An Interactive Numerical Analysis Code for Linear Electroacoustic Systems

1. Introduction

In much electroacoustic work there is a need for applying techniques of mathematical linear system analysis to specific arrangements of components. Traditionally, these analyses are regarded as a branch of circuit theory, and progress through three identifiable steps. First, an equivalent electrical schematic diagram is constructed, with electroacoustic analogies being applied where necessary. Second, a set of simultaneous linear equations is written, based on applying Kirchoff's laws to the network diagram. Third, the equations are solved to eliminate all internal (non-terminal) variables and expressions for the system response and terminal impedances are found analytically. When the resulting expressions are complicated, digital computer codes are useful in evaluating and displaying system characteristics for given values of the constant parameters in the system (see, for example, [1]) and can be interactively employed to optimize adjustable parameters. Such codes can also be combined with parameter-space search routines to generate families of parameter sets meeting desired specifications (e.g., [2]), thereby permitting synthesis of systems achieving given requirements.

A substantial reduction in algebraic drudgery could be realized by extending the role of the computer to encompass or eliminate the three tasks just discussed. System performance is, after all, a well-defined function of its component elements' properties, and suitable software should in principle allow any desired diagnostic functions to be constructed from a mathematically complete system de.
scription. With such software, the designer could pass directly from a block-diagram-level description to an assessment of its performance. In this way, the designer's role could become more clearly focused in the areas of devising and evaluating new configurations — creative areas in which human imagination is far more powerful than digital computer techniques. In this paper, I shall describe the present status of a Fortran-language code that realizes many of these objectives.

Program RAD and its associated subroutines constitute a means for evaluating the end-to-end amplitude and phase characteristics of a given linear system. The system's topology is specified simply by a list of CALL statements within the program. This list corresponds to the sequence of stages in the system's block diagram. The number of elements in the system is limited only by the available space in the computer's memory. Individual elements include lumped-constant devices such as amplifiers, simple electrical and acoustic inductances, and transducers; distributed-constant elements such as ducts, chambers and horns are also included. Furthermore, coupling by free-space radiation can be regarded as a component of certain kinds of system, and can be represented in the system specification. After compilation, the code runs interactively to allow complete user control of all parameter arrays, frequency display ranges, and plot formats. Compatibility with most common input-output devices (CRT's, teletypes, and line printers) is made possible through the exclusive use of printer plots for diagnostic function display. Thus the many advantages of graphic diagnostic presentation are retained, without the need for a vector graphics terminal facility and specialized support software. Core memory requirements are easily met by medium-size minicomputers.

The implementation of this program could, in principle, have been based on any of several well-developed electrical network analysis codes. However, these algorithms typically require a system specification in the form of numbered junctions or nodes and a list of admittances connecting these nodes, i.e., a kind of schematic diagram. With this basis, the engineer would be required to formulate the necessary electroacoustic analogs and prepare the diagram. Moreover, network algorithms are often not adaptable to distributed elements, which comprise infinitely many nodes and consequently have transcendental rather than algebraic response functions. For these reasons, the software described here is based on the T-matrix method ([3], hereinafter referred to as Paper 1). This technique offers a number of advantages, one of which is that the terminal properties of each stage or element of a system are consolidated into a identifiable mathematical entity: the T-matrix.

2. The T-matrix technique

Any linear source-free two-port device can be mathematically represented by a transmission matrix or T-matrix defined by

\[
\begin{pmatrix}
\text{e}_1 \\
\text{i}_1
\end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix}
\text{e}_2 \\
\text{i}_2
\end{pmatrix} = T \begin{pmatrix}
\text{e}_2 \\
\text{i}_2
\end{pmatrix}
\]

(1)

where \((e_1, i_1)\) and \((e_2, i_2)\) are the complex (voltage, current) variables at the input and output ports respectively. The representation exists whenever the output variables are dependent upon the input variables. The four elements of the matrix are, in general, complex-valued functions of frequency. They are respectively the reciprocal of the open circuit voltage gain, the reciprocal of the short circuit transadmittance, the reciprocal of the open circuit transadmittance, and the reciprocal of the short circuit current gain. The matrix elements depend only upon the internal parameters of the device and not on the source or load impedances presented to it. They are thus invariant under interconnection. Consequently, the overall end-to-end T-matrix of a sequence of cascaded stages is simply given by the product of the individual stages' T-matrices.

The generalization of the T-matrix method to include mechanical and acoustic variables, and the application to branched rather than simply cascaded systems, was discussed in paper 1 and references therein. Briefly, transducers are represented as matrices connecting the electrical regime (voltage, current) with the mechanical (force, velocity) or acoustical (pressure, volume velocity) regimes. Two-way branches are dealt with by replacing them with the appropriate combination T-matrix, which is a straightforward function of the T-matrices of the separate branches. N-way branches are represented as N-1 successive two-way branches.

Once the T-matrix of a complete system has been built up, its response may be found by identifying the single desired input and output variable and calculating the reciprocal of the corresponding matrix element. For example, suppose a microphone model has been computed and one wishes to determine the open-circuit voltage response to a given sound pressure \(p_1\), from the definition of its T-matrix,

\[ p_1 = A e_2 + B i_2, \]

while for an open circuit load \(i_2 = 0\); hence the desired voltage response \(e_2/p_1\) is given by \(1/A\). Similarly, for loudspeakers the ratio of short-circuit output volume velocity \(u_3\) to drive voltage \(e_1\) is given by \(1/B\). Arbitrary load impedances may be treated as in paper 1, by incorporating the load into the T-matrix as a final matrix factor.

Other useful diagnostics are the input and output impedances of a system and its reciprocity index. The open circuit and short circuit input impedances are \(A/C\) and \(B/D\), while the corresponding output impedances are \(D/C\) and \(B/A\). The reciprocity index of a system is the determinant of its T-matrix, \(AD - BC\).

3. Program RAD

A listening of one version of RAD in given in Fig. 1. This version was written as part of a study of acoustic resonances in transmission-line and bass-reflex loudspeaker enclosures. It provides examples of techniques useful in interactively modifying parameter arrays, building a system T-matrix, and displaying the resulting diagnostics.

Nongraphical input/output functions are handled by RAD directly, while plots are constructed through the use of two subroutines. In this version, it has been assumed that three input/output devices are available: a user CRT, a keyboard, and a line printer for permanent hard copy output. These devices are assigned logic unit addresses 6, 5, and 3.

The program itself has four chief divisions: the declaration and data statements, the parameter specification procedure, the evaluation of diagnostics, and the preparation of output. These divisions are delineated by the comment cards appearing in the code.

Complex arrays are provided for several T-matrices and three diagnostic functions of frequency: response, input impedance, and reciprocity index. A few complex inmitances are also declared for possible use in calculating lumped-constant series and shunt immittance T-matrix routines. Real constant parameter arrays are declared, which de-
Acoustica
Vol. 42 (1979)

RESEARCH NOTES

fault alternatives are expressed by entering zeros or blanks. The entire array is left with its default values by simply entering a carriage return at the keyboard, since this creates a null record which is interpreted as all blanks. Any number of subsequent arrays can be brought to the CRT for possible updating in this fashion. SI units (mks) are employed throughout this code, including the parameter arrays.

The lower and upper limits for the frequency range, in hertz, are displayed at statement 170, and can at this point be modified via the user keyboard. Again, a zero or blank entry leaves the previous value unchanged.

The overall evaluation process takes place within the DO 299 loop, in which individual frequencies between FMIN and FMAX are generated and the diagnostics are calculated. For most purposes, a logarithmic frequency scale is desirable; this is achieved by obtaining each frequency from an exponential function of the DO loop index, I, as shown. For some purposes such as subsequent Fourier analysis into the time domain, equally spaced frequencies are required, and would be obtained by substituting an appropriate linear function (avoiding zero frequency) for statement 210. The angular frequency \( \omega \) is the parameter actually passed to the subroutines employed here.

The actual T-matrix buildup is, of course, the most system-dependent procedure in the code, since it is the sequence of calls to the matrix-generating routines which specifies the system. This buildup is simplest in unbranched (cascaded) systems, where one can proceed directly from input to output. As an example, suppose that the consecutive stages are representable by \( T \)-matrices generated by successive subroutines MAT1, MAT2, . . . , MATn. Then one would write

```
CALL MAT1(T)
CALL MAT2(U)
CALL MULT(T, U, T)
CALL MAT3(U)
CALL MULT(T, U, T)
.
.
.
CALL MATn(U)
CALL MULT(T, U, T)
```

which leaves the final system matrix stored in \( T \). The multiplication routine MULT(X,Y, XY) has internal buffering to allow its output array to safely overwrite either input array.

Branched topologies are dealt with by storing the \( T \)-matrix of each branch and then calling the appropriate matrix combination function. The result is a \( T \)-matrix which is ready to be multiplied into the buildup at the branch location. An example, the basic reflex loudspeaker system, was discussed in paper 1; in Fig. 1, this is implemented by combining the identity matrix, UNITY, with the enclosure/vent matrix using the SIPO rule as in paper 1.

When writing the system buildup statements, it must be borne in mind that matrix multiplication is not commutative, although it is associative. Thus the individual factors must form the same sequence as the corresponding system stages, although they may be arbitrarily grouped together for computational convenience. Of course it is also essential to join a mechanical port only to another mechanical port, and so forth.

Diagnostic functions specified by the user are evaluated once the system matrix is complete. The example shown in

```
Fig. 1. Fortran listing of program RAD.
```

scribe an electrodynamic loudspeaker and two ducts. The first of these ducts, ENCL, represents a chamber surrounding the rear of the loudspeaker; the second, VENT, connects the enclosure volume to the ambient medium. The number of desired frequency steps, NUM, must not exceed the dimensions of the diagnostic arrays. In this version, NUM equals 101.

In operation, the program title, user instructions, and the first parameter array are transmitted to the user CRT. Beneath each parameter value, a reminder as to its name is printed. At this point the user must enter new chosen values or, by default, allow the old values to remain. De-
Fig. 1 illustrates the computation of the loudspeaker response based on the reciprocal of its T(2) or "B" matrix element, its electrical input impedance, and, for verifying the smallness of computational errors, the reciprocity index.

When the frequency step loop is complete, control transfers to the 300-series statements which ask the user to specify the plot options desired. Each question includes a list of possible answers from which subsequent CALL decisions are made. One exception in the example of Fig. 1 is the possibility to obtain a complex-plane plot of the determinant by offering a D response to the first question. Default responses (e.g., a carriage return or a blank) lead to the first-listed option following each question.

After any plot has been displayed on the CRT, an option exists to request a hard copy of the same plot from the line printer. Printed pages usually accommodate a somewhat larger format than the 80 x 25 character display common to most CRT's, and accordingly the plot calling specifications transmit a larger format request for line printer output than for CRT output. These formats should, of course, be chosen to provide a convenient scale on the user's hardware.

After finishing with the plot requests, the code offers four alternatives: additional plots of any of the existing diagnostic arrays, a reexamination of the same system in another frequency range, a modification of any of the system parameters, or termination of the job. This branching is again handled with IF statements acting on the user's answer.

4. Subroutines

Fortran-language subroutines for use with RAD are of two kinds: the matrix routines, which construct or manipulate four-element complex arrays, and the diagnostic routines, which create specific printer plots of given complex arrays. Variables are not transmitted by a COMMON declaration. Instead, the first portion of each subroutine's argument list is used to transmit numbers to the subroutine when it is called, while the remaining variables in the list return results to the calling program. A number of useful routines are listed in Fig. 2.

The routines for matrix multiplication, series impedance, and shunt admittance are reasonably straightforward. The exponential horn routine EXPF is derived from paper 1, eq. (5-16). It can be employed to simulate constant cross-section ducts by setting the flare rate equal to zero. In this way, one-dimensional chambers may be modeled. The electrodynamic loudspeaker routine, SPKR, computes the transducer T-matrix from the seven parameters listed in its comment cards, using eq. (4-2) of paper 1. An error flag is set to 1 if any parameters are invalid, and in the main program this condition causes the parameter array to be transmitted to the user CRT for correction. Subroutine FREE generates the T-matrix representing the free-field acoustic coupling between two points separated by a given distance, D. The matrix is the elementary negative immittance inverter form identified in paper 1 (see discussion following eq. (6-6))

The combination matrix subroutines such as SIPO (see paper 1, eq. (7-1) to (7-4)) are equipped with error flags which sense a zero-denominator fault condition. A warning: if the combination routines are to be called upon to substitute a result for a given input matrix, as SIPO is by RAD, it is essential to either compute an intermediate temporary matrix result (as in MULT) or to calculate the four elements in a non-disruptive sequence. Recommended sequences are for PIPO 4321; for PIPO 3412; for SISO 2143; and for SISO 1234.

Two additional combination matrices are useful in describing radiation coupled transducers whose electrical input ports are in series or parallel. Here, the radiation field is represented as a general three-port network connecting the transducers' outputs to the system output port. This network is specified by the nine elements of the Z matrix, while the transducers and other components specific to each branch are described by individual branch T-matrices. This combination is calculated by SIZO for series input connections, or PIZO for parallel input connections. The subroutines are listed in Fig. 3. They are proving useful in modelling differential delay phenomena in two way loudspeaker systems.
Fig. 3. Listing of the matrix combination routines SIZO and PIZO.

Fig. 4. Listing of the plot routines BODE and CPLN.

Fig. 5. BODE plot of a vented enclosure system between 10 and 1000 Hz. The amplitude scale is in dB relative to 1 pascal per volt at one metre. The phase scale is in degrees.
Subroutine BODE constructs a bar-graph histogram of the logarithm of a given complex array, and superposes a linear-scale point-plot of the corresponding phase angles. The argument list is defined in the comment cards. The vertical full scale span is set to 25 dB with the top of the chart set to the largest value in the array. The routine automatically can plot a list of NDAT data on a grid of NX carriage spaces even if NDAT ≠ NX, by means of periodically repeating or skipping data as required. The subroutine listing is given in Fig. 4, and an example of its output appears in Fig. 5.

Subroutine CPLN plots the locus of an array in the complex plane. The array is first searched to determine the extreme values of the real and imaginary components. These become the boundaries of the plot. The output field array, LINE, is initialized to the Hollerith symbol "" everywhere, and then is overwritten by a numerical character at locations determined by the given complex array. These characters help to distinguish the increasing frequency direction along the locus. An example of the output from CPLN is shown in Fig. 6.

5. Applications

Several versions of RAD have been written. The first cases examined were essentially tests of the code, in which loudspeaker alignments having known response forms were reproduced. Subsequent runs simulated aspects of the transmission-line loudspeaker problem, and have been instructive in clarifying the consequences of midrange enclosure and vent resonances. The T-matrix technique is applicable to a broad range of analysis problems, including equalizer and crossover design, optimization of multi-way phase coherent loudspeakers, and microphone design. Since individual machine evaluations require only about one second of computation per iteration, the tasks of pilot design and tolerance analysis can be effectively conducted on an exploratory basis.

(Received July 1st, 1977.)

References