

WORKING PAPER 61

QUALITATIVE KNOWLEDGE, CAUSAL REASONING,
AND THE LOCALIZATION OF FAILURES

by

ALLEN L. BROWN

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

March, 1974

Abstract

A research program is proposed, the goal of which is a computer system that embodies the knowledge and methodology of a competent radio repairman.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0005.

Working Papers are informal papers intended for internal use.

Qualitative Knowledge, Causal Reasoning,
and the Localization of Failures
--A Proposal for Research

Whither Debugging?

The recent theses of Fahlman <Fahlman>, Goldstein <Goldstein>, and Sussman <Sussman> document programs which exhibit formidable problem-solving competences in particular micro-worlds. In the last instance, the problem-solving system begins as a rank novice and proceeds, with suitable "homework," to improve its skill. In these problem-solving systems, a number of distinct, but by no means independent, components are identifiable. Among these are planning, model-building, the formulation and testing of hypotheses, and debugging. It is debugging as a problem-solving faculty, that will be the focus of attention of our research.

Goldstein and Sussman, in automatic programming, and Papert <Papert>, in education, have more than adequately argued the importance of a bug as a Powerful Idea, and debugging as an important intellectual process. Despite these important insights into the role of debugging in various intellectual endeavors, we are left, nonetheless, with inadequate models of the debugging process and a less than complete characterization of any category of bugs. In order to make significant progress in the theory of debugging, we should attempt to study in depth some particular, well-defined class of bugs. Indeed the problem with Goldstein's and Sussman's efforts from our point of view is that they attempted the full gamut of problem-solving activity rather than simply debugging, thereby muddying the issues that we should like to see clarified. We further contend that the debugging of programs is still

too difficult a task for us to deal with successfully. First of all, program debugging unavoidably requires design, as the structure of the patch does not follow immediately from the underlying cause of the bug. Furthermore, bugs that involve shared components (with which programs are rife) seem sufficiently complicated such that we should postpone investigating them until we thoroughly understand the bugs of systems in which there is no such sharing.

Desiderata for Debuggability

How then are we to do a deep study of bugs? Let us first consider what sort of debugging we should like to investigate and then, attempt to design a micro-world that supports it! The aspect of debugging to which we should like to address our attention is the localization of failures to a particular module of some mechanism, without concerning ourselves with how to fix it (i.e. the patch is essentially the replacement of the module). Now we are in a position to enumerate some desiderata of the micro-world in which we wish to do debugging. The essential problem-solving task in this world is to be the localization of failures in some class of mechanisms. Consequently, those mechanisms must admit local failures. Bugs in these mechanisms should be separable into failures of design and failures of components. The components themselves should not be type-shared.

We distinguish two kinds of sharing: type-sharing and token-sharing. Subroutines are examples of type-sharing, wherein a given chunk of code is activated from many different points. A line of code with several purposes for several different (other) lines is an example of token-sharing.

Component failures should be an important class of bugs among these mechanisms, thereby enabling us to study local debugging techniques.

The micro-world should come equipped with a ready-made technical

language that supports structural and teleological descriptions of the class of mechanisms in the micro-world.

The Electronics Repairman

Recently Sussman <Winston> has proposed the implementation of an electronics repairman. Briefly, the problem domain is that of electronic circuits which realize various devices. Initially, at least, we imagine that the members of the class of electronic devices in question can be assumed to be of correct design and have become inoperative due to component failure. The repairman is given a schematic of the failing device, plus suitable description of the device's public extrinsic and intrinsic properties.

By an "extrinsic description" of a module we shall mean a description that explains the use or purpose of that module in some larger structure. By an "intrinsic description" of a module we shall mean a description that tells what a module is by virtue of its physics. For example, if the device were an amplifier, its extrinsic description might entail "audio amplifier for a public address system," while its intrinsic description might include "audio amplifier" with various graphs showing its frequency response, power output, output impedance, etc.

From this schematic, the repairman should be able to produce the plan of the device (à la Goldstein). He knows about typical electronic subsystems, having extrinsic and intrinsic descriptions of them.

Consider the single tube AM radio receiver diagrammed in Figure 1. We may identify several components, which we have given subscripted letter labels. A description of this device is given by the part(s) name and specification (intrinsic description) followed by its(their) purpose (extrinsic description):

- L_0 , several turns of wire, acting as a link coupling the input signal to the input tank;
- L_1 , an inductance (200 μ H), acting as the inductance of the input tank;
- L_0-L_1 , transformer, acting as a port between the input signal and the input tank;
- C_1 , variable capacitor (365 pf Max.), acting as the tuning capacitor of the input tank;

the input tank--selects the signal;
 C_2 , fixed capacitor (120 pf), acting as the grid blocking capacitor (passes the signal but not the grid DC);
 R_1 , resistor (5 M Ω), acting as the grid leak resistances;
 R_1-C_2 , act as grid bias circuit--a grid leak detector "trick;"
 V_1 , triode tube, acting as a nonlinear element (AM detector)
 R_L , load resistor, acting as internal resistance of earphone;
 E_1 , earphone (crystal), acting as the signal output;
 B, battery (22.5 V), acting as anode bias.
 This is the sort of knowledge that the repairman must have at his finger-tips in order to understand a particular electronic device.

An additional part of the description of any subsystem is its expansion into component subsystems, again with appropriate descriptions attached. (Observe that L_1-C_1 above is such a subsystem.) By suitable pattern-matching techniques, the repairman resolves the modules he knows against the schematic he has been given and produces a parse-tree of the device. The parse-tree corresponds to the expansion of the device into sub-modules (the nodes of the tree) where each sub-module has been assigned (by the parsing) appropriate extrinsic and intrinsic descriptions. The terminal nodes of the tree correspond to atomic components such as resistors, capacitors, wires, etc. The problem-solving task is to localize the failure(s) to some atomic component(s) (barring the moving of wires, etc.). The problem is clearly solvable since an exhaustive search of all atomic components, examining each to see if it matches its intrinsic description, must eventually isolate the fault(s). The debugging question here concerns the use of extrinsic descriptions of the various sub-modules to guide the search, thereby drastically limiting the effective size of the search space.

The Debuggability of Electronic Devices

Does the world of the electronics repairman satisfy the criteria set forth earlier? We claim that it does. Issues of design and repair

have been separated by removing the design process from the scene altogether (at least in the initial stages of the repairman's career). The kinds of bugs that the early repairman will encounter are of a local nature owing to the fact that a well-designed electronic device with component failure is analogous in behavior to an open-coded program, some of whose instructions are failing to behave as specified in the documentation.

Sharing in the repairman's world is limited to token-sharing because there is no design occurring. In the world of the electronics designer a change in a module design procedure will produce a change in every instance of those modules, hence type-sharing.

There is a large class of bugs of this sort among electronic devices to be catalogued along with procedural information concerning their detection.

In an actual protocol for trouble-shooting superheterodyne receivers <Tepper> we find an enumeration of symptoms paired with a possibilities list of underlying (local) causes. For example:

If the receiver is not dead and there is no audible signal other than audio-frequency noise of the "hum" variety, then the trouble is likely to be a defective filter or open grid return.

If the receiver is dead, the tube heaters and pilot lights are not lit, and the receiver in question is AC, then the trouble is likely to be a single tube heater that has opened.

We have available a large collection of such manifestation-of-bug/underlying-(local)-cause pairs waiting to be organized into a structure useable by a program.

The debugging system would use the natural structural hierarchy of the parse-trees, presumably attempting to guess as high as possible in the tree (at the highest possible level of abstraction) the underlying cause of the bug (the lowest level of abstraction). We may view the repairman's initial debugging methodology to be extensible to other classes of bugs: errors or incomplete design specification and bugs over electronic systems with "extended" token-sharing.

Surely the same sort of qualitative reasoning that allows us to discover malfunctioning components will also allow us to discover local design errors that take the form of incorrect values of resistances, capacitances, inductances, etc. Furthermore, tutorial texts such as <Marcus> would indicate that a great deal of design (such as the "invention" of the grid blocking capacitor and the grid leak resistor) can be done by the sort of qualitative reasoning that we intend to use in the debugging of local failures. Similarly, we think that bugs in electronic systems with extended token-sharing (such as the use of the BFD in the product detector of a communications receiver both for single-side-band and CW reception) will yield to local debugging techniques as well, since such shared components have only one purpose at a given time.

We assert that the world of electronic devices is semantically rich.

By a "semantically rich" world we mean one which can be described from many points of view. The canonical example of such a world is elementary plane geometry having the following representations: physics (mechanistic), algebra (analytic), Euclidean (synthetic), and LOGO turtle drawings (differential). Another example of a semantically rich world is thermodynamics which classically has had macroscopic and microscopic viewpoints, but is further enriched by "heat engine," statistical, and quantum mechanical notions of thermodynamic processes.

There is a seventy-five year development of technical language whose purpose is to render qualitative descriptions of electronic devices.

Recall our description of the single tube AM receiver diagrammed in Figure 1. The language used in the extrinsic descriptions is the traditional technical language. Given this qualitative description and qualitative physical knowledge about what various components do, a reasonable technician can, in fact, deduce the behavior of the device so described.

By suitable readings in the electronics literature and by observing our own debugging techniques, we should be able to design effective data structures that reflect this technical language and embody the underlying knowledge.

A Methodological Example

In the following, we should like to present a scenario of the electronics repairman in action. This scenario is meant to illustrate

our current thinking about the repairman's top-level procedure. The scenario will emphasize causal reasoning in fault isolation and techniques of fault isolation (including such methods as signal injection, signal tracing, resistance readings, voltage readings, isolation of faulty stages, isolation of faulty circuits, etc.)

We suppose that the repairman is faced with an AM superheterodyne radio receiver (realized by the standard five-tube AC-DC circuit) whose presenting symptom is that loud signals are distorted. We suppose further that the underlying cause of this problem is a shorted AVC filter capacitor. In an attempt to discover the underlying cause, the repairman examines his fully expanded model (as per our previous discussion) of the superheterodyne receiver. The highest level of abstraction in this model is presented in Figure 2. We shall not worry here about how he acquired this expanded model. Let it suffice to say that this level of abstraction is quite general and, indeed, covers a very large class of AM radio receivers. As we noted earlier, this model (parse tree) will distinguish submodules, each of which has ports and further internal structure. We see that the model distinguishes the basic signal path from control paths and power paths. Each submodule is similarly specified. We show two possible converter modules (Figures 3 and 4). The first is appropriate to most broadcast receivers, while the second might be found in communications equipment. This hierarchical model has further expansion, and will include all of the descriptive and connective information that we have mentioned previously.

Let us now consider the reasoning of our repairman. Distorted sounds on loud signals indicate that some module in the basic signal path is not linear with respect to the audio component of the signal. Where is the distortion introduced? The repairman instructs his human

assistant to set up the radio on the workbench under operating conditions. The radio is tuned to the output of an amplitude modulated RF signal generator which is adjusted to produce a signal strong enough to cause the problem. The repairman now traces the signal path back from the speaker. He asks his assistant to "scope" the audio output and determine if distortion is present--it is. He then asks about the output of the detector, which again yields an affirmative answer. In fact, he finds that the distortion originates in the IF amplifier because its output, but not its input, is distorted. Is the problem in the IF amplifier?

He next requests his assistant to examine the other inputs (prerequisites) to the IF amplifier so that he may decide if they are reasonable. He finds that the power supplied is correct, but that the AVC line is at 0 Volts. The repairman asks his assistant to vary the output strength of the signal generator. Something is wrong since the repairman discovers the AVC signal to remain invariant. He considers the consequences of the AVC being held at 0 Volts and reasons that this could indeed cause the problem: The AVC controls the gain of the converter and the IF amplifier. At 0 Volts everything is at maximum gain. A strong signal would then be amplified enough to force the IF amplifier into a non-linear operating region, hence the distortion.

What is the problem with the AVC bus? The repairman proposes two possibilities: either the AVC voltage is not being generated, or it is being bypassed to ground in the converter or IF amplifier. The repairman checks these hypotheses by having his assistant disconnect the AVC line from the converter and IF stages and measure the line's voltage in isolation. He finds that it is still zero! Thus the repairman determines that the voltage is not being generated. He then looks at

the detector in more detail (Figures 5 and 6). Since there is an audio component emerging from the audio filter (with a DC component) the problem must be in the DC filter. This leaves only two components to test: the 3.3 MOhm resistor and the .1 μ f capacitor. He finds that the capacitor is shorted, solving the problem.

What Is the AI Content?

The electronics repairman will necessitate our addressing Artificial Intelligence issues much broader than that which we have chosen to emphasize thus far. We may view an electronics repairman system as the interaction of two knowledge bases: qualitative knowledge of the individual behavior of, and interactions of electronic components (We emphasize the qualitative nature of this system in contrast to the quantitative nature of the typical electronics circuit design program.), and knowledge about bugs and debugging. The locus of interaction of these two knowledge bases is at the yet to be determined interface of plans for devices and bugs of devices. This problem domain is distinctly non-trivial, as is evidenced by the fact that humans require significant time to master the understanding and repair of electronic devices. We believe that the electronics repairman can be made to exhibit to some large degree that desirable quality known as "common sense," since his problem-solving task will require him to do qualitative causal and teleological reasoning. Such a repairman would be intelligently "alive" to the extent that he exhibits purposeful experimental and exploratory behavior. In fact, a reasonable long-term research goal might be the achievement of sufficiently good descriptive languages for electronic devices and debugging so as to make it possible to ask the repairman questions about his behavior and to offer him

advice.

A Research Protocol

We will set forth a number of stages of research that we foresee in the development of an electronics repairman. Although it is expected that the stages will evolve sequentially in the order described, it is inconceivable that later stages will not cause the re-thinking of earlier ones (and that earlier stages will not motivate modified or additional later stages).

The first phase of research in the development of the electronics repairman is the epistemological and linguistic phase. We shall have to deal with the following sorts of questions: What knowledge is necessary to give a complete teleological description of the operation of a circuit? What is the language appropriate to the representation of that knowledge in a hierarchy (or perhaps heterarchy?) of abstractions of electronic circuit concepts?

The next phase is concerned with the debugging process. It will be necessary for us to become adept at electronic debugging, and to abstract protocols of our own debugging techniques. We shall also make use of the electronics literature to obtain protocols of standard trouble-shooting procedures. After suitable digestion of these protocols, we will then have to pose such questions as: What is the catalogue of possible bugs? What is the language appropriate to the description of bugs? What is the catalogue of debugging techniques? What is the language appropriate to the description of debugging techniques? Is there some natural underlying structure to these catalogues?

At this point we should be ready to take a collection of

schematics and annotate them both from the teleological point of view and from the point of view of bugs that might possibly arise. (This exercise should be the hand-coding of the parse-tree that the electronics repairman should eventually produce for himself.) These annotated schematics will be the prelude to the final phase of development of the initial repairman. This will be the realization of a program that embodies the answers to the questions posed above. Moreover, this program will be capable of carrying out the debugging process described in Sussman's proposal.

The development of the repairman will not end here, of course. But beyond this point we see dimly a more advanced repairman able to fill in local design details of a partially specified (in the sense of a parse-tree) electronic device. This repairman would also be capable of patching local design errors. It is reasonable for us to surmise that the early repairman will extend naturally to the advanced repairman. The further extension of the repairman to one capable of debugging circuits having type-shared, multi-purpose sub-modules, is less obvious, and hence probably less natural. This latter extension would undoubtedly reveal much about the relationship of local bugs and global consequences, a topic which we barely begin to understand at present.

DATE	SYM	REVISION RECORD	AUTH.	DR.	CK.

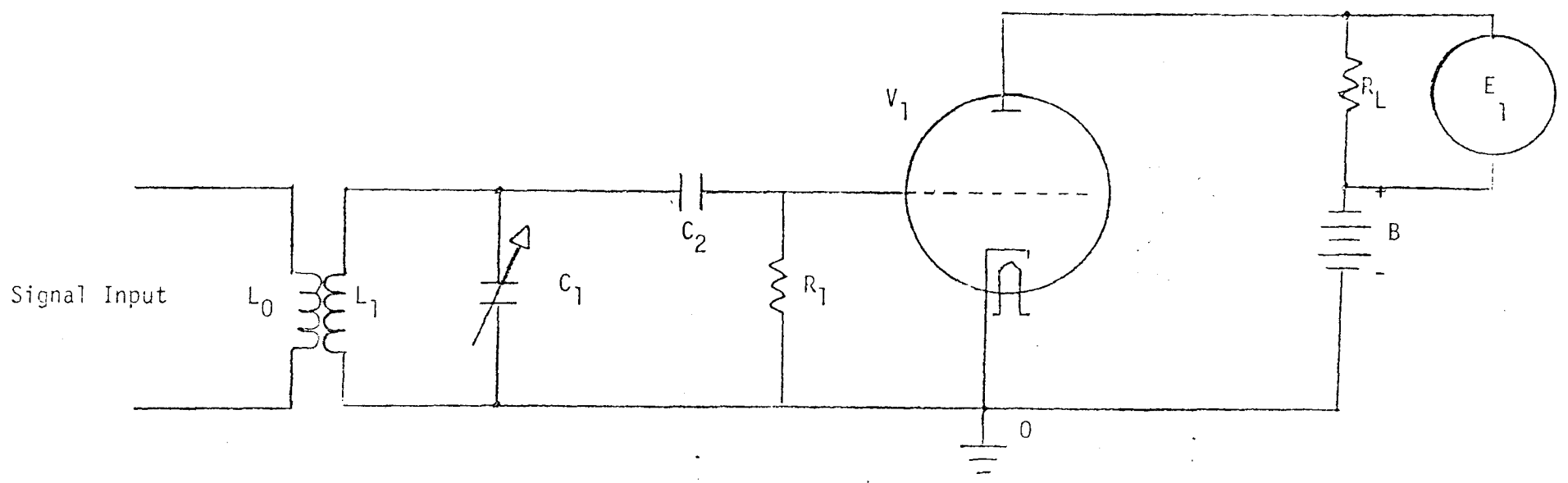


Figure 1
Single-Tube AM Radio Receiver

TOLERANCES (EXCEPT AS NOTED)			
DECIMAL		SCALE	DRAWN BY
±			APPROVED BY
FRACTIONAL	TITLE		
±			
ANGULAR	DATE	DRAWING NUMBER	
±			

DATE	SYM	REVISION RECORD	AUTH.	DR.	CK.

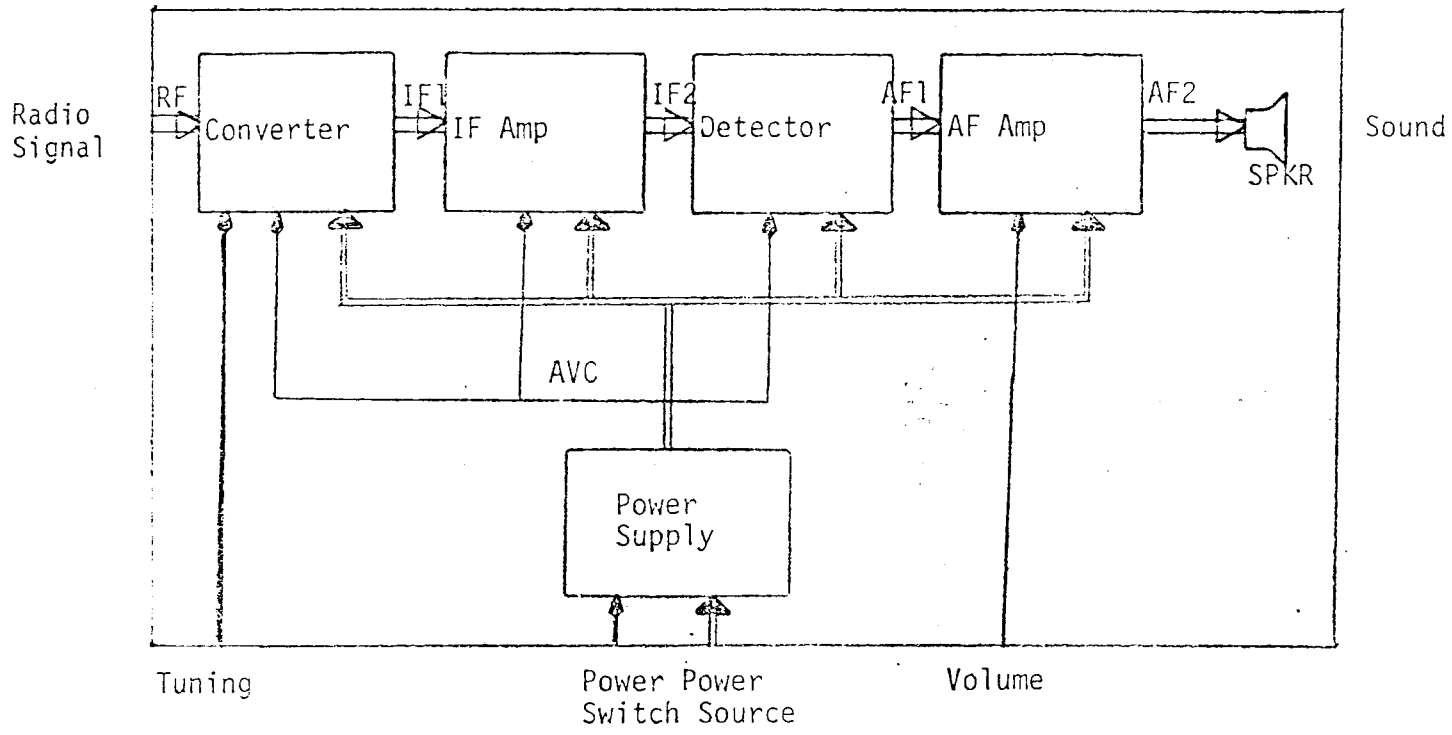


Figure 2
Superhet #1

TOLERANCES (EXCEPT AS NOTED)			
DECIMAL		SCALE	DRAWN BY
±			APPROVED BY
FRACTIONAL	TITLE		
±			
ANGULAR	DATE	DRAWING NUMBER	
±			

DATE	SYM	REVISION RECORD	UTH.	DR.	CK.

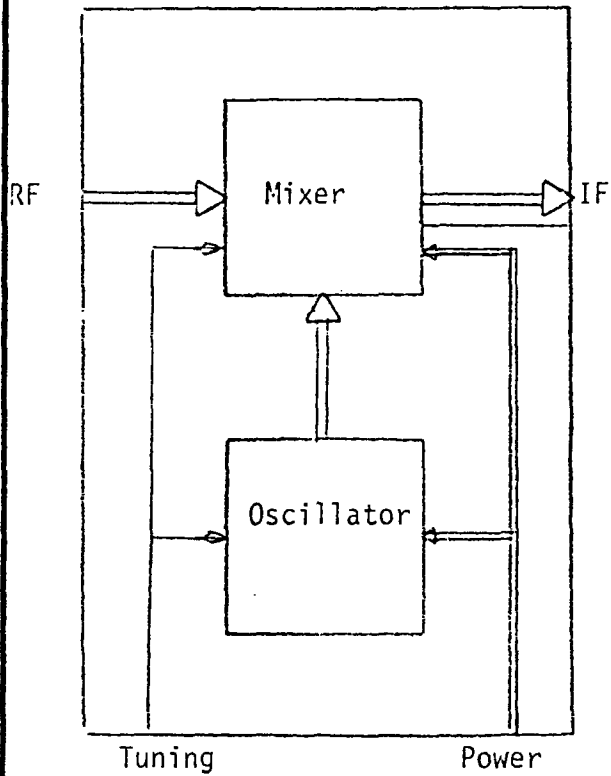
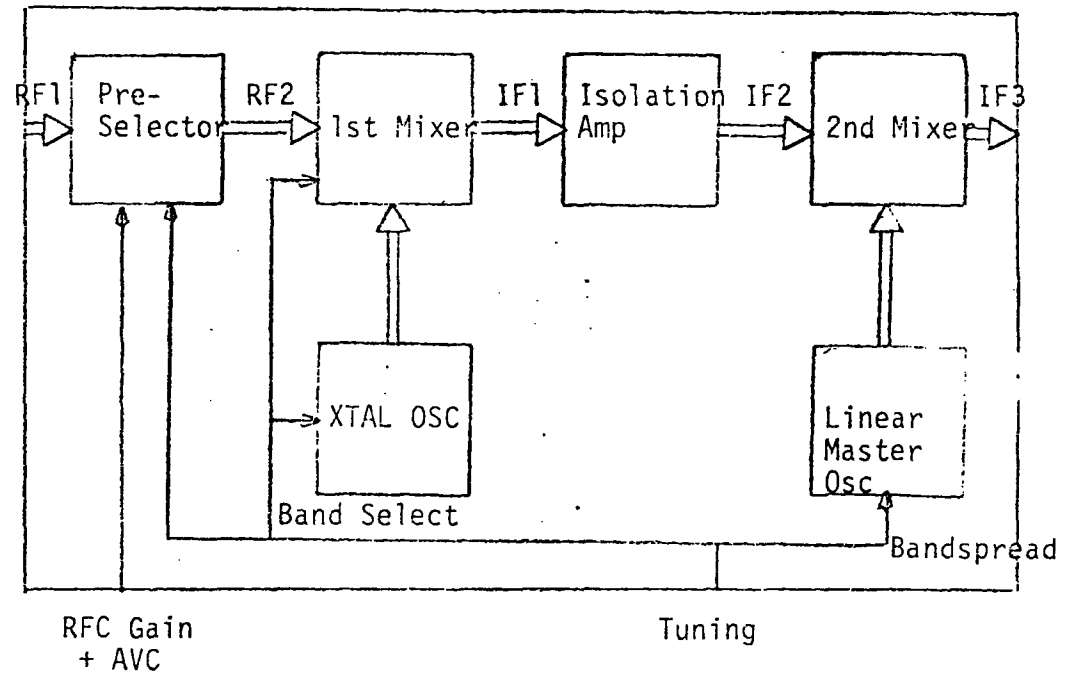


Figure 3
Simple Converter

Figure 4 Communications Receiver Converter



TOLERANCES (EXCEPT AS NOTED)			
DECIMAL		SCALE	DRAWN BY
±			APPROVED BY
FRACTIONAL	TITLE		
±			
ANGULAR	DATE	DRAWING NUMBER	
±			

DATE	SYM	REVISION RECORD	AUTH.	DR.	CK.

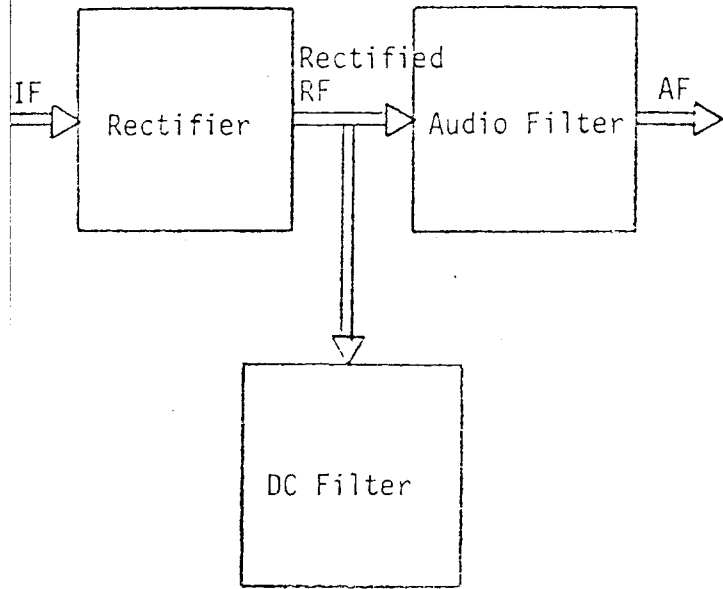


Figure 5

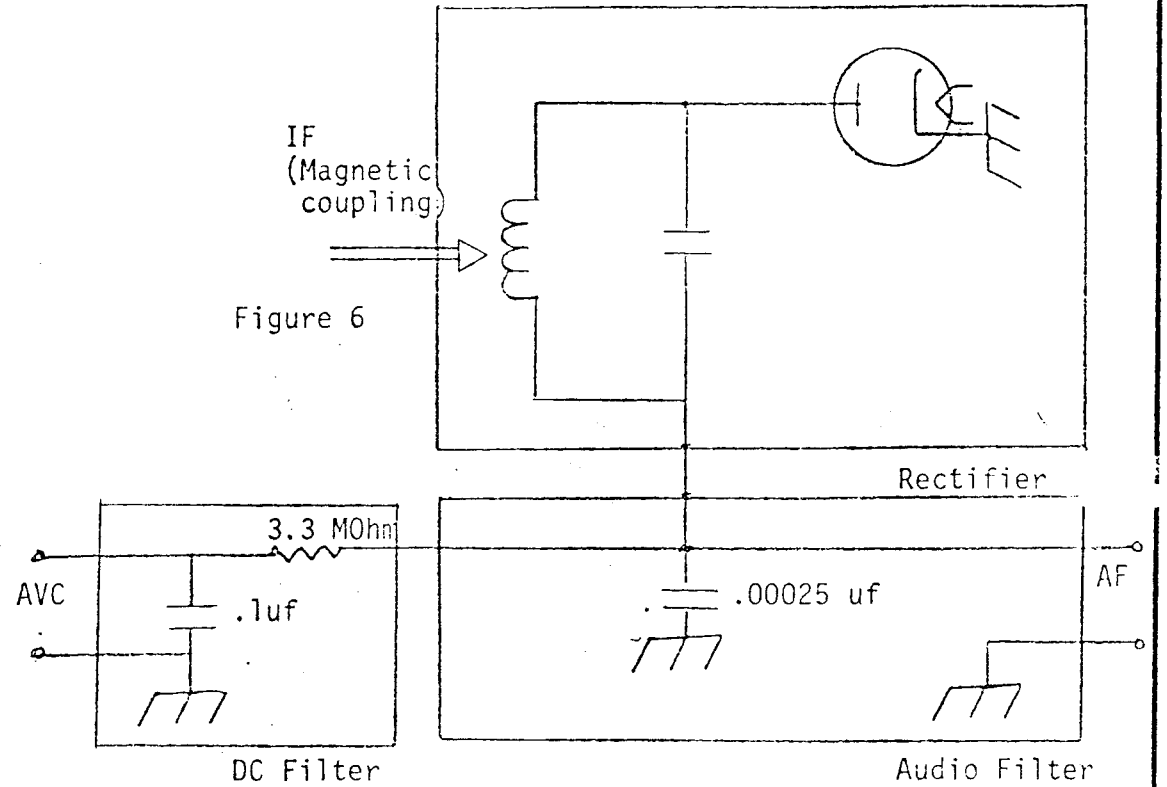


Figure 6

TOLERANCES (EXCEPT AS NOTED)			
DECIMAL		SCALE	DRAWN BY
±			APPROVED BY
FRACTIONAL	TITLE		
±			
ANGULAR	DATE	DRAWING NUMBER	
±			

Cited References

<Fahlman>

MIT Artificial Intelligence Laboratory, A Planning System for Robot Construction Tasks, by Scott Fahlman, AI-TR-283, Cambridge, MA, Feb 1973

<Goldstein>

An Intelligent Monitor for LOGO, by Ira P. Goldstein, AI-TR-294, Cambridge, MA, Sept 1973

<Marcus>

Marcus, Abraham and William Marcus, Elements of Radio, Englewood Cliffs, NJ, Prentice-Hall, 1959

<Papert>

MIT Artificial Intelligence Laboratory, Uses of Technology to Enhance Education, by Seymour Papert, AI-Memo-298, Cambridge, MA, June 1973

<Sussman>

MIT Artificial Intelligence Laboratory, A Computational Model of Skill Acquisition, by Gerald J. Sussman, AI-TR-297, Cambridge, MA, Aug 1973

<Tepper>

Tepper, Marvin, Basic Radio Repair, New York, John F. Rider Publisher, Inc. 1963

<Winston>

MIT Artificial Intelligence Laboratory, 1974 Proposal to the Advanced Research Projects Agency for Research on Intelligent Automata, by Patrick Winston, Marvin Minsky, and Seymour Papert, Sept 1973

Uncited References

<ARRL>

American Radio Relay League, The Radio Amateur's Handbook, 47'th edition, Doug DeMauw, ed., Newington, CT, ARRL, 1970

<Benedict>

Benedict, R. Ralph, Electronics for Scientists and Engineers, Englewood Cliffs, NJ, Prentice-Hall, 1967