
$$R'' = R_1$$

$$X_c = X_{L''}$$

(parallel resonance)

(The parallel combination of the capacitor and the apparent inductor is equivalent to an open circuit at the resonant frequency!)

With this understanding, the problem of finding appropriate values of L and C is significantly simplified. We can hope to have algebraic manipulators, both human and mechanical, which can solve the problem.

If we set R" = R_1 we can easily solve for X_L in terms of R_1 and R_2 at our desired frequency.

$$X_L = \sqrt{R_1 R_2 - R_2^2}$$

We then plug this result in to get the net parallel inductive reactance of network N".

$$X_{L''} = \frac{R_1 R_2}{\sqrt{R_1 R_2 - R_2^2}}$$

We must provide an equal capacitive reactance to balance this inductive reactance at the desired frequency. This condition determines the value of C.

$$X_c = \frac{R_1 R_2}{\sqrt{R_1 R_2 - R_2^2}}$$

Notice how smoothly the equations solve themselves. The trick is not being good at algebra but rather knowing what algebra to do. Each component in a circuit is used to provide for only part of the goal of that circuit. It is necessary to look at each part in its restricted context. In electrical circuits, this is often conveniently specified by means of equivalent circuits.

Power and Phasors

Another point of view which explicates the operation of our L-network involves power arguments and reasoning with phasors. We intend that N be such that N" looks like a 50 Ohm resistor R_1. But no power is dissipated in N; in fact all of the power apparently dissipated in R_1 is really dissipated in the 1 Ohm resistor R_2.

Now suppose the RMS voltage across N' is $|V_1|$. Then the power dissipated is $|V_1|^2/R_1$. But this power is really going into R_2 hence the RMS voltage across R_2 is:

$$\sqrt{\frac{R_2}{R_1}} \, |V_1|$$

R_2 and L are in series so the RMS voltage across L is:

$$\sqrt{|V_1|^2 - \frac{R_2}{R_1}|V_1|^2}$$

In addition, the current through R_2 (and L) is:

$$\sqrt{\frac{1}{R_2 R_1}} \; |V_1|$$

Hence, we can derive the reactance:

$$X_L = \sqrt{R_1 R_2 \left(1 - \frac{R_2}{R_1}\right)} = \sqrt{R_1 R_2 - R_2^2}$$

We could continue to extract the capacitive reactance...but later.


Expressing this with Slices

The moral of this story is that the activity of assigning component values for the parts of a circuit whose topology is known is not a simple task. To be able to accomplish it without unobtainable algebraic power requires an understanding of how the circuit works. This combines a diverse set of reasoning strategies and overlapping points of view. Some of these points of view are expressed by useful electrical port equivalences, and some are more global statements about the identity of parameters.

Electrical Engineering has provided us with the key concept of an equivalent circuit and several theorems which tell us about some useful types of equivalence[Equivalence] but it does not supply us with a notation for drawing the simultaneous multiple views of a circuit that make the use of equivalence so powerful. The best that one ever sees in engineering books is block diagrams, but the boxes I wish to draw would be so overlapped as to be difficult to draw in that way. I will now introduce a notation for formally describing these simultaneous slices,[Grey Boxes] so that they can be communicated to either a person or a computer involved in the design or analysis of circuits.[Pity Students]

Equivalence Slices

Usually in thinking of electrical networks we only consider one kind of connection -- Terminals of components are "soldered" together to form nodes which have definite node voltages and which conserve current. If I want to specify, in a schematic diagram, both a circuit and its equivalent, I cannot just connect them with the usual kind of solder because that would indicate that they are connected in parallel. I am forced to introduce a new kind of connection: I define 2 "wires" as being identified if they have the same voltage and the same current (in corresponding directions). Thus, we can indicate that two circuits are equivalent by identifying the corresponding terminals. For example, to indicate that network N has a Thevenin equivalent I would draw:
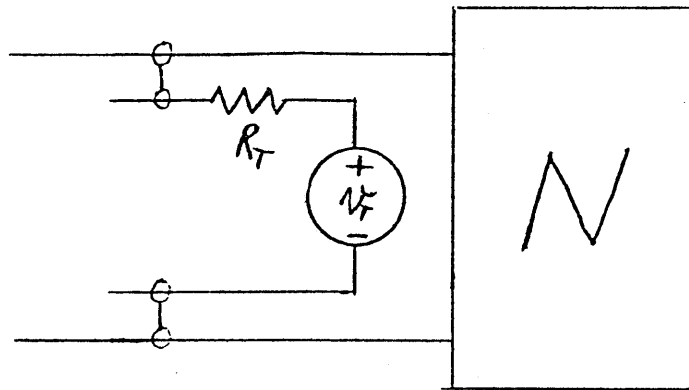
Any particular circuit diagram may contain several slices. Each slice involves two or more identifications of wires. Thus, to keep things straight, we label each identification with the name of the slice it is part of. Thus network N may have both a Thevenin (slice a) and Norton (slice b) equivalent:
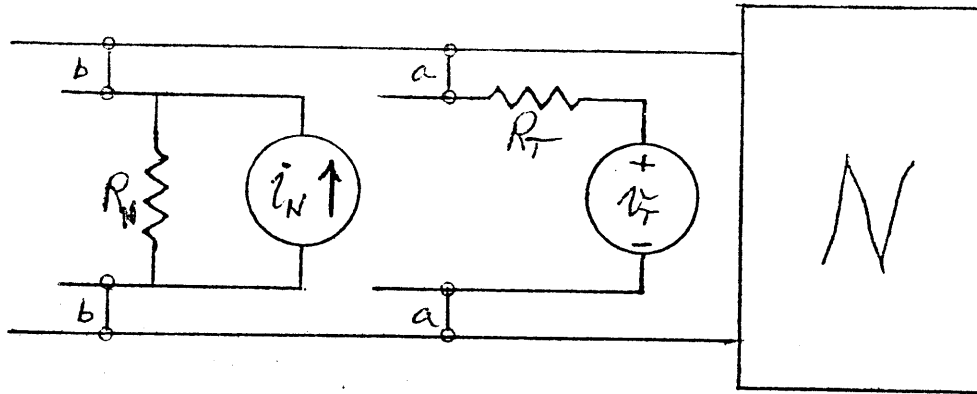
Parameter Slices

        Identifications of wires is not enough to express many constraints.
Sometimes I need to be able to say that two circuit parameters are to be identified. For
example, if two resistances are constrained to be the same or if the power dissipated in one
part is the same as the power dissipated in some other part. I use dotted lines to express
identifications between parameters. For example, to express the truth that $R_T = R_N$ in our
network above I would draw:
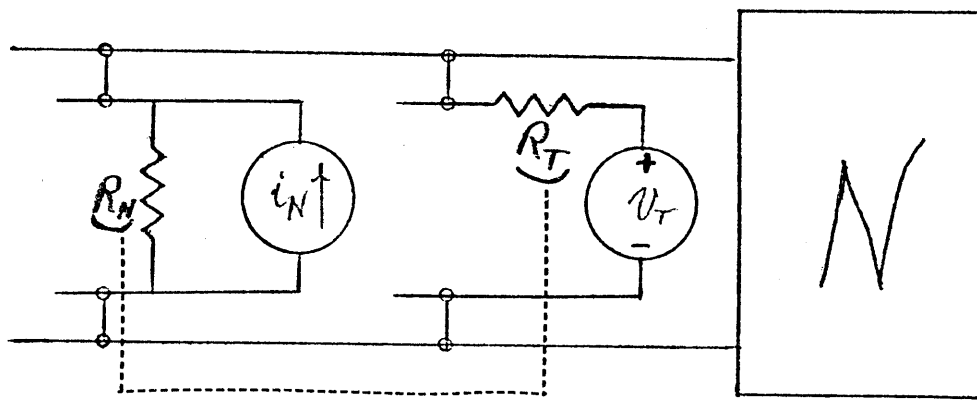
As you can see, here we have an identification which spans two slices. This is perfectly
legitimate. Once we have placed an slice on the circuit diagram it becomes part of the

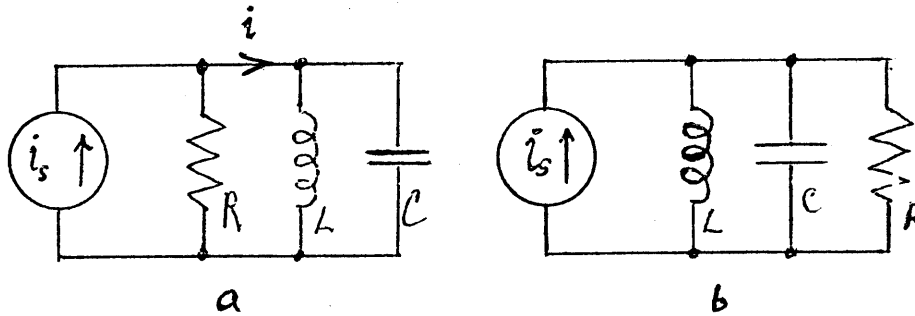diagram and further slices can include parts from it. We cannot construct, however, two distinct slices, each of which refers to parts "inside" the other.

The Problem of Nodes

There is another problem in standard electrical circuit notation which has to be overcome. In formal network theory the following diagrams are considered identical:

This identity is not reasonable to one who is interested in the current, i, indicated on diagram a. The topological abstraction of a node which conserves current and which has a definite potential is not the correct representation of the circuit from his point of view. I thus consider connection points to be components with simple laws, and I give wires formal status. Thus we may have several equivalent circuits of the "same" (network theoretic) node which are used when we wish to talk about different groupings of the currents.

The L-Network Plan

I now use these techniques to present the complete set of slices which constitute the rationale for the design of an L-network impedance matcher.
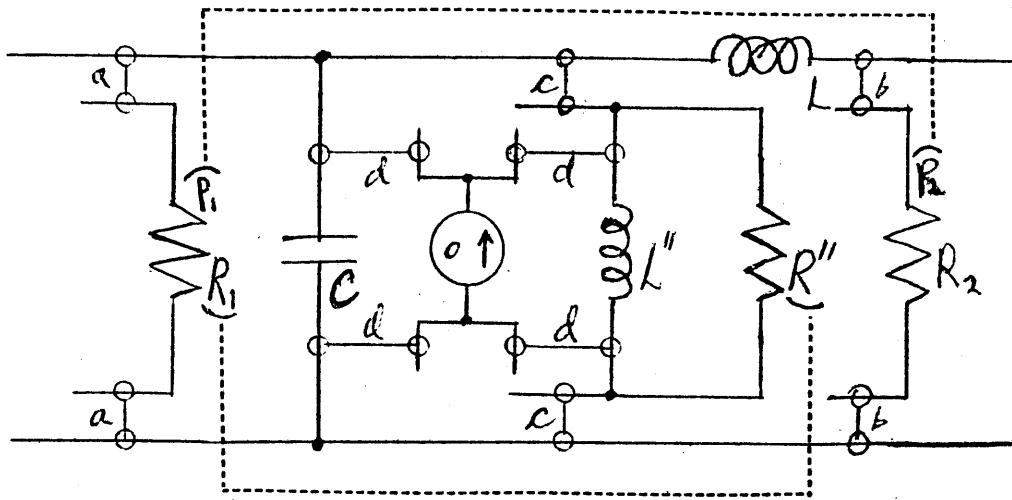
   I have tried to capture in this diagram all of the basic relationships which explain how the L-network is intended to work. We see the following facts indicated:

     The network is expected to be connected on its right
     to a circuit which is equivalent to a resistance R_2.

     The network presents a resistance R_1 to the port on
     its left.

     The series combination of L and R_2 is equivalent to
     the parallel combination of L" and R".

     R" has the same value as R_1.

     The parallel combination of L" and C is equivalent
     to a current source of zero current (an open circuit).

     The power dissipated in R_1 is the power dissipated
     in R_2.

   Some of the slices express information not available from the basic circuit diagram. For example, the slices a and b, which attach the equivalent resistances R_1 and R_2 to the circuit form the specifications of the L-network. The power relationship and the parallel equivalence are redundant as they could be deduced by a sufficiently powerful circuit analyzer.

Synthesis by Analysis

Aided by the slices we can now apply an automatic analytic procedure, Analysis by Propagation of Constraints, to determine the component values of the L-network.[EL Can't] We terminate the L-network with a 1 Ohm resistor on its right and we specify that its input impedance be a 50 Ohm resistance:
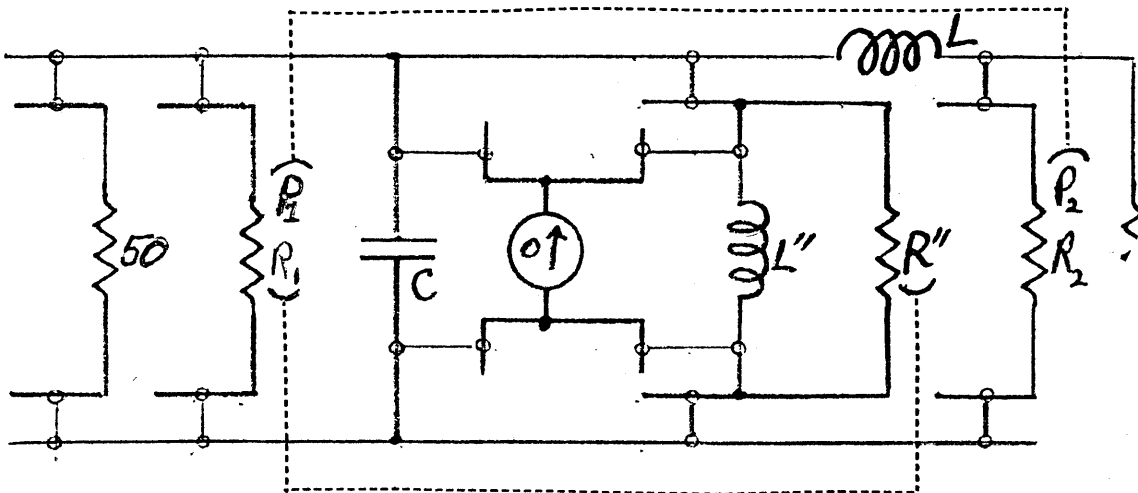
Suppose we knew the voltage-amplitude, V_2, (relative to the ground indicated) at the top of the 1 Ohm load, then the current (amplitude) flowing down through the load is V_2. This same voltage and current appear on the terminals of R_2 so we can deduce in one step that $R_2 = 1$ Ohm. This current also flows through the inductor, L; it enters the left side of L. (Thus the parallel slice has this same current entering it.) Now the power dissipated in R_2, $P_2 = |V_2|^2$. This power must be dissipated in R_1. Suppose we also knew the voltage V_1 at the top of R_1. Thus by a similar argument to the one above, we can deduce that $R_1 = 50$ Ohms. Since we know the power dissipated in this 50 Ohm resistance, we can deduce that the magnitude of the voltage at the top of R_1 is SQRT(50) * $|V_2|$.

At this point we know the voltage amplitude at the right of the inductor, L, the current amplitude through it, and the magnitude of the voltage on its left. From this we deduce its reactance:

$$X_L = 7 \text{ Ohms.}$$

Using this we can get the full voltage amplitude on the left of L in terms of that on the right:

$$V_1 = V_2 * (1 + 7 * j)$$

This voltage also appears at the top of R_1. We use this to get the current through R_1 (which is the same as the current into N'). It is:

$$V_2 * (1/50 + (7/50) * j)$$

From this we compute the current into the capacitor:

$$V_2 * (- 49/50 + (7/50) * j)$$

But we know the voltage across the capacitor and the current through it; hence the reactance

$$X_C = 50/7 \text{ Ohms.}$$

Note that we didn't use the parallel slice. Other interesting derivations pass through that route. The slices are redundant descriptions -- the same truths from a variety of viewpoints. The way that the slices do their job, however, is <u>by providing redundant paths</u> for information to travel in the process of analysis. Thus, in the power argument above, the power dissipation slice, though redundant (It is derivable using only the local knowledge that inductors and capacitors dissipate no net power.), made it possible for enough information to get from the right side of the inductor to its left side to determine the inductive reactance.

Knowing the Form of the Answer

I have shown how slices can be used to express some of the knowledge of how a circuit is intended to work by describing alternate views of sections of the circuit. I have shown how this knowledge can help in the process of synthesis <u>if we know the topology and the correct slices to use.</u> Where does this information come from?

To answer this question in detail requires describing a theory of the design process. I can outline such a theory in general terms. The key idea is that of <u>knowing the form of the answer.</u> When presented with a simple problem, like a radio frequency impedance match, an engineer is likely to be able to retrieve such plausible solutions as the L-network and the transformer directly. If a more complex problem is posed, such as a 30 dB gain broadband amplifier with an input and output impedance of 600 Ohms, capable of output signals with 1 Volt of swing, the engineer does not retrieve a detailed answer but rather a high-level plan such as:
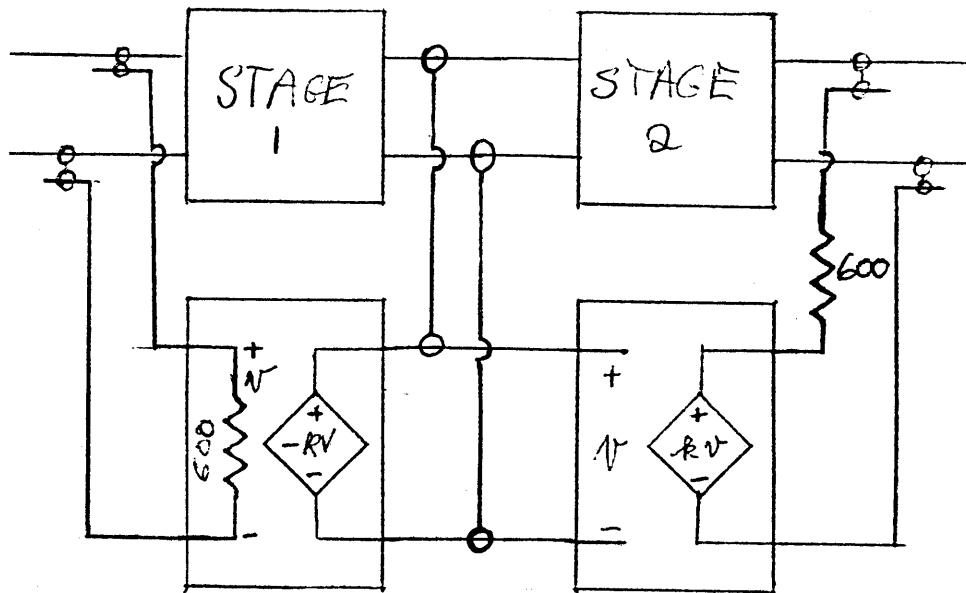
This has the effect of decomposing the problem into a set of relatively independent subproblems. In fact, there are always complex interactions among the parts which lead to bugs and debugging.[PSBDARP] But the form of the answer is set. The problem is reduced to one of finding a feasible solution for each subproblem represented by the boxes. Each subproblem is constrained by the laws of electricity and the slices present on the plan. Each subproblem is further expanded until one gets to atomic problems.[Refinement]

I believe that the slices are an integral part of the forms of solution for the different problem classes stored in the engineer's bag-of-tricks. When solutions to subproblems are combined, as above, the slices on the subsolutions are combined with the slices on the higher level plan. In addition, new patterns are formed by debugging of designs and by certain window optimizations. For example, if the solution of each of two subproblems leads to an inductor in series with a port, and the two subsolutions are combined according to a Cascade plan
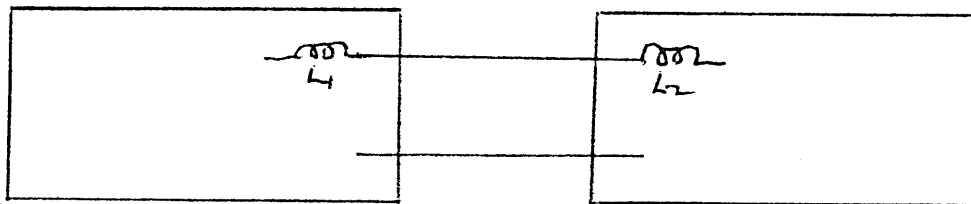
it is recognized as an optimizable pattern and replaced by one inductor which performs the duties of both subinductors:
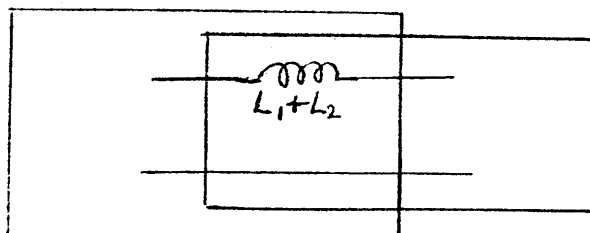
The old circuit remains as a set of slices explaining the new part!

Knowing the form of the answer is not unique to engineering problem solving. It is very common in mathematics.[Edge] For example, if we have a Homogeneous Linear Differential Equation with Initial Conditions we know that the form of the answer is $A\,e^{kt}\cos(wt + p)$. We can plug this in and solve for A,k,w,p. The algebra is, however, pretty rough. If we know how the parts of the answer are related to the problem -- that w and k are functions of the differential equation only and A, p are derived from the effects of the initial conditions -- things are algebraically more feasible. These facts can be represented as "algebraic slices".

In conclusion, I believe that the concepts of "The Method of Knowing the Form of the Answer" coupled with and refined by "Slices" is a step toward developing an Epistemology of Engineering appropriate for aiding synthesis and design.

Notes and Relation to Other Work

Context

This work is part of a larger effort to apply artificial intelligence methods to computer-aided design at the MIT AI Laboratory. An overview of the work is in [Sussman 1977]. Drew McDermott [1976, 1977] has made considerable progress on the problems of rephrasing problem descriptions and proposing plausible designs. Allen Brown [1975] and Johan de Kleer [1976] have investigated the problems involved in troubleshooting circuits -- with and without access to a plan. Richard Stallman and I have designed and implemented some novel circuit analysis programs [Sussman and Stallman 1975] [Stallman and Sussman 1976]. Johan de Kleer [1977] is considering the problem of recognition of the plan of a circuit given the circuit diagram. We are not solely interested in computer-aided circuit design, but rather in an understanding of the general epistemology of engineering. We are also engaged in a substantial effort on the computer-aided design of programs. See [Rich and Shrobe 1977], [Waters 1976].

EL

Sussman and Stallman [1975, 1976] have developed a rule-based system for computer-aided circuit analysis. The set of rules, called EL, is written in a rule language called ARS. Rules are implemented by ARS as pattern-directed invocation demons monitoring an associative data base. Deductions are performed in an antecedent manner, giving EL's analysis a catch-as-catch-can flavor suggestive of the behavior of expert circuit analyzers. We call this style of circuit analysis propagation of constraints. The system threads deduced facts with justifications which mention the antecedent facts and the rule used. These justifications may be examined by the user to gain insight into the operations of the set of rules as they apply to a problem. The same justifications are used by the system to determine the currently active data-base context for reasoning in hypothetical situations. They are also used by the system in the analysis of failures to reduce the search space. This leads to effective control of combinatorial search which we call dependency-directed backtracking.

De Kleer, Doyle, Steele, and Sussman [1977] have been developing an even more powerful rule-based language which we call AMORD in which the EL rules can be expressed in a more hierarchical form. AMORD depends upon a generalization of the ARS dependency structure developed by Doyle [1977] to enable the exercise of more explicit control of the reasoning process.

Class

I am one of the faculty in charge of the introductory subject in Electrical Engineering at MIT, Introductory Network Theory. I find the relationship between teaching introductory engineering subjects and research in Artificial Intelligence rewarding. Teaching novel ideas to humans is like teaching them to machines. Often a new way of explaining an old idea becomes a program and the reverse is also true. Observing expert circuit analyzers solve networks, so that students could be taught their expert methods, led

to the development of analysis by propagation of constraints. This became EL. EL style analysis is now being taught to our students.

Algebra

Although there are some very powerful algebraic manipulators available [MACSYMA 1973], it is unreasonable to expect them to help us out by symbolically inverting the equations of analysis. Suppose that the problem was an 8-section lattice filter (32 part values to determine -- nonlinearly related to each other and to the specifications)? In general, it is more helpful to have knowledge which tells us how to avoid manipulation than it is to have powerful manipulators.

Believing in the ultimate power of mathematical manipulation is one of the most common difficulties encountered by students learning electrical circuit analysis. Students often grind out "impossible" algebra in the course of solving a homework problem, even though a little thought will reveal an algebraically feasible approach which depends upon a small insight into the operation of the network being analyzed. They then complain that we give them too much homework!

Equivalence

In electrical engineering, most equivalences, such as the Thevenin/Norton theorems and the two-port results depend critically on the linearity of the circuits they summarize. Thus they are severely limited in applicability. Engineers often compensate for this limitation by considering perturbations. This "small signal analysis" is critical to the design of complex signal processing systems.

Grey Boxes

Stallman and Sussman [1976] introduced the "grey box", a predecessor of the slice idea. Grey boxes also attached to a circuit diagram by identification of terminals. Grey boxes are additional "laws" which bypass the deductive machinery and present the answer for certain otherwise unfeasible deductions. For example, the incremental gain of an emitter-coupled pair is impossible to deduce without exponential algebra. These additional laws allow EL to "know" some facts about a circuit construct in an implicit manner. Grey boxes, however, have no substructure so they cannot really provide an alternate electrical view of a circuit. Slices, on the other hand are "real" electrical subdiagrams. They can contain further slices describing their structure. In addition, slices can share parts and can identify wires in two different other slices. One of the primary reasons for designing AMORD is to investigate the applicability of multi-level slices which are inherited with a circuit type.

Pity Students

One difficulty with current teaching in Electrical Engineering (and other disciplines) is the lack of formal ways of communicating "intuitive" knowledge. Much of this knowledge is procedural in nature. The spread of "computer culture" will enhance our ability to express this kind of knowledge. Slices are a way of expressing more traditional

intuitive knowledge about how a circuit works.

EL Can't

No current implementation of EL can actually handle the propagations described in this section because the algebraic manipulation section of EL cannot yet deal with complex numbers.

PSBDARP

We believe that many bugs are just manifestations of powerful strategies of creative thinking -- that creation and removal of bugs are <u>necessary</u> steps in the normal process of solving a complex problem. Following the work of Polya [Polya 1962], recent research [Fahlman 1973] [Sussman 1973] [Goldstein 1974] predicated on this belief has resulted in the development of a paradigm for problem solving which we call <u>Problem Solving by Debugging Almost-Right Plans</u> (PSBDARP). We believe that the PSBDARP theory is a good foundation for building expert problem-solving systems for such diverse kinds of engineering as circuit design [Sussman 1977] and computer programming.

Refinement

The idea of successive refinement of plans appears as a key dogma of "Structured Programming" [Dijkstra 1970] [Wirth 1971] [Dahl et al 1972], although it also appears in the Artificial Intelligence problem-solving literature. The idea of relaxation of a hierarchy of constraints comes from [Freeman & Newell 1971]. There are also versions of GPS [Ernst & Newell 1969] which were purported to do reasoning in a hierarchy of abstraction spaces. ABSTRIPS [Sacerdoti 1973] showed how refinement of abstract plans could be used to guide a problem solver past problems which would otherwise be combinatorially explosive. Recently the NOAH system [Sacerdoti 1975] has developed this idea to great depth.

Edge

Perhaps the best example of a program which uses The Method of Knowing the Form of the Answer is SIN [Moses 1967]. This symbolic integration program classified an expression to be integrated according to certain criteria which allowed it to select a general form for the answer and then solve for the details of the undetermined coefficients. SIN did not, however, use slices to help in the solution process.

Bibliography

Brown, Allen (1975) <u>Qualitative Knowledge, Causal Reasoning, and the Localization of</u>
<u>Failures</u>, Cambridge: MIT Ph.D. Thesis, also MIT Artificial Intelligence
Laboratory Technical Report ?

Brown, John Seely and Richard R. Burton (1975) "Multiple Representations of Knowledge
for Tutorial Reasoning" in Daniel Bobrow and Allan Collins (Eds.),
<u>Representation and Understanding: Studies in Cognitive Science</u>, New York:
Academic Press.

Dahl, O.J., E.W. Dijkstra and C.A.R. Hoare (1972) <u>Structured Programming</u>, London:
Academic Press.

de Kleer, Johan (1976) <u>Local Methods for Localization of Faults in Electronic Circuits</u>,
Cambridge: MIT Artificial Intelligence Laboratory Memo 394.

de Kleer, Johan (1977) <u>A Theory of Plans for Electronic Circuits</u> MIT Artificial Intelligence
Laboratory Working Paper 144. (May 1977)

de Kleer, Doyle, Steele, and Sussman (1977) <u>Explicit Control of Reasoning</u>, MIT Artificial
Intelligence Memo 427. (June 1977), To be delivered at AI&PL, Rochester, NY.
(August 1977)

Dijkstra, Edsgar (1970) "Structured Programming" in <u>Software Engineering Techniques</u>,
J.N. Buxton and B. Randell (Eds.), NATO Scientific Affairs Division,
Bruseels, Belgium, 1970.

Doyle, Jon (1977) <u>Truth Maintenance Systems for Problem Solving</u>, MIT Artificial
Intelligence Laboratory Technical Report 419 (June 1977).

Ernst, George W. and Allen Newell (1969) <u>GPS: A Case Study in Generality and Problem-</u>
<u>Solving</u>, New York: Academic Press.

Fahlman, Scott (1973) <u>A Planning System for Robot Construction Tasks</u>, Cambridge: MIT
Master's Thesis.

Freeman, P. and Allen Newell (1971) "A Model for Functional Reasoning in Design,"
<u>Proceedings of International Joint Conference on Artificial Intelligence II</u>, p.
621.

Goldstein, Ira (1974) "Summary of MYCROFT: A System for Understanding Simple Picture

Programs," <u>Artificial Intelligence</u>, Vol. 6, No. 3, Fall, 1975, pp. 249-288.

Latombe, Jean-Claude (1976) <u>Artificial Intelligence in Computer-Aided Design: The Tropic
System</u>, Menlo Park, CA: Stanford Research Institute Artificial Intelligence
Center Technical Note 125.

MACSYMA (1973)
Bogen, R. et al, <u>The MACSYMA User's Manual</u>, MIT Project MAC
publication.

McDermott, Drew (1976) <u>Flexibility and Efficiency in a Computer Program for Designing
Circuits</u>, Cambridge: MIT Department of Electrical Engineering and Computer
Science Ph.D. Thesis.

McDermott, Drew (1977) "Vocabularies For Problem Solver State Descriptions", <u>Proc. IJCAI
V.</u> (August 1977)

Moses, Joel (1969) <u>Symbolic Integration</u>, MIT Project MAC Technical Report 47.

Polya, George (1962) <u>Mathematical Discovery: on Understanding, Learning, and Teaching
Problem Solving, Vols. I and II</u>, New York: John Wiley and Sons, Inc.

Rich, Charles and Howie Shrobe (1976) <u>Initial Report on a LISP Programmer's Apprentice</u>,
Cambridge: MIT Artificial Intelligence Laboratory Technical Report 354.

Sacerdoti, Earl (1973) "Planning in a Hierarchy of Abstraction Spaces," <u>Proceedings of the
Third International Joint Conference on Artificial Intelligence</u>, August, 1973, pp.
412-422.

Sacerdoti, Earl (1975) <u>A Structure for Plans and Behavior</u>, Menlo Park, CA, Stanford
Research Institute Artificial Intelligence Group Technical Note 109, also New
York: American Elsevier Publishing Company (1977)

Simon, Herbert (1969) <u>The Sciences of the Artificial</u>, Cambridge: MIT Press.

Stallman, Richard and Gerald Jay Sussman (1976) <u>Forward Reasoning and Dependency-
Directed Backtracking in a System for Computer-Aided Circuit Analysis</u>,
Cambridge: MIT Artificial Intelligence Laboratory Memo 380. (To appear in
the AI Journal)

Sussman, Gerald Jay (1977) "Electrical Design -- a Problem for Artificial Intelligence
Research", MIT Artificial Intelligence Memo 425. Also in <u>Proc. IJCAI V.</u>,
(August 1977)

Sussman, Gerald Jay (1974) "The Virtuous Nature of Bugs", Proceedings of the Artificial Intelligence Simulation of Behavior Conference, Sussex, U.K.

Sussman, Gerald Jay (1973) A Computational Model of Skill Acquisition MIT Dept. of Math. PhD Thesis, MIT Artificial Intelligence Laboratory Technical Report 297 (August 1973), also A Computer Model of Skill Acquisition, New York: American Elsevier Publishing Company (1975).

Sussman, Gerald Jay and Richard Stallman (1975) "Heuristic Techniques in Computer-Aided Circuit Analysis," IEEE Transactions on Circuits and Systems, Vol. CAS-22, No. 11, Nov., 1975.

Waters, Richard C. (1976) A System for Understanding Mathematical FORTRAN Programs, MIT Artificial Intelligence Laboratory Memo 368 (August 1976)

Wirth, N. (1971) "Program Development by Stepwise Refinement," Communications of the ACM, Vol. 14, No. 4, April 1971.